FAST ALGORITHM FOR CONDITIONAL SEPARATION AND DEREVERBERATION

Takuya Yoshioka, Tomohiro Nakatani, and Masato Miyoshi

NTT Communication Science Laboratories, NTT Corporation
2-4, Hikari-dai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan
phone: + (81) 774 93 5326, fax: + (81) 774 93 5158, email: takuya@cslab.kecl.ntt.co.jp
web: http://www.kecl.ntt.co.jp/icl/signal/takuya/index.html

ABSTRACT

Last year, we proposed a conditional separation and dereverberation method (the CSD method) for simultaneously achieving blind source separation and dereverberation of sound mixtures. This paper presents a novel fast algorithm for implementing the CSD method. Most of the computation time of the CSD method is spent on calculating what we call modified correlation matrices. The proposed algorithm can calculate these modified correlation matrices much faster than the original algorithm. This improvement is realized by capitalizing on the particular structure of the modified correlation matrices. Experimental results obtained using 672 test samples revealed that the CSD method provided a much better signal-to-interference ratio than a frequency-domain blind source separation method. The real time factor of the proposed algorithm was between 4 and 6, which is less than one tenth that of the original algorithm.

1. INTRODUCTION

The issues of blind source separation (BSS) [1] and blind dereverberation (BD) [2] have attracted a lot of attention over the past decade. The aim of BSS is to separate mixtures of multiple sounds while that of BD is to remove the effect of reverberation from sounds picked up by microphones. Although there have been many advances as regards these two issues, there have been few attempts to achieve BSS and BD simultaneously. We refer to this as blind source separation and dereverberation (BSSD). On one hand, it is known that the performance of conventional BSS methods gradually deteriorates as the reverberation effect becomes severe [3]. On the other hand, most of the existing BD methods are based on the assumption that only one sound source is active at a time. Therefore, simply cascading existing BSS and BD methods could not accomplish BSSD.

At EUSIPCO 2008, we presented a method for BSSD, which we call the Conditional Separation and Dereverberation (CSD) method, to overcome the above limitation [4]. The CSD method provided high separation and dereverberation performance even when the reverberation time of the room was longer than 0.5 sec. Unfortunately, however, its computation time was very long, which renders the CSD method useless in practice.

We found that most of the computation time was spent on the calculation of matrices with particular structures, which we call modified correlation matrices. Thus, in this paper, we propose a fast algorithm for calculating those matrices based on a fast Fourier Transform (FFT). This is realized by capitalizing on the structures of the modified correlation matrices. Furthermore, large-scale experimental results are reported whereas, by contrast, the experiment reported in [4] was quite limited.

The remainder of this paper is organized as follows: Section 2 reviews the original algorithm of the CSD method; Section 3 describes the novel algorithm for modified correlation matrix calculation; Section 4 reports the experimental results; and Section 5 concludes this paper.

2. CONDITIONAL SEPARATION AND DEREVERBERATION

2.1 Task Specification

We assume that we capture sounds in a room using M_{M} microphones. We also assume that there are M_{S} sound sources in the room and that $M_{\mathrm{S}} \leq M_{\mathrm{M}}$.

Let $s^m_{t,l}$ denote the signal emitted from the mth sound source represented in the short-time Fourier transform (STFT) domain, where t and l are time frame and frequency bin indices, respectively. The typical frame size and frame shift for the STFT are 30 msec and 15 msec, respectively. This is in contrast to the frequency-domain BSS method, which often uses such a long time frame that covers the reverberation time. We represent all source signals in vector form as

$$\boldsymbol{s}_{t,l} = [s^1_{t,l}, \cdots, s^{M_{\mathrm{S}}}_{t,l}]^T, \tag{1}$$

where superscript T represents a non-conjugate transpose. Source signals $s^1_{t,l}, \cdots s^{M_{\mathrm{S}}}_{t,l}$ are reverberated and mixed with each other while propagating from the sound sources to the microphones. We denote the signal observed by the mth microphone as $y^m_{t,l}$ and the vector of the observed signals as

$$\boldsymbol{y}_{t,l} = [y^1_{t,l}, \cdots, y^{M_{\mathrm{M}}}_{t,l}]^T. \tag{2}$$

Now, we assume that $\boldsymbol{y}_{t,l}$ is observed over N consecutive time frames for all l values from 0 to $L-1$, where L is the number of frequency bins. Let $\mathscr{S}$ and $\mathscr{Y}$ be sets of $\boldsymbol{s}_{t,l}$ and $\boldsymbol{y}_{t,l}$, respectively, over all t and l so that we have

$$\mathscr{S} = \{\boldsymbol{s}_{t,l}\}_{0 \leq t \leq N-1, 0 \leq l \leq L-1} \tag{3}$$

$$\mathscr{Y} = \{\boldsymbol{y}_{t,l}\}_{0 \leq t \leq N-1, 0 \leq l \leq L-1}. \tag{4}$$

Then, the task to be solved is defined as estimating the source data, $\mathscr{S}$, when the observed data, $\mathscr{Y}$, are given. This task may be restated as achieving BSS and BD simultaneously, i.e. BSSD.

2.2 Method Description

The CSD method calculates an estimate, $\hat{\boldsymbol{s}}_{t,l}$, of source signal vector $\boldsymbol{s}_{t,l}$ according to the following formulae for each

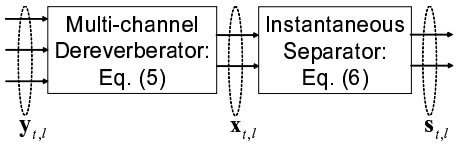

Figure 1: Diagram of source signal estimator.

frequency bin index l:

$$\boldsymbol{x}_{t,l} = \boldsymbol{y}_{t,l} - \sum_{k=1}^{K_l} G_{k,l}^H \boldsymbol{y}_{t-k,l} \tag{5}$$

$$\hat{\boldsymbol{s}}_{t,l} = W_l^H \boldsymbol{x}_{t,l}, \tag{6}$$

where $G_{k,l}$ is an M_{M}-by-M_{S} matrix, W_l is an M_{S}-by-M_{S} matrix, and superscript H represents a conjugate transpose. Therefore, we want to optimize $G_{1,l}, \cdots, G_{K_l,l}$ and W_l for all l values so that $\hat{\boldsymbol{s}}_{t,l}$ can best approximate the unobservable true source signal vector, $\boldsymbol{s}_{t,l}$. Reflecting the forms of (5) and (6), $G_{k,l}$ and W_l are referred to as a regression matrix and a separation matrix, respectively. Figure 1 shows a diagram of the source signal estimator defined by (5) and (6). As shown in the figure, the process for obtaining $\boldsymbol{x}_{t,l}$ according to (5) serves as a multi-channel dereverberator while (6) corresponds to an instantaneous separator. The CSD method jointly optimizes these two systems, namely the regression matrix $G_{k,l}$ and separation matrix W_l.

Note that the process of calculating $\hat{\boldsymbol{s}}_{t,l}$ based on (5) and (6) is equivalent to filtering $\boldsymbol{y}_{t,l}$ with a causal multiple-input multiple-output (MIMO) finite impulse response (FIR) filter of order K_l. Hence, the task we are going to perform is almost equivalent to a standard convolutive BSS task. The only difference is that we aim at cancelling the effect of reverberation in addition to separating sound mixtures while the standard convolutive BSS task ignores the reverberation.

The CSD method finds the values of the regression and separation matrices that minimize a cost function, which is derived based on a time-varying all-pole source signal model. Let P, $a_{t,k}^m$, and γ_t^m denote the number of poles, the kth linear prediction coefficient (LPC), and the prediction residual power (PRP) of the mth source signal at the tth time frame, respectively. Then, the cost function is described as

$$F = \sum_{l=0}^{L-1} \sum_{t=0}^{N-1} \left\{ \left(\boldsymbol{y}_{t,l} - \sum_{k=1}^{K_l} G_{k,l}^H \boldsymbol{y}_{t-k,l} \right)^H W_l \, {}_{\mathrm{S}}\Lambda_{t,l}^{-1} \, W_l^H \right.$$
$$\left. \times \left(\boldsymbol{y}_{t,l} - \sum_{k=1}^{K_l} G_{k,l}^H \boldsymbol{y}_{t-k,l} \right) - \log \left| W_l \, {}_{\mathrm{S}}\Lambda_{t,l}^{-1} \, W_l^H \right| \right\}. \tag{7}$$

M_{S}-by-M_{S} matrix ${}_{\mathrm{S}}\Lambda_{t,l}$ is a diagonal matrix, of which the mth diagonal element is the all-pole spectral component of the mth source signal at time frequency point (t,l). To be more precise, ${}_{\mathrm{S}}\Lambda_{t,l}$ is defined based on the LPCs and PRPs of the source signals as follows:

$${}_{\mathrm{S}}\Lambda_{t,l} = \begin{bmatrix} {}_{\mathrm{S}}\lambda_{t,l}^1 & & O \\ & \ddots & \\ O & & {}_{\mathrm{S}}\lambda_{t,l}^{M_{\mathrm{S}}} \end{bmatrix} \tag{8}$$

$${}_{\mathrm{S}}\lambda_{t,l}^m = \frac{\gamma_t^m}{\left| 1 - a_{t,1}^m e^{-j\frac{2\pi l}{L}} - \cdots - a_{t,P}^m e^{-j\frac{2\pi lP}{L}} \right|^2}. \tag{9}$$

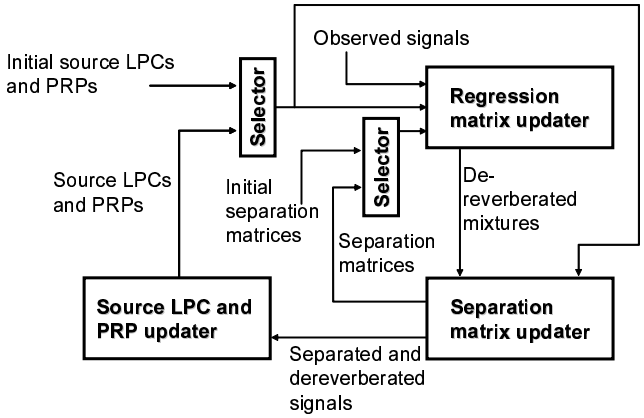

Figure 2: Diagram of parameter optimization. The selectors output the initial parameters only for the first iteration. For the subsequent iterations, updated parameters are selected.

All parameters (i.e. regression matrix $G_{k,l}$, separation matrix W_l, source LPC $a_{t,k}^m$, and source PRP γ_t^m) are optimized so as to minimize the value of cost function F.

The CSD method alternately updates regression matrix $G_{k,l}$, separation matrix W_l, and source-related parameters $\{a_{t,k}^m, \gamma_t^m\}$ until convergence. Figure 2 is a diagram of the parameter optimization scheme.

The source LPC and PRP, $a_{t,k}^m$ and γ_t^m, are updated by minimizing the cost function, F, for the fixed regression matrix $G_{k,l}$ and separation matrix W_l. This partial minimization is accomplished by the short-time linear predictive analysis of the tentative source signal estimates that are calculated based on the fixed values of $G_{k,l}$ and W_l. These tentative source signal estimates are indicated by the arrow labeled "Separated and dereverberated signals" in Figure 2.

The separation matrix, W_l, is updated by minimizing F for fixed $G_{k,l}$, $a_{t,k}^m$, and γ_t^m. A convenient way to achieve this partial minimization is to apply an independent component analysis (ICA) algorithm to the dereverberated, but still mixed, signals that are calculated using the fixed value of $G_{k,l}$ and indicated by the arrow labeled "Dereverberated mixtures" in Figure 2. Although this update procedure does not necessarily minimize F, we found that it sufficed in practice. An algorithm that exactly minimizes F, which is based not only on the dereverberated mixtures but also on $a_{t,k}^m$ and γ_t^m as shown in Figure 2, is described in [4].

Finally, the update rule for the regression matrix, $G_{k,l}$, may be described as follows. For each l, all regression matrices for the lth frequency bin are jointly updated. Let us define vector $\boldsymbol{g}_l$, which consists of the elements of the regression matrices for the lth frequency bin, as

$$\boldsymbol{g}_l = [(\boldsymbol{g}_{1,l}^1)^T \ \cdots \ (\boldsymbol{g}_{1,l}^{M_{\mathrm{S}}})^T \ | \ \cdots \ | (\boldsymbol{g}_{K_l,l}^1)^T \ \cdots \ (\boldsymbol{g}_{K_l,l}^{M_{\mathrm{S}}})^T]$$
$$(M_{\mathrm{S}} M_{\mathrm{M}} K_l\text{-dimensional row vector}), \tag{10}$$

where $\boldsymbol{g}_{k,l}^m$ denotes the mth column of $G_{k,l}$. $\boldsymbol{g}_l$ is referred to as a regression coefficient vector. Furthermore, let matrix $\boldsymbol{Y}_{t-1,l}$

be defined as

$$\boldsymbol{Y}_{t-1,l} = \left[\begin{array}{ccc|c|ccc} \boldsymbol{y}_{t-1,l}^H & & O & & \boldsymbol{y}_{t-K_l,l}^H & & O \\ & \ddots & & \cdots & & \ddots & \\ O & & \boldsymbol{y}_{t-1,l}^H & & O & & \boldsymbol{y}_{t-K_l,l}^H \end{array}\right] \quad (M_{\mathrm{S}}\text{-by-}M_{\mathrm{S}}M_{\mathrm{M}}K_l \text{ matrix}). \quad (11)$$

Then, the regression coefficient vector is updated according to the following rule:

$$\boldsymbol{g}_l \leftarrow (R_l^{-1}\boldsymbol{r}_l)^H \quad (12)$$

$$R_l = \sum_{t=0}^{N-1} \boldsymbol{Y}_{t-1,l}^H W_{l\,\mathrm{S}} \Lambda_{t,l}^{-1} W_l^H \boldsymbol{Y}_{t-1,l} \quad (13)$$

$$\boldsymbol{r}_l = \sum_{t=0}^{N-1} \boldsymbol{Y}_{t-1,l}^H W_{l\,\mathrm{S}} \Lambda_{t,l}^{-1} W_l^H \tilde{\boldsymbol{y}}_{t,l}, \quad (14)$$

where $\tilde{\boldsymbol{y}}_{t,l}$ is the vector of the signals observed by the first M_{S} microphones, that is to say,

$$\tilde{\boldsymbol{y}}_{t,l} = [y_{t,l}^1, \cdots, y_{t,l}^{M_{\mathrm{S}}}]^T. \quad (15)$$

It is noteworthy that (12) is an extension of the well-known Yule-Walker equation solution. This is a corollary of the fact that the multi-channel dereverberator (5) is based on the multi-channel autoregressive (AR) model. In light of this, matrix R_l and vector $\boldsymbol{r}_l$ are called a modified correlation matrix and a modified correlation vector, respectively. The sizes of R_l and $\boldsymbol{r}_l$ are $M_{\mathrm{S}}M_{\mathrm{M}}K_l$-by-$M_{\mathrm{S}}M_{\mathrm{M}}K_l$ and $M_{\mathrm{S}}M_{\mathrm{M}}K_l$-by-1, respectively.

The following naive algorithm for calculating the modified correlation matrices is very time consuming, which renders the CSD method useless:

$R_l = O$;
for $t = 0 : N-1$
  $R_l = R_l + \boldsymbol{Y}_{t-1,l}^H W_{l\,\mathrm{S}} \Lambda_{t,l}^{-1} W_l^H \boldsymbol{Y}_{t-1,l}$;
end

This naive algorithm may be computationally redundant because most parts of $\boldsymbol{Y}_{t-1,l}$ and $\boldsymbol{Y}_{t,l}$ have the same values. In the next section, we propose a novel algorithm that runs much faster than the naive algorithm.

3. NOVEL ALGORITHM FOR MODIFIED CORRELATION MATRIX CALCULATION

In the following, frequency bin index l is omitted for brevity because regression coefficient vector $\boldsymbol{g}_l$ is updated on a frequency bin by frequency bin basis. In addition, we express $W_{\mathrm{S}}\Lambda_t^{-1}W^H$ as Λ_t, whose (q_1, q_2)th element is denoted by $\lambda_t^{q_1 q_2}$, so that we can have

$$\Lambda_t = W_{\mathrm{S}}\Lambda_t^{-1}W^H = \begin{bmatrix} \lambda_t^{11} & \cdots & \lambda_t^{1M_{\mathrm{S}}} \\ \vdots & \ddots & \vdots \\ \lambda_t^{M_{\mathrm{S}}1} & \cdots & \lambda_t^{M_{\mathrm{S}}M_{\mathrm{S}}} \end{bmatrix}. \quad (16)$$

As shown later, modified correlation matrix R and modified correlation vector $\boldsymbol{r}$ consist of the cross correlation coefficients between two time series $\lambda_t^{q_1 q_2}$ and $y_{t-m}^{p_1 *} y_t^{p_2}$ for all m from 0 to K, all p_1 and p_2 from 0 to M_{M}, and all q_1 and q_2 from 0 to M_{S}, where superscript $*$ represents a complex conjugate. *Therefore, the basic idea behind the proposed algorithm is that we first calculate these cross correlation coefficients and then rearrange them to form R and* $\boldsymbol{r}$.

Indeed, the modified correlation matrix and vector are structured as follows:

$$R = \sum_{t=0}^{N-1} \boldsymbol{Y}_{t-1}^H \Lambda_t \boldsymbol{Y}_{t-1} = \begin{bmatrix} R_{11} & \cdots & R_{1K} \\ \vdots & \ddots & \vdots \\ R_{K1} & \cdots & R_{KK} \end{bmatrix} \quad (17)$$

$$\boldsymbol{r} = \sum_{t=0}^{N-1} \boldsymbol{Y}_{t-1}^H \Lambda_t \tilde{\boldsymbol{y}}_t = \overrightarrow{\sum} \tilde{R} = \overrightarrow{\sum} \begin{bmatrix} \tilde{R}_1 \\ \vdots \\ \tilde{R}_K \end{bmatrix}, \quad (18)$$

where $\overrightarrow{\sum}$ represents the sums of the elements of a matrix over each row, and submatrix R_{mn} and submatrix $\tilde{R}_m$ are given by (19) and (20), respectively. ((19) appears on the top of the next page.) The detail of the proposed algorithm is described below.

$$\tilde{R}_m = \left[\begin{array}{ccc} \sum_{t=0}^{N-1} \lambda_t^{11} y_{t-m}^{1*} y_t^1 & \cdots & \sum_{t=0}^{N-1} \lambda_t^{1M_{\mathrm{S}}} y_{t-m}^{1*} y_t^{M_{\mathrm{S}}} \\ \vdots & & \\ \sum_{t=0}^{N-1} \lambda_t^{11} y_{t-m}^{M_{\mathrm{M}}*} y_t^1 & \cdots & \sum_{t=0}^{N-1} \lambda_t^{1M_{\mathrm{S}}} y_{t-m}^{M_{\mathrm{M}}*} y_t^{M_{\mathrm{S}}} \\ \hline \vdots & & \\ \hline \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}1} y_{t-m}^{1*} y_t^1 & \cdots & \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}M_{\mathrm{S}}} y_{t-m}^{1*} y_t^{M_{\mathrm{S}}} \\ \vdots & & \\ \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}1} y_{t-m}^{M_{\mathrm{M}}*} y_t^1 & \cdots & \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}M_{\mathrm{S}}} y_{t-m}^{M_{\mathrm{M}}*} y_t^{M_{\mathrm{S}}} \end{array}\right] \quad (M_{\mathrm{S}}M_{\mathrm{M}}\text{-by-}M_{\mathrm{S}} \text{ matrix}). \quad (20)$$

First of all, it should be noted that we have only to calculate the lower triangular part of R because R is an Hermitian matrix. We define sequence $\vec{y}_m^{p_1 p_2}$ of length N as

$$\vec{y}_m^{p_1 p_2} = (y_{N-1-m}^{p_1 *} y_{N-1}^{p_2}, \cdots, y_0^{p_1 *} y_m^{p_2}, 0, \cdots, 0) \quad (21)$$

for all m from 0 to K, all p_1 from 0 to M_{M}, and all p_2 from 0 to M_{M}. Likewise, we define sequence $\vec{\lambda}^{q_1 q_2}$ of length N as

$$\vec{\lambda}^{q_1 q_2} = (\lambda_{N-1}^{q_1 q_2}, \cdots, \lambda_0^{q_1 q_2}) \quad (22)$$

for all q_1 and q_2 from 0 to M_{S}. Since $\vec{y}_m^{p_1 p_2}$ depends only on the observed data, it can be calculated prior to parameter optimization. On the other hand, $\vec{\lambda}^{q_1 q_2}$ is calculated according to (16) by using the fixed separation matrices, source LPCs, and source PRPs every time regression coefficient vector $\boldsymbol{g}$ is to be updated in the iterative optimization process shown in Figure 2.

Now, let us look at how R and $\tilde{R}$ are built from the cross correlation coefficients between $\vec{y}_k^{p_1 p_2}$ and $\vec{\lambda}^{q_1 q_2}$. We focus on the submatrices, $R_{(m+1)1}, \cdots, R_{K(K-m)}$, on the $(m+1)$th

$$R_{mn} = \left[\begin{array}{ccc|c|ccc} \sum_{t=0}^{N-1} \lambda_t^{11} y_{t-m}^{1*} y_{t-n}^{1} & \cdots & \sum_{t=0}^{N-1} \lambda_t^{11} y_{t-m}^{1*} y_{t-n}^{M_{\mathrm{M}}} & & \sum_{t=0}^{N-1} \lambda_t^{1M_{\mathrm{S}}} y_{t-m}^{1*} y_{t-n}^{1} & \cdots & \sum_{t=0}^{N-1} \lambda_t^{1M_{\mathrm{S}}} y_{t-m}^{1*} y_{t-n}^{M_{\mathrm{M}}} \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \sum_{t=0}^{N-1} \lambda_t^{11} y_{t-m}^{M_{\mathrm{M}}*} y_{t-n}^{1} & \cdots & \sum_{t=0}^{N-1} \lambda_t^{11} y_{t-m}^{M_{\mathrm{M}}*} y_{t-n}^{M_{\mathrm{M}}} & & \sum_{t=0}^{N-1} \lambda_t^{1M_{\mathrm{S}}} y_{t-m}^{M_{\mathrm{M}}*} y_{t-n}^{1} & \cdots & \sum_{t=0}^{N-1} \lambda_t^{1M_{\mathrm{S}}} y_{t-m}^{M_{\mathrm{M}}*} y_{t-n}^{M_{\mathrm{M}}} \\ \hline \vdots & & \vdots & \ddots & \vdots & & \vdots \\ \hline \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}1} y_{t-m}^{1*} y_{t-n}^{1} & \cdots & \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}1} y_{t-m}^{1*} y_{t-n}^{M_{\mathrm{M}}} & & \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}M_{\mathrm{S}}} y_{t-m}^{1*} y_{t-n}^{1} & \cdots & \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}M_{\mathrm{S}}} y_{t-m}^{1*} y_{t-n}^{M_{\mathrm{M}}} \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}1} y_{t-m}^{M_{\mathrm{M}}*} y_{t-n}^{1} & \cdots & \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}1} y_{t-m}^{M_{\mathrm{M}}*} y_{t-n}^{M_{\mathrm{M}}} & & \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}M_{\mathrm{S}}} y_{t-m}^{M_{\mathrm{M}}*} y_{t-n}^{1} & \cdots & \sum_{t=0}^{N-1} \lambda_t^{M_{\mathrm{S}}M_{\mathrm{S}}} y_{t-m}^{M_{\mathrm{M}}*} y_{t-n}^{M_{\mathrm{M}}} \end{array}\right]$$

($M_{\mathrm{S}}M_{\mathrm{M}}$-by-$M_{\mathrm{S}}M_{\mathrm{M}}K_l$ matrix). (19)

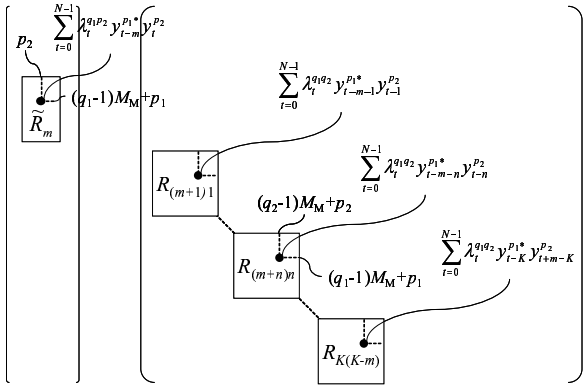

Figure 3: Structures of R and $\tilde{R}$. The left and right matrices are $\tilde{R}$ and R, respectively.

diagonal, where the first diagonal refers to the main diagonal. As shown in Figure 3, the $((q_1-1)M_{\mathrm{M}}+p_1,(q_2-1)M_{\mathrm{M}}+p_2)$th elements of these matrices consist of the cross correlation coeffcients between $\vec{y}_m^{p_1p_2}$ and $\vec{\lambda}^{q_1q_2}$ at lags 1 to $K-m$. Moreover, the $((q_1-1)M_{\mathrm{M}}+p_1,p_2)$th element of $\tilde{R}_m$ is the zeroth-lag cross correlation coeffcient between $\vec{y}_k^{p_1p_2}$ and $\vec{\lambda}^{q_1p_2}$. Therefore, the cross correlation coefficients are required at most for lags of 0 to K for any m. The parts in R and $\tilde{R}$ at which these cross correlation coefficients should be placed is obvious from Figure 3.

The above discussion leads to the following algorithm for calculating the modified correlation matrix, R, and the modified correlation vector, $\boldsymbol{r}$.

/* Prior to parameter optimization. */
for $p_1,p_2=0:M_{\mathrm{M}}, m=0:K$
  Calculate $\vec{y}_m^{p_1p_2}$ according to (21).
end
/* Whenever regression matrices are to be updated during parameter optimization.*/
for $q_1,q_2=1:M_{\mathrm{M}}$
  Calculate $\vec{\lambda}^{q_1q_2}$ according to (22).
  for $p_1,p_2=0:M_{\mathrm{M}}, m=0:K$
    Calculate the cross correlation coefficients between $\vec{y}_m^{p_1p_2}$ and $\vec{\lambda}^{q_1p_2}$ as

$$\vec{y}_m^{p_1p_2} \textcircled{K} \vec{\lambda}^{q_1q_2} = \Big(\sum_{t=0}^{N-1} y_{t-m}^{p_1*} y_t^{p_2} \lambda_t^{q_1q_2}, \cdots, \sum_{t=0}^{N-1} y_{t-K-m}^{p_1*} y_{t-K}^{p_2} \lambda_t^{q_1q_2} \Big),$$

    where Ⓚ denotes the cross correlation coefficients of two sequences at lags of 0 to K. Then, rearrange the cross correlation coefficients in R and $\tilde{R}$ according to Figure 3.
  end
end
Make the upper triangular part of R from the calculated lower triangular part.
Calculate $\boldsymbol{r}$ from $\tilde{R}$ according to (18).

We close this section by describing two points regarding the above algorithm.

- The cross correlation coefficients can be efficiently calculated by using an FFT. Because the proposed algorithm uses the cross correlation coefficients only at lags of 0 to K, the FFT point must be $K+N$ or larger. For our experimental system, we set the FFT point at the smallest power of two greater than or equal to $K+N$.
- The efficiency of the above algorithm can be further improved by capitalizing on the fact that the source power spectrum matrix ${}_{\mathrm{S}}\Lambda_t$ is diagonal. Indeed, Λ_t, defined by (16), is expressed as

$$\Lambda_t = \sum_{r=1}^{M_{\mathrm{S}}} \frac{\boldsymbol{w}_r \boldsymbol{w}_r^H}{{}_{\mathrm{S}}\lambda_t^r}, \tag{23}$$

  where $\boldsymbol{w}_r$ is the rth column of W and ${}_{\mathrm{S}}\lambda_t^r$ is the rth diagonal element of ${}_{\mathrm{S}}\Lambda_t$ (see Equation (9)). Therefore, instead of directly calculating $\vec{y}_m^{p_1p_2} \textcircled{K} \vec{\lambda}^{q_1q_2}$, we may calculate $\vec{y}_m^{p_1p_2} \textcircled{K}\, {}_{\mathrm{S}}\vec{\lambda}^r$, where

$${}_{\mathrm{S}}\vec{\lambda}^r = (1/{}_{\mathrm{S}}\lambda_{N-1}^r, \cdots, 1/{}_{\mathrm{S}}\lambda_0^r), \tag{24}$$

  and then obtain $\vec{y}_m^{p_1p_2} \textcircled{K} \vec{\lambda}^{q_1q_2}$ based on the following re-

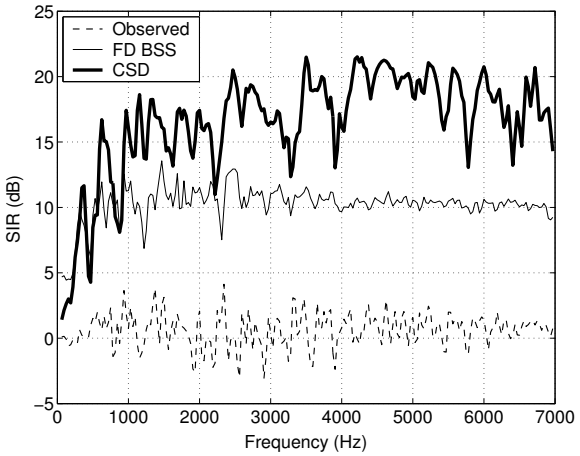

Figure 4: Average SIR as a function of frequency.

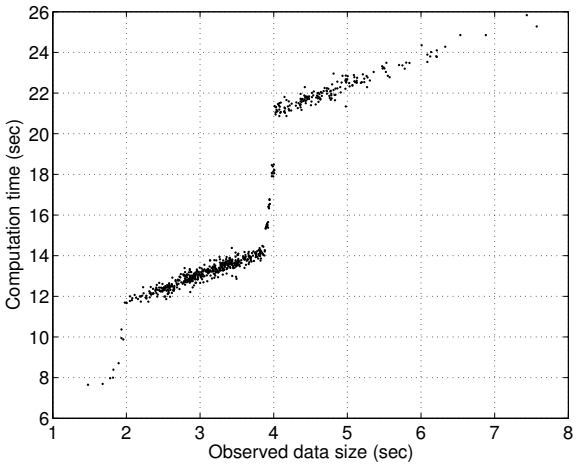

Figure 5: Computation time plotted against observed data size.

lation:

$$\vec{y}_m^{p_1 p_2} \textcircled{K} \vec{\lambda}^{q_1 q_2} = \sum_{r=1}^{M_S} w_{rq_1} w_{rq_2}^* (\vec{y}_m^{p_1 p_2} \textcircled{K}_S \vec{\lambda}^r). \qquad (25)$$

4. EXPERIMENTAL RESULTS

We conducted an experiment to evaluate the performance of the CSD method and the computation efficiency of the algorithm described above. The experiment assumed that we observed mixtures of two sounds with four microphones. We used the TIMIT complete test set, which includes 112 male speakers, 56 female speakers, 624 texts, and 1344 utterances. These utterances were band-limited to a frequency range of 50 Hz to 7 kHz. We formed 672 pairs of utterances. Then, for each pair, the acoustic signals of the two utterances were convolutively mixed by using room impulse responses measured in a room with a reverberation time of 0.5 sec. Thus, we had a total of 672 test samples.

The proposed and reference algorithms were implemented as MATLAB programs, which were run on a PC equipped with a 2.4-GHz quad core processor. The frame size and frame shift were set at 256 points (16 msec) and 128 points (8 msec), respectively. The number of poles, P, was set at 20. The regression orders, K_l, were set depending on frequency as follows: $K_l = 25$ for $f_l < 800$; $K_l = 20$ for $800 \leq f_l < 1500$; $K_l = 15$ for $1500 \leq f_l < 3000$; $K_l = 10$ for $f_l \geq 3000$, where f_l is the lth bin's frequency in Hz. The regression matrices, separation matrices, source LPCs, and source PRPs were updated two times.

Figure 4 shows the input and output signal-to-interference ratio (SIR) curves against frequency averaged over all the test samples. The SIR curve obtained by the frequency-domain BSS method described in [5] is also shown for reference. We can see that the CSD method greatly improved the average SIR over the whole frequency range. Moreover, the degree of improvement obtained with the CSD method was much higher than that obtained with the frequency-domain BSS method.

Figure 5 shows the computation time as a function of the observed data size. We can see that the computation time curve is discontinuous at about 2 and 4 secs. This is because the FFT point for calculating the cross correlation coefficients became a higher power of two at these data sizes. The real time factor (RTF) of the proposed algorithm was between 4 and 6, which was less than one tenth of the RTF of the naive algorithm. The RTF is expected to be further decreased by employing implementation techniques that use the multiple cores more effectively.

5. CONCLUSION

This paper described a fast algorithm for calculating modified correlation matrices, which are required by the CSD method for BSSD [4]. The proposed algorithm takes advantage of the particular structure of the modified correlation matrices shown in Figure 3. Experimental results obtained using the TIMIT complete test set revealed the efficiency of the proposed algorithm as well as the great advantage of the CSD method over the frequency-domain BSS method.

REFERENCES

[1] J. Benesty, S. Makino, and J. Chen, *Speech enhancement*. Springer, 2005.

[2] E. A. P. Habets, "Multi-channel speech dereverberation based on a statistical model of late reverberation," in *Proc. Int'l Conf. Acoust. Speech, Signal Process.*, vol. IV, 2005, pp. 173–176.

[3] S. Araki, R. Mukai, S. Makino, T. Nishikawa, and H. Saruwatari, "The fundamental limitation of frequency domain blind source separation for convolutive mixtures of speech," *IEEE Trans. Speech, Audio Process.*, vol. 11, no. 2, pp. 109–116, 2003.

[4] T. Yoshioka, T. Nakatani, and M. Miyoshi, "An integrated method for blind separation and dereverberation of convolutive audio mixtures," in *Proc. Eur. Signal Process. Conf.*, 2008, CD-ROM Proceedings.

[5] H. Sawada, S. Araki, and S. Makino, "Measuring dependence of bin-wise separated signals for permutation alignment in frequency-domain BSS," in *Int'l Symp. Circ., Syst.*, 2007, pp. 3247–3250.