EVALUATION AND REAL-TIME IMPLEMENTATION OF BLIND SOURCE SEPARATION SYSTEM USING TIME-DELAYED DECORRELATION

Futoshi Asano

Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, Japan
asano@etl.go.jp

Shiro Ikeda

RIKEN,
Hirosawa 2-1, Wako-shi,
Saitama, 351-0198 Japan
Shiro.Ikeda@brain.riken.go.jp

ABSTRACT

Blind source separation based on the time-delayed decorrelation, which had been extended to a convolved mixture problem, is evaluated and implemented in a DSP system. The evaluation is conducted using the data recorded in an anechoic chamber and those in a listening room with moderate reverberation. The crosstalks and the scores of automatic speech recognition are measured. The hardware consists of dual DSPs and a host PC. The results of the benchmark is shown.

1. INTRODUCTION

Currently, blind source separation (BSS) is intensively being studied for various applications (e.g., [1].) BSS for a convolved mixture such as overlapping of voices by multiple talkers in an acoustical environment known as cocktail party problem is one of the challenging areas [2, 3, 4, 5]. A method of BSS based on the time-delayed decorrelation (TDD) had been extended to a convolved mixture of acoustical signals [6]. TDD had originally been developed for an instantaneous mixture problem [7, 8, 9]. In [6], the multi-channel input signal is transformed into the frequency domain by DFT, and, by doing this, the instantaneous TDD can be applied to a mixed signal at each frequency independently. Then, the permutation problem (swapping of output channels) is solved by maximizing the inter-frequency correlation, and the final separated output is obtained.

In this paper, the evaluation and the real-time implementation of the extended TDD is discussed. The evaluation is conducted using the data recorded in an anechoic chamber and a listening room with reverberation time of 0.2 s. The crosstalks and the scores of automatic speech recognition are measured. The hardware consists of dual DSPs (Texas Instruments,TMS-320C6701, 150MHz) and the host PC (Pentium III, 600MHz). The results of the benchmark are shown.

2. METHOD

In this section, TDD extended to a convolved mixture problem is briefly explained.

2.1. Time-delayed correlation

The multi-channel input signal is first transformed into the frequency-domain by DFT. Let us denote the DFT of the mth microphone at the time frame t as $X_m(t,f)$. The input vector at the time frame t is then defined as

$$\mathbf{x}(t,f) = [X_1(t,f), \cdots, X_M(t,f)]^T, \tag{1}$$

where $\cdot^T$ denotes the transpose. For the sake of simplicity in notation, the frequency index f is omitted hereafter except where it is necessary. Using this input vector, the time-delayed correlation (TDC) is defined as

$$\mathbf{R}(\tau) = E[\mathbf{x}(t)\mathbf{x}^H(t-\tau)] \quad \text{for } \tau = 0, \cdots, L, \tag{2}$$

where $\cdot^H$ denotes the conjugate transpose. The symbol, τ, denotes the time delay expressed in the number of frames, and corresponds to the actual time delay of nT_s where T_s denotes the time interval between the frames in calculating DFT. TDC can be estimated iteratively in a real-time system as

$$\mathbf{R}(\tau,t) = (1-\alpha)\mathbf{R}(\tau,t-1) + \alpha\mathbf{x}(t)\mathbf{x}^H(t-\tau). \tag{3}$$

where α denotes the forgetting factor.

2.2. Separation filter

Suppose that a complex input signal $\mathbf{s}(t) = [S_1(t), \cdots, S_N(t)]^T$ is mixed by a complex mixing matrix $\mathbf{A}$ as

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{4}$$

The separation system is written as

$$\mathbf{y}(t) = \mathbf{B}\mathbf{x}(t). \tag{5}$$

The separation filter $\mathbf{B}$ can be decomposed into

$$\mathbf{B} = \mathbf{U}^H \mathbf{W}. \tag{6}$$

The matrix $\mathbf{W}$ is determined by the principle component analysis (PCA) using $\mathbf{R}(0)$ as

$$\mathbf{W} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{E}^H. \tag{7}$$

where $\mathbf{\Lambda}$ and $\mathbf{E}$ denote the eigenvalue matrix and the eigenvector matrix of $\mathbf{R}(0)$, respectively, i.e., $\mathbf{R}(0) = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^{-1}$. This stage is termed sphering.

The matrix $\mathbf{U}$ is determined by the joint-diagonalization problem[8, 9]:

$$\mathbf{U} = \arg\ \min_{\mathbf{U}} \sum_{\tau=1}^{L} \sum_{i \neq j} |(\mathbf{U}\bar{\boldsymbol{R}}(\tau)\mathbf{U}^T)_{ij}|^2 \tag{8}$$

where

$$\bar{\boldsymbol{R}}(\tau) = \mathbf{W}\mathbf{R}(\tau)\mathbf{W}^H. \tag{9}$$

This stage is termed rotation.

2.3. Permutation and Compensation of Frequency Spectrum Distortion

In usual instantaneous BSS, arbitrary permutation and scaling of the output is allowed. However, in the frequency-domain processing for a convolved mixture, different permutation at different frequencies leads to mixing of signal in the final output. Also different scaling in different frequencies leads to distortion of the frequency spectrum of the output signal.

The permutation problem is solved by swapping the output channels so that the inter-frequency correlation of the envelop of the output $\mathbf{y}$ is maximized. This is realized by the following iteration for all frequency f:

$$\hat{P}(f) = \arg\ \max_{P(f)} \sum_{t=1}^{T} \sum_{j=1}^{f-1} [P(f)\bar{\boldsymbol{y}}(t,f)]^H \bar{\boldsymbol{y}}(t,j) \tag{10}$$

where $P(f)$ denotes the permutation matrix. The symbol, $\bar{\boldsymbol{y}}(t)$, denotes the envelop of $\mathbf{y}(t)$. The symbol, T, denotes the number of spectrum used in solving permutation.

The scaling problem can be solved by filtering each component of the output $\mathbf{y}(t)$ by the inverse of $\mathbf{B}$ separately:

$$\tilde{\boldsymbol{y}}_n(t) = \begin{pmatrix} B_{1,n}^{-1} \\ \cdots \\ B_{M,n}^{-1} \end{pmatrix} y_n(t) \tag{11}$$

where $(B_{1,n}^{-1}, \cdots, B_{M,n}^{-1})^T$ denotes the nth column of the inverse of $\mathbf{B}$. The symbol $y_n(t)$ denotes the nth component of $\mathbf{y}$. The output $\tilde{\boldsymbol{y}}_n(t)$ corresponds to the estimate of the signal that is emitted by the nth source and is observed at the microphones.

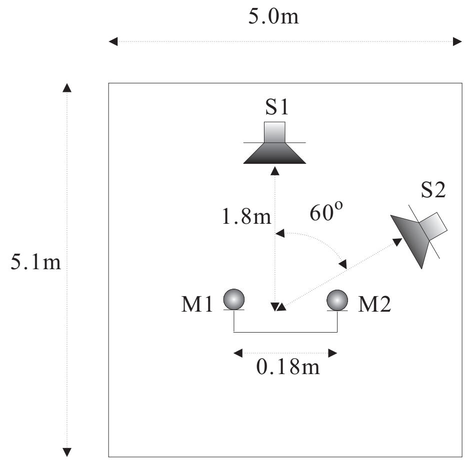

Figure 1: Configuration of sources and microphones in the evaluation experiment

Table 1: Parameters of the system.

Sampling Frequency	16 kHz
Number of Microphones M	2
Length of DFT	256
Frame Shift T_s	32
Number of TDCs L	32
Number of spectra for solving premutation T	512
Window for DFT	hamming
Forgetting Factor α	1/1024

3. EVALUATION

3.1. Condition

The evaluation experiment was conducted in an anechoic chamber and a listening room. The configuration of the sound sources (speakers) and the microphones in the listening room is depicted in Fig. 1. The reverberation time of the listening room was $T_{60} = 0.2$ [s]. The room has a carpet floor and cloth-coated walls. The configuration in the anechoic chamber is the same as that of the listening room except the wall location and the room size.

The BSS system used in this experiment was implemented in a workstation. The parameters of the system, that is listed in Table 1, is the same as that of the real-time system described in the next section, since this is a preliminary evaluation of the final real-time system.

Table 2: ASR rate.

	Anechoic Chamber		Listening Room	
	(a) Perm. Solved	(b) Perm. Given	(c) Perm. Solved	(d) Perm. Given
Ch.1	39.6%	58.7%	16.7%	25.2
Ch.2	40.0%	58.1%	14.4%	20.9

3.2. Results

In the evaluation, crosstalk and the score of the automatic speech recognition (ASR) were measured.

The crosstalk is defined as the power ratio of each output signal when only one of the sound sources is active. For measuring the crosstalk, the filter coefficients were fixed after learning with two sound sources active, then one of the two sources was deactivated and the power ratio of the cross channel and the straight channel was measured. The input signals were pairs of Japanese words. The number of pairs of words was 492. The word duration was 1-2 [s]. Figure 2(a) shows the measured crosstalk for 492 words in an anechoic chamber, shown in descending order of the performance. In the best case, 20 [dB] of crosstalk cancellation, that is sufficient for a practical use, was achieved. Figure 2(b) shows the crosstalk when the permutation is solved by matching the output spectrogram with the correct spectrogram of the sources that is unknown in a real operation. This was done to evaluate the performance of solving permutation with the algorithm described in Section 2.3. Comparing (a) and (b), the performance was reduced to some extent by employing the algorithm described in Section 2.3. Figure 2(c) and (d) shows the results for the listening room. The performance was reduced to around half of that obtained in the anechoic chamber.

Table 2 shows the score of ASR. The input was the same as that in the crosstalk evaluation. As a speech recognizer, HTK [10] was employed. In the anechoic chamber, the ASR score was around 40 %. Comparing (a) in which the permutation was solved with (b) in which the correct premutation was given, the score was reduced by around 20 %. This is due to the error in solving permutation. From the results obtained in the listening room shown in (c) and (d), it can be seen that the performance was reduced to less than half of that obtained in the anechoic chamber.

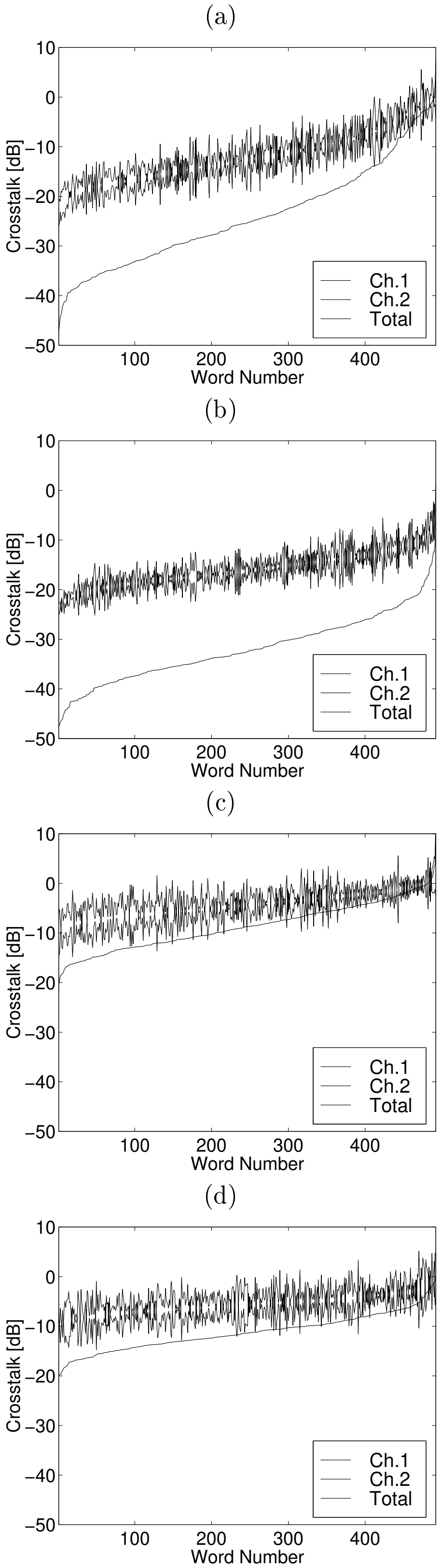

Figure 2: Crosstalk cancellation. (a):Anechoic chamber, permutation solved; (b):Anechoic chamber, permutation given; (c):Listening room, permutation solved; (d):Listening room, permutation given;

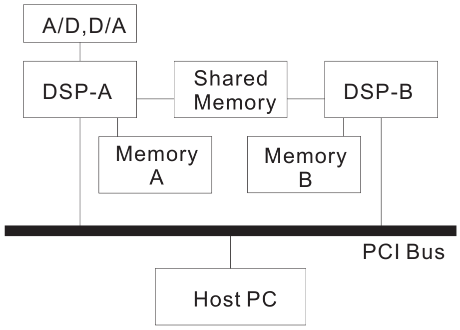

Figure 3: Hardware overview of the real-time system.

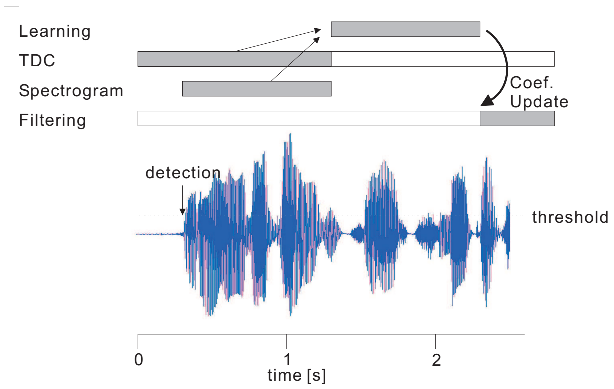

Figure 4: Timing chart of each module.

4. REAL-TIME SYSTEM

The architecture of the real-time system is depicted in Fig. 3. The system consists of dual DSPs (Texas Instruments, TMS320C6701, 150MHz) and a host PC (Pentium III, 600MHz). Each DSP has its local RAM (16 Mbytes.) For communication between the two DSPs, the system has a shared RAM (16 Mbytes.) DSP-A covers the real-time processing such as calculating the spectrogram and TDC, and filtering. TDC is always kept updated with (3). The spectrogram data are stored in a buffer, and when the sufficient spectrogram data for solving permutation is obtained, TDC and the spectrogram data are sent to the the learning module that calculates the separation filter. Currently, spectrogram data of around 1 s (512 input vectors) are used for solving permutation.

The learning module can be placed either in DSP-B or the host CPU. Table 3 shows the results of the benchmark of the learning module. When using the host CPU, the calculation time of the learning module is 1.02 s for single-word-duration data (1.17s). Therefore, in terms of the hardware performance, the system can process data in real time. However, since the system employs batch processing for calculating the separation filter, the system reflects the update of the filter coefficients with 1s of delay as depicted in Fig. 4. When the learning is conducted in DSP-B, the calculation time is about 4 times larger than that of the host CPU. This is mainly due to the large data and program size of the learning module. Also, a complicated algorithm of the learning module prevents easy optimization of the DSP code. However, assembly-level optimization of the code for DSP-B could improve the execution time dramatically, that may enable us to build up the stand-alone system.

Table 3: Results of the benchmark for calculating the separation filter.

Module	Pentium III, 600MHz [s]	C67, 150MHz [s]
Sphering	0.09	0.24
Rotation	0.03	0.31
Permutation	0.80	3.49
Filter Coef.	0.10	0.02
Total	1.02	4.06
Word Duration	1.17	

5. DISCUSSIONS AND SUMMARY

In this paper, the blind source separation based on the extended TDD was evaluated and implemented in a DSP system.

The evaluation was conducted as a preliminary one with the system implemented in a workstation, the parameters of which were the same as that in the real-time system. The results for the anechoic chamber was satisfactory to some extent. However, for some pairs of words, no crosstalk cancellation was achieved. The reason is not clear at this stage. One of the reason might be related to the ergodicity of the signals. During the estimation of TDC, the input signals are assumed to be ergodic. This assumption holds in a practical sense when vowel portions of the inputs overlaps. In some pairs of words, this assumption might not hold even in a practical sense. However, further analysis of data is required.

From the evaluation of solving permutation, it was found that the performance was reduced to tome extent due to the error of solving permutation using the inter-frequency correlation of spectrogram. The reason for this is mainly considered to be that there were pair of input signals which has similar envelops. In the current system, the permutation is solved using the envelop of the spectrogram for around 1 s, due to the restriction in memory and execution time. However, by employing longer spectrogram for solving permutation, the probability of having similar envelop will decrease. This issue should be considered in the future work.

In the real environment with reflections, the performance of separation was reduced to around half of that in the anechoic chamber. One of the main reason is considered to be the length of DFT, or equivalently the length of the separation filter. It was reported that longer filter length is required for reflective fields for deconvolving the reflections [1]. In this report, the length of DFT was also limited by the resource of the real time system. This issue must also be treated in the future.

In the real time system, calculation of the spectrogram, TDC, and the filtering is done in real time by a single DSP. The calculation of the separation filter is conducted in another CPU, which can either be the other DSP or the CPU in the host PC. Currently, the processing is 4 times faster by using the CPU in the host PC than by using another DSP. The results of the benchmark shows that, by employing the host CPU for learning module, the system can track the environmental change with 1 s of delay.

6. REFERENCES

[1] T.-W. Lee, *Independent Component Analysis*, Kluwer Academic Publishers, Boston, 1998.

[2] N. Murata, S. Ikeda, and A. Ziehe, "An Approach to Blind Source Separation Based on Temporal Structure of Speech Signals,", submitted to Neurocomputing, BSIS Tech. Report, http://www.bsis.brain.riken.go.jp/, 1998.

[3] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22(1-3), pp. 21–34, 1998.

[4] M. Kawamoto, K. Matsuoka, and N. Ohnishi, "A method of blind separation for convolved nonstationary signals," *Neurocomputing*, vol. 22(1-3), pp. 157–171, 1998.

[5] S. C. Douglas and A. Cichocki, "Neural Networks for Blind Decorrelation of Signals," *IEEE Trans. Signal Processing*, vol. 45(11), pp. 2829–2842, nov 1997.

[6] S. Ikeda and N. Murata, "A method of blind separation based on temporal structure of signals," In *Proceedings of The Fifth International Conference on Neural Information Processing (ICONIP'98 Kitakyushu)*, pp. 737–742, 1998.

[7] L. Tong, R.-W. Liu, V. C. Soon, and Y.-F. Huang, "Indeterminacy and identifiability of blind identification," *IEEE Trans. Circuits and Systems*, vol. 38, pp. 499–509, May 1991.

[8] L. Molgedey and H. G. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Physical Review Letters*, vol. 72(23), pp. 3634–3637, 1994.

[9] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Process*, vol. 45(2), pp. 434–443, February 1997.

[10] http://www.entropic.com/.