A METHOD OF BLIND SEPARATION ON TEMPORAL STRUCTURE OF SIGNALS

Shiro Ikeda and Noboru Murata
Email:{Shiro.Ikeda,Noboru.Murata}@brain.riken.go.jp

RIKEN Brain Science Institute
Hirosawa 2-1, Wako, Saitama 351-0198, Japan

ABSTRACT

In this article, we propose an Blind Source Separation algorithm for convolutive mixture of signals. We propose a method of separating signals in the time-frequency domain. We apply the decorrelation method proposed by Molgedey and Schuster on spectrogram and reconstruct separated signals focusing on the temporal structure of the signals. We show some results of experiments with both artificially controlled data and speech data recorded in the real environment.

KEYWORDS: Convolutive mixtures, Windowed Fourier Transform

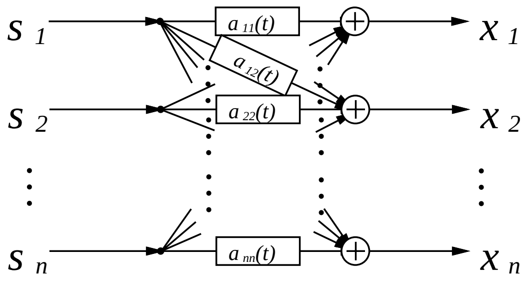

Figure 1: The problem: Convolutive Mixtures

1. Introduction

In this paper, we propose a blind source separation (BSS) method for speech signals recorded in a real environment. Speech signals have a temporal structure that it is stationary for a short period but not stationary for a long term[6]. We use this time structure to build an algorithm.

The problem of BSS[2, 5] is defined as follows. Source signals are denoted by a vector $\boldsymbol{s}(t) = (s_1(t), \cdots, s_n(t))^T$ and it is assumed that each component of $\boldsymbol{s}(t)$ is independent to each other and mean 0. Recorded signals are $\boldsymbol{x}(t) = (x_1(t), \cdots, x_n(t))^T$, where they are instantaneous mixture which can be written with an $n \times n$ real matrix A as,

$$\boldsymbol{x}(t) = A\boldsymbol{s}(t).$$

But for the recorded speech signals, we usually simulate a real-room recording with FIR filters, s.t. the observations are convolutive mixtures of source signals,

$$\begin{aligned} \boldsymbol{x}(t) = A * \boldsymbol{s}(t) &= \left(\sum_k a_{ik} * s_k(t) \right), \\ a_{ik} * s_k(t) &= \sum_{\tau=0}^{\tau_{max}} a_{ik}(\tau) s_k(t-\tau), \end{aligned} \tag{1}$$

where $A(t)$ is a function of time, $a_{ik} * s_k(t)$ is the convolution of $a_{ik}(t)$ and $s_k(t)$, where $a_{ik}(t)$ is the impulse response from source signal k to sensor i. The goal of BSS is to separate signals into the components which are mutually independent without knowing operator A and source signals $\boldsymbol{s}(t)$.

Basic BSS approaches have been developed for instantaneous mixtures. For convolutive mixtures, there are some trials[3]. We use the windowed-Fourier transform, which is known as the name spectrogram, as in [7, 9] to transform mixed source signals into the time-frequency domain. After that we apply Molgedey and Schuster's decorrelation algorithm[8] to the signals of each frequency independently. Most of the BSS approaches usually ignore the ambiguities of the amplitude and the permutation, but we have to remove these ambiguities to reconstruct the separated signals. Our idea is to use the inverse of the decorrelating matrices and the envelope of the speech signal.

2. Decorrelation Algorithm for Instantaneous Mixture

First we explain the decorrelation algorithm by Molgedey and Schuster [8] which was proposed for instantaneous mixtures, i.e. $\boldsymbol{x}(t) = A\boldsymbol{s}(t)$ where A is an $n \times n$ matrix. The goal is to find a matrix B which is equivalent to the inverse matrix of A with the ambiguity of amplitude and permutation.

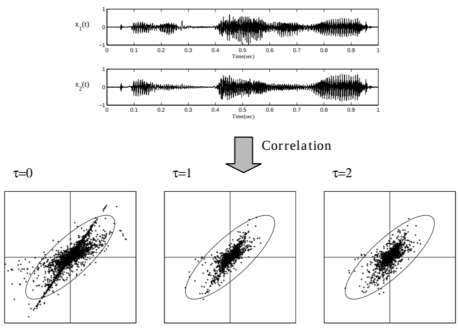

Figure 2: Correlation of original inputs: On the top, the original inputs are shown. On the bottom side, these signals are plotted with different time delays

The correlation matrix of observations is written as

$$\begin{aligned} &\left\langle \boldsymbol{x}(t)\boldsymbol{x}(t+\tau)^T \right\rangle = R_{xx}(\tau) \\ =&A \left\langle \boldsymbol{s}(t)\boldsymbol{s}(t+\tau)^T \right\rangle A^T = AR_{ss}(\tau)A^T, \end{aligned} \tag{2}$$

where $R_{xx}(\tau)$ and $R_{ss}(\tau)$ are correlation matrices. Since each component of $\boldsymbol{s}(t)$ is independent, $R_{ss}(\tau)$ is diagonal for any

τ. Molgedey and Schuster showed that the BSS problem of finding B is reduced to solve the eigenvalue problem

$$\left(R_{xx}(\tau_1)R_{xx}(\tau_2)^{-1}\right) B = B\left(\Lambda_1 \Lambda_2^{-1}\right). \tag{3}$$

This problem can also be solved by simultaneous diagonalization of matrices, where the number of the matrices doesn't have to be 2 but any number,

$$BR_{xx}(\tau_i)B^T = \Lambda_i, \quad i = 1, \ldots, r. \tag{4}$$

Although from the effect of the noise and small correlations among the source signals, (4) does not hold in practice. We implemented in the way to minimize the off-diagonal components of the matrices $BR_{xx}(\tau_i)B^T$. In order to obtain B, we use the algorithm which only needs straightforward calculations[10]. It consists of two procedures, sphering and rotation(Fig.3).

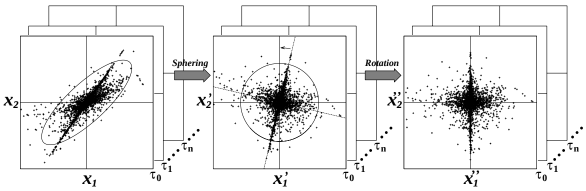

Figure 3: Decorrelation algorithm

Sphering is a procedure to obtain a matrix V which satisfies,

$$VR_{xx}(0)V^T = I. \tag{5}$$

This procedure corresponds to Principal Component Analysis(PCA). Figure 4 shows the signals after sphering.

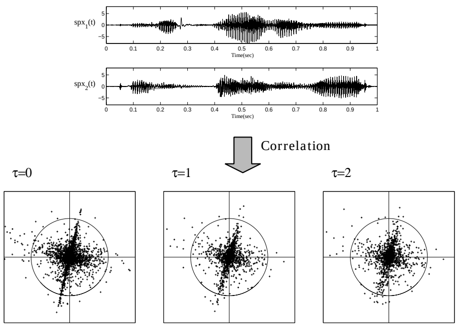

Figure 4: Correlations of the signals after sphering

Rotation is a procedure to remove off-diagonal elements of correlation matrices with an orthogonal transformation. This can be realized by an orthogonal matrix C which minimizes

$$\sum_{l=1}^{r}\sum_{i\neq k}\left|(CVR_{xx}(\tau_l)V^TC^T)_{ik}\right|^2, \tag{6}$$

where $(*)_{ik}$ is the ik-element of a matrix. Cardoso and Souloumiac gave an implementation [1] with Jacobi-like algorithm to obtain C. Finally, matrix B is given by $B = CV$. The decorrelated signals are shown in Fig.5. An advantage of this method is that it uses only the second order statistics and fixed amount of computation.

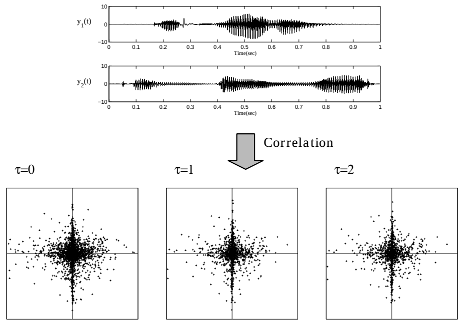

Figure 5: Correlations of the outputs

3. Proposed Method

In this section, the detail of the algorithm is shown along with the flow of it. First, the windowed-Fourier transform is applied to convolutive mixed signals,

$$\begin{aligned}\hat{\boldsymbol{x}}(\omega, t_s) &= \sum_t e^{-j\omega t}\boldsymbol{x}(t)w(t-t_s), \\ \omega &= 0, \tfrac{1}{N}2\pi, \ldots, \tfrac{N-1}{N}2\pi, \quad t_s = 0, \Delta T, 2\Delta T, \ldots\end{aligned} \tag{7}$$

where ω denotes the frequency and N denotes the number of points of the discrete Fourier transform, t_s denotes the window position, w is a window function (we used Hamming window) and ΔT is the shifting interval of moving windows. Let us redefine an $\hat{\boldsymbol{x}}(\omega, t_s)$ for a fixed frequency ω as $\hat{\boldsymbol{x}}_\omega(t_s) = \hat{\boldsymbol{x}}(\omega, t_s)$. If the window length is long enough compared to the impulse response of $A(t)$, the relationship between observations and sources can be approximated as,

$$\hat{\boldsymbol{x}}_\omega(t_s) = \hat{A}(\omega)\hat{\boldsymbol{s}}(\omega, t_s),$$

where $\hat{A}(\omega)$ is the Fourier transform of operator $A(t)$, and $\hat{\boldsymbol{s}}(\omega, t_s)$ is the windowed-Fourier transform of $\boldsymbol{s}(t)$. This shows that for fixed ω, a convolutive mixture is simply an instantaneous mixture. We extend the algorithm in the last section to complex values by substituting a Hermite matrix and a unitary matrix for a symmetric matrix and an orthogonal matrix respectively, and apply it for each frequency. As a result, we have a separated time sequence for each frequency,

$$\hat{\boldsymbol{u}}_\omega(t_s) = B(\omega)\hat{\boldsymbol{x}}_\omega(t_s).$$

Since BSS algorithms cannot solve the ambiguity of amplitude and permutation, even if we put each component of $\hat{\boldsymbol{u}}_\omega(t_s)$ along with ω, amplitudes are irregular and different independent sources will be mixed up. The problem of irregular amplitude can be solved by putting back the separated independent components to the sensor input with the inverse matrices $B(\omega)^{-1}$. Let us define $\hat{\boldsymbol{v}}_\omega(t_s; i)$ as,

$$\hat{\boldsymbol{v}}_\omega(t_s; i) = B(\omega)^{-1}(0\ldots0, \hat{u}_{i,\omega}(t_s), 0\ldots0)^T, \quad i = 1, \ldots, n$$

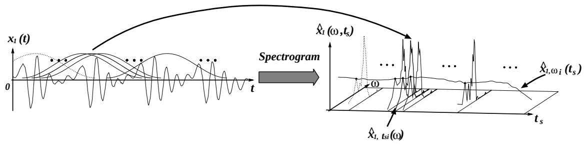

Figure 6: Windowed-Fourier Transform (spectrogram)

where $\hat{v}_{k,\omega}(t_s; i)$ represents the input of i-th independent component of $\hat{\boldsymbol{u}}_\omega(t_s)$ into the k-th ($k = 1, \ldots, n$) sensor. We applied $B(\omega)$ and $B(\omega)^{-1}$ to obtain $\hat{\boldsymbol{v}}_\omega(t_s; i)$, therefore $\hat{\boldsymbol{v}}_\omega(t_s; i)$ has no ambiguity of amplitude.

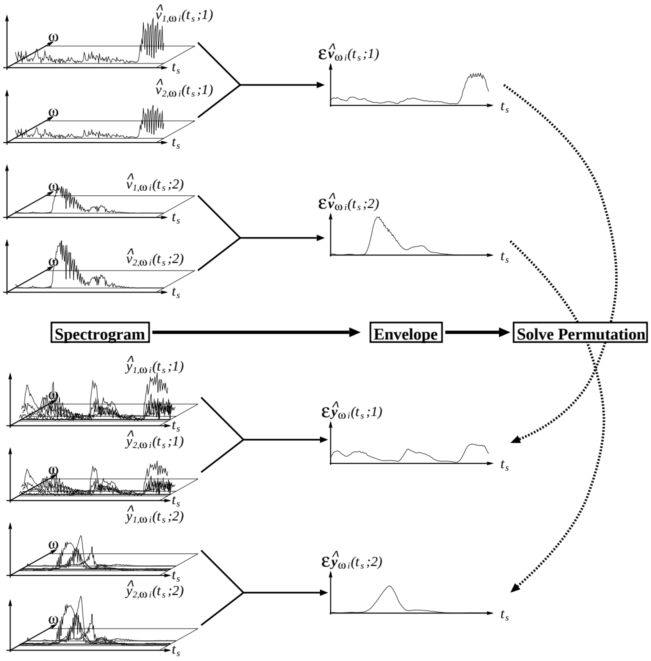

Figure 7: Solving the permutation ambiguity

Remaining problem is permutation. We assumed that even for different frequencies, if the original source is the same, the envelopes are similar, and utilize this idea for solving the permutation. $\mathcal{E}$ is an operator to take the envelope as,

$$\mathcal{E}\hat{\boldsymbol{v}}_\omega(t_s; i) = \frac{1}{2M} \sum_{t'_s = t_s - M}^{t_s + M} \sum_{k=1}^{n} |\hat{v}_{k,\omega}(t'_s; i)|, \tag{8}$$

where M is a positive constant and $\hat{v}_{k,\omega}(t_s; i)$ denotes the k-th element of $\hat{\boldsymbol{v}}_\omega(t_s; i)$. Inner product and norm are defined as

$$\mathcal{E}\hat{\boldsymbol{v}}_\omega(i) \cdot \mathcal{E}\hat{\boldsymbol{v}}_{\omega'}(k) = \sum_{t_s} \mathcal{E}\hat{\boldsymbol{v}}_\omega(t_s; i)\mathcal{E}\hat{\boldsymbol{v}}_{\omega'}(t_s; k), \tag{9}$$

$$\|\mathcal{E}\boldsymbol{v}_\omega(i)\| = \sqrt{\mathcal{E}\hat{\boldsymbol{v}}_\omega(i) \cdot \mathcal{E}\hat{\boldsymbol{v}}_\omega(i)}, \tag{10}$$

and we define the similarity among all the envelopes in the same frequencies by,

$$\mathrm{sim}(\omega) = \sum_{i \neq k} \frac{\mathcal{E}\hat{\boldsymbol{v}}_\omega(i) \cdot \mathcal{E}\hat{\boldsymbol{v}}_\omega(k)}{\|\mathcal{E}\hat{\boldsymbol{v}}_\omega(i)\| \|\mathcal{E}\hat{\boldsymbol{v}}_\omega(k)\|}. \tag{11}$$

With these operations, the permutation is solved (see Fig.7):

- Sort ω in order of the weakness of correlation between independent components in ω. This is done by sorting in increasing order of $\mathrm{sim}(\boldsymbol{\omega})$ as,

$$\mathrm{sim}(\omega_1) \leq \mathrm{sim}(\omega_2) \leq \cdots \leq \mathrm{sim}(\omega_N). \tag{12}$$

- For ω_1, assign $\hat{\boldsymbol{v}}_{\omega_1}(t_s; i)$ to $\hat{\boldsymbol{y}}_{\omega_1}(t_s; i)$ as it is:

$$\hat{\boldsymbol{y}}_{\omega_1}(t_s; i) = \hat{\boldsymbol{v}}_{\omega_1}(t_s; i), i = 1, \ldots, n \tag{13}$$

- For ω_k, find the permutation $\sigma(i)$ which maximizes the correlation between the envelope of ω_k and the aggregated envelope from ω_1 through ω_{k-1}. This is achieved by maximizing

$$\sum_{i=1}^{n} \mathcal{E}\hat{\boldsymbol{v}}_{\omega_k}(\sigma(i)) \cdot \left(\sum_{j=1}^{k-1} \mathcal{E}\hat{\boldsymbol{y}}_{\omega_j}(i) \right) \tag{14}$$

within all the possible permutations σ of $i = 1, \ldots, n$.

- Assign the appropriate permutation to $\hat{\boldsymbol{y}}_{\omega_k}(t_s; i)$:

$$\hat{\boldsymbol{y}}_{\omega_k}(t_s; i) = \hat{\boldsymbol{v}}_{\omega_k}(t_s; \sigma(i)). \tag{15}$$

As a result, we obtain separated spectrograms as $\hat{\boldsymbol{y}}_\omega(t_s; i)$. Applying inverse Fourier transform, finally we get a set of separated sources

$$\boldsymbol{y}(t; i) = \frac{1}{2\pi} \cdot \frac{1}{W(t)} \sum_{t_s} \sum_{\omega} e^{j\omega(t - t_s)} \hat{\boldsymbol{y}}_\omega(t_s; i), \quad i = 1, \ldots, n$$

where $W(t) = \sum_{t_s} w(t - t_s)$. Note that each $y_k(t; i)$ represents a separated independent component i on sensor k, and $\sum_i \boldsymbol{y}(t; i) = \boldsymbol{x}(t)$ holds. And finally we obtain $n \times n$ signals from n dimensional inputs.

4. Experimental Results

4.1. Artificial Data

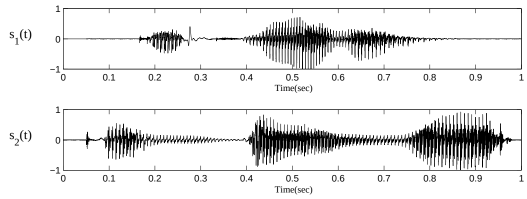

Figure 8: The source signals: each signal was spoken by a different male and recorded with sampling rate of 16kHz. $s_1(t)$ is a recorded word of “good morning” and $s_2(t)$ is a Japanese word “konbanwa” which means “good evening”.

There are some parameters in the proposed algorithm, and we have to define appropriate values for them. We used a set of data which were recorded separately and mixed on a computer, and tested source separation with different parameters. The parameters to be set are:

$\varDelta T$: Shifting time of the window function (in (7)).

r : Number of the matrices used for the simultaneous diagonalization (in (4)).

M : The number of the time steps for taking the moving average (in (8)).

WindowLength : The length of the window function used in windowed-Fourier Transform.

$\varDelta T$, r, and M are mutually related. Speech signals can be regarded as stational for a several 10msecs, but not station if it is longer than 100msec. Therefore, $\varDelta T \times r$ should stay within a several 10msecs. Also the time for taking the moving average, which is defined as $\varDelta T \times (2M + 1)$ should also be around a few 10msecs.

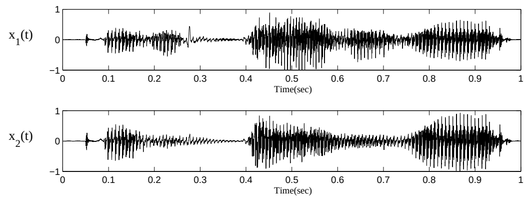

Figure 9: The instantaneous mixture

a) Instantaneous Mixtures Figure 8 shows the sources which were recorded separately on the computer. First, we made instantaneous mixtures with these source signals with a matrix A as follows. The mixed signals are shown in Fig.9.

$$\boldsymbol{x}(t) = A\boldsymbol{s}(t) = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \boldsymbol{s}(t) = \begin{pmatrix} 1 & 0.7 \\ 0.3 & 1 \end{pmatrix} \boldsymbol{s}(t),$$

Since we know the true sources and the mixing rates, we can evaluate the performance using the Signal to Noise Ratio (SNR) which is defined as

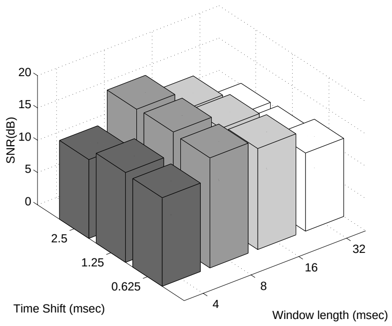

Figure 10: The SNR_{11} for instantaneous mixtures: the window lengths and ΔT were varied, r (the number of the matrices for simultaneous diagonalization) was fixed to 30.

$$\begin{aligned} \mathrm{signal}_i(t;j) &= a_{ij}s_j(t) && (16) \\ \mathrm{error}_i(t;j) &= y_i(t;j) - \mathrm{signal}_i(t;j) && (17) \\ \mathrm{SNR}_{ij} &= 10\log_{10}\frac{\sum_t \mathrm{signal}_i(t;j)^2}{\sum_t \mathrm{error}_i(t;j)^2}. && (18) \end{aligned}$$

First, we tried to choose the right values of the window length and ΔT. The SNR_{ij}'s are measured by changing the window length from 4msec to 32msec and ΔT from 0.625msec to 2.5msec. We used the Hamming window for the window function. Results of SNR_{11} is shown in Fig.10. It is clear from the graph that the window length with 8msec gave results better than the others and we confirmed this fact for other ij and r with experiments not shown here. Therefore, we defined window length as 8msec for this experiment.

We still have to define ΔT and r. Theoretically, r can be 2 or any larger number. However, small r gives an unstable solution, and large r leads to a wrong solution because time difference between correlation matrices will be too larger for the stationarity of speech signals. We changed ΔT and r and calculated SNR_{ij}. Figure 11 shows the result of SNR_{11} with changing ΔT from 0.625msec to 2.5msec and r from 2 to 70. There is a peak on each row. $\Delta T \times r$ is the interval of

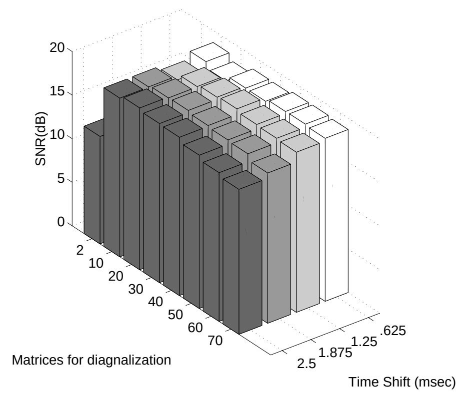

Figure 11: The SNR_{11} for instantaneous mixtures: the window lengths was fixed to 8msec, ΔT and r were varied.

time within which the matrices are diagonalized. This value should not go beyond the stationarity of the speech signals. We can see there is a peak between 30 and 50msec. It is said that speech signals are stationary around 40msec, and this matches the result we obtained here. This feature is also true for other ij's. We can see that the combination of ΔT = 1.25msec and $r = 40$ is the best. The position of the peaks are almost the same for other ij's, and we decided to use these values for ΔT and r. SNR_{11} for this combination is 18.9 (crosstalk is 1/77.6).

The parameter M in (8) is used to make the moving average of signals to make the envelopes. From some results, we found that if the time for taking the moving average is longer than 20msec, permutation ambiguity was solved properly, and we used $\Delta T \times (2M+1)$ to be around 40msec. Because ΔT was defined as 1.25msec, M was set to be 15, Finally, we obtained the separated signals which is shown in Fig.12.Crosstalk is small and it is hard to see them.

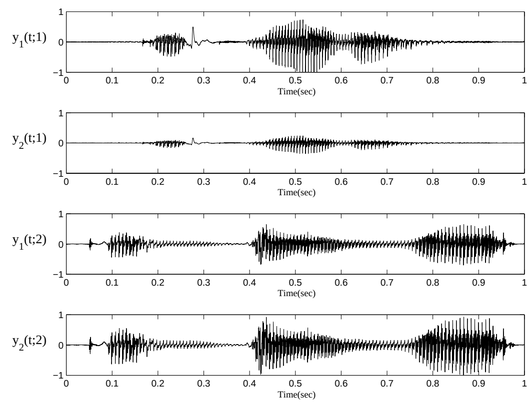

Figure 12: The separated signals using the proposed algorithm: the window length was 8msec, ΔT = 1.25msec and $r = 40$.

b) Separating convolutive mixtures Our main aim of this paper is not to separate instantaneous mixtures, but to separate convolutive mixtures as in (1). We also made convolutive mixture signals on the computer and used these signals for experiment to set the parameters and to assess how our

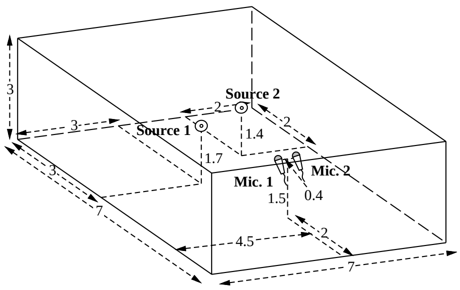

Figure 13: Virtual room for making convolutive mixtures: unit for the length is meter, and the sonic speed is 340m/sec. The strength of the reflection is 0.1 in power for any frequency, and the strength of sounds varies in proportion to the inverse square of the distance. Because the second reflection is small, only the first reflection are taken into accounted.

algorithm works.

We wanted to simulate the general problem of recording sounds in a real environment. In order to simulate it, we built a virtual room as Fig.13 and calculated reflections and delays. In Fig.14, the impulse response from source 1 to microphone 2 is shown. Also we show the window function with different lengths in the graph. The impulse response is rather long. We have to set the window length longer than the impulse response. But if we make it long, the SNR will be worse. There is a trade-off between the window length and SNR for the convolutive mixture. The source signals are the same as Fig.8. The convolutive mixtures in this virtual room are shown in Fig.15.

For the separation, we applied our algorithm, changing the window length from 8msec to 32msec, and evaluated the result with the SNRs. In this case, (16) was modified as,

$$\mathrm{signal}_i(t;j) \quad = \quad a_{ij}(t) * s_j(t). \tag{19}$$

The SNRs of these results are shown in Tab.1. Our approach with the window length of 32msec gave the best SNRs. Separated signals are shown in Fig.16(window length was 32msec, ΔT was 1.25msec, r was 40.).

4.2. Real-room Recorded Data

The proposed algorithm is applied to the data recorded in a real environment. The data was provided by Prof. Kota Takahashi in the University of Electro-Communications. Two males were repeating different phrases simultaneously in a room and their voices were recorded with two micro-

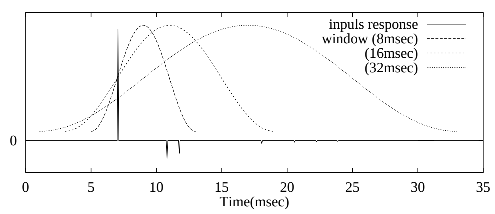

Figure 14: The impulse response from the source 1 to the microphone 2 in a virtual room

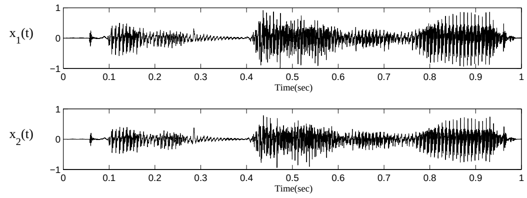

Figure 15: The mixed signals in a virtual room

phones with 44.1kHz for 5sec then down-sampled to 16kHz. Inputs are shown in Fig.17. Window length was 32msec (512 points), ΔT was 1.25msec and r was 40. The result is shown in Fig.18. We heard them and they were separated clearly.

5. Discussion and Conclusion

We proposed a blind source separation algorithm based on the temporal structure of speech signals. Our algorithm only uses straightforward calculations, and it includes only a few parameter to be tuned. This is possible because of the short range and long range temporal structure of natural acoustic signals. On the experiments, the algorithm worked very well for the artificial data and the real-room-recorded data.

Blind source separation of convolutive mixtures is a problem of estimating a filter matrix from sources to each sensors. In our algorithm, we use the parameterization of a filter matrix in the frequency domain (cf. [4, 9]) as

$$A(\omega) = \sum_{k=0}^{K} A_k \delta(\omega - \omega_k) \quad \delta(\omega - \omega_k) = \left\{ \begin{array}{ll} 0 & \omega \neq \omega_k \\ 1 & \omega = \omega \end{array} \right. . \tag{20}$$

A_k is a matrix estimated for each frequency independently. This is the reason we can build our algorithm with only straightforward calculations. In order to separate the signals, we need the inverse filter, and we can estimate the inverse process easily by calculating A_k^{-1}. A problem with this parameterization occurs when one of the source signals does not have any frequency component on a frequency ω_k. In our algorithm, we cannot estimate A_k^{-1} since A_k is a singular matrix. We have to treat these cases separately.

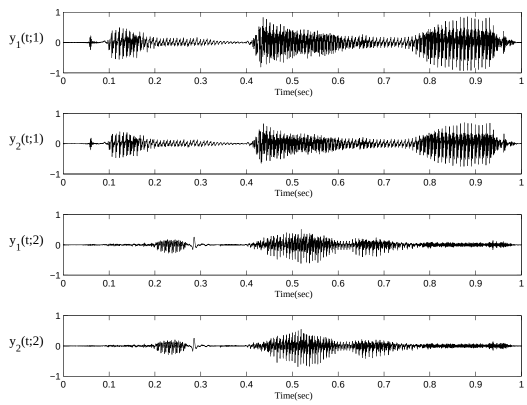

Figure 16: The separated signals: the proposed algorithm was applied with the window length of 32msec, ΔT = 1.25msec and $r = 40$.

Table 1: SNRs (dB) for Separated Signals (ΔT is 1.25msec and r in (4) is 40)

		SNR_{11}	SNR_{12}	SNR_{21}	SNR_{22}
Window length	8msec	4.36	6.32	11.94	11.66
	16msec	4.72	6.65	12.66	12.52
	32msec	6.47	7.30	14.40	13.19

Another major approach for convolutive mixtures is to parameterize the impulse response from each source to each sensor with an FIR filter. This approach estimates the parameters of the filters and build inverse filters to separate the signals (e.g. [3]). The impulse responses from sources to sensors are defined in matrix form,

$$A_{\mathrm{FIR}}(t) = \sum_{l=0}^{L} A_{\mathrm{FIR}l}\,\delta(t - t_l), \tag{21}$$

where $A_{\mathrm{FIR}l}$ is a matrix which corresponds to the t_l-time delayed component of the mixing process. The Fourier transform of the filter is defined as

$$\widehat{A}_{\mathrm{FIR}}(\omega) = \sum_{l=0}^{L} A_{\mathrm{FIR}_l} e^{\sqrt{-1}\omega t_l}. \tag{22}$$

One of the advantages to this parameterization is that some sort of continuity across different frequencies can be included naturally because the Fourier transform $\widehat{A}_{\mathrm{FIR}}(\omega)$ for any ω depends on all the mixing matrices $\widehat{A}_{\mathrm{FIR}_l}$'s. A disadvantage of the approach is that, in order to estimate the parameters $\widehat{A}_{\mathrm{FIR}_l}$, usually some kind of iterative procedures is necessary, and after estimating $\widehat{A}_{\mathrm{FIR}_l}$, it needs inverse filters to separate the signals. The inverse filters are defined as

$$\widehat{A}^{-1}_{\mathrm{FIR}}(\omega) = \left(\sum_{l=0}^{L} A_{\mathrm{FIR}_l} e^{\sqrt{-1}\omega t_l} \right)^{-1} \tag{23}$$

This inverse filter generally doesn't have finite impulse response and we also have to be careful with their causality. And, we cannot calculate this for each frequency independently.

There are some problems to be solved in our algorithm. We have three major parameters, the window length, the shift ΔT and the number of correlation matrices r. We showed that window length can be defined independent of the other two parameters, but if the mixing process has a long impulse response, window length has to be correspondingly longer, and it will make the performance worse because it will go beyond the stationary range of the source signals. There is another problem, that is, the sampling rate. We only used a sampling rate of 16kHz. From the sampling theorem, it follows that the data includes signals whose frequency component is below 8kHz. Usually, speech signals have some power for every component under 8kHz. Our algorithm applies the decorrelation algorithm for every frequency component, but if even one component doesn't have any power, the decorrelation algorithm fails. Therefore, if we use 44.1kHz for the sampling rate of speech signals, there will be a lot of components which cannot be separated correctly. We need some other technique to solve this problem.

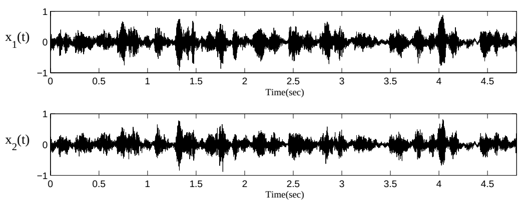

Figure 17: Recorded Signals in a Real Room

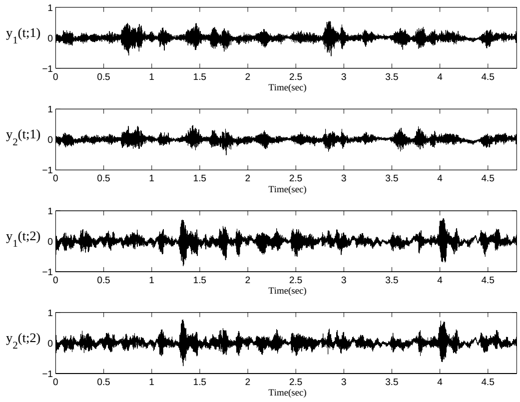

Figure 18: Separated Signals

References

[1] Jean-François Cardoso and Antoine Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM J. Mat. Anal. Appl.*, 17(1):161–164, jan 1996.

[2] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, apr 1994.

[3] Scott C. Douglas and Andrzej Cichocki. Neural networks for blind decorrelation of signals. *IEEE Trans. Signal Processing*, 45(11):2829–2842, nov 1997.

[4] F. Ehlers and H.G. Schuster. Blind separation of convolitive mixtures and an application in automatic speech recognition in noisy environment. *IEEE Transactions on Signal Processing*, 45(10):2608–2609, 1997.

[5] Christian Jutten and Jeanny Herault. Separation of sources, part i. *Signal Processing*, 24(1):1–10, jul 1991.

[6] Hideki Kawahara and Toshio Irino. Exploring temporal feature representations of speech using neural networks. Technical Report SP88-31, IEICE, Tokyo, 1988. (in Japanese).

[7] Te-Won Lee, Andreas Ziehe, Reinhold Orglmeister, and Terrence Sejnowski. Combining time-delayed decorrelation and ICA: towards solving the cocktail party problem. In *Proceedings of ICASSP'98*, 1998.

[8] L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72(23):3634–3637, 1994.

[9] Paris Smaragdis. Blind separation of convolved mixtures in the frequency domain. In *International Workshop on Independence & Artificial Neural Networks*, University of La Laguna, Tenerife, Spain, feb 1998.

[10] Andreas Ziehe. Statistische verfahren zur signalquellentrennung. Master's thesis, Humboldt Universität, Berlin, 1998. (in German).