An Approach to Blind Source Separation Based on Temporal Structure of Speech Signals

Noboru Murata, Shiro Ikeda
Lab. for Information Synthesis
RIKEN Brain Science Institute
mura,shiro@brain.riken.go.jp

Andreas Ziehe
GMD FIRST, Berlin
ziehe@first.gmd.de

Abstract

In this paper we introduce a new technique for blind source separation of speech signals. We focus on the temporal structure of the signals in contrast to most other major approaches to this problem. The idea is to apply the decorrelation method proposed by Molgedey and Schuster in the time-frequency domain. We show some results of experiments with both artificially controlled data and speech data recorded in the real environment.

1 Introduction

Recently, blind source separation, or BSS, within the framework of independent component analysis has attracted a great deal of attention in engineering field. It has been widely noticed that there are many possible applications such as removing additive noise from signals and images, separating crosstalk in telecommunication, preprocessing for multi-probed radar-sonar signals, and analyzing EEG (Electroencephalograph) or MEG (Magnetoencephalograph) data (see for example [12]).

Blind source separation is the problem to separate independent sources given a mixed signal where the mixing process is unknown. We want to extract each source from the mixed signals using some technique. Even if the mixing process is unknown, we can separate the sources if they are independent to each other.

The major approaches of blind source separation use higher order statistics but not the temporal structure of input signals. These algorithms also need iterative calculations for estimating the source signals because in most cases, they require non-linear optimization. On the other hand, Molgedey and Schuster have shown that it is possible to separate signals by using second order statistics as a form of correlation function. Their approach does need neither higher order statistics, nor iterative calculations.

In this paper, we propose a blind source separation method for temporally structured signals, in particular speech signals. Blind source separation of speech signals is often called the "cocktail party effect problem". This name comes from the fact that we can hold a conversation at a cocktail party even though we are surrounded by loud voices and boisterous music. Speech signals have a temporal structure that can be regarded as stationary for short time-scales, although for longer time-scales, it is non-stationary. We can build an algorithm which uses this temporal structure.

The difficulty of separating recorded speech signals is due to the delays and reflections of the real environment. Those mixed signals are not instantaneous mixtures but convolutive mixtures. We solve the problem of this convolution by applying a windowed Fourier transform. The time signals are transformed to time-frequency signals, and we apply Molgedey and Schuster's decorrelation algorithm to the signals of each frequency components. Molgedey and Schuster's decorrelation algorithm cannot solve the ambiguity of permutation and this can be a big problem in our approach when we reconstruct the time-frequency signal into separated time signals. We solve this ambiguity by using the time structure of the speech signals. In particular, we use the envelope of each frequency signal to group the sources.

Our algorithm has some advantages for hardware implementation, because the calculation is considerably straightforward, it includes only a few parameter to be tuned, and it is easy for parallel processing.

This paper is organized as follows: in section 2, we describe some basic approaches to blind source separation of instantaneous mixed signals. In section 3, we propose an algorithm for blind source separation of convolutive mixtures and, in section 4 some results of our algorithm will be shown. Finally, we give a brief summary and concluding remarks in section 5.

2 Blind Source Separation Problem

In this section, we give a formulation of the basic problem of blind source separation and describe major approaches.

Source signals are denoted by a vector

$$\boldsymbol{s}(t) = (s_1(t), \cdots, s_n(t))^T, \quad t = 0, 1, 2, \dots \tag{1}$$

and there is the assumption that each component of $\boldsymbol{s}(t)$ is independent of each other. But it is difficult to define the independence of the sources accurately, and we use the following notion of pairwise independence: at any time, the joint distribution of any two different signals can be factorized by their marginals

$$p(s_i(t), s_j(t')) = p(s_i(t))p(s_j(t')), \quad \forall t, t', i, j (i \neq j). \tag{2}$$

Without loss of generality, we assume the source signal $\boldsymbol{s}(t)$ to be zero mean.

Observations are represented by

$$\boldsymbol{x}(t) = (x_1(t), \cdots, x_n(t))^T. \tag{3}$$

They correspond to the recorded signals. In the basic blind source separation problem, we assume that observations are linear mixtures of source signals:

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

where A is an unknown linear operator. A typical example of linear operators is an $n \times n$ real valued matrix. This formulation represents non-delayed (instantaneous) linear mixing, such as MEG measurements. Another example is a matrix of FIR filters, which is commonly used as a model of real-room recording. In this paper we wish to focus on the latter case; we describe the problem more fully in the next section.

The goal of blind source separation is to find a linear operator B such that the components of the reconstructed signals

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

are mutually independent, without knowing operator A and the probability distribution of source signal $\boldsymbol{s}(t)$. Ideally we expect B to be the inverse of operator A, but since we lack information about the amplitude of the source signals and their order, there remains indefiniteness of permutation and dilation factors:

$$BA = PD, \tag{6}$$

where P is a permutation matrix, i.e. all the elements of each column and row are 0 except for one element with value 1, and D is a diagonal matrix.

Popular approaches of blind source separation are based on the following three strategies:

- factorizing the joint probability density of the reconstructed signals by its marginals,
- decorrelating the reconstructed signals through time, that is, diagonalizing the covariance matrices at every time, and
- eliminating the cross-correlation functions of the reconstructed signals as much as possible.

In the rest of this section, we will survey these strategies individually.

2.1 Factorizing Joint Probability

Under the assumption that the source signals are stationary and non-Gaussian, the independence of the reconstructed signals can be measured by a statistical distance between the joint distribution

$$p(y_1, \cdots, y_n) \tag{7}$$

and the product of its marginals

$$\prod_{i=1}^{n} p(y_i). \tag{8}$$

The Kullback-Leibler divergence is often used as the statistical distance:

$$
\begin{aligned}
KL(B) &= \int p(\boldsymbol{y}) \log \frac{p(\boldsymbol{y})}{\prod_{i=1}^{n} p(y_i)} d\boldsymbol{y} && (9) \\
&= -H(\boldsymbol{Y}; B) + \sum_{i=1}^{n} H(Y_i; B), && (10)
\end{aligned}
$$

where $H(\boldsymbol{Y}; B)$ is the entropy of the joint distribution $p(\boldsymbol{y})$ and $H(Y_i; B)$ is the entropy of marginal distribution $p(y_i)$, both of which are calculated from the distribution of observation $p(\boldsymbol{x})$ and de-mixing operator B in this case. In particular, when operator B is a matrix, the entropy of the joint may be simply written as

$$H(\boldsymbol{Y}; B) = H(\boldsymbol{X}) + \log |B|, \tag{11}$$

where $|B|$ is the determinant of matrix B. Note that from the assumption of stationarity, the distributions $p(\boldsymbol{s})$, $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are time independent. As we assume that the sources are not Gaussian, $KL(B)$ vanishes if and only if reconstructed signals $\boldsymbol{y}(t)$ are mutually independent. Basically, we can estimate B by minimizing $KL(B)$, but the important problem which remains with this approach is how to approximate the entropy term $H(Y_i; B)$. Since it is impossible to calculate $H(Y_i; B)$ directly, these values are usually estimated in the following manner: first, a stationarity and a sort of ergodicity are assumed on the source signals, and the ensemble average (phase average) is replaced by the time average over observations, then parametric families of nonlinear functions and statistical expansions are used to estimate the entropies or the derivatives of the entropies with respect to the operator B. Jutten and Herault [9] first proposed the former approach by adopting polynomials. However their discussion was not from this point of view. Bell and Sejnowski [3] used sigmoidal functions. Common [6] studied the latter approach, and proposed an algorithm based on the Edgeworth expansion. Amari et al. [2] used the Gram-Charlier expansion. There are a lot of variations depending on how one approximates entropies.

We should note that ergodicity is usually not explicitly mentioned in the literature. However, in order to estimate probability densities only from observations, we need some sort of ergodic assumption. Strict ergodicity might be too strong for the blind source separation problem, but we need an assumption that allows us to use the time average instead of the ensemble average,

$$\langle f(x(t)) \rangle \sim \frac{1}{T} \sum_{t=1}^{T} f(x(t)), \tag{12}$$

where $f(x)$ is a certain function and $\langle \cdot \rangle$ denotes the expectation under probability distribution of $x(t)$. Therefore, there is a problem when this approach is applied to non-stationary signals, because of the poor estimation of probability densities. Theoretically, it is impossible to estimate the probability distribution of non-stationary stochastic process from one observation, because the probability distribution changes at each time, but we hope there still is a way to estimate the demixing matrix B. Some sketchy studies on the problem have been done along with on-line learning under the assumption of "weak" non-stationarity, such as slow drifting and stepwise change from one stationarity process to another.

Another problem is that this approach is not robust to noise. Consider the case that the observations are noisy,

$$\boldsymbol{x}(t) = A\boldsymbol{s}(t) + \boldsymbol{\xi}(t). \tag{13}$$

In this case, the divergence $KL(B)$ won't be 0 for any B and $B = A^{-1}$ does not give the minimum of $KL(B)$ in general. Hence estimation based on the divergence cannot avoid crosstalk. This problem should be reformulated from the perspective of factor analysis with the EM algorithm and so on (Amari, personal communication).

2.2 Decorrelating through Time

If some signals are mutually independent, off-diagonal elements of the covariance matrix vanish. Although the reverse of this statement is not always true, if the signals are non-stationary we can utilize the covariance to estimate the demixing matrix B. Matsuoka et al. [13] proposed an algorithm to estimate the matrix B by minimizing

$$Q(B;t) = \frac{1}{2}\left\{\mathrm{tr}\left(\log\left\langle \boldsymbol{y}(t)\boldsymbol{y}(t)^T\right\rangle\right) - \log\left(\det\left\langle \boldsymbol{y}(t)\boldsymbol{y}(t)^T\right\rangle\right)\right\}, \tag{14}$$

where $\langle\cdot\rangle$ denotes the expectation at each time, and log operates on the matrix component-wise. For fixed t, there does not exist a unique B which minimize $Q(B;t)$ and we define a subset of the domain of B:

$$\mathcal{B}_t = \left\{B_*; Q(B_*;t) = \min_B Q(B;t)\right\}. \tag{15}$$

In other words, there is ambiguity when determining B from $Q(B;t)$ at each time t. However, subsets at different time t's are basically different because of non-stationarity, and as a result, a proper B is determined as an intersection of all subsets, i.e.

$$B \in \bigcap_t \mathcal{B}_t. \tag{16}$$

Whether B can be determined as one point or not depends on the non-stationarity of the signals, and if signals are strongly or weakly stationary, the ambiguity of B does not disappear.

An advantage of this method is that it only uses the second order statistics. Generally speaking, estimation of higher order statistics is easily influenced by outliers, and therefore some of the previous methods, which rely on higher order statistics, are prone to errors in noisy conditions. On the other hand, because of the ambiguity mentioned above, good estimation is not guaranteed for stationary signals. Also, in practice the ensemble average is replaced by the running average (leaky average)

$$\langle f(t)\rangle_{\text{leaky}} = (1-\varepsilon)\langle f(t-1)\rangle_{\text{leaky}} + \varepsilon f(t), \tag{17}$$

where ε is a small positive number. Since the running average is an operation based on stationarity, and we need a proper assumption that source signals have both properties of "stationarity" and "non-stationarity" to validate this operation.

2.3 Eliminating Cross-correlation

The following method is what we use in our experiments. We are going to explain the original idea of the method in this subsection.

Let us assume that the sources are weakly stationary signals and observations are instantaneous mixtures, i.e. A is a constant matrix. The correlation matrix of observations is written as

$$\begin{aligned}\left\langle \boldsymbol{x}(t)\boldsymbol{x}(t+\tau)^T\right\rangle &= A\left\langle \boldsymbol{s}(t)\boldsymbol{s}(t+\tau)^T\right\rangle A^T \\ &= A\begin{pmatrix} R_{s_1}(\tau) & 0 & \dots & 0 \\ 0 & R_{s_2}(\tau) & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & R_{s_n}(\tau)\end{pmatrix} A^T,\end{aligned} \tag{18}$$

where $R_{s_i}(\tau)$ is the auto-correlation function of the source signal $s_i(t)$. As a proper matrix B satisfies Equation (6), the correlation matrix of the reconstructed signals becomes

$$\begin{aligned}\left\langle \boldsymbol{y}(t)\boldsymbol{y}(t+\tau)^T\right\rangle &= \left\langle (PD\boldsymbol{s}(t))\left(PD\boldsymbol{s}(t+\tau)\right)^T\right\rangle \\ &= \begin{pmatrix} \lambda_{1'}^2 R_{s_{1'}}(\tau) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_{n'}^2 R_{s_{n'}}(\tau)\end{pmatrix}\end{aligned} \tag{19}$$

where $1', 2', \dots, n'$ denotes a permutation of the indices $1, 2, \dots, n$ determined by matrix P and λ_i is the i-th diagonal element of matrix D. Including the ambiguity of matrices P and D, an optimal B can be characterized as a matrix which diagonalizes the correlation matrix at any time difference τ.

Molgedey and Schuster [14] simplified this concept as the simultaneous diagonalization of the correlation matrix of observations at several time delays, i.e. find B in a certain class of matrices, such that

$$B\langle \boldsymbol{x}(t)\boldsymbol{x}(t+\tau_i)^T\rangle B^T = \mathit{\Lambda}_i, \quad i = 1, \dots, r, \tag{20}$$

where $\mathit{\Lambda}_i$'s are diagonal matrices.

There are some algorithms for calculating the solution to Equation (20). For examples, Molgedey and Schuster proposed a gradient-based method which minimizes the error function

$$L(B) = \sum_{i=1}^{r} \sum_{j \neq k} \left| B \langle \boldsymbol{x}(t) \boldsymbol{x}(t+\tau_i)^T \rangle B^T \right|_{jk}^2 . \tag{21}$$

We are going to use only the straightforward calculation [17, 18]. The algorithm consists of two procedures, "sphering" and "rotation" (see Figure 1).

Sphering is an operation for orthogonalizing the source signals in the observing coordinate. Let us define a covariance matrix of observations

$$V = \left\langle \boldsymbol{x}(t) \boldsymbol{x}(t)^T \right\rangle , \tag{22}$$

and define its square root inverse

$$\sqrt{V^{-1}} = \sqrt{\Lambda^{-1}} S^T \tag{23}$$

where S and Λ are an orthogonal matrix and a diagonal matrix, which satisfy

$$V = S \Lambda S^T , \tag{24}$$

and $\sqrt{\Lambda^{-1}}$ denotes a diagonal matrix, each of whose elements are the square root of Λ^{-1}'s elements. Note that from weak stationarity, the covariance matrix is time independent. By transforming the observation vector as

$$\boldsymbol{x}'(t) = \sqrt{V^{-1}} \boldsymbol{x}(t), \tag{25}$$

the covariance matrix of the new vector $\boldsymbol{x}'(t)$ is orthogonalized, i.e.

$$\left\langle \boldsymbol{x}'(t) \boldsymbol{x}'(t)^T \right\rangle = \sqrt{V^{-1}} V \sqrt{V^{-1}}^T = I, \tag{26}$$

where I is the identity matrix. Intuitively speaking, an observation is thought as a projecting the source signals in a certain direction. Although the directions of the original observations are in general not orthogonal, sphering rearranges them to be orthogonal to each other.

Even after sphering observed signals, there is still an ambiguity of rotation. The correct rotation is determined by removing the off-diagonal elements of the correlation matrix at several time delays. A possible implementation is to find an orthogonal matrix C which minimizes

$$\sum_{k=1}^{r} \sum_{i \neq j} \left| (C M_k C^T)_{ij} \right|^2 , \tag{27}$$

where $(C M_k C^T)_{ij}$ denotes the ij-element of matrix $C M_k C^T$ and

$$M_k = \left\langle \boldsymbol{x}'(t) \boldsymbol{x}'(t+\tau_k)^T \right\rangle , \quad k = 1, \ldots, r. \tag{28}$$

To solve this approximate simultaneous diagonalization problem, Cardoso and Souloumiac [5] proposed a Jacobi-like algorithm, and we use their method in our implementation.

With these two operations, matrix B is given by

$$B = C\sqrt{V^{-1}}. \tag{29}$$

An advantage of this method is that it uses only the second order statistics. Moreover it could be applied even when the observations contain white noises. White noise can be avoided by using only the cross-correlations at $\tau \neq 0$. Basically this method is applied when the sources are weakly stationary signals with different auto-correlation. However, even when the signals are non-stationary, it is applicable if the non-stationarity is not strong and we can well approximate the averaged correlation functions by using the observations

$$\frac{1}{T}\sum_{t=0}^{T}\left\langle \boldsymbol{x}'(t)\boldsymbol{x}'(t+\tau_k)^T \right\rangle \sim \frac{1}{T}\sum_{t=0}^{T}\boldsymbol{x}'(t)\boldsymbol{x}'(t+\tau_k)^T \tag{30}$$

for appropriately chosen T and τ_k.

Molgedey and Schuster also mentioned an algorithm with directly solves the eigenvalue problem of the two correlation matrices

$$M_1^{-1}M_2B^T = B^T\Lambda_1^{-1}\Lambda_2, \tag{31}$$

but from a practical point of view, this procedure is sensitive to the estimation error of the correlation matrices, such as observation noise. Hence simultaneous diagonalization at several time delays is more feasible in practice.

3 Proposed Method on Convolutive Mixtures

We have shown some approaches to the blind source separation problem for basic instantaneous mixtures. In this section, we focus on convolutive mixtures of time-structured signals, such as speech signals, and we propose an algorithm for blind source separation. In this paper, we use the term, time-structured signal, as a model of a natural signal, which we characterize as follows:

- signals are supposed to be stationary within a short time-scale,
- signals are intrinsically non-stationary in a long range because of amplitude modulation and so on.

In the case of speech signals, it is said that the human voice is stationary for a period shorter than a few 10msecs[10]. If it is longer than a few 10msecs and around 100msec, the frequency components of the speech will change its structure. This means that the speech signal is not stationary.

Our idea is to transform the mixed signals to the time-frequency domain, which is typically called a spectrogram. After that, we perform blind source separation for each frequency. Finally, we reconstruct the separated signals from the spectrograms of the separated signals.

The first property allows us to apply the windowed Fourier transform at a short time range. The length of the time window should be determined with prior knowledge. We used the second property to properly combine the decomposed frequency components. Usually, blind source separation does not care about the permutation of the separated signals, but because we apply blind source separation to every frequency separately, we have to know how to combine them. From the assumptions it follows that there won't be a drastic change of the distributions of frequency components, and the spectrums are assumed to be continuous. Using this continuity, we can combine the separated frequency elements and construct the separated spectrograms.

We also assume that observations are time independent convolutive mixtures, i.e.

$$\boldsymbol{x}(t) = A * \boldsymbol{s}(t), \tag{32}$$

where each element of $A(t)$ is an unknown transfer function.

Relying on the first assumption of signals, we apply the Fourier transform with moving windows (see Figure 2)

$$\begin{gathered}\hat{\boldsymbol{x}}(\omega, t_s) = \sum_t e^{-\sqrt{-1}\omega t}\boldsymbol{x}(t)w(t-t_s), \\ \omega = 0, \tfrac{1}{N}2\pi, \ldots, \tfrac{N-1}{N}2\pi, \quad t_s = 0, \mathit{\Delta}T, 2\mathit{\Delta}T, \ldots\end{gathered} \tag{33}$$

where ω denotes the frequency and N denotes the number of points in the discrete Fourier transform, t_s denotes the window position, w is a window function, such as Hamming, Hanning or Kaiser, and $\mathit{\Delta}T$ is defined as the shifting time of the moving windows. The length of windows should be shorter than the duration within which signals are mostly stationary. Also the inversion of Equation (33) is defined by

$$\boldsymbol{x}(t) = \frac{1}{2\pi} \cdot \frac{1}{W(t)} \sum_{t_s} \sum_{\omega} e^{\sqrt{-1}\omega(t-t_s)} \hat{\boldsymbol{x}}(\omega, t_s), \tag{34}$$

where

$$W(t) = \sum_{t_s} w(t - t_s). \tag{35}$$

The relationship between observations and sources is approximated as

$$\hat{\boldsymbol{x}}(\omega, t_s) = \hat{A}(\omega)\hat{\boldsymbol{s}}(\omega, t_s), \tag{36}$$

where $\hat{A}(\omega)$ is the Fourier transform of the operator $A(t)$, and $\hat{\boldsymbol{s}}(\omega, t_s)$ is the windowed Fourier transform of the source $\boldsymbol{s}(t)$.

We now show the details of our algorithm. It is schematically shown in Figure 3.

If we fix the frequency as ω for spectrograms,

$$\hat{\boldsymbol{x}}_\omega(t_s) = \hat{\boldsymbol{x}}(\omega, t_s) \tag{37}$$

is a time series of t_s, and hence any blind source separation algorithm for non-delayed mixtures can be applied for estimating $\hat{A}(\omega)$. In our case, Molgedey-Schuster's method is adopted. We have to note that $\hat{\boldsymbol{x}}_\omega(t_s)$ is complex valued, and therefore the method in Section 2.3 should be extended to complex values. This can be easily done by substituting a Hermite matrix and a unitary matrix for a symmetric matrix and an orthogonal matrix respectively.

After the algorithm is applied, we have an estimated time sequence whose components are mutually independent for each frequency ω

$$\hat{\boldsymbol{u}}_\omega(t_s) = B(\omega)\hat{\boldsymbol{x}}_\omega(t_s). \tag{38}$$

Since a general blind source separation algorithm cannot solve the ambiguity of permutation and dilation, in each frequency channel, the signals are permutated and amplified independently. This means that, even if we put $\hat{\boldsymbol{u}}_\omega(t_s)$ along with ω, those spectrograms is mixed up with the different independent sources and the amplitude are irregular.

To solve the permutation and dilation problem, we disassemble the spectrograms exploiting the independent components at each frequency channel. Let us define split spectrograms by

$$\hat{\boldsymbol{v}}_\omega(t_s; i) = B(\omega)^{-1} \begin{pmatrix} 0 \\ \vdots \\ \hat{u}_{i,\omega}(t_s) \\ \vdots \\ 0 \end{pmatrix}, \tag{39}$$

where index i denotes the dependence of the spectrograms at ω on the i-th independent component of $\hat{\boldsymbol{u}}_\omega(t_s)$. Note that implicitly, i is a function of the frequency ω, i.e. $i = i(\omega)$. In order to obtain $\hat{\boldsymbol{v}}_\omega(t_s; i)$, we apply $B(\omega)$ and $B(\omega)^{-1}$, and therefore $\hat{\boldsymbol{v}}_\omega(t_s; i)$ does not have an ambiguity of dilation.

The remaining problem is permutation. When the signal is stationary, any Fourier components at different frequencies are uncorrelated, so that it is impossible to find an appropriate sorting. However, we can utilize the non-stationarity of the source signals. Based on the second assumption of the source signals, if the split band-passed signals $\hat{\boldsymbol{v}}_\omega(t_s; i)$ originate from the same source signal, it is natural to assume that they are under the influence of a similar modulation in amplitude. To put it concretely, we define an operator $\mathcal{E}$ as

$$\mathcal{E}\hat{\boldsymbol{v}}_\omega(t_s; i) = \frac{1}{2M} \sum_{t'_s = t_s - M}^{t_s + M} \sum_{j=1}^{n} |v_{j,\omega}(t'_s; i)|, \tag{40}$$

where M is a positive constant and $v_{j,\omega}(t_s; i)$ denotes the j-th element of $\hat{\boldsymbol{v}}_\omega(t_s; i)$. Also we define its inner product and norm as

$$\mathcal{E}\hat{\boldsymbol{v}}_\omega(i) \cdot \mathcal{E}\hat{\boldsymbol{v}}_{\omega'}(j) = \sum_{t_s} \mathcal{E}\hat{\boldsymbol{v}}_\omega(t_s; i)\mathcal{E}\hat{\boldsymbol{v}}_{\omega'}(t_s; j), \tag{41}$$

$$\|\mathcal{E}\boldsymbol{v}_\omega(i)\| = \sqrt{\mathcal{E}\hat{\boldsymbol{v}}_\omega(i) \cdot \mathcal{E}\hat{\boldsymbol{v}}_\omega(i)}. \tag{42}$$

We solve the permutation by sorting them. Sorting was determined with the correlation between the envelopes of band-passed signals (see Figure 4):

- Sort ω in order of the weakness of correlation between independent components in ω. This is done by sorting in increasing order of

$$\mathrm{sim}(\omega) = \sum_{i \neq j} \frac{\mathcal{E}\hat{\boldsymbol{v}}_\omega(i) \cdot \mathcal{E}\hat{\boldsymbol{v}}_\omega(j)}{\|\mathcal{E}\hat{\boldsymbol{v}}_\omega(i)\| \|\mathcal{E}\hat{\boldsymbol{v}}_\omega(j)\|}, \tag{43}$$

$$\mathrm{sim}(\omega_1) \leq \mathrm{sim}(\omega_2) \leq \cdots \leq \mathrm{sim}(\omega_N). \tag{44}$$

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

$$\hat{\boldsymbol{y}}_{\omega_1}(t_s; i) = \hat{\boldsymbol{v}}_{\omega_1}(t_s; i), i = 1, \ldots, n \tag{45}$$

- 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

$$\sum_{i=1}^{n} \mathcal{E}\hat{\boldsymbol{v}}_{\omega_k}(\sigma(i)) \cdot \left(\sum_{j=1}^{k-1} \mathcal{E}\hat{\boldsymbol{y}}_{\omega_j}(i) \right) \tag{46}$$

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

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

$$\hat{\boldsymbol{y}}_{\omega_k}(t_s; i) = \hat{\boldsymbol{v}}_{\omega_k}(t_s; \sigma(i)). \tag{47}$$

As a result, we can solve the permutation ambiguity and obtain separated spectrograms

$$\hat{\boldsymbol{y}}(\omega, t_s; i) = \hat{\boldsymbol{y}}_\omega(t_s; i). \tag{48}$$

Applying the inverse Fourier transform defined by Equation (34), we finally get a set of separated sources

$$\boldsymbol{y}(t; i) = \frac{1}{2\pi} \cdot \frac{1}{W(t)} \sum_{t_s} \sum_{\omega} e^{\sqrt{-1}\omega(t - t_s)} \hat{\boldsymbol{y}}(\omega, t_s; i), \quad i = 1, \ldots, n \tag{49}$$

where i denotes the index of independent components. Note that we have separated the sources $\boldsymbol{y}(t; i)$ which has the same dimension as $\boldsymbol{x}(t)$ where i is the number of sources. For example, if inputs are 2, the separated sources are 4 in our method. And also note that from its definition, $\sum_i \boldsymbol{y}(t; i) = \boldsymbol{x}(t)$.

4 Experimental Results

In this section, we show some results of the proposed algorithm. First, the sources are mixed on the computer and our algorithm was applied to those mixed data. Since the true sources were available, we can evaluate the performance of the algorithm. Also another result with the data recorded in the real environment will be shown. In this case, we cannot know the true sources. We show the result with graphs. The data is available on our web page:

```
http://www.islab.brain.riken.go.jp/~shiro/blindsep.html
```

4.1 Artificial Data

4.1.1 Separating instantaneous mixtures with the basic de-correlation approach

In this subsection, we show the result of an experiment on a set of data which was mixed instantaneous on a computer. Figure 5 shows the sources which were recorded separately on the computer.

We mixed these source signals without delay as in Equation (4) where the matrix A is shown below. The mixed signals are shown in Figure 6.

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

Since $\boldsymbol{x}(t)$ is not a convolutive mixture, we can apply the original technique in Subsection 2.3. The results are shown in Figure 7. We used 30 matrices for simultaneous diagonalization in the experiment. Since we know the true sources and the mixing rates, we can evaluate the performance using the SNR (Signal to Noise Ratio) which is defined as

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

SNR_{ij} for this experiment is shown in Table 1. Every SNR_{ij} is more than 27.0 which means crosstalk is less than 1/500.

4.1.2 Separating instantaneous mixtures with the proposed method

In this subsection, our algorithm described in section 3 was applied to the same problem in the last subsection.

In order to use our algorithm, we have to first define some parameters. Our algorithm needs to apply the windowed Fourier transform which was defined in Equation (33). There are two parameters for a windowed Fourier transform, one is the window length and the other is the shifting time ΔT. We also have another parameter r in Equation (28) which is the number of matrices to be diagonalized simultaneously. We made some preliminary experiments to set these parameters.

From some trials, we found that ΔT and r are strongly related, but that window length is relatively independent of these values. The SNR_{ij}'s are measured by changing the window length from 4msec to 32msec and ΔT from 0.0625msec to 2.5msec. We used the Hamming window for the window function. Results of SNR_{11} is shown in Figure 8. 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. Signals were all recorded with a sampling rate of 16kHz and 8msec corresponds to 128 points of samples.

This set the window length, but we still had 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 9 shows the result of SNR_{11} with changing ΔT from 0.0625msec to 2.5msec and r from 2 to 70. From this graph, we see that there is a peak on each row. To see this more clearly, we replot SNR_{11} versus $\Delta T \times r$. $\Delta T \times r$ is the interval of time within which the matrices are diagonalized. This value should not go beyond the stationarity of the speech signals. The result is shown in Figure 10. 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 $\Delta T = 1.25$msec 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 Equation (40) is used to make the moving average of signals to make the envelopes. We used $\Delta T \times (2M + 1)$ to be around 40msec. We defined M as 15, because ΔT was defined as 1.25msec.

Finally, we show the separated signals in Figure 11, and SNR's in Table 2. Crosstalk is small and it is hard to see them in the graph.

4.1.3 Separating convolutive mixtures

Our main aim of this paper is not to separate instantaneous mixtures, but to separate convolutive mixtures. We also made convolutive mixture signals on the computer and used these signals for experiment to assess how our algorithm works. As in Equation (32), a convolutive mixture is defined as,

$$\begin{aligned} \boldsymbol{x}(t) = A * \boldsymbol{s}(t) &= \begin{pmatrix} \sum_j a_{1j} * s_j(t) \\ \sum_j a_{2j} * s_j(t) \end{pmatrix} \\ a_{ij} * s_j(t) &= \sum_{\tau=0}^{\infty} a_{ij}(\tau) s_j(t-\tau). \end{aligned}$$

$a_{ij} * s_j(t)$ is the convolution of $a_{ij}(t)$ and $s_j(t)$.

We wanted to simulate the general problem of recording sounds in a real environment. When sound signals are recorded, the major factors causing convolutions are reflections and delays. In order to simulate these factors, we built a virtual room as Figure 12 and calculated reflections and delays.

We supposed that each wall, floor and ceiling reflects the sound. The strength of the reflection is 0.1 in power for any frequency. We also supposed the strength of the sounds varies in proportion to the inverse square of the distance. Because the second reflection of a sound is really small, we only counted the first reflection. In Figure 13, the impulse response from source 1 to microphone 2 is shown. Also we show the window function with different lengths in the graph.

Apparently, the impulse response is rather long and if the window length is 8msec, all the reflections within one window cannot be included. If all the reflections are not included within a window, our new approach won't work well. Hence we have to set the window length longer than the impulse response. But as shown in Subsection 4.1.2, if we make the window length 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 Figure 5. The convolutive mixtures in this virtual room are shown in Figure 14.

For the separation, we first applied the original de-correlation algorithm. Of course it didn't work for a convolutive mixture. We also applied our algorithm, changing the window length from 8msec to 32msec. The SNRs of these results are shown in Table 3. Our approach with the window length of 32msec gave the best SNRs. Separated signals are shown in Figure 15(window length was 32msec, $\varDelta T$ was 1.25msec, r was 40.).

4.2 Real-room Recorded Data

In this subsection, we will show a result of our algorithm applied to data recorded in a real environment. This data was recorded by Prof. Kota Takahashi in the the University of Electro-Communications. Two males were repeating different phrases simultaneously in a room and they were recorded with two microphones with 44.1kHz for 5sec. Data was processed with low pass filter with 500 taps, and the sampling rate was reduced to 16kHz. Inputs to the microphones were shown in Figure 16.

We applied our algorithm to this data. The parameters are set as in the last subsection. Window length was 32msec (512 points), $\varDelta T$ was 1.25msec and r was 40. The result is shown in Figure 17.

In this experiment, we don't know the source signals precisely, and we cannot calculate the SNRs. The only way to evaluate the performance is to see the graphs and to listen to the results. A part of the separated signals $y_1(t, 1)$ and $y_2(t, 2)$ are shown in Figure 18. These two signals seem to be independent in the graphs. We listened to them and they were separated clearly.

5 Conclusion

We proposed a blind source separation algorithm based on the temporal structure of speech signals. Our algorithm has a feature that it 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 data mixed on the computer and also for the real-room-recorded data. We haven't shown other results but we have also applied our algorithm to other data, and they are available under our home page. The results are also pretty good.

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. [8, 15]) as

$$A(\omega) = \sum_{k=0}^{K} A_k \delta(\omega - \omega_k) \quad \delta(\omega - \omega_k) = \left\{ \begin{array}{ll} 0 & \omega \neq \omega_k \\ 1 & \omega = \omega \end{array} \right. . \tag{50}$$

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.

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. [7]). The impulse responses from sources to sensors are defined in matrix form,

$$A_{\mathrm{FIR}}(t) = \sum_{l=0}^{L} A_{\mathrm{FIR}l} \delta(t - t_l), \tag{51}$$

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

$$\widehat{A}_{\mathrm{FIR}}(\omega) = \sum_{l=0}^{L} A_{\mathrm{FIR}l} e^{\sqrt{-1}\omega t_l}. \tag{52}$$

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

$$\widehat{A}^{-1}_{\mathrm{FIR}}(\omega) = \left(\sum_{l=0}^{L} A_{\mathrm{FIR}l}\, e^{\sqrt{-1}\omega t_l}\right)^{-1} \tag{53}$$

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 still some problems to be solved in our algorithm. We have three major parameters, the window length, $\mathit{\Delta} T$ and r. We showed that window length can be defined independent of the other two parameters, but window length has a strong relation to the impulse response of its mixing process. If the mixing process has a long impulse response, window length has to be correspondingly longer, but 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. In our experiments, 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.

Finally, we want to say that this is only the first trial of this approach. The results were pretty good, and the algorithm is easy for hardware implementation; we are now working for it. We are also working on realizing its on-line version. An on-line algorithm will make it possible to follow the changing environment, such as tracking walking speakers. Our algorithm is based on the windowed Fourier transformation, but we can use filter banks or Wavelets instead of Fourier transformations. We are investigating the possibility of these modifications. After separating each frequency component, we rebuild them to construct new spectrograms. We use the envelope of the signals to classify each component, but here, it may be possible to use the continuity of de-mixing matrices between close frequency channels. This is also one of the problems which have not been solved.

Acknowledgment

We thank Shun-ichi Amari, Klaus-Robert Müller, Andrew Cichocki, and Colin Fyfe for very useful discussions on this work. We are grateful to Kota Takahashi for a set of data and comments on the manuscript. We also thank Te-Won Lee, Paris Smaragdis, and Scott C. Douglas for comments on our work and demonstrations which can be found in the web contents. We appreciate Daishi

Harada for thorough proofreading our our manuscript. A.Z. was partly funded by DFG contracts JA 379/51 and 379/71.

References

[1] Shun-ichi Amari and Jean-François Cardoso. Blind source separation – semiparametric statistical approach. *IEEE Trans. Signal Processing*, 45(11):2692–2700, nov 1997.

[2] Shun-ichi Amari, Andrzej Cichocki, and Howard Hua Yang. A new learning algorithm for blind signal separation. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 757–763. MIT Press, Cambridge MA, 1996.

[3] Anthony J. Bell and Terrence J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[4] Jean-François Cardoso and Beate Laheld. Equivariant adaptive source separation. *IEEE Trans. Signal Processing*, 44(12):3017–3030, December 1996.

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

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

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

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

[9] Christian Jutten and Jeanny Herault. Separation of sources, Part I. *Signal Processing*, 24(1):1–10, July 1991.

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

[11] Te-Won Lee, Anthony J. Bell, and Russell H. Lambert. Blind separation of delayed and convolved sources. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 758–764. MIT Press, Cambridge MA, 1997.

[12] Scott Makeig, Tzyy-Ping Jung, Anthony J. Bell, Dara Ghahremani, and Terrence J. Sejnowski. Blind separation of auditory event-related brain responses into independent components. *Proc. Natl. Acad. Sci. USA*, (94):10979–10984, 1997.

[13] Kiyotoshi Matsuoka, Masahiro Ohya, and Mitsuru Kawamoto. A neural net for blind separation of nonstationary signals. *Neural Networks*, 8(3):411–419, 1995.

[14] 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.

[15] 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, February 1998.

[16] Ch. von der Malsburg and W. Schneider. A neural cocktail-party processor. *Biological Cybernetics*, 54:29–40, 1986.

[17] A. Ziehe, K.-R. Müller, G. Nolte, B.-M. Mackert, and G. Curio. ICA analysis of MEG data. NIPS 97: Functional brain imaging workshop (workshop talk), 1997.

[18] Andreas Ziehe. Statistische Verfahren zur Signalquellentrennung. Master's thesis, Humboldt Universität, Berlin, 1998. (in German).

List of Figures

List of Tables

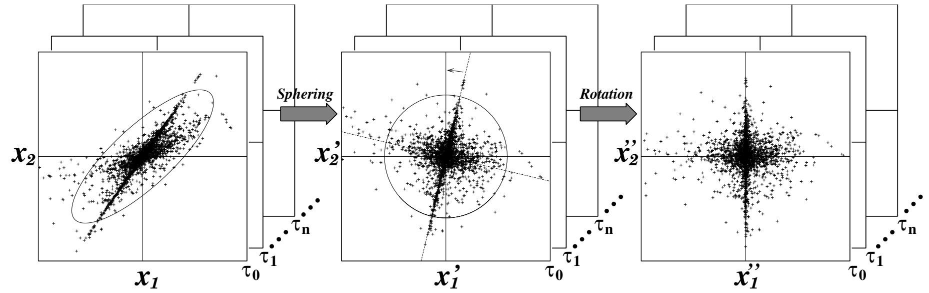

Figure 1: Decorrelation method using temporal structure of the signals

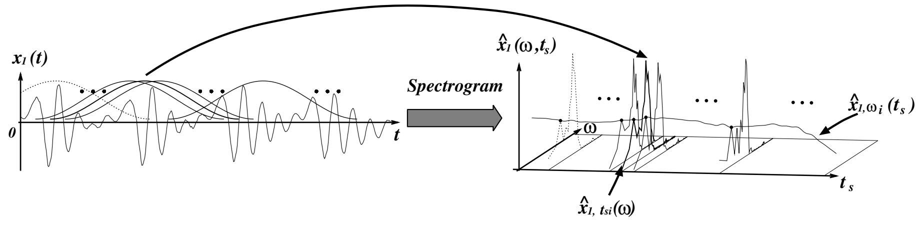
$x_l\,(t)$
0
t
. . .
Spectrogram
$\hat{x}_l\,(\omega, t_s)$
ω
$\hat{x}_{l,\omega_i}\,(t_s)$
$\hat{x}_{l,\,t_{si}}(\omega)$
t_s

Figure 2: The windowed Fourier transformation

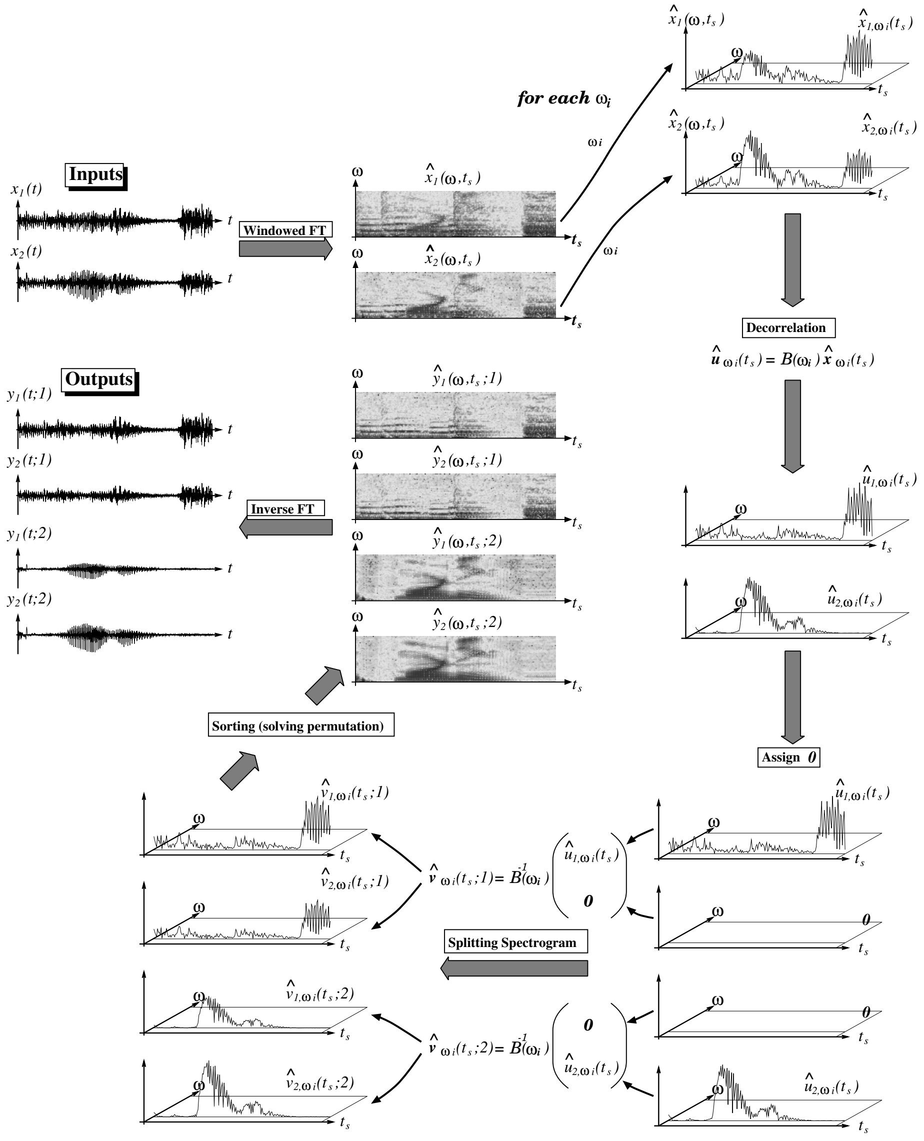
Inputs
$x_1(t)$
$x_2(t)$
t
Windowed FT
$\hat{x}_1(\omega,t_s)$
$\hat{x}_2(\omega,t_s)$
ω
t_s
for each ω_i
ω_i
$\hat{x}_{1,\omega_i}(t_s)$
$\hat{x}_{2,\omega_i}(t_s)$
Decorrelation
$\hat{u}_{\omega_i}(t_s) = B(\omega_i)\hat{x}_{\omega_i}(t_s)$
$\hat{u}_{1,\omega_i}(t_s)$
$\hat{u}_{2,\omega_i}(t_s)$
Assign 0
Outputs
$y_1(t;1)$
$y_2(t;1)$
$y_1(t;2)$
$y_2(t;2)$
Inverse FT
$\hat{y}_1(\omega,t_s;1)$
$\hat{y}_2(\omega,t_s;1)$
$\hat{y}_1(\omega,t_s;2)$
$\hat{y}_2(\omega,t_s;2)$
Sorting (solving permutation)
$\hat{v}_{1,\omega_i}(t_s;1)$
$\hat{v}_{2,\omega_i}(t_s;1)$
$\hat{v}_{1,\omega_i}(t_s;2)$
$\hat{v}_{2,\omega_i}(t_s;2)$
$\hat{v}_{\omega_i}(t_s;1)= B^{-1}(\omega_i)\begin{pmatrix}\hat{u}_{1,\omega_i}(t_s)\\ 0\end{pmatrix}$
$\hat{v}_{\omega_i}(t_s;2)= B^{-1}(\omega_i)\begin{pmatrix}0\\ \hat{u}_{2,\omega_i}(t_s)\end{pmatrix}$
Splitting Spectrogram
0

Figure 3: Flowchart of the proposed algorithm

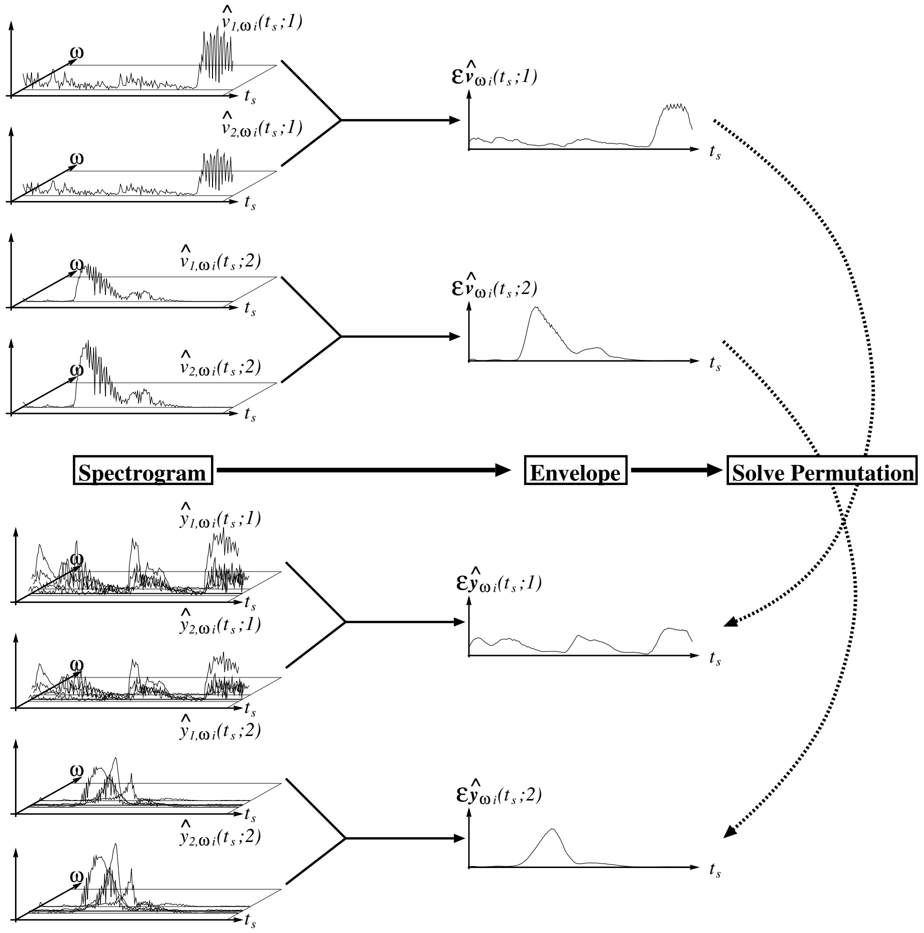
$\hat{v}_{1,\omega_i}(t_s;1)$
ω
t_s
$\hat{v}_{2,\omega_i}(t_s;1)$
ω
t_s
$\varepsilon\hat{v}_{\omega_i}(t_s;1)$
t_s
$\hat{v}_{1,\omega_i}(t_s;2)$
ω
t_s
$\hat{v}_{2,\omega_i}(t_s;2)$
ω
t_s
$\varepsilon\hat{v}_{\omega_i}(t_s;2)$
t_s
Spectrogram
Envelope
Solve Permutation
$\hat{y}_{1,\omega_i}(t_s;1)$
ω
t_s
$\hat{y}_{2,\omega_i}(t_s;1)$
ω
t_s
$\varepsilon\hat{y}_{\omega_i}(t_s;1)$
t_s
$\hat{y}_{1,\omega_i}(t_s;2)$
ω
t_s
$\hat{y}_{2,\omega_i}(t_s;2)$
ω
t_s
$\varepsilon\hat{y}_{\omega_i}(t_s;2)$
t_s

Figure 4: Solving permutation problem based on the correlation of the envelopes

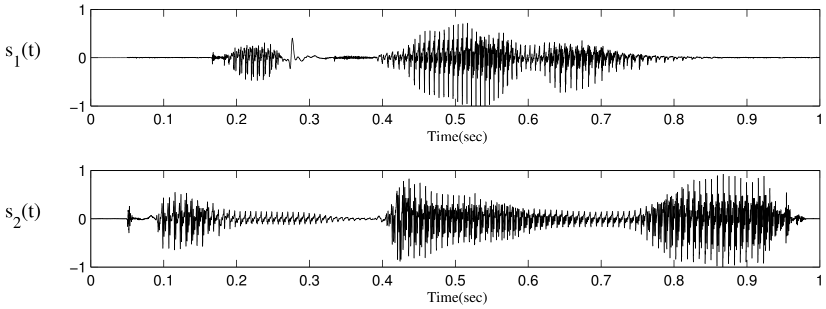

Figure 5: The source signals: each signal was spoken by a different male and recorded with sampling rate of 16kHz. $s_1(t)$ is a recorded word of "good morning" and $s_2(t)$ is a Japanese word "konbanwa" which means "good evening".

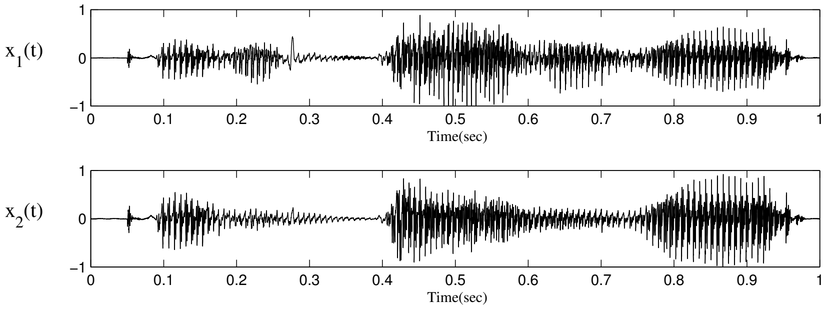

Figure 6: The mixed signals: the source signals were linearly mixed on a computer.

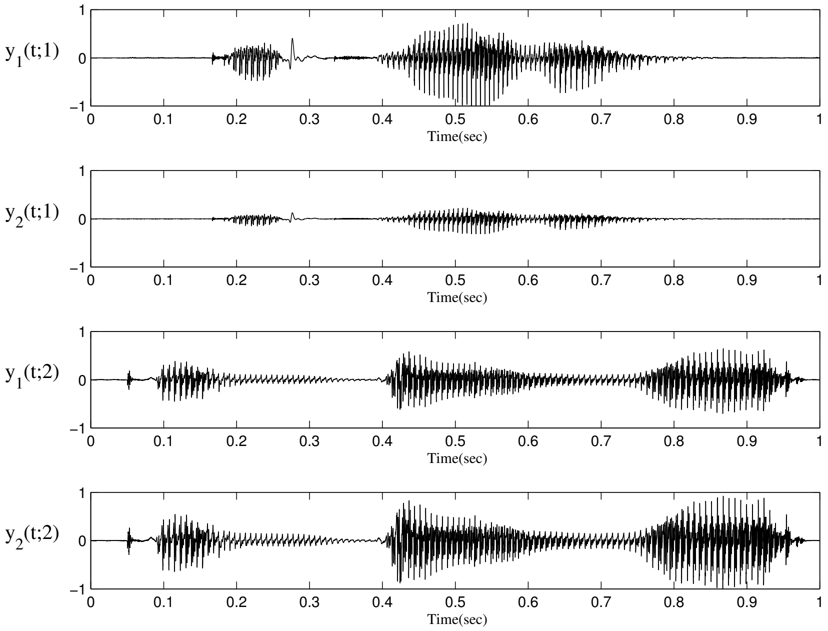

Figure 7: The separated signals using the basic decorrelation algorithm: 30 matrices were used for simultaneous diagonalization.

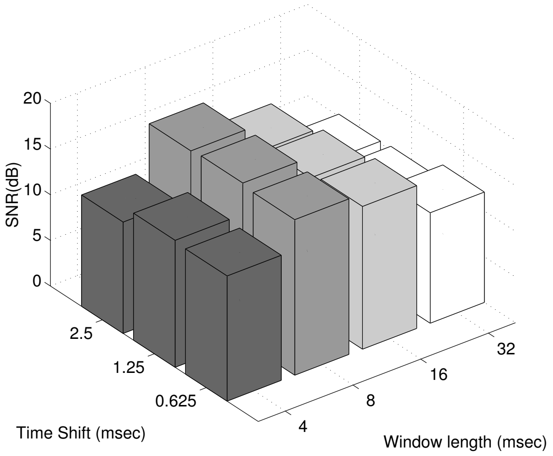

Figure 8: The SNR_{11} for instantaneous mixtures: the window lengths and $\varDelta T$ were varied, r (the number of the matrices for simultaneous diagonalization) was fixed to 30.

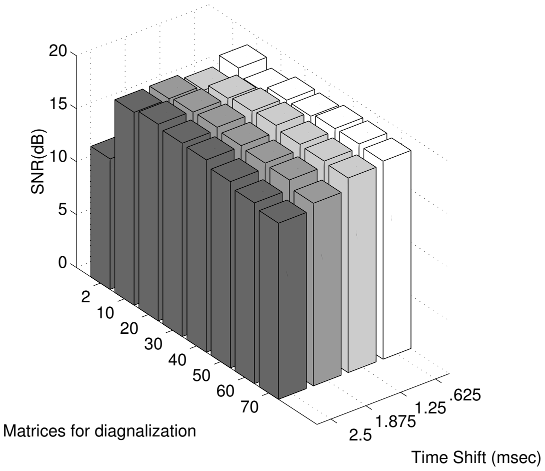

Figure 9: The SNR$_{11}$ for instantaneous mixtures: the window lengths was fixed to 8msec, $\varDelta T$ and r were varied.

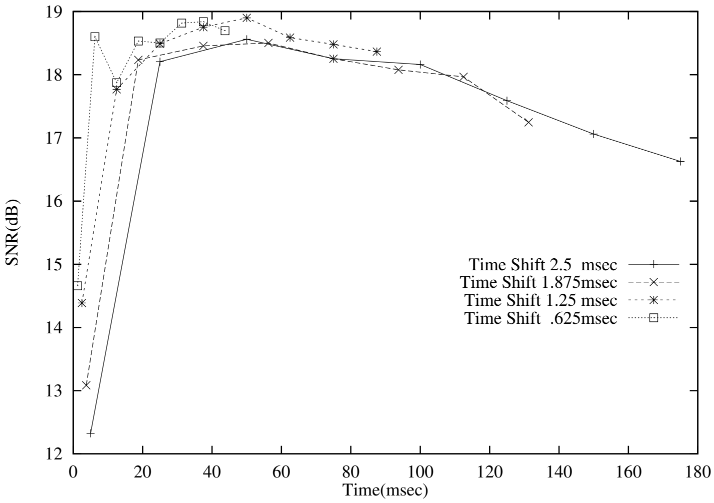

Figure 10: The SNR$_{11}$: data in Figure 9 were replot against $\varDelta T \times r$.

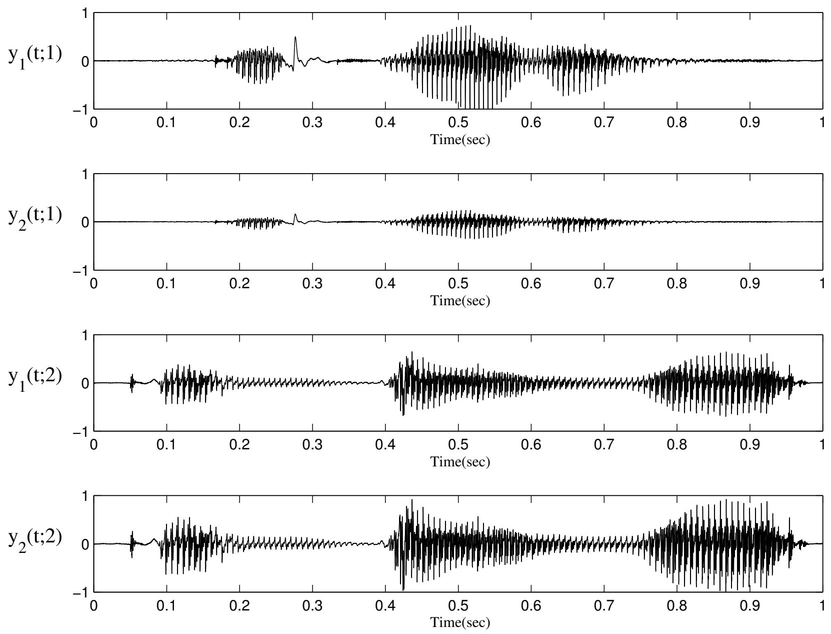

Figure 11: The separated signals using the proposed algorithm: the window length was 8msec, $\varDelta T = 1.25$msec and $r = 40$.

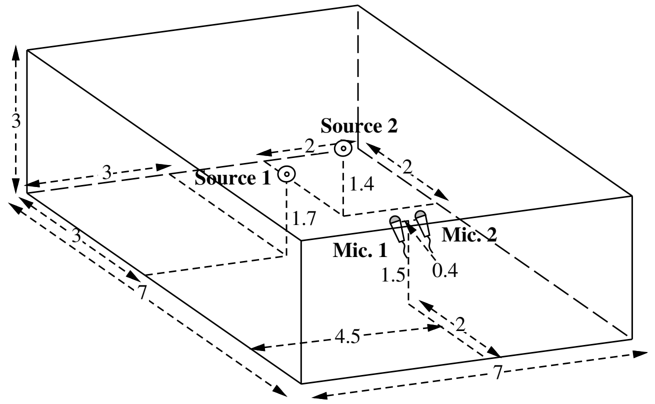

Figure 12: Virtual room for making convolutive mixtures: the length in the figure is m, and the sonic speed is 340m/sec. The strength of the reflection is 0.1 in power for any frequency, and the strength of sounds varies in proportion to the inverse square of the distance. We only counted first reflection.

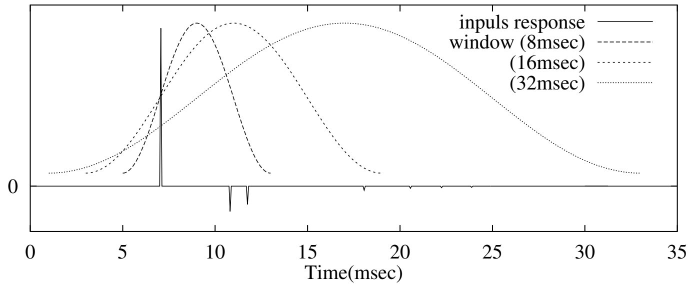

Figure 13: The impulse response from the source 1 to the microphone 2 in a virtual room

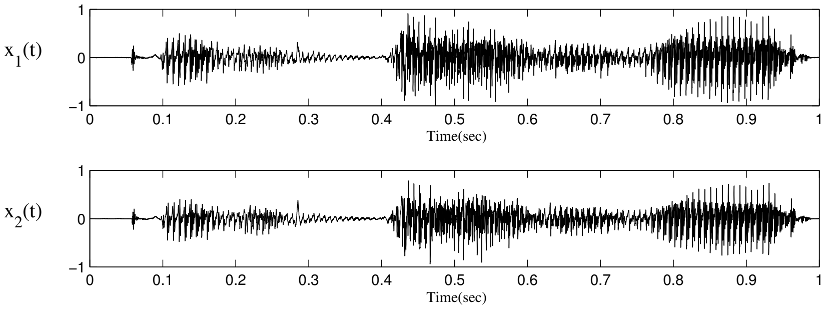

Figure 14: The mixed signals in a virtual room

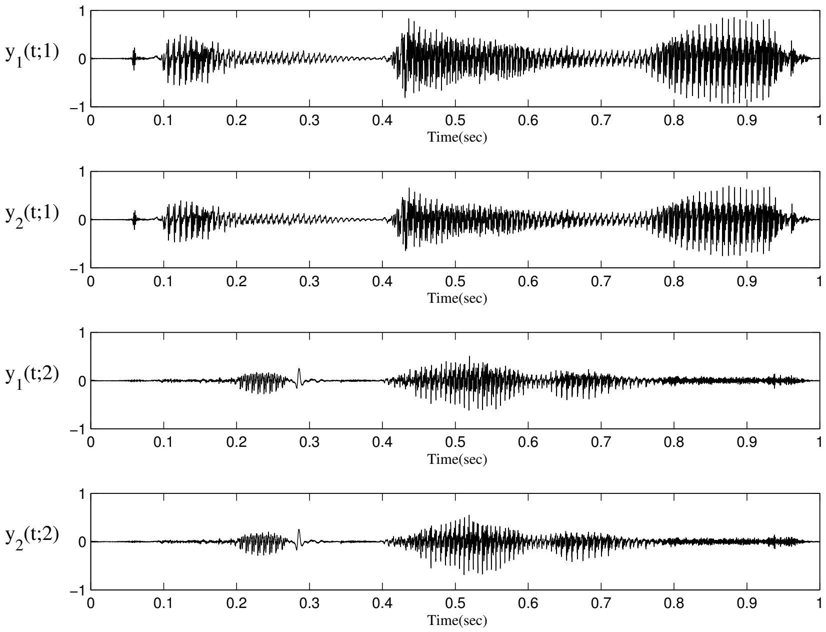

Figure 15: The separated signals: the proposed algorithm was applied with the window length of 32msec, $\varDelta T = 1.25$msec and $r = 40$.

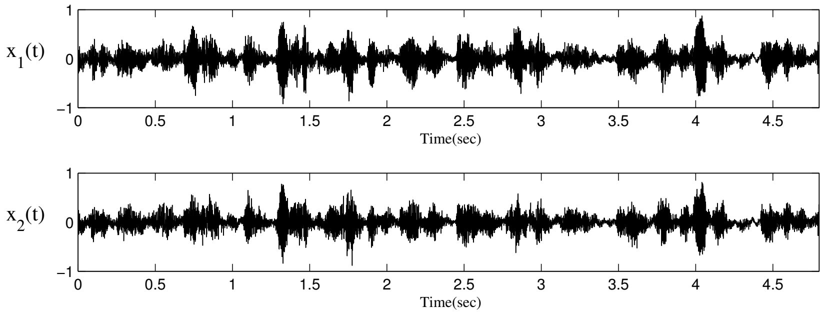

Figure 16: The recorded signals in a real room: two males were speaking different phrases independently. Speaker 1 repeated Japanese name of their university, and speaker 2 repeated the title of their research in Japanese.

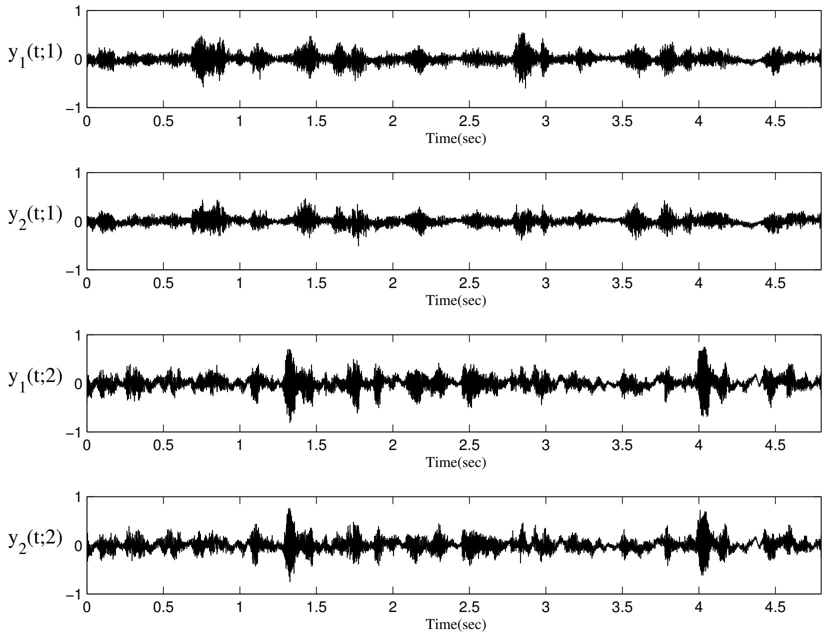

Figure 17: The separated signals: the proposed algorithm was applied with the window length of 32msec, $\varDelta T = 1.25$msec and $r = 40$.

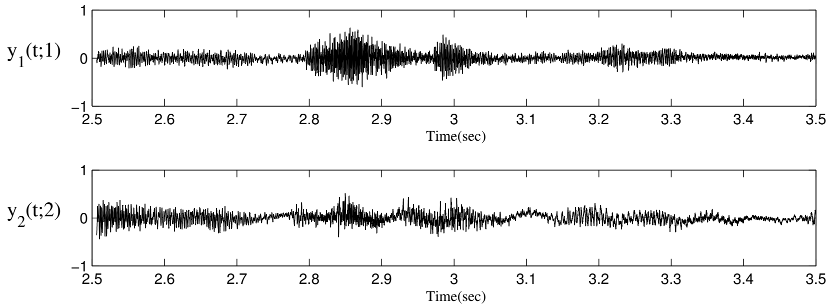

Figure 18: A part of $y_1(t, 1)$ and $y_2(t, 2)$ in Figure 17: they were zoomed up from t =2.5sec to 3.5.

Table 1: The SNRs (dB) for linear mixture using the basic decorrelation algorithm: 30 matrices were used for simultaneous diagonalization.

SNR$_{11}$	SNR$_{12}$	SNR$_{21}$	SNR$_{22}$
40.98	27.45	41.54	41.56

Table 2: The SNRs (dB) for linear mixture using the proposed algorithm: the window length was 8msec, $\Delta T = 1.25$msec and $r = 40$.

SNR$_{11}$	SNR$_{12}$	SNR$_{21}$	SNR$_{22}$
18.90	11.13	19.71	21.14

Table 3: The SNRs (dB) for convolutive mixtures in a virtual room: $\Delta T = 1.25$msec and $r = 40$.

		SNR$_{11}$	SNR$_{12}$	SNR$_{21}$	SNR$_{22}$
Original de-correlation		-1.87	-1.22	2.93	2.74
Our approach (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