A New Method for Blind Source Extraction

Tsubasa YOSHIHARA * and Kiyoshi MATSUOKA *

Abstract: Blind source separation (BSS) is a method for recovering a set of statistically independent signals from the observation of their mixtures without any prior knowledge about the mixing process. If, as a special case, only one source component is to be extracted, it is called blind source extraction (BSE). Since BSE involves a smaller number of parameters to be estimated than BSS, it requires less computation time. In this paper we propose a new algorithm for BSE of the convolutive mixture. The algorithm determines the extractor by evaluating independence between a target signal and other signals. It has some good properties. First, since it is formulated in the time domain, we do not need to worry about the so-called permutation problem. Second, by applying a particular constraint on the extractor the signal quality at the sensors is preserved through the extraction process. A couple of experiments are shown in which the proposed algorithm is applied to the mixture of five voice signals.

Key Words: blind source extraction, blind source separation, independent component analysis, convolutive mixture, minimal distortion.

1. Introduction

Blind source separation (BSS) or independent component analysis (ICA) is a method for recovering a set of statistically independent signals from the observation of their mixtures without any prior knowledge about the mixing process [1]. The mixing process can be classified into two types: the instantaneous mixture and the convolutive mixture [2]. While early works for BSS dealt with the former type, recent works are more concerned with the latter type, which is much more difficult from theoretical and computational points of view. In this paper we deal with convolutively mixed sources.

To recover source components from $N$ observations of $N$ sources, two schemes can be considered:

1. To separate all $N$ source components.
2. To extract only one source component.

This paper proposes a method for the second case, which is called blind source extraction (BSE). In the first case, i.e. in the case of the usual BSS, we need to find $N^2$ parameters while in BSE we have only to determine $N$ parameters. BSE is accordingly superior to BSS with respect to the computation time. There are many cases where we have only to extract one signal, for example, biomedical signal analysis, geophysical data processing, data mining, wireless communications, speech and image recognition and enhancement [3],[4].

BSE for the convolutive mixture is sometimes performed in the frequency domain [5]–[7]. The observation signals are decomposed into a set of frequency bins and a BSE algorithm for the instantaneous mixture is applied to the data in each bin. The frequency-domain BSE, however, has the so-called permutation problem; the obtained extractor can extract a pure source component for each frequency, but the components for different frequencies may be originated from different sources. This inter-frequency permutation problem is much more crucial in BSE than in BSS because, once the permutation occurs, there is no way to solve it. In the case of BSS, since all source components for each frequency are obtained, the problem can be solved by appropriately aligning the permutation, using some information about the signals.

The algorithm proposed in this paper has some particular features:

1. It is a completely time-domain method. So, we can avoid the permutation problem that can appear in the frequency-domain approach. It therefore fits real-time processing.
2. It evaluates independence between a target signal and other signals; the conventional methods employ maximization or minimization of a contrast function for a single signal [8].
3. A source signal incoming to a sensor can be extracted without any distortion [9], i.e., the signal quality is preserved through the extraction process [10].
4. The obtained extractor is irrelevant to nonstationarity of the sources.

2. Mixing and Extracting Processes

Let us consider a situation where statistically independent random signals $s_i(t) (i = 1, \ldots, N)$ are generated by $N$ sources and their mixtures are observed by $N$ sensors. It is assumed that source signals $s_i(t)$ are random processes with zero mean, and the sensor’s outputs $x_i(t) (i = 1, \ldots, N)$ are given by a linear mixing process:

$$x(t) = \sum_{\tau=0}^{\infty} A_s(t-\tau),$$

where $s(t) = [s_1(t), \ldots, s_N(t)]^T$ and $x(t) = [x_1(t), \ldots, x_N(t)]^T$. It can be rewritten as $x(z) = A(z)s(z)$, where $x(z)$, $s(z)$, and $A(z)$

* Department of Brain Science and Engineering, Kyushu Institute of Technology, Hibikino 1–1, Wakamatsu, Kitakyushu, Japan
E-mail: yoshihara-tsubasa@edu.brain.kyutech.ac.jp, matsuoka@brain.kyutech.ac.jp

(Received August 8, 2008)
(Revised February 18, 2009)
are \( z \)-transforms of \( X(t) \), \( s(t) \), and \( A_r \), respectively. It is known that, to realize BSS or BSE, at most one source signal is allowed to be Gaussian.

To extract a source component from the sensor signals we consider the following process, which is called an extractor:

\[
y(t) = \sum_{\tau=\infty}^{\infty} W_{\tau} x(t - \tau),
\]

where \( y(t) \equiv [y_1(t), \ldots, y_N(t)]^T \) and \( W_{\tau} \) are \( N \times N \) square matrices. In the \( z \)-domain eq. (2) is expressed as \( y(z) = W(z)x(z) \). The impulse response \( \{W_{\tau}\} \) may take a non-causal form in general, i.e., \( W_{\tau} \neq 0 \) \((\tau < 0)\). The problem of noncausality can be solved by designing \( W(z) \) so that the source signals may be reproduced with a time lag.

In the conventional methods the extractor takes the form \( y(z) = W(z)x(z) \), where \( y(z) \) is a scalar function and \( W(z) \) is an \( N \)-dimensional row vector. In contrast our extractor has the same structure as the separator, but a certain constraint is given to \( W(z) \). Since our task is to extract one independent component, we are only concerned about independence between \( y_1(t) \) and \( y_2(t) = y_2(t), \ldots, y_N(t) \); we do not care about independence among \( y_2(t), \ldots, y_N(t) \).

### 3. A Constraint for the Extractor

In order to perform the task, the extractor must at least take a form of

\[
W(z) = \begin{bmatrix} d(z) & 0^T \\ 0 & D(z) \end{bmatrix} PA^{-1}(z),
\]

where \( 0 \) is the \((N - 1)\)-dimensional zero vector and \( P \) is a permutation matrix. Henceforth, without loss of generality, \( P \) is assumed to be the identity matrix, \( I \). Function \( d(z) \) is an arbitrary scalar transfer function, and \( D(z) \) is an arbitrary transfer function matrix of size \( N \times 1 \). The key point is that \( D(z) \) is not diagonal. So, if a scalar function of \( z \) is counted as one, we are given \((1 + (N - 1))^2\) degrees of freedom for \( W(z) \).

In order to eliminate the indeterminacy, which is referred to as filtering indeterminacy, we introduce a particular constraint on \( W(z) \):

1. \( w_{ij}(z) = 0 \) for \( i \neq 1, j \neq 1 \), and \( i \neq j \)
2. \( \sum_{j=1}^{N} w_{1j}(z) = 1 \)
3. \( w_{1j}(z) + w_{jj}(z) = 0 \) for \( j \neq 1 \)

Namely, the extractor \( W(z) \) takes the following form:

\[
W(z) = \begin{bmatrix} 1 - \sum_{i=2}^{N} w_{i1}(z) & w_{12}(z) & w_{13}(z) & \cdots & w_{1N}(z) \\ w_{21}(z) & -w_{12}(z) & 0 & \cdots & 0 \\ w_{31}(z) & 0 & -w_{13}(z) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ w_{N1}(z) & 0 & \cdots & 0 & -w_{1N}(z) \end{bmatrix}
\]

Figure 1 shows the extractor for \( N = 3 \). Obviously the total number of the constraints on \( W(z) \) is \( 1 + (N - 1)^2 \), which is equivalent to the total freedom in \( d(z) \) and \( D(z) \). Thus, the numerator of the remaining parameters we must determine based on independence between \( y_1(t) \) and \( y_2(t) \) is \( 2(N - 1) \). In the case of \( N = 8 \), for example, the substantial number of variables to be determined is 14, being much less than \( N^2 = 64 \).

The constraint on \( W(z) \) gives an important characteristic to the extractor. It is easy to show that \( W(z) \) satisfies

\[
\begin{bmatrix} 1 & e^{T} \\ 0 & D(z) \end{bmatrix} W(z) = \begin{bmatrix} 1 & 0^{T} \\ 0 & D(z) \end{bmatrix}
\]

where \( e = [1, \ldots, 1]^{T} \). Substituting \( W(z) \) in this equation with the right-hand side of eq. (3), we obtain

\[
\begin{bmatrix} 1 & e^{T} \\ 0 & D(z) \end{bmatrix} [d(z) 0^{T}] = [1 0^{T}] A(z)
\]

This leads to

\[
d(z) = a_{11}(z).
\]

Also, we can show

\[
D(z) = \text{diag} \left( a_{22}(z) \left( A_{22}(z) - a_{21}(z) a_{11}(z) a_{12}(z) \right)^{-1} \right)
\]

where

\[
A(z) = \begin{bmatrix} a_{11}(z) & a_{12}(z) \\ a_{21}(z) & a_{22}(z) \end{bmatrix}
\]

or more generally \( y_1(t) = a_{11}(z) s_1(t) \), where \( j \) is an arbitrary source number. Namely a source signal that would be observed at a sensor \( x_j(t) \) in the absence of interference appears at \( y_1(t) \). In this sense the obtained signal receives no distortion through the extraction process.

Though the indeterminacy in \( d(z) \) and \( D(z) \) can be removed as eqs. (8) and (9), we still have indeterminacy due to permutation. In actual applications, however, the problem can be solved by appropriately selecting an initial value of \( W(z) \) based on some information about the direction of the target source.

Another important point is that the constraint is applied directly to \( W(z) \), not to \( y(z) \), so that the extractor is independent of statistical properties of sources. That is, the obtained extractor does not involve \( s(t) \), and depends only on \( A(z) \) eqs. (8) and (9)). Therefore, even for such a nonstationary signal as speech, the proposed extractor is invariant with time. It is a nice property in many applications.
4. A New Algorithm for Blind Source Extraction

Here we show how to achieve independence between \( y_1(t) \) and \( y_2(t) = [y_2(t), \ldots, y_N(t)]^T \). We start with a time-domain BSS algorithm proposed by Amari et al. [12],[13]. In the algorithm the separator \( W(z) \) is updated as

\[
\Delta W(z) \propto -\text{off-diag} \left( \varphi(y(t)) y^T(t, z^{-1}) \right) W(z),
\]

where

\[
y(t, z) = \sum_T y(t + \tau) z^{-\tau}, \quad \varphi(y(t)) = [\varphi(y_1(t)), \ldots, \varphi(y_N(t))]^T
\]

and nonlinear function \( \varphi(\cdot) \) is determined according to the type of non-gaussianity of the sources. Off-diag(*) sets the diagonal entries of matrix (*) to be zero. In this algorithm every independence among \( y_1(t), \ldots, y_N(t) \) are evaluated.

Our concern in this paper is only independence between \( y_1(t) \) and \( y_2(t) = [y_2(t), \ldots, y_N(t)]^T \). To achieve this, we modify eq. (12) as

\[
\Delta W(z) \propto -
\begin{bmatrix}
0 & \varphi(y_1(t)) y_2^T(t, z^{-1}) \\
\varphi(y_2(t)) y_1(t, z^{-1}) & 0
\end{bmatrix}
\begin{bmatrix}
w_{11}(z) & w_{12}(z) \\
w_{21}(z) & w_{22}(z)
\end{bmatrix},
\]

where \( w_{21}(z) = [w_{21}(z), \ldots, w_{N1}(z)]^T \) and \( w_{12}(z) = [w_{12}(z), \ldots, w_{1N}(z)]^T \).

Taking into account the proposed constraint, we further modify eq. (13) as

\[
\Delta W(z) \propto -
\begin{bmatrix}
g(z) & \varphi(y_1(t)) y_2^T(t, z^{-1}) \\
\varphi(y_2(t)) y_1(t, z^{-1}) & G(z)
\end{bmatrix}
\begin{bmatrix}
w_{11}(z) & w_{12}(z) \\
w_{21}(z) & w_{22}(z)
\end{bmatrix},
\]

where

\[
g(z) = -e^T \varphi(y_2(t)) y_1(t, z^{-1})
\]

\[
G(z) = \text{off-diag} \left( \varphi(y_1(t)) y_2^T(t, z^{-1}) \right)
\]

Terms \( g(z) \) and \( G(z) \) has totally \( 1 + (N-1)^2 \) entries. They are determined so that \( \Delta W(z) \) will satisfy a condition derived from the constraint on \( W(z) \).

In order for \( W(z) \) to satisfy eq. (6) at every step, the following equation must hold:

\[
\begin{bmatrix}
e & e^T
\end{bmatrix}
\begin{bmatrix}
W(z) + \Delta W(z)
\end{bmatrix} =
\begin{bmatrix}
e & e^T
\end{bmatrix}
\begin{bmatrix}
0 & 0
\end{bmatrix},
\]

or

\[
\begin{bmatrix}
e & e^T
\end{bmatrix}\Delta W(z) =
\begin{bmatrix}
0 & 0
\end{bmatrix}.
\]

So as to make this condition satisfied, we choose \( g(z) \) and \( G(z) \) as

\[
g(z) = -e^T \varphi(y_2(t)) y_1(t, z^{-1})
\]

\[
G(z) = \text{off-diag} \left( \varphi(y_1(t)) y_2^T(t, z^{-1}) \right)
\]

To actually implement this algorithm it is more efficient to introduce new variables

\[
u_{ij}(t, z^{-1}) = y_i(t, z^{-1}) w_{ij}(z), \quad (i = 1 \text{ or } j = 1 \text{ or } i = j)
\]

Then, eq. (13) becomes

\[
\Delta W(z) \propto
\begin{bmatrix}
\Delta w_{11}(z) & \Delta w_{12}(z) & \cdots & \Delta w_{1N}(z) \\
\Delta w_{21}(z) & \Delta w_{22}(z) & \cdots & 0 \\
0 & \Delta w_{21}(z) & \cdots & 0 \\
\Delta w_{N1}(z) & 0 & \cdots & \Delta w_{NN}(z)
\end{bmatrix}.
\]

where

\[
\Delta w_{11}(z) =
\begin{bmatrix}
-a \left( \sum_{i=2}^N \varphi(y_1(t)) u_{11}(t, z^{-1}) + \varphi(y_1(t)) \sum_{i=2}^N u_{i1}(t, z^{-1}) \right)
\end{bmatrix}
\]

\[
\Delta w_{12}(z) =
\begin{bmatrix}
-a \left( \varphi(y_1(t)) u_{12}(t, z^{-1}) - \varphi(y_1(t)) u_{11}(t, z^{-1}) \right)
\end{bmatrix}
\]

\[
\Delta w_{21}(z) =
\begin{bmatrix}
-a \left( \sum_{i=2}^N \varphi(y_1(t)) u_{21}(t, z^{-1}) + \varphi(y_1(t)) u_{11}(t, z^{-1}) \right)
\end{bmatrix}
\]

\[
\Delta w_{22}(z) = -\Delta w_{12}(z) (j = 2, \ldots, N)
\]

It should be noted that the amount of calculation required for eqs. (17) and (18) is very small compared to the case that \( \Delta w_{ij} \) need to be calculated for every \( i \) and \( j \).

After \( \Delta W(z) \) is calculated, \( W(z) \) is updated by \( W(z) \leftarrow W(z) + \Delta W(z) \). After updating \( W(z) \), we must further rectify the extractor so as to satisfy the constraint. This manipulation may apparently seem unnecessary because eqs. (18) and (19) have already satisfied the constraint. However, it should be incorporated to avoid the accumulation of small round-off errors; for \( w_{ij}(z) (i = 1, \ldots, N) \)

\[
\tilde{w}_{ij}(z) \leftarrow \frac{1}{N} \left( \sum_{j=1}^N w_{ij}(z) - 1 \right).
\]

and for \( w_{ij}(z) \) and \( w_{jj}(z) (j = 2, \ldots, N) \)

\[
\tilde{w}_{ij}(z) \leftarrow \frac{1}{2} (w_{ij}(z) + w_{jj}(z)),
\]

\[
\tilde{w}_{ij}(z) \leftarrow w_{ij}(z) - \tilde{w}_{ij}(z) \quad \text{and} \quad w_{jj}(z) \leftarrow w_{jj}(z) - \tilde{w}_{jj}(z).
\]

5. Actual Implementation

In actual implementation, we use an FIR filter for the extractor:

\[
W(z) = \sum_{\tau=-L_1}^{L_2} \tilde{w}_\tau z^{-\tau}.
\]

Moreover we need to take into account non-causality of the desired extractor. It is solved by designing it so that the output of the extractor will be generated with a time delay \( L_1 \).

\[
y_1(t - L_1) = \sum_{\tau=1}^{L_2} \sum_{j=1}^N w_{j1} x_j(t - L_1 - \tau)
\]

\[
y_i(t - L_1) = \sum_{\tau=1}^{L_2} (w_{1i} x_1(t - L_1 - \tau) + w_{ii} x_i(t - L_1 - \tau))
\]

where \( i = 2, \ldots, N \). Here, for example, \( y_1(t - L_1) \) indicates an estimate of component of \( s_1(t) \) in \( x_i(t-L_1) \) estimated 'at time t'.

Corresponding to eq. (17), we have

\[
u_{ij}(t - L_0) = \sum_{\tau=-L_1}^{L_2} y_i(t - L_0 + \tau) w_{ij}(z),
\]
where \( L_0 = L_1 + L_2 \), and so the update rule for \( w_{ij,t} \) \((\tau = -L_1,\ldots,L_2)\) becomes as follows:

\[
\Delta w_{1,t} = -\alpha_r \left\{ -\sum_{i=2}^{N} \varphi(y_i(t-L_0 + \tau))u_{11}(t-L_0) + \varphi(y_1(t-L_0 + \tau))u_{11}(t-L_0) \right\} \Delta w_{1,1} = -\alpha_r \left\{ -\sum_{i=2}^{N} \varphi(y_i(t-L_0 + \tau))u_{j1}(t-L_0) + \varphi(y_1(t-L_0 + \tau))u_{j1}(t-L_0) \right\} \Delta w_{j,j} = -\alpha_r \left\{ -\sum_{i=2}^{N} \varphi(y_i(t-L_0 + \tau))u_{j1}(t-L_0) + \varphi(y_1(t-L_0 + \tau))u_{j1}(t-L_0) \right\} \Delta w_{j,j} = -\Delta w_{j,j} \] (25)

Parameter \( \alpha_r \) is a learning coefficient that depends on lag time \( \tau \). This is introduced to increase the robustness [14].

\[
\alpha_r = \begin{cases} 
\alpha \left( 1 + \frac{\tau}{L_1 + 1} \right) & \text{for } -L_1 \leq \tau < 0 \\
\alpha \left( 1 - \frac{\tau}{L_2 + 1} \right) & \text{for } 0 \leq \tau \leq L_2 
\end{cases} \] (26)

After \( w_{ij,t} \) are updated as \( w_{ij,t} = w_{ij,t} + \Delta w_{ij,t} \), they are rectified so as to satisfy the constraints: for \( w_{i1,t} (i = 1,\ldots,N) \)

\[
\bar{w}_{i1,t} = \frac{1}{N} \sum_{i=1}^{N} w_{i1,t} - \delta(\tau), \quad \text{for } \tau = 0 \quad \text{and} \quad \delta(\tau) = 0 \text{ otherwise}.
\] (27)

where \( \delta(\tau) \) is defined as \( \delta(\tau) = 1 \) for \( \tau = 0 \) and \( \delta(\tau) = 0 \) otherwise, and for \( w_{ij,t} \) and \( w_{jj,t} (j = 2,\ldots,N) \)

\[
\bar{w}_{j,t} = \frac{1}{2} (w_{j,t} + w_{j,t}), \quad w_{i1,t} = \bar{w}_{i1,t} - \bar{w}_{1,t} \quad \text{and} \quad w_{i,j,t} = \bar{w}_{i,j,t} - \bar{w}_{j,t} \] (28)

6. The Experimental Results

To demonstrate the effectiveness of the algorithm, we conducted an experiment in a sound proof room. Five microphones and five loudspeakers were put as shown in Fig. 2; the microphones were omnidirectional. The sources were voices of three males and two females. They were sampled at 10kHz; see Figs. 4 and 5. For nonlinear function \( \varphi, \varphi(u) = \text{sgn}(u) \) was used in consideration of super-Gaussianity of the sound signals.

In the experiment (a), the microphones were arranged at equal intervals on a circle with radius \( r_1 = 0.2 \text{m} \), and the loudspeakers are similarly arranged on a circle of radius \( r_2 = 1.2 \text{m} \). Figure 5 shows the result, which means \( y_1(t) \) in the figure corresponds to \( x_1(t) \) in Fig. 3. To evaluate the performance of the extraction, we calculated a signal-to-noise ratio improvement (SNR) between the source signal and the extracted signal. It is defined as
SNRI = 10\log_{10} \left( \frac{\|s_1(t) - a_{11}(z)s_1(t)\|^2}{\|y_1(t) - a_{11}(z)s_1(t)\|^2} \right)

In the case of experiment (a), the SNRI was 16.6 [dB].

Next, an additional experimental setup is shown in Fig. 6. The microphones were arranged as well as the experiment (a), i.e., \( r_1 = 0.2 \)m but the loudspeakers were put at random in experiment (b).

Figures 7, 8, and 9 show the source signals, the observed signals, and the extraction result, respectively. In this case the SNRI was 15.3 [dB]; almost the same degree of SNRI was attained.

Here, let us consider the computation time in the experiment, i.e., \( N = 5 \). As the number of required parameters is large, a longer computation time is required. In the case of this experiment, since the length of the filter was \( (L_1 = 360, L_2 = 480) \), totally \( 8 \times 841 \) parameters are used. In contrast, in the case of BSS, we need to evaluate independence not only between \( y_1(t) \) and \( y_2(t) = [y_2(t),...,y_N(t)]^T \) but also among \( y_i(t) (i = 2,\ldots,N) \), and so \( (N^2 \times \text{filter length}) = 25 \times 841 \) parameters are required to be found. Table 1 shows the number of multiplications in the proposed BSE algorithm and that in the BSS algorithm with the same property \cite{15}. In the proposed algorithm, the number of multiplications is smaller by about 50\% than in BSS.

Table 1 The number of times of the multiplication.

<table>
<thead>
<tr>
<th>proposed method</th>
<th>BSS algorithm based on \cite{15}</th>
</tr>
</thead>
<tbody>
<tr>
<td>((19N - 11)L)</td>
<td>(7N^2L)</td>
</tr>
</tbody>
</table>

7. Conclusion

In this paper we have shown a novel method for BSE, which preserves the quality of the extracted signal. The extraction result showed that the proposed method provides a good extraction performance for nonstationary sources. The constraints are easy to implement and the one on the extractor eliminates the filtering indeterminacy. Also, the constraints reduce the computation time particularly in the case with a large number of sources.

References


**Appendix**

The extractor $W(z)$ has the following form:

$$W(z) = \begin{bmatrix} w_{11}(z) & w_{12}(z) \\ w_{21}(z) & W_{22}(z) \end{bmatrix},$$

(A.1)

where $W_{22}(z)$ is a diagonal matrix of size $N - 1$.

The inverse of $A(z)$ is

$$A^{-1}(z) = \begin{bmatrix} a'_{11}(z) & -a_{12}(z)a_{12}(z) \\ -a_{21}(z)a'_{12}(z) & a'_{12}(z) \end{bmatrix},$$

(A.2)

where $a'_{11}(z) = (a_{11}(z) - a_{12}(z)a'_{22}(z)a_{21}(z))^{-1}$ and $a'_{12}(z) = (A_{22}(z) - a_{21}(z)a'_{11}(z)a_{12}(z))^{-1}$.

From eqs. (A.1) and (A.2), we have

$$D(z) = W_{22}(z)(A_{22}(z) - a_{21}(z)a'_{11}(z)a_{12}(z))^{-1}$$

(A.3)

From eqs. (3), (6), and (A.1), we have

$$\begin{bmatrix} 1 & e^T \\ 0 & D(z) \end{bmatrix} \begin{bmatrix} d(z) \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a_{11}(z) & a_{12}(z) \\ a_{21}(z) & A_{22}(z) \end{bmatrix}$$

(A.4)

leading to

$$d(z) = a_{11}(z),$$

$$e^T D(z) = a_{12}(z)$$

(A.5)

From eqns. (A.3) and (A.5), we have

$$e^T W_{22}(z)(A_{22}(z) - a_{21}(z)a'_{11}(z)a_{12}(z))^{-1} = a_{12}(z)$$

(A.6)

This leads to

$$W_{22}(z) = \text{diag}(e^T W_{22}(z)) = \text{diag}(a_{12}(z)(A_{22}(z) - a_{21}(z)a'_{11}(z)a_{12}(z)))^{-1}$$

(A.7)

Substituting this into eq. (A.3), we obtain

$$D(z) = \text{diag}(a_{12}(z)(A_{22}(z) - a_{21}(z)a'_{11}(z)a_{12}(z)))^{-1} \cdot (A_{22}(z) - a_{21}(z)a'_{11}(z)a_{12}(z))^{-1}$$

(A.8)

Thus, the desired extractor satisfying the constraint is

$$W(z) = d(z) \begin{bmatrix} 0^T \\ 0 \end{bmatrix} D(z) \begin{bmatrix} 1 \\ A^{-1}(z) \end{bmatrix}$$

(A.9)

---

**Tsubasa Yoshihara (Student Member)**

He received his B.E. degree from Yatsushiro National College of Technology, Japan in 2004, and his M.E. degree from Kyushu Institute of Technology, Japan in 2006. He is currently a candidate for the Ph. D. at Kyushu Institute of Technology. His research interests are in blind source separation and blind source extraction.

**Kiyotoshi Matsuoka (Member)**

He holds his Master’s and Ph.D. Degrees, both in Mechanical Engineering from the University of Tokyo. He is now a Professor of the School of Life Science and Systems Engineering at Kyushu Institute of Technology, Japan. His research interests are in signal processing such as independent component analysis and neural networks. He received an excellent paper award from Japan Neural Network Society in 1996 and the best paper award from the Research Institute of Signal Processing Japan in 2008.