# Title
Partial separation method for solving permutation problem in frequency domain blind source separation of speech signals

# Author(s)
Reju, Vaninirappuputhenpurayil Gopalan; Koh, Soo Ngee; Soon, Ing Yann

# Citation

# Date
2008

# URL
http://hdl.handle.net/10220/7006

# Rights
© 2008 Elsevier. This is the author created version of a work that has been peer reviewed and accepted for publication by Neurocomputing, Elsevier. It incorporates referee’s comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [DOI: http://dx.doi.org/10.1016/j.neucom.2007.08.030].
Partial Separation Method for Solving
Permutation Problem in Frequency Domain
Blind Source Separation of Speech Signals

V.G.Reju, Soo Ngee Koh, Ing Yann Soon

School of Electrical and Electronic Engineering, Nanyang Technological University,
Singapore 639798

Abstract

This paper addresses the well known permutation problem in frequency domain blind source separation. The proposed method uses correlation between two signals in each DFT bin to solve the permutation problem. One of the signals is partially separated by a time domain blind source separation method and the other is obtained by the frequency domain blind source separation method. Two different ways of configuring the time and frequency domain blocks, i.e., in parallel or cascade, have been studied. The cascaded configuration not only achieves a better separation performance but also reduces the computational cost as compared to the parallel configuration.

Key words: Blind source separation, convolutive mixture, direction of arrival, independent component analysis, permutation problem.
1 Introduction

Blind source separation (BSS) is the technique of separating the sources from their mixtures without any prior knowledge about the sources or the mixing process. Independent component analysis (ICA) is one of the techniques commonly used for blind source separation. There are many algorithms for separation of linear instantaneous mixtures where the mixing matrix consists of simple scalars and so is the unmixing matrix. However in real acoustic environments, the mixing is very complex convolutive mixing, so that finite impulse response (FIR) filter matrix is required in place of simple matrix of scalars. This makes the problem of blind source separation a challenging task. For example, the speech signal mixed in a real acoustic environment normally requires thousands of filter taps for the unmixing filter.

There are two main approaches for the BSS of convolutive mixtures. The first one is the time domain approach and the second is the frequency domain approach. Also there is a combination of the above two, where the computational complexity of the time domain method is reduced by implementing the time domain convolution operation in the frequency domain [13,8]. The main disadvantage of the time domain method is that it is computationally intensive and convergence speed is low for long filters. Whereas, the frequency domain method is computationally efficient as the convolution in time domain becomes simple element-wise multiplication in the frequency domain and hence the complex valued ICA algorithm can be applied to each discrete Fourier transform (DFT) bin. However, the method has the disadvantage of inconsistent permutation in the DFT bins after separation by ICA. This is the popularly known permutation problem of frequency domain BSS. The algo-
rithms used for solving the permutation problem will align the permutation in the DFT bins so that the separated signal in the time domain will contain the frequency components from the same source signals.

Various methods have been proposed to solve the permutation problem in the frequency domain BSS. L. Parra et al [18] suggested a method which constrains the length of the filter but this is not suitable for real acoustic environments where the length of the separation filter is of the order of thousands. Smoothening of the separation matrix is another method [24,5]. The property that for speech signals, the adjacent bands are highly correlated is utilized in [15] to solve the permutation problem. In [19] the reliability of the correlation method is improved with the dyadic permutation sorting algorithm. Direction of arrival (DOA) is another popular method, where the directivity pattern formed by the separation matrix in the frequency domain is used to estimate the direction of the sources and hence the permutation problem is solved [14,22,11,12]. The DOA method cannot solve the permutation problem in all the DFT bins especially for lower frequency bins. H. Sawada et al [23] combined the DOA method with the correlation methods to form a robust method for solving the permutation problem which is suitable for all the cases except when the sources are very close to each other or collinear.

In Section 2, the problem of BSS is introduced and Section 3 summarizes the drawbacks of the existing methods and a new method is proposed in Section 4, which is applicable to collinear sources also, provided a time domain BSS method which can separate the signals at least partially so that the spectra of the separated signals will be closer to those of the sources. Section 5 summarizes the experimental results and conclusions are drawn in Section 6.
The problem of BSS can be explained as follows. Suppose there are \( P \) sources mixed convolutively to obtain \( Q \) mixtures, then the \( Q \) mixtures at the sensor output can be written as

\[
x(t) = H * s(t)
\]

where \( x(t) = [x_1(t), x_2(t), \cdots, x_Q(t)]^T \) are the sensor outputs and \( s(t) = [s_1(t), s_2(t), \cdots, s_P(t)]^T \) are the mutually independent sources. The superscript \( T \) represents the matrix transpose and the operator \( * \) denotes the convolution. The matrix \( H \) is the mixing matrix of order \( Q \times P \) whose \( qp^{th} \) element, \( h_{qp}(l) \) is the impulse response from source \( p \) to sensor \( q \) so that

\[
x_q(t) = \sum_{p=1}^{P} \sum_{l=0}^{\infty} h_{qp}(l)s_p(t-l), \quad \text{for } q = 1, \cdots, Q.
\]

Even though in real acoustic environments the length of the mixing filter is infinity, the filter coefficients will decay to negligibly small values after a certain time and hence the length of the unmixing filter can be limited to a finite value.

Unlike the blind deconvolution, the objective of BSS is to separate the sources \( s(t) \) from the mixtures \( x(t) \) in such a way that they are mutually independent [26]. Hence the separated signals \( y_r(t), r = 1, \cdots, P \) will be the filtered version of the original sources \( s_\Pi(r)(t) [23] \)

\[
\begin{array}{c|c}
\text{Mixing filter} & \text{Unmixing filter} \\
\hline
s_1 & h_{11} & h_{12} \\
h_{21} & h_{22} & \hline
s_2 & w_{11} & w_{12} \\
w_{21} & w_{22} & y_1 \\
\hline
x_1 \oplus & y_1 \oplus \\
\hline
x_2 \oplus & y_2 \oplus \\
\end{array}
\]

Fig. 1. BSS for convolutive mixtures in the case of two sources and two sensors
\[ y_r(t) = \sum_l \alpha_r(l) s_{\Pi(r)}(t-l) \]  

where \( \alpha_r(l) \) is a filter, \( \Pi : \{1, \cdots, P\} \rightarrow \{\Pi(1), \cdots, \Pi(P)\} \) is the permutation and \( s_{\Pi(r)}(t) \) represents \( r^{th} \) signal with permutation. Since the mixing filters can be considered as finite length filters, the sources can be separated using the unmixing filters of finite length \( L \) and hence the output will be

\[ y(t) = W \ast x(t) \]  

where \( y(t) = [y_1, y_2, \cdots, y_P]^T \) are the separated signals and \( W \) is a \( P \times Q \) matrix of FIR filters with elements \( w_{rq}(l), r = 1, \cdots, P, q = 1, \cdots, Q \) and \( l = 0, 1, \cdots, L \). The mixing and unmixing processes are depicted in Fig.1, for the case of two sources and two sensors. Using the convolution multiplication property of DFT, (1) and (3) can be written as shown in (4) and (5).

\[ X(f,t) = H(f) S(f,t) \]  

\[ Y(f,t) = W(f) X(f,t) \]

where \( X(f,t), S(f,t) \) and \( Y(f,t) \) are the discrete time-frequency transformation vectors of \( x(t), s(t) \) and \( y(t) \) respectively. \( H(f) \) and \( W(f) \) are respectively the mixing and unmixing filters in the frequency domain. By observing (5), we can see that, conceptually any linear ICA algorithm for complex numbers can now be applied to each DFT bin. In this paper, the information maximization approach [6] combined with natural gradient (NG) [3] and FastICA algorithms [7] are used in each DFT bin. The information maximization
combined with NG algorithm is used to initialize the unmixing matrix for the
FastICA algorithm, which can improve the separation performance. The NG
algorithm is initialized with unit matrix.

It is a well known fact that, in BSS the sources can at best be obtained up to
a scaling and permutation [27], i.e.,

\[ W(f)H(f) = \Gamma(f)D(f) \] (6)

where \( \Gamma(f) \) is the permutation matrix and \( D(f) \) is the scaling diagonal matrix.

Many methods have been proposed to solve the scaling problem. In this paper,
the method suggested in [15] is used. The permutation problem in the time
domain method, called global permutation, is not a serious issue, because it is
not affecting the quality of the separated signals. In contrast, the permutation
problem in the frequency domain, called local permutation, is complicated as
it is in each of the frequency bins, which will affect the separation performance.
Hence the permutations in each of the DFT bins is to be aligned in such a
way that the separated signals in the time domain will contain the frequency
components of the same source signals. The following section summarizes the
drawbacks of the two popular methods, out of the many existing methods.

3 Drawbacks of the existing methods

The most commonly used approaches for solving the permutation problem
are the DOA approach and the correlation approach. For the convenience of
readers, the basic principles and pitfalls of these methods are summarized
3.1 Direction Of Arrival approach

The Fourier transform of the impulse response from source $s_p(t)$ at direction $\theta_p$ to the $q^{th}$ sensor, $h_{qp}(t)$, if we neglect the room reverberation, can be approximated as [14,22,23]

$$H_{qp}(f) = e^{j2\pi f \tau_{qp}} \quad \tau_{pq} = \frac{1}{c} d_q \sin \theta_p$$

(7)

where $\tau_{qp}$ is the time lag for the source with respect to the $q^{th}$ sensor placed at position $d_q$ (assuming that the sensors are placed in a linear array and the direction orthogonal to the array is $0^\circ$) and $c$ is the velocity of the signal. Using (4) and (5), the frequency response from $p^{th}$ source to $r^{th}$ separated signal can

![Diagram](image_url)

Fig. 2. Directivity pattern of the two sources shown in Fig.5 at two different frequencies.
be written as [14,22,23]

\[
U_{rp}(f) = \sum_{q=1}^{Q} W_{rq}(f) H_{qp}(f) = \sum_{q=1}^{Q} W_{rq}(f) e^{j2\pi fc^{-1}d_{q}\sin\theta_{p}}
\]  

If we consider the direction \( \theta_{p} \) as a variable, say \( \theta \), the modified equation will be [14,22,23]

\[
U_{r}(f, \theta) = \sum_{q=1}^{Q} W_{rq}(f) e^{j2\pi fc^{-1}d_{q}\sin\theta}
\]

The gain \( |U_{r}(f, \theta)| \) will vary according to the direction of the angle \( \theta \) and hence it is called the directivity pattern. The value of the gain \( |U_{r}(f, \theta)| \) will be sufficiently high when \( \theta \) is equal to the direction of the source corresponding to the separated signal \( y_{r} \) and low in the directions of the other sources. Hence the directions of all the sources can be estimated by the rows of the unmixing filter, \( W_{rq}(f), r = 1, \cdots, P \), for all the frequency bins. The permutation problem in each bin can be fixed by proper clustering of the above estimated directions. However, as shown in Fig.2, for lower frequencies we will not be able to estimate the directions of the sources and hence we are unable to fix the permutation for these bins. Also, under heavy reverberant conditions we may not be able to solve the permutation problem for all the frequency bins, even for the higher frequencies and hence, the permutations in these bins have to be fixed by some other methods. Another major disadvantage of using DOA method is that, when the sources are collinear or at very small angle,
the method will fail.

3.2 Correlation approach

For speech signals there is a strong correlation between the adjacent frequency bands. This inter-frequency correlation is utilized to solve the permutation problem in correlation methods (see [15] for more details). In [23], the algorithm proposed in [15] is simplified so that the envelop $v_f^f(t) = |S_{\Pi(f)}(f, t)|$ is used to measure the correlation between the neighboring bands. The correlation between the envelops in the neighboring frequency bins will be high, if the separated signal belongs to the same source signal. If $\Pi_f$ is the permutation corresponding to $\Gamma^{-1}(f)$, the permutation at frequency $f$, can be estimated by maximizing the sum of the correlation between the neighboring frequencies within the frequency distance $\delta$ [23].

$$\Pi_f = \arg \max_{\Pi} \sum_{|g-f| \leq \delta} \sum_{r=1}^{P} \text{cor}\left(v_{\Pi(r)}^f, v_{\Pi_g(r)}^g\right)$$

(10)

where $\Pi_g$ is the permutation at frequency $g$. The correlation between two signals $x(t)$ and $y(t)$ is defined as [23]

$$\text{cor}(x, y) = \frac{(\mu_{xy} - \mu_x\mu_y)}{\sigma_x\sigma_y}$$

(11)

where $\mu_x$ is the mean and $\sigma_x$ is the standard deviation of $x$. The main disadvantage of this method is that a mistake in one frequency bin will lead to complete misalignment beyond that frequency bin.
In the work of H. Sawada et al [23], the DOA and correlation approaches were combined so that the permutation is fixed for certain frequency bins where the confidence of DOA method was sufficiently high. For the remaining bins the permutation was decided based on the correlation between the neighboring or harmonic frequency bins without changing the permutation fixed by the DOA method. Harmonic correlation method utilizes the fact that, for speech signals, the envelop $v_f(t)$ at frequency $f$ has strong correlation to that at its harmonic frequencies $2f, 3f$ and so forth.

Hence, if the permutation is not fixed for frequency $f$, but fixed for its harmonic components, then the permutation at $f$, $\Pi_f$, can be fixed by maximizing the correlation between $v_f(t)$ at frequency $f$ and its harmonic bins [23], i.e.,

$$\Pi_f = \arg \max_{\Pi} \sum_{g \in (\text{Harmonics of } f)} \sum_{r=1}^{P} \text{cor}(v_{\Pi(r)}, v_{\Pi_g(r)})$$  (12)

The reason for using the harmonic correlation method in addition to the neighboring bands correlation method is that, the DOA approach does not provide sufficient confidence to fix the permutation, continuously for a certain range of frequencies, which is at the lower frequency bins. In such case, the use of neighboring correlation method alone may result in consecutive misalignment.
The proposed system basically consists of two blocks, namely, the time domain and frequency domain blocks. The time domain block partially separates the signal mixture using the time domain algorithm for BSS, whereas the frequency domain block separates the signals using the frequency domain BSS algorithm. The permutation problem in the DFT bins of the frequency domain BSS block is solved by utilizing the correlation between the envelop, $|\hat{S}_q(f, t)|$, in the DFT bins of the partially separated signal (partially separated signal is converted to the DFT domain time series signals, $\hat{S}_q(f, t)$) and that of the frequency domain method.

The input to the frequency domain stage can either be the output from the microphones or the partially separated signals from the time domain stage. Correspondingly there are two configurations for the proposed system, namely the parallel configuration [20] and the cascaded configuration [21]. The detailed explanations for each of these configurations are given in the following sections. Before proceeding to the next section, the time domain BSS algorithm used in this paper is summarized below for the convenience of the readers.

The time domain algorithm used in this paper is the computationally efficient implementation [1] of the algorithm proposed in [8,9]. The algorithm is based on the second order statistics which utilizes the non-stationarity and non-whiteness property of the speech signals. The cost function for the algorithm is [8,1,9].
\[ J(m) = \sum_{i=0}^{m} \beta(i,m) \left\{ \log \left( \det \left( \text{bdiag} \left( Y^T(i) Y(i) \right) \right) \right) - \log \left( \det \left( Y^T(i) Y(i) \right) \right) \right\} \] (13)

where \( \beta \) is the weighing function, \( m \) is the block index (the speech signal is divided into different blocks) and bdiag represents the block diagonal operation. \( Y(m) = [Y_1(m), \cdots, Y_P(m)] \) is the block output signal matrix, the columns of \( Y_r(m) \) contains blocks of the \( r \)-th output signal, \( y_r(t) \), of length \( N \) samples and each column is delayed by one sample as

\[
Y_r(m) = \begin{bmatrix}
y_r(ml_1) & \cdots & y_r(ml_1 - L_t + 1) \\
y_r(ml_1 + 1) & \cdots & y_r(ml_1 - L_t + 2) \\
\vdots & \ddots & \vdots \\
y_r(ml_1 + N - 1) & \cdots & y_r(ml_1 - L_t + N)
\end{bmatrix}
\] (14)

where \( N \) is the length of the output block and \( L_t \) is the length of the time domain unmixing filter. Hence the size of the matrix \( Y(m) \) will be of the order of \( N \times L_t P \). The natural gradient of (13) with respect to the unmixing filter in time domain, \( W_t \), gives [1,9]:

\[
\nabla^{NG}_{W_t} J(m) = 2 \sum_{i=0}^{m} \beta(i,m) W_t(i) \left\{ R_{YY}(i) - \text{bdiag} \left( R_{YY}(i) \right) \right\} \text{bdiag}^{-1}(R_{YY}(i))
\] (15)

where the \( L_t P \times L_t P \) correlation matrix \( R_{YY} \) consists of the correlation ma-
trices $\mathbf{R}_{y_p y_q}(m) = \mathbf{Y}_p^T(m) \mathbf{Y}_q(m)$ of size $L_t \times L_t$. When the output signals are mutually independent $\mathbf{R}_{y_p y_q}(m) = 0$, for $p \neq q$. Since the temporal correlation of the speech signal is also taken into consideration, the algorithm is free from whitening problem. The direct computation of the (15) is complex but an efficient and approximated version of the algorithm, which is given in [1], is used in this paper.

### 4.1 Parallel configuration

The block diagram of the proposed method for the parallel configuration with two sources is shown in Fig.3. For simplicity, we assume that the number of sources is equal to the number of mixed signals, i.e., $P = Q$. The time domain signals at the microphone outputs are first converted to the frequency domain time series signals, $X_q(f, t)$, $q = 1, \ldots, Q$, using $K$ point DFT. Complex valued ICA algorithm for linear mixtures is then applied to all the $K$ frequency

![Block diagram for the proposed method (parallel configuration).](image-url)
bins so as to obtain the separated signals in each bin. The separated signals in
different frequency bins may have different permutations that have to be solved
before they are combined by the inverse discrete Fourier transform (IDFT) to
obtain the separated signals, $y_r(t), r = 1, \ldots, P$. To solve this permutation
problem, we use partially separated signals, $\hat{s}_p(t), p = 1, \cdots, P$, separated
in the time domain as shown in Fig.3. The listening quality of the partially
separated signal need not be very good but the spectra of the separated signals
must be closer to the corresponding sources.

Each partially separated signal in the time domain, $\hat{s}_p(t)$, is then transformed
to the frequency domain using DFT of the same length as that for the micro-
phone output, i.e., $K$. Now $k^{th}$ bin of $p^{th}$ partially separated signal, $\hat{S}_p(f_k, t)$,
will be more correlated to the corresponding bin of the fully separated signal
$S_{\Pi(p)}(f_k, t)$. So the frequency bin among, $S_{\Pi(q)}(f_k, t), q = 1, \cdots, P$, having the
highest correlation with $\hat{S}_p(f_k, t)$ is identified, which is $S_{\Pi(q)}(f_k, t)$, for $q = p$,
and is assigned to $Y_p(f_k, t)$. Since the adjacent bins of a speech signal are
highly correlated, instead of taking a single bin from the partially separated
signal, the average of $k^{th}$ and its adjacent bins is used. Hence the permutation,
$\Pi_{f_k}(p)$, for the $k^{th}$ bin is

$$
\Pi_{f_k}(p) = \arg\max_{\Pi(q), q=1, \cdots, P} \text{cor}
\left(\frac{1}{2\Delta k + 1} \sum_{b = k - \Delta k}^{k + \Delta k} \left| \hat{S}_p(f_b, t) \right|, \left| S_{\Pi(q)}(f_k, t) \right| \right)
$$

The same procedure is repeated for all $\hat{S}_p(f_k, t), p = 1, \cdots, P$, and $k =
1, \cdots, K$. Subsequently, these assignments $Y_p(f_k, t)$ have to be converted back
to the time domain signal $y_p(t)$ using IDFT. One problem with this method
is that we may not be able to solve the permutations for all the bins, with full confidence, which can be explained as follows:

Let $c_{pq,k}$, $p, q = 1, \cdots, P$, be the correlation between $k^{th}$ bin of the $p^{th}$ partially separated signal and the $q^{th}$ fully separated signal. For $P$ sources there will be $P \times P$ such correlations for each bin. Considering $c_{pq,k}$ is an element in the $p^{th}$ row and $q^{th}$ column of a $P \times P$ matrix and if the highest $P$ values are distributed in that matrix in such a way that there is no column or row, common to each other, we can say that each partially separated signal has one and only one correlation with the fully separated signal. In that case $S_{\Pi(q)}(f_k, t)$ is assigned to $Y_p(f_k, t)$ with confidence. If not, leave that bin as it is and use the correlation between neighboring or harmonic bins to solve the permutation problem. For these bins also, permutations are fixed after checking the confidence. To check the confidence we use the same procedure that has been explained above. For the remaining bins, the neighboring correlation approach [23], which is shown in (10), is used to solve the permutation problem.

For clarification the confidence checking procedure can be illustrated with an example as follows: Let the number of sources be $P = 2$ and the correlations at bin $k$ are $c_{11,k} = 0.6$, $c_{22,k} = 0.7$, $c_{12,k} = 0.3$ and $c_{21,k} = 0.2$. Here, the highest two correlations are $c_{11,k}$ and $c_{22,k}$, which are the diagonal elements, if we consider the correlations as the elements of a matrix and hence there is no common row and column for the highest elements. In this case we can solve the permutation problem with confidence. Instead, if $c_{11,k} = 0.8$, $c_{22,k} = 0.1$, $c_{12,k} = 0.3$ and $c_{21,k} = 0.4$, we leave that bin unaltered during the first round of solving the permutation (the round which uses confidence check), because the highest two correlations, $c_{11,k}$ and $c_{21,k}$, lie in the same column of the matrix. During the second round, where we use the correlation without confidence
check, we only check either $c_{11,k} + c_{22,k} \geq c_{12,k} + c_{21,k}$ or $c_{11,k} + c_{22,k} < c_{12,k} + c_{21,k}$, and solve the permutation problem accordingly, which is the method adopted in [23]. Hence the proposed method consists of the following three steps:

- Fix the permutation using the correlation between the partially separated signal and the fully separated signal for the bins where it can be fixed with confidence.
- Fix the permutation either using the adjacent bins correlation or harmonic correlation method with confidence, without changing the permutation fixed in the previous step.
- For the remaining bins, fix the permutation using (10).

### 4.2 Cascaded configuration

Certain time domain algorithms will distort the spectrum of the separated signals. For example, the algorithm which does not consider the temporal correlation of the speech signal (e.g. [2]) may whiten the spectrum of the sep-

![fig4](image-url)

Fig. 4. Block diagram for the proposed method (cascaded configuration).
arated signal. For such time domain algorithms the parallel configuration will be better [20], otherwise the output of the frequency domain stage also will remain distorted because of its distorted input. On the other hand, if the algorithm is not distorting the spectrum of the separated signals, the cascaded configuration will be the best option as it not only reduces the computational cost but also improves the overall separation performance. The block diagram for the cascaded configuration is shown in Fig.4. By comparing the two block diagrams, parallel and cascaded, it can be seen that the number of fast Fourier transform (FFT) blocks for the cascaded configuration is only 2, whereas for the parallel configuration is 4. Hence the computational cost is reduced. Since the input signal to the frequency domain stage in the case of cascaded configuration is the partially separated signals, which is already separated to a certain level, the overall performance is higher than that of the parallel configuration. In [16,17] it is shown that the cascaded configuration of the time domain and the frequency domain stages will improve the separation performance. Unlike in [16,17], where the DOA method is used to solve the permutation problem in the frequency domain stage, the proposed algorithm uses the signals from the time domain stage to solve the permutation problem and hence the computational cost is optimally utilized.

5 Experimental results

For performance analysis of the proposed algorithms, both simulated and measured room impulse responses are used. The simulated room impulse responses are used in Sections 5.1 and 5.2. The wall reflections up to $10^{th}$ order is taken and humidity, temperature, absorption of sound due to air etc., are consid-
Table 1
Experimental Conditions

<table>
<thead>
<tr>
<th>Source signals</th>
<th>Speech of 15sec, except in Section 5.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direction of sources</td>
<td>As shown in the respective figures</td>
</tr>
<tr>
<td>Distance between</td>
<td>20cm for PS and 4cm for DOA</td>
</tr>
<tr>
<td>two microphones</td>
<td></td>
</tr>
<tr>
<td>Sampling rate $f_s$</td>
<td>16kHz</td>
</tr>
<tr>
<td>Frame size of DFT $K$</td>
<td>4096</td>
</tr>
<tr>
<td>Frequency resolution</td>
<td>$\Delta f = f_s/K = 3.90625$Hz</td>
</tr>
<tr>
<td>Distance $\Delta k$</td>
<td>$18\Delta f$</td>
</tr>
<tr>
<td>Distance $\delta$</td>
<td>$9\Delta f$</td>
</tr>
<tr>
<td>Number of filter taps for time domain algorithm</td>
<td>512 (unless otherwise specified)</td>
</tr>
</tbody>
</table>

![Diagram of source-microphone configuration for the room impulse responses simulation.](image)

Fig. 5. The source-microphone configuration for the room impulse responses simulation.

For calculating the impulse responses [28]. Whereas, in Sections 5.3, 5.4 and 5.5, the measured real room impulse response is used. For all the experiments, the average performance of 10 speech utterances are used to evaluate the performance. The algorithms are also tested with real recorded speech mixtures.
Fig. 6. Separation performance of the proposed method (reverberation time $T_{R_{60}} = 86$ms): Partial Separation approach PS, Correlation approach C1 and the combined approaches PS+C1, PS+C2+C1 and PS+C2+Ha+C1

Fig. 7. NRR at different frequencies for the 4th set of speech utterances in Fig.6.

5.1 Performance evaluation for collinear and non-collinear sources

The major disadvantages of the time domain BSS method for convolutive mixtures are the statistical interdependence among the filter coefficients, which
hinders the convergence, and the heavy computational cost involved for long filter taps [25]. Furthermore, good separation of convolutive mixture using a time domain method when the sources are collinear is a formidable task [10]. To show that the proposed algorithm can solve the permutation problem even when the quality of the partially separated signal is very poor, we simulated the impulse response between the sources and microphones for the configurations shown in Fig.5. The sources are placed in two different configurations: i) collinear (Sources 2 and 2’), for which the time domain separation is very poor ii) non-collinear (Sources 3 and 3’), for which the time domain separation is good. The mixed signals are generated by convolving the impulse responses obtained for the above configurations with the speech signals so that the performance can be evaluated by calculating the noise reduction rate (NRR) [16]

\[
NRR = \frac{1}{P} \sum_{q=1}^{P} \left( SNR_{q}^{(O)} - SNR_{q}^{(I)} \right)
\] (17)

\[
SNR_{q}^{(O)} = 10 \log_{10} \frac{\sum_{f} |A_{qq}(f)S_{q}(f)|^2}{\sum_{f} |A_{qp}(f)S_{p}(f)|^2}
\] (18)

\[
SNR_{q}^{(I)} = 10 \log_{10} \frac{\sum_{f} |H_{qq}(f)S_{q}(f)|^2}{\sum_{f} |H_{qp}(f)S_{p}(f)|^2}
\] (19)

where \(SNR_{q}^{(O)}\) and \(SNR_{q}^{(I)}\) are the output SNR and the input SNR respectively, \(q \neq p\). \(A_{qp}(f)\) is the \(q^{th}\) row and \(p^{th}\) column of the matrix \(A(f)\), as given in (20)

\[
A(f) = W(f)H(f)
\] (20)

For each configuration of the source positions shown, both parallel and cascaded configurations of the proposed methods are used to solve the permuta-
tion problem. In each case, the permutation is solved in five different ways:

- Partial Separation method (PS). Bins whose permutation could not be solved with confidence are left unaltered.
- Correlation between adjacent bins (C1) using (10)
- Combination of PS and C1 (PS+C1), i.e., PS followed by C1.
- PS followed by C2, that solves permutation by using correlation between adjacent bins with confidence, and finally followed by C1 (i.e., PS+C2+C1)
- Combination of PS and C2, followed by Ha which solves permutation using correlation between the harmonic components with confidence, and finally followed by C1 (i.e., PS+C2+Ha+C1).

Fig.6 shows the performance of the proposed method for both collinear and non-collinear sources. For clarity, only the average performance of the 10 sets of speech utterances are shown, except for the collinear sources in the parallel configuration where the performances of all the 10 sets of speech utterances are shown. For the parallel configuration, since the time domain NRR is not directly observable at the output, it is shown with ‘dash-dot’ line. As reported in [23] and from Fig.6, it can be seen that C1 can solve the permutation problem in most of the cases but it is not stable and sometimes it results in very poor performance. The performance of PS alone is not good but stable as PS solves the permutation for the frequency bins where the confidence is sufficiently high. However, for PS+C1, the performance is improved in terms of NRR compared to PS and stability compared to C1 alone. After solving permutation by PS followed by C2 and before C1, the performance is again improved. Also it can be seen that PS+C2+Ha+C1 offers almost the same performance as that without Ha. This is because Ha is used to solve permutation problem at lower frequencies where the problem is already solved by
PS method. The NRR measured at each frequency bin for a pair of speech utterances (set 4) using different methods are shown in Fig.7. It shows the effectiveness of the proposed method in solving the permutation misalignment problem. There are large regions of permutation misalignment when C1 alone is used. Since PS can provide correct permutation for certain frequencies in these regions, the problem is solved when PS is combined with C1. Unlike the DOA approach, where it is very hard to estimate the direction of arrival at lower frequencies and hence difficult to solve the permutation with confidence, PS solves the permutation problem almost uniformly irrespective of the frequencies as shown in Fig.7. Note that the case shown in Fig.7 is for the worst case, where the NRR for the time domain method is only 3.9812dB and also it is for the parallel configuration. From Fig.6, it can be seen that even for collinear sources with poor separation in time domain, if we use cascaded configuration the performance will be significantly better. In Fig.6 for the cascaded configurations, the performance shown for the correlation method alone (C1) is better than the all other methods, because C1 was successful for all the 10 sets of speech utterances. Whereas for the parallel configuration, the method failed for some of the sets. This is not because of the configuration but because of the reason that the correlation method alone is highly unreliable.

5.2 Performance evaluation under different reverberation times

For further analysis, the performance of the algorithm is compared with the DOA method. Since DOA utilizes the direction of arrival of the signal for solving permutation problem, the distance between microphones must be less than \( c/2f \), where \( c \) is the velocity and \( f \) is the frequency of the signal, to avoid
spatial aliasing. However, when the microphones are very close, the problem reduces to the single channel mixing case. For the time domain method used in this paper more spacing between the microphones is required to achieve separation. Table 2 shows the performance of the time domain and DOA method for different microphone spacings, while keeping all other parameters such as the position of the sources, central point of the microphones, room surface absorption, size of the room etc., same as that in other experiments. Since the performance of the PS method depends on the quality of the partially separated signal, the NRR of the time domain method is shown instead of the NRR of the PS method. Also while calculating the NRR for the DOA method, the bins where the DOA method could not solve the permutation with confidence are left unaltered. This is because, the robustness of the DOA method followed by correlation methods depends on the success of the DOA method alone in solving the permutation problem. From the table we can see that 4cm is the optimum distance for the DOA method. For the PS method 20cm is taken. Note that this comparison is done only to find out the optimum microphone distance for the DOA method. Since the distances between the

Table 2
NRR for the time domain method and DOA method for different microphone spacings (Room surface absorption = 0.5)

<table>
<thead>
<tr>
<th>Spacing (cm)</th>
<th>Time domain (dB)</th>
<th>DOA (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.9785</td>
<td>6.3786</td>
</tr>
<tr>
<td>4</td>
<td>2.7985</td>
<td>6.8414</td>
</tr>
<tr>
<td>8</td>
<td>4.2190</td>
<td>6.2397</td>
</tr>
<tr>
<td>12</td>
<td>6.2419</td>
<td>5.2858</td>
</tr>
<tr>
<td>16</td>
<td>7.8860</td>
<td>4.3331</td>
</tr>
<tr>
<td>20</td>
<td>8.2101</td>
<td>3.6543</td>
</tr>
</tbody>
</table>
Fig. 8. Room impulse response for different values of surface absorption: 0.3 ($TR_{60} = 235\text{ms}$), 0.5($TR_{60} = 130\text{ms}$), 0.7($TR_{60} = 86\text{ms}$) and 0.9($TR_{60} = 63\text{ms}$). Only the impulse responses from Source 1 to Microphone 1 are shown.

Fig. 9. Performance comparison of PS method alone with DOA method alone as a function of room surface absorption.

For the experiments in this section, room impulse response is simulated [28] for different values of room surface absorption. The impulse responses thus obtained are shown in Fig.8. The other experimental conditions are shown in Table. 1. The positions of the sources are shown in Fig.5 (Sources 1 and 1').
Fig. 10. Performance comparison of PS method alone without confidence check with PS method after confidence check followed by the methods which utilizes the correlation between adjacent and harmonic bins, for parallel and cascaded configurations. The DOA method after confidence check followed by correlation methods are also shown.

The performance of the two methods, PS and DOA, when used alone for different values of reverberation time are shown in Fig.9. For both the cases of ‘DOA only’ and ‘PS only with confidence check’ the bins whose permutation could not be fixed with confidence are left unaltered.

Unlike the correlation approach discussed in Section 3.2, which utilizes the inter-frequency correlation of the signal for permutation alignment [15], the PS method utilizes the correlation with the partially separated signal for solving permutation. So, even if the permutation fixed in one bin is wrong, the permutation alignment in the remaining bins will not be affected by the wrongly aligned one. Hence the reliability of the method is high and we can fix the permutation in all the bins without confidence check. This is shown as ‘PS only without confidence check’ in Fig.9, where the permutation of all the bins are fixed without checking the confidence as
Fig. 11. The source-microphone configuration for the measurement of real room impulse response

\[ \prod_{f_k} = \arg \max_{\Pi} \sum_{p=1}^{P} \text{cor} \left( \frac{1}{2\Delta k + 1} \sum_{b=k-\Delta k}^{k+\Delta k} |\hat{S}_p(f_b, t)|, |S_{\Pi(p)}(f_k, t)| \right) \] (21)

for all the bins, where \( \prod_{f_k} \) is the permutation at \( k^{th} \) frequency bin and \( 2\Delta k + 1 \) is the total number of adjacent bins of the partially separated signal taken to obtain the average. From Fig.10, it can be seen that the difference in performance of PS alone without confidence check (‘PS only’) and PS with confidence check followed by the methods utilizing correlation between adjacent and harmonic bins (‘PS followed by others’) are very small. Therefore, the PS method alone can be used to solve the permutation problem which reduces the computational cost at the expense of very small performance reduction.

In Fig.10, the performance of the DOA method followed by other correlation methods (‘DOA followed by others’) are also shown for comparison. The reasons for the high deviation in performance of the time domain method, for the changes in room reverberation time, as shown in Fig.9 and 10 are explained in Section 5.5.
Fig. 12. NRR for various algorithms using real room impulse response. PS - Partial Separation method with confidence check, C1 - Correlation between the adjacent bins without confidence check, C2 - Correlation between adjacent bins with confidence check, Ha - Correlation between the harmonic components with confidence check, PS1 - Partial separation method alone without confidence check.

5.3 Performance evaluation using the measured real room impulse response

The performance of the PS method is also evaluated using the measured impulse response of a real furnished room. The reverberation time of the room ($T_{R_{60}}$) is 187ms and is measured with the help of an acoustic impulse response measuring software “Sample Champion” [29]. The microphone and loud speaker transfer function are neglected in the measurement. The positions of the sources and sensors are shown in Fig.11. The other experimental conditions are same as those used in the previous sections. The calculated performances are shown in Fig.12, from where it can be seen that the performance of PS alone without confidence check is very close to that of PS+C2+Ha+C1. Therefore the PS method alone, which is reliable and independent of the positions of the sources and better than the DOA approach, can be used to solve the permutation problem in frequency domain BSS of speech signals. Note that the impulse responses for the PS method and DOA methods are not ex-
Fig. 13. NRR for different lengths of speech utterances when the NRR of the partially separated signals used for solving permutation problem are of different levels. Actually the same because of the difference in microphone spacing and hence, for C1, the NRRs are different.

5.4 Robustness test for short speech utterances

The accuracy of the correlation method for solving the permutation problem depends on the length of the speech utterances [22,14] because as the speech length becomes shorter they tends to resemble each other. Since the proposed method is based on the correlation between the partially separated and fully separated signals in the frequency bins, the robustness of the algorithm for shorter speech segments is tested in this experiment. The algorithm is tested for 1, 2, 3, 4, 5 and 10 seconds speech utterances using different levels of partial separation (2, 4, 6 and 8 dB). Fig.13 shows the test results and it can be seen that the algorithm works even for smaller speech lengths and 3 sec speech is sufficient to get a good separation. For these experiments the length of the data in the frequency bins is fixed at 500 for all speech lengths, by adjusting
the overlapping of the sliding window, while converting into frequency domain. The unmixing filter length of the time domain stage was 512 in all the cases and the number of iterations are adjusted to get different levels of partial separation.
5.5 Effect of combination order in cascaded configuration

In [16,17] the time domain and frequency domain stages were cascaded for improving the NRR. Even though the objectives of [16,17] were not to solve the permutation problem, to improve the NRR, we study the performance difference between the orders of the time domain and frequency domain stages when they are swapped. To find out the optimum filter length for the time domain stage in the proposed cascaded configuration, we have conducted experiments for different values of filter lengths. The result are shown in Fig.14. To model the complete reflections in the room, the length of the unmixing filter must be at least equal to the length of the mixing filter [9]. From Fig.14 it can be seen that, smaller the filter length, poorer the performance. This is because of the reason that the length of the unmixing filter is less than the required. However as the unmixing filter length increases due to the interdependency of the filter coefficients [25], the convergence will become poorer, which is ev-
ident from the Fig.14. As the filter length increases from 64 to 512, the NRR is increasing and then decreasing. Here, out of the different filter lengths used, the best performance is achieved at 512. Hence for the remaining parts of the experiments 512 taps are used for the time domain stage, unless otherwise it is specified. The NRR for the cascaded configuration of frequency domain stage followed by the time domain stage with different values of filter length for the time domain stage is shown in Fig.15, the 100% data learning case (the remaining cases will be explained later). For the frequency domain stage the permutation problem is solved using the proposed parallel configuration with only a difference that, instead of using the partially separated signals, the clean signals are used to solve the permutation problem in the best possible way, which is the ideal case. The results showed that for the cascaded configuration with frequency domain stage followed by the time domain stage, the NRR is poorer than that of the frequency domain stage alone. This is due to: i) For further separation, after the frequency domain separation, fine tuning of the unmixing filter coefficients with longer filter taps is needed. But as discussed above, when the filter taps increases, because of the interdependency of the filter coefficients, the convergence is poor. ii) In the frequency domain stage even if we perfectly separate the signals in all the bins, fails to completely solve the permutation problem, the resultant mixing will be too complex for the time domain algorithm to separate and is even worse, if the signals in the frequency bins remain mixed in addition to the permutation. This is clear from the simulation results shown in Fig.16, where in one case the clean signals are first converted into frequency domain and every $B^{th}$ bin is permuted, where $B = 2^b$, $b = 2, 3, \ldots, 11$, so that the signals in all the bins are fully separated and then after converting into time domain they are mixed signals, because of the permutation. In the second case, instead of using clean signals, the mixed
(convolutive mixing using real room impulse response) signals are used so that after converting into the time domain, the resultant signals are more complex mixtures than that of the simple convolutive mixtures. When applied these two sets of signals to the time domain algorithm, it is found that even for the case where the mixing was only because of the permutation, the separation is very difficult and when the mixing is because of the convolutive as well as that of the permutation, the resultant is worse. In Fig.16, for smaller values of $B$, the NRR for the “mixture permuted” (permuted using mixed signals) signals is higher than that of the “clean permuted” (permuted using clean signals) signals. This is because, at the $k^{th}$ frequency bin, $X_1(f_k,t)$ and $X_2(f_k,t)$ are more dependent than $S_1(f_k,t)$ and $S_2(f_k,t)$ [4], where $X_n(f_k,t)$ and $S_n(f_k,t)$ are the data in the $k^{th}$ frequency bin for $n^{th}$ mixed and clean signals respectively. Hence the mixing effect caused due to the permutation of $S_1(f_k,t)$ and $S_2(f_k,t)$ is more than that of permuting $X_1(f_k,t)$ and $X_2(f_k,t)$.

In the cascaded configuration of frequency domain stage followed by the time domain stage, even if the permutation is fully solved and the signals are not completely separated in the frequency bins, the time domain stage requires fine tuning with long filter taps, which is a difficult task because of the reasons mentioned in Section 5.1. Moreover, if the separation in each bin is not perfect, the chance for getting wrong permutations is high and this will further worsen the result.

The length of the data in the frequency bin will affect the separation of the frequency domain stage, because, as the data length decreases the independence assumption will collapse [4] and hence the separation will be poor. Fig.15 also shows the variation in NRR for different data lengths. The maximum length of the data in each bin is 117 (15 sec speech, 16 kHz sampling frequency, 4098
point FFT and 50% overlap). Since the mixing filter is constant over the full
length of the speech signals, only a fraction of the data (20%, 40%, 60% and
80%) in the frequency bins is used for learning, i.e., to estimate the unmixing
filter. Then the full length data is separated using the estimated filter. As ex-
pected [4], smaller the data length for learning, poorer the separation, which
is shown in Fig.15. It can also be seen that the time domain stage could not
improve the separation further, for any of these cases. Hence we conclude that
it is better to place the time domain stage in front of the frequency domain
stage because i) the time domain block used to generate the partially sepa-
rated signals for solving permutation problem can be fully utilized, otherwise
we need a separate time domain block, only for solving permutation ii) The
overall performance is better than that of the individual blocks iii) Efficient
and approximate algorithms like [1] can be used with smaller taps, as we need
only partial separation.

In [16,17], the performance of the time domain stage followed by the frequency
domain stage configuration was poor compared to its swapped configuration
because to initialize and to solve the permutation problem, direction of arrival
and beam forming techniques were used. Hence when the time domain block
was in the first stage the directional information was not available for the
frequency domain stage.

6 Conclusions

We have proposed a method for solving the permutation problem in frequency
domain BSS of speech signals. The method uses correlation between the signals
in each frequency bin, one of which is partially separated by a time domain
BSS method and the other is obtained by a frequency domain BSS method. The algorithm does not require the knowledge of the directions of arrival of the sources. Hence, if the time domain BSS algorithm can partially separate the signals so as to make the spectrum of the resultant signals closer to their corresponding sources, the proposed method can be used to separate collinear sources also. Unlike the correlation method which utilizes the inter-frequency correlation of the signals, the reliability of the proposed method is high as it utilizes the correlation with partially separated signals. The additional computational cost of the proposed method in the cascaded configuration is due to the time domain stage. Since it is cascaded with the frequency domain stage, the performance improvement is due to the correctly fixed permutation as well as due to the achievable cascaded configuration. Hence, the additional computational cost is optimally utilized.

References


fundamental limitation of frequency domain blind source separation for
convolutive mixtures of speech, IEEE Trans. Speech Audio Process., 11(2)

of array processing and independent component analysis for blind separation of


component analysis of complex-valued signals, Int. J. of Neural Systems 10(1)

source separation algorithms for convolutive mixtures, In Proc. ICA (2003) 945-
950.

separation algorithms for convolutive mixtures based on second-order statistics,
IEEE Transactions on speech and audio processing 13(1) (2005) 120-134.

placement in blind source separation, IEEE Workshop on Applications of Signal
Processing to Audio and Acoustics (2001) 95-98.

alignment for multichannel frequency domain blind speech separation, In Proc.

separation: Investigation and solutions, IEEE Transactions on speech and audio
543-548.

of blind signal separation method using directivity pattern under reverberant


[16] T. Nishikawa, Blind source separation based on multistage independent

signal based on multistage ica combining frequency-domain ica and time-domain

[18] L. Parra and C. Spence, Convolutive blind separation of non-stationary sources,

[19] K. Rahbar and J. P. Reilly, A frequency domain method for blind source
separation of convolutive audio mixtures, IEEE Transactions on Speech and

in blind source separation of speech signals: A method applicable for collinear

Permutation Problem in Frequency Domain Blind Source Separation of Speech

[22] H. Saruwatari, S. Kurita, K. Takeda, F. Itakura, T. Nishikawa, and K. Shikano,
Blind Source Separation Combining Independent Component Analysis and


