uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

QiongFeng Pan

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical Engineering)

GRADE / DEGREE

School of Information Technology Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Efficient Blind Speech Signal Separation using Independent Component Analysis

TITRE DE LA THÈSE / TITLE OF THESIS

Tyseer Aboulnasr

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Martin Bouchard

Rafik Goubran

Saeed Gazor

Wail Gueaieb

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Efficient Blind Speech Signal Separation using Independent Component Analysis

By

Qiongfeng Pan, MSc.

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the PhD degree in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

August 2006

Your file Votre référence
ISBN: 978-0-494-49387-8
Our file Notre référence
ISBN: 978-0-494-49387-8

Abstract

Blind speech signal separation has a wide range of potential applications in our life, such as speech enhancement for speech recognition, teleconference application, hearing aids etc. The ultimate aim of blind speech signal separation is to mimic the action of a human in a cocktail party situation, where our hearing system can focus on any specific audio source of interest while suppressing all other sources present even in noisy environments. Blind source separation (BSS) provides a good tool to approach this problem. In blind source separation, source signals are estimated using only information observed at receivers and the estimation is performed blindly, without information on source signals and the mixing system.

In this thesis, we concentrate on issues of relevance to convolutive blind speech signal separation based on the general frame work of Independent Component Analysis (ICA). Both time and frequency domain convolutive BSS algorithms are investigated in our thesis.

First we conduct our convolutive speech signal separation in the frequency domain. We propose a convolutive blind signal separation approach for joint speech signal separation and echo cancellation. Then, we suggest a simple means to using the psychoacoustic properties of human auditory system to improve the quality of separated speech signals.

Next, we propose to combine convolutive blind source separation with beamforming to deal with speech separation in heavy reverberation acoustic environment. By exploiting spatial information from beamforming, we maintain the speech separation performance with lower computational complexity.

Because of the inherent problems in frequency domain BSS algorithms, we investigate novel algorithms to improve the convergence and reduce the complexity of time domain convolutive BSS algorithm. We propose the application of MMax partial update algorithm to the time domain convolutive BSS (MMax BSS) to demonstrate that the partial update scheme applied in the MMax LMS algorithm for single channel can be extended to

multichannel time domain convolutive BSS with little deterioration in performance and possible computational complexity saving. Also we propose exclusive maximum selective-tap time domain convolutive BSS algorithm (XMax BSS) that reduces the interchannel coherence of the tap-input vectors and improves the conditioning of the autocorrelation matrix resulting in improved convergence rate and reduced misalignment. Moreover, the computational complexity is reduced since only half tap inputs are selected for updating.

献给我的妈妈易格容和爸爸潘明杰

Acknowledgements

I would like to give my big thanks to the following persons, as my thesis might not have been possible without them.

First of all, I would like to thank my supervisor Dr. Tyseer Aboulnasr for her persistent support and guidance through the course of this work. Especially, I would like to express my sincere appreciation of her great advice, feedback and financial support. She is a role model of mine even outside my research work.

I am also grateful to the committee members of my PhD program, for their insightful comments on the dissertation, and for their precious time and efforts as the committee members.

I gratefully acknowledge the financial support of the Ontario Ministry of Education and Training (Scholarship OGS), and the University of Ottawa (Admission Scholarship, Excellence Scholarship).

Finally, I give my great thanks to my Mom for her support and sacrifices. Without her help, I can not concentrate on my research. I also really appreciate the help, encouragement and understanding from my husband, Guangze, my son, Yi and my little daughter, Annie. Without their great love and endless support, I never could have finished my study. Many thanks also go to my father and my siblings for their constant support and encouragement throughout these years.

Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

x	variable
$\mathbf{x}$	vector
$\mathbf{X}$	matrix
$\mathbf{S}(f)$	signal in frequency domain
F_x	cumulative distribution function of random variable x.
$p_x(x)$	pdf of random variable x.
$\mathbf{m_x}$	mean vector of the random vector $\mathbf{x}$.
$\mathbf{R_x}$	correlation matrix of the random vector $\mathbf{x}$.
$\mathbf{C_x}$	covariance matrix of the random vector $\mathbf{x}$.
$\kappa_{x,k}$	cumulants of random variable x.
$\mathbf{x}^T$	transpose of a vector
$\mathbf{x}^{-1}$	inverse of a vector
$\mathbf{x}^H$	Hermitian (complex-conjugate transpose of a vector
$\mathbf{W}^T$	transpose of a matrix
$\mathbf{W}^{-1}$	inverse of a square matrix
$\mathbf{W}^H$	Hermitian (complex-conjugate transpose) of a matrix
kurt(.)	kurtosis
$\approx$	approximate to
$\propto$	proportional to
det(**A**)	determinant of a square matrix
$\|\|$	absolute value
$E\{.\}$	expectation
fft(.)	Fourier Transform operator
diag(.)	diagonal matrix

BSS	Blind Signal (Source) Separation
ICA	Independent Component Analysis
DOA	Direction of Arrival
PCA	Principal Component Analysis
ML	Maximum Likelihood
FIR	Finite Impulse Response
i.i.d.	Independent and identically-distributed
ISI	Intersymbol Interference
SIR	Signal to Interference Ratio
PESQ	Peceptual Evaluation of Speech Quality
MOS	Mean Opinion Score
LMS	Least Mean Square
SPL	Sound Pressure Level
FFT	Fast Fourier Transform
SNR	Signal to Noise Ratio

MUSIC	Multiple Signal Classification
FD-BSS	Frequency Domain Blind Source Separation
XMax BSS	Exclusive Maximum Blind Source Separation

Chapter 1 Introduction

1.1 Blind Source Separation

In the real world, we frequently encounter cases in which signal observations are mixtures of separate independent signals from different sources. It is desirable to process these observations such that the source signals can be extracted or separated. This problem is known as ***Blind Source (Signal) Separation (BSS).*** The term 'blind' refers to two aspects [Cardoso 1998]: (1) the source signals are not observed directly; (2) there is no information available about the mixing system. These weak assumptions make BSS a very powerful tool for modeling lots of situations in real environments.

The efforts to resolve the BSS problem can be traced back to a paper by Herault et al. in 1985 [Herault 1985]. The basic idea proposed in this paper is now referred as Independent Component Analysis (ICA). At that time, it did not attract much attention. In 1994, P. Comon published a paper on independent component analysis [Comon 1994], and in 1995, T. Bell and T. Sejnowski published a paper [Bell 1995] on the Infomax algorithm for blind signal separation based on the ICA in [Comom 1994]. By then, independent component analysis and blind signal separation began to become a popular research area and found many potential applications in a variety of diverse fields. Essentially, blind source separation and independent component analysis refer to the same problem and they can be used interchangeably under some conditions.

Blind source separation was initially used to deal with instantaneous linear mixtures. In this case, source signals are assumed to be mutually independent; the observed signals are the simultaneously mixed source signals without time delays and reverberation effects. Frequently, the number of sensors is assumed to be equal to the number of sources. This is the very basic blind source separation model and independent component analysis serves as the most important tool for this instantaneous BSS. Many methods [Amari 1996][Cardoso

1996][Hyvarinen 1997] have been proposed to deal with this problem and a unified theoretic framework [Cardoso 1998][Amari 1998] has been constructed for it.

Because of the limitations in the assumptions of basic instantaneous BSS model, it cannot be used to deal with many real world situations. Thus, it has been extended in several directions such as more sensors than sources, noisy blind source separation, non-stationary blind source separation etc. A very challenging extension is the convolutive blind source separation [Torkkola 1996][Smaragdis 1998] since it was proposed to deal with realistic situations by taking into account propagation delays and multipath effects.

BSS has a large number of potential applications to a variety of diverse signals, such as image processing, biomedical signal processing, financial data analysis, wireless communication etc. However, one very useful application is the blind source separation of speech signals.

In this thesis, we focus on blind speech signal separation in real-world environments by using convolutive BSS model since it provides a very useful tool to simulate realistic acoustic environment. The cocktail party problem is a real-time illustration of the blind speech signal separation problem. In a party, our ears can still focus on a specific sound source and separate it from all the sound sources presented in the room. All these tasks are conducted automatically by our ears and brain. In our research, we attempt to achieve automatic speech signal separation using a computer system by studying the way the humans tackle this problem and reproducing it as much as possible.

1.2 Thesis Overview

In this thesis, we explore approaches to blindly separate speech mixtures in convolutive environments using methods based on independent component analysis. Both frequency domain and time domain methods are investigated for speech separation.

In Chapter 2, we describe fundamental knowledge necessary for understanding blind speech signal separation. Related concepts in probability theory, statistical processes, optimization methods, estimation theory and information theory are reviewed. Principal component analysis is also described in this chapter since it is also a basic technique used to perform source separation.

In Chapter 3, the blind source separation problem is divided into two classes, instantaneous BSS and convolutive BSS, based on the nature of the mixing system. Both instantaneous and convolutive BSS system models and their corresponding estimation methods are described. Then, blind speech signal separation is emphasized and existing algorithms are reviewed.

In Chapter 4, we provide simulation results for some of the existing instantaneous and convolutive blind source separation algorithms in order to achieve a better understanding of blind source separation. We focus on algorithms that influence our approaches on source separation later on.

In Chapter 5, we propose a convolutive blind signal separation approach for joint speech signal separation and echo cancellation. A unifying blind source separation architecture for network and acoustic echo cancellation is constructed. The frequency domain convolutive blind source separation algorithm is exploited for speech separation and echo cancellation.

In Chapter 6, we propose to combine blind source separation and the psychoacoustic properties of human auditory system. First, we propose a post-filter based perceptual convolutive source separation system to deal with speech signal separation. Masking properties of human auditory system are exploited in the post-filter system to shape speech spectrum attempting to cancel interference remaining after convolutive blind source separation. Next, we propose a perceptual convolutive blind source separation algorithm by incorporating the absolute hearing threshold property into frequency domain convolutive

blind source separation algorithm. By emphasizing frequencies that human ears are sensitive to and deemphasizing frequencies that human ears are not sensitive to, or are even inaudible, we attempt to improve speech signal separation quality.

In Chapter 7, we propose to combine convolutive blind source separation with beamforming to deal with speech separation in heavy reverberation acoustic environments. First we propose a system with an adaptive beamformer cascaded with a BSS system. In the first stage, the adaptive beamformer is exploited to isolate signals from selected directions and reduce most of the reverberation effects. In the second stage, a convolutive blind source separation algorithm is used to further separate the remaining interference with low complexity and better convergence. Since we need some prior information for the adaptive beamforming system, we then propose a completely blind beamformer system cascaded with a BSS system, which can blindly estimate source signal directions and separate speech signal with low complexity and good separation performance compared with existing methods.

In Chapter 8, we investigate novel algorithms to improve the convergence and reduce the complexity of time domain convolutive BSS algorithm. First, we propose the application of MMax partial update algorithm to the time domain convolutive BSS (MMax BSS). We demonstrate that the partial update scheme applied in the MMax LMS algorithm for single channel can be extended to a multichannel time domain convolutive BSS with little deterioration in performance and a possible reduction in computation complexity. Next, we propose exclusive maximum selective-tap time domain convolutive BSS algorithm (XMax BSS) that reduces the interchannel coherence of the tap-input vectors and improves the conditioning of the autocorrelation matrix resulting in an improved convergence rate and reduced misalignment. Moreover, the computational complexity is reduced since only half tap inputs are selected for updating. Simulation results have shown a significant improvement in convergence rate compared to existing techniques.

In Chapter 9, we summarize the issues that we studied in this thesis and emphasize the contributions we have made. Then, we discuss some of the current open questions in blind speech signal separation.

1.3 Publications derived from this work

The following publications have been derived from this work.

- Qiongfeng Pan and Tyseer Aboulnasr, "Time Domain Convolutive Blind Source Separation Employing Selective-tap Adaptive Algorithms," Accepted as Invited Paper to EURASIP Journal on Audio, Speech and Music Processing 2007.

- Qiongfeng Pan, Tyseer Aboulnasr, "Blind Speech Signal Separation Combining Independent Component Analysis and Beamforming," to be submitted to EURASIP Journal on Audio, Speech and Music Processing 2007.

- Qiongfeng Pan and Tyseer Aboulnasr, "Time Domain Convolutive Blind Source Separation Employing Selective-tap Adaptive Algorithms," To be submitted to EUSIPCO 2007,

- Qiongfeng Pan and Tyseer Aboulnasr, "Blind Speech Signal Separation Combining Independent Component Analysis and Beamforming," To be submitted to EUSIPCO 2007

- Qiongfeng Pan and Tyseer Aboulnasr, "Combined Spatial/Beamforming and Time/Frequency Processing for Blind Source Separation," Invited paper for 13th European Signal Processing Conference, EUSIPCO 2005.

- Tyseer Aboulnasr and Qiongfeng Pan, "Data Dependant Partial Update Adaptive Algorithms for Linear and Nonlinear Systems", invited paper for 13th European Signal Processing Conference, EUSIPCO 2005.

- Qiongfeng Pan and Tyseer Aboulnasr, "A Post Filter Perceptual Convolutive Blind Source Separation Approach For Speech Signals," 7th International Conference on Signal Processing, ICSP, 2004.

- Qiongfeng Pan and Tyseer Aboulnasr, "A New Perceptual Convolutive Blind Source Separation Algorithm for Speech Separation," 7th International Conference on Signal Processing, ICSP, 2004.

- Qiongfeng Pan and Tyseer Aboulnasr, "A Convolutive Blind Signal Separation Method for Joint Speech Signal Separation and Echo Cancellation", IEEE 46-th Midwest Symposium on Circuits and Systems, 2003.

Chapter 2 Fundamentals of Blind Source Separation and Independent Component Analysis

In this chapter, we summarize the fundamental background knowledge that is necessary for proper understanding of blind source separation algorithms.

2.1 Probability Theory and Stochastic Processes [Papoulis 1991]

2.2.1 Basic Definitions

Assume that x is a random variable, its cumulative distribution function (cdf) F_x at point $x = x_0$ is defined as the probability that $x \le x_0$:

$$F_x(x_0) = P(x \le x_0) \tag{2.1}$$

Its probability density function (pdf) $p_x(x)$ is obtained as the derivative of the cdf F_x:

$$p_x(x_0) = \frac{dF_x(x)}{dx}\Big|_{x=x_0} \tag{2.2}$$

On the other hand, the cdf can also be computed from the known pdf by using the inverse relationship:

$$F_x(x_0) = \int_{-\infty}^{x_0} p_x(\xi)d\xi \tag{2.3}$$

Assume that $\mathbf{x}$ is an n-dimensional random vector $\mathbf{x} = (x_1, x_2, ..., x_n)^T$, where T denotes the transpose. Its cdf at point $\mathbf{x} = \mathbf{x_0}$ is defined as:

$$F_{\mathbf{x}}(\mathbf{x}_0) = P(\mathbf{x} \le \mathbf{x}_0) \tag{2.4}$$

The pdf $p_{\mathbf{x}}(\mathbf{x})$ of $\mathbf{x}$ is defined as the derivative of the cdf $F_{\mathbf{x}}(\mathbf{x})$ with respect to all components of the random vector $\mathbf{x}$:

$$p_{\mathbf{x}}(\mathbf{x}_0) = \frac{\partial}{\partial x_1}\frac{\partial}{\partial x_2}...\frac{\partial}{\partial x_n}F_{\mathbf{x}}(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_0} \tag{2.5}$$

And

$$F_{\mathbf{x}}(\mathbf{x}_0) = \int_{-\infty}^{x_0} p_{\mathbf{x}}(\mathbf{x})d\mathbf{x} = \int_{-\infty}^{x_{01}} \int_{-\infty}^{x_{02}} \cdots \int_{-\infty}^{x_{0n}} p_{\mathbf{x}}(\mathbf{x})dx_1 dx_2 \ldots dx_n \tag{2.6}$$

Assume that $\mathbf{y}$ is another vector with dimension m which is different from the dimension n of $\mathbf{x}$. The joint cdf of $\mathbf{x}$ and $\mathbf{y}$ at $\mathbf{x} = \mathbf{x}_0, \mathbf{y} = \mathbf{y}_0$ is given by

$$F_{\mathbf{x},\mathbf{y}}(\mathbf{x}_0, \mathbf{y}_0) = P(\mathbf{x} \le \mathbf{x}_0, \mathbf{y} \le \mathbf{y}_0) \tag{2.7}$$

The joint pdf $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})$ of $\mathbf{x}$ and $\mathbf{y}$ is defined by differentiating the joint cdf $F_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})$ with respect to all components of the random vectors $\mathbf{x}$ and $\mathbf{y}$. Also we can get the cdf by the following inverse relationship:

$$F_{\mathbf{x},\mathbf{y}}(\mathbf{x}_0, \mathbf{y}_0) = \int_{-\infty}^{x_0} \int_{-\infty}^{y_0} p_{\mathbf{x},\mathbf{y}}(\xi, \eta)d\eta d\xi \tag{2.8}$$

The marginal probability densities $p_{\mathbf{x}}(\mathbf{x})$ of $\mathbf{x}$ and $p_{\mathbf{y}}(\mathbf{y})$ of $\mathbf{y}$ are obtained by integrating over the other random vector in their joint pdf $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})$:

$$p_{\mathbf{x}}(\mathbf{x}) = \int_{-\infty}^{\infty} p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \eta)d\eta \tag{2.9}$$

$$p_{\mathbf{y}}(\mathbf{y}) = \int_{-\infty}^{\infty} p_{\mathbf{x},\mathbf{y}}(\xi, \mathbf{y})d\xi \tag{2.10}$$

Assuming that the joint pdf $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})$ of $\mathbf{x}$ and $\mathbf{y}$ and their marginal densities exist, the conditional probability density of $\mathbf{x}$ given $\mathbf{y}$ is defined as

$$p_{\mathbf{x}/\mathbf{y}}(\mathbf{x}/\mathbf{y}) = \frac{p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})}{p_{\mathbf{y}}(\mathbf{y})} \tag{2.11}$$

Based on the definition of the conditional probability density, we can describe the Bayes' rule as following:

$$p_{\mathbf{y}/\mathbf{x}}(\mathbf{y}/\mathbf{x}) = \frac{p_{\mathbf{x}/\mathbf{y}}(\mathbf{x}/\mathbf{y})p_{\mathbf{y}}(\mathbf{y})}{p_{\mathbf{x}}(\mathbf{x})} \tag{2.12}$$

where the denominator can be computed as

$$p_{\mathbf{x}}(\mathbf{x}) = \int_{-\infty}^{\infty} p_{\mathbf{x}/\mathbf{y}}(\mathbf{x}/\eta)p_{\mathbf{y}}(\eta)d\eta \tag{2.13}$$

It means that we can estimate the posterior pdf $p_{\mathbf{y}/\mathbf{x}}(\mathbf{y}/\mathbf{x})$ of the vector $\mathbf{y}$ if we know the pdf of the observation vector $\mathbf{x}$ and know the prior pdf $p_{\mathbf{y}}(\mathbf{y})$. Bayes' rule is widely used in the estimation theory.

2.2.2 *Expectation and Second-order Moments*

Let $g(\mathbf{x})$ be any function of the random vector $\mathbf{x}$. The expectation of $g(\mathbf{x})$ is defined by

$$E\{g(\mathbf{x})\} = \int_{-\infty}^{\infty} g(\mathbf{x}) p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{2.14}$$

Moments of a random vector $\mathbf{x}$ are obtained when $g(\mathbf{x})$ consists of products of components of $\mathbf{x}$. The first moment of a random vector $\mathbf{x}$ is defined as

$$\mathbf{m}_{\mathbf{x}} = E\{\mathbf{x}\} = \int_{-\infty}^{\infty} \mathbf{x} p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{2.15}$$

which is also called the mean vector $\mathbf{m}_{\mathbf{x}}$ of $\mathbf{x}$.

The second moment of $\mathbf{x}$ is defined as

$$r_{ij} = E\{x_i x_j\} \tag{2.16}$$

which is also called the correlation between the i th and j th component of $\mathbf{x}$.

The correlation matrix $\mathbf{R}_{\mathbf{x}} = E\{\mathbf{x}\mathbf{x}^T\}$ of the vector $\mathbf{x}$ is a matrix which has the element in row i and column j as the second moment r_{ij}.

The covariance matrix $\mathbf{C}_{\mathbf{x}}$ of $\mathbf{x}$ is given by

$$\mathbf{C}_{\mathbf{x}} = E\{(\mathbf{x} - \mathbf{m}_{\mathbf{x}})(\mathbf{x} - \mathbf{m}_{\mathbf{x}})^T\} \tag{2.17}$$

The cross-correlation matrix of two different random vectors $\mathbf{x}$ and $\mathbf{y}$ is

$$\mathbf{R}_{\mathbf{xy}} = E\{\mathbf{x}\mathbf{y}^T\} \tag{2.18}$$

And the cross-covariance matrix is

$$\mathbf{C}_{\mathbf{xy}} = E\left\{(\mathbf{x} - \mathbf{m}_{\mathbf{x}})\left(\mathbf{y} - \mathbf{m}_{\mathbf{y}}\right)^T\right\} \tag{2.19}$$

2.2.3 *Uncorrelatedness and Independence*

Two random vectors $\mathbf{x}$ and $\mathbf{y}$ are uncorrelated if their cross-covariance matrix $\mathbf{C}_{\mathbf{xy}}$ is a zero matrix

$$\mathbf{C}_{\mathbf{xy}} = E\left\{(\mathbf{x}-\mathbf{m}_{\mathbf{x}})(\mathbf{y}-\mathbf{m}_{\mathbf{y}})^{T}\right\} = \mathbf{0} \tag{2.20}$$

A random vector is said to be white if it has zero mean and unit covariance matrix, possibly multiplied by a constant variance.

The random variables x and y are said to be independent if and only if

$$p_{x,y}(x,y) = p_x(x)p_y(y) \tag{2.21}$$

Assume that $\mathbf{x},\mathbf{y},\mathbf{z},...$ are random vectors which may in general have different dimensions. The independence condition for $\mathbf{x},\mathbf{y},\mathbf{z},...,$ is then

$$p_{\mathbf{x},\mathbf{y},\mathbf{z},...}(\mathbf{x},\mathbf{y},\mathbf{z},...) = p_{\mathbf{x}}(\mathbf{x})p_{\mathbf{y}}(\mathbf{y})p_{\mathbf{z}}(\mathbf{z})... \tag{2.22}$$

A very basic property that follows from Eq. (2.21) for independent random variables x and y is that they satisfy the following equation:

$$E\{f(x)g(y)\} = E\{f(x)\}E\{g(y)\} \tag{2.23}$$

where $f(x)$ and $g(y)$ are arbitrary integrable function. From this, we can see that statistical independence Eq. (2.21) is a much stronger property than uncorrelatedness of Eq. (2.20).

2.2.4 *Random Gaussian Vector*

The probability density function of an n-dimensional random Gaussian vector $\mathbf{x}$ is

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}(\det \mathbf{C}_{\mathbf{x}})^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m}_{\mathbf{x}})^{T}\mathbf{C}_{\mathbf{x}}^{-1}(\mathbf{x}-\mathbf{m}_{\mathbf{x}})\right) \tag{2.24}$$

where n is the dimension of $\mathbf{x}$, $\mathbf{m}_\mathbf{x}$ is its mean vector, $\mathbf{C}_\mathbf{x}$ is its covariance matrix, $\det \mathbf{C}_\mathbf{x}$ is the determinant of matrix $\mathbf{C}_\mathbf{x}$.

From Eq. (2.24), we can see that the first and second order statistics can uniquely describe the probability density function of the Gaussian vector. Higher-order statistics do not provide new information for Gaussian signals.

2.2.5 Probability of the Transformed Variables

Assume that $\mathbf{x}$ and $\mathbf{y}$ are n-dimensional random vectors that are related by the transformation function vector $\mathbf{g}(.)$

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) \tag{2.25}$$

And the transformation is invertible as

$$\mathbf{x} = \mathbf{g}^{-1}(\mathbf{y}) \tag{2.26}$$

The pdf $p_\mathbf{y}(\mathbf{y})$ of $\mathbf{y}$ is obtained from the pdf $p_\mathbf{x}(\mathbf{x})$ of $\mathbf{x}$ as follows:

$$p_\mathbf{y}(\mathbf{y}) = \frac{1}{\left|\det J\mathbf{g}\left(\mathbf{g}^{-1}(\mathbf{y})\right)\right|} p_\mathbf{x}\left(\mathbf{g}^{-1}(\mathbf{y})\right) \tag{2.27}$$

where $J\mathbf{g}$ is the Jacobian matrix

$$J\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_1} & \frac{\partial g_2(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial g_n(\mathbf{x})}{\partial x_1} \\ \frac{\partial g_1(\mathbf{x})}{\partial x_2} & \frac{\partial g_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial g_n(\mathbf{x})}{\partial x_2} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial g_1(\mathbf{x})}{\partial x_n} & \frac{\partial g_2(\mathbf{x})}{\partial x_n} & \cdots & \frac{\partial g_n(\mathbf{x})}{\partial x_n} \end{bmatrix} \tag{2.28}$$

where $g_i(\mathbf{x})$ is the ith element of $g(\mathbf{x})$.

When the transformation is linear and non-singular, i.e. $\mathbf{y} = \mathbf{A}\mathbf{x}$ and $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$, a widely used transformation in instantaneous BSS, the pdf of $\mathbf{y}$ can be simplified to

$$p_{\mathbf{y}}(\mathbf{y}) = \frac{1}{|\det \mathbf{A}|} p_{\mathbf{x}}\left(\mathbf{A}^{-1}\mathbf{y}\right) \tag{2.29}$$

Higher-Order Statistics

Assume that x is a real-valued zero mean, continuous scalar random variable with probability density function $p_x(x)$. The first characteristic function $\varphi_x(\omega)$ of x is defined as the continuous Fourier transform of the pdf $p_x(x)$:

$$\varphi_x(\omega) = E\{\exp(j\omega x)\} = \int_{-\infty}^{\infty} \exp(j\omega x) p_x(x)\, dx \tag{2.30}$$

where ω is the transformed variable corresponding to x. Expanding the characteristic function $\varphi_x(\omega)$ into its Taylor series yields

$$\varphi_x(\omega) = \int_{-\infty}^{\infty} \left(\sum_{k=0}^{\infty} \frac{x^k (jw)^k}{k!} \right) p_x(x)\, dx = \sum_{k=0}^{\infty} E\{x^k\} \frac{(jw)^k}{k!} \tag{2.31}$$

The coefficient terms of this expansion are moments $E\{x^k\}$ of x and that is why the characteristic function $\varphi_x(\omega)$ is also called the moment generating function.

The natural logarithm of the characteristic function $\varphi_x(\omega)$ is the cumulant generating function as follows:

$$\phi_x(\omega) = \ln(\varphi_x(\omega)) = \ln\left(E\{\exp(j\omega x)\}\right) \tag{2.32}$$

The cumulants $\kappa_{x,k}$ of x are defined as the coefficients of the Taylor series expansion of the cumulant generating function:

$$\phi_x(\omega) = \sum_{k=0}^{n} \kappa_{x,k} \frac{(j\omega)^k}{k!} \tag{2.33}$$

where the k th cumulant is obtained as

$$\kappa_{x,k} = (-j)^k \frac{d^k \phi_x(\omega)}{d\omega^k}\Big|_{\omega=0} \tag{2.34}$$

The list below is the expression of the first four cumulants:

$$
\begin{aligned}
\kappa_{x,1} &= E\{x\} \\
\kappa_{x,2} &= E\left\{\left(x-E\{x\}\right)^2\right\} = E\left\{x^2\right\}-\left[E\{x\}\right]^2 \\
\kappa_{x,3} &= E\left\{\left(x-E\{x\}\right)^3\right\} = E\left\{x^3\right\}-3E\left\{x^2\right\}E\{x\}+2\left[E\{x\}\right]^3 \\
\kappa_{x,4} &= E\left\{\left(x-E\{x\}\right)^4\right\} = E\left\{x^4\right\}-3\left[E\left\{x^2\right\}\right]^2-4E\left\{x^3\right\}E\{x\}+12E\left\{x^2\right\}\left[E\{x\}\right]^2-6\left[E\{x\}\right]^4
\end{aligned}
\tag{2.35}
$$

For a zero mean random variable x, the first four cumulants are simplified as:

$$
\begin{aligned}
\kappa_{x,1} &= 0 \\
\kappa_{x,2} &= E\left\{x^2\right\} \\
\kappa_{x,3} &= E\left\{x^3\right\} \\
\kappa_{x,4} &= E\left\{x^4\right\}-3\left[E\left\{x^2\right\}\right]^2
\end{aligned}
\tag{2.36}
$$

The above definition can be easily extended to the vector case. For a random vector $\mathbf{x}$ whose probability density function is $p_{\mathbf{x}}(\mathbf{x})$, the characteristic function of $\mathbf{x}$ is the Fourier transform of its pdf

$$
\varphi_{\mathbf{x}}(\boldsymbol{\omega}) = E\left\{\exp\left(j\boldsymbol{\omega}\mathbf{x}\right)\right\} = \int_{-\infty}^{\infty}\exp\left(j\boldsymbol{\omega}\mathbf{x}\right)p_{\mathbf{x}}(\mathbf{x})d\mathbf{x} \tag{2.37}
$$

where $\boldsymbol{\omega}$ is a row vector having the same dimension as $\mathbf{x}$ and the integral is computed over all components of $\mathbf{x}$. The moments of $\mathbf{x}$ are coefficients of the Taylor series expansion of the characteristic function $\varphi(\boldsymbol{\omega})$ and the cumulants of $\mathbf{x}$ are coefficients of the Taylor series expansion of the characteristic function $\phi(\boldsymbol{\omega}) = \ln\left(\varphi(\boldsymbol{\omega})\right)$. The second, third and fourth order cumulants for a zero mean random vector $\mathbf{x}$ are

$$
\begin{aligned}
cum\left(x_i, x_j\right) &= E\left\{x_i x_j\right\} \\
cum\left(x_i, x_j, x_k\right) &= E\left\{x_i x_j x_k\right\} \\
cum\left(x_i, x_j, x_k, x_l\right) &= E\left\{x_i x_j x_k x_l\right\} - E\left\{x_i x_j\right\}E\left\{x_k x_l\right\} \\
&\quad - E\left\{x_i x_k\right\}E\left\{x_j x_l\right\} - E\left\{x_i x_l\right\}E\left\{x_j x_k\right\}
\end{aligned}
\tag{2.38}
$$

Both moments and cumulants contain the same statistical information. However, it is usually preferable to work with cumulants because they present the additional information provided by higher-order statistics in a clearer way. The cumulants satisfy the linear addition property

in which the cumulant of the sum of two independent random vectors is equal to the sum of their cumulants. Moreover, all cumulants of order three or higher of multivariate Gaussian random vector are identically zero. Thus, higher order statistics can be used to measure the departure of a random vector from a Gaussian random vector with an identical mean vector and covariance matrix.

There are some drawbacks associated with higher-order statistics. One is that reliable estimation of higher-order moments and cumulants requires many more samples compared to second-order statistics. Another disadvantage is that higher-order statistics can be very sensitive to outliers in the data.

Despite these drawbacks, higher-order statistics are used in blind signal separation either explicitly or implicitly via nonlinearities. The reason for using a non-linear function of the random variable instead of direct higher-order statistic estimation is that when using Taylor series expansion, the non-linear function can be expanded as the sum of higher-order functions. For example the $\tanh(\cdot)$ function can be expanded as

$$\tanh(x) = x - \frac{1}{3}x^3 + \frac{2}{15}x^5 + \ldots \tag{2.39}$$

Thus by using a non-linear function, we are exploiting the higher-order statistics implicitly and in a more robust way since the values of these functions can be contained in an limited range, such as (-1,1) for $\tanh(\cdot)$.

2.2.6 Stochastic Processes

Stochastic processes are random functions of time. Assume a stochastic process $\{x_j(t)\}$ defined at discrete times $t_1, t_2, \ldots, t_k$. The stochastic process is said to be stationary in the strict sense if its joint density depends only on the time difference but not directly on the time instants.

Stochastic processes are usually characterized in terms of their mean and autocorrelation or autocovariance functions. The mean function of the stochastic process $\{x(t)\}$ is

$$m_x(t) = E\{x(t)\} = \int_{-\infty}^{\infty} x(t) p_{x(t)}(x(t))\, dx(t) \tag{2.40}$$

which becomes a constant mean m_x independent of time for a stationary process.

The variance function of the stochastic process $\{x(t)\}$ is

$$\sigma_x^2(t) = E\left\{\left[x(t) - m_x(t)\right]^2\right\} = \int_{-\infty}^{\infty} \left[x(t) - m_x(t)\right]^2 p_{x(t)}(x(t))\, dx(t) \tag{2.41}$$

which becomes a constant σ_x^2 for a stationary process.

The autocovariance function of the stochastic process $\{x(t)\}$ is

$$c_x(t,\tau) = \mathrm{cov}\left[x(t), x(t-\tau)\right] = E\left\{\left[x(t) - m_x(t)\right]\left[x(t-\tau) - m_x(t-\tau)\right]\right\} \tag{2.42}$$

which becomes $c_x(\tau)$ independent of the time for a stationary process.

The autocorrelation function of the stochastic process $\{x(t)\}$ is

$$r_x(t,\tau) = E\{x(t)x(t-\tau)\} \tag{2.43}$$

which is $r_x(\tau)$ independent of the time t for a stationary process.

For two different stochastic processes $\{x(t)\}$ and $\{y(t)\}$, the cross-correlation function is defined as:

$$r_{xy}(t,\tau) = E\{x(t)y(t-\tau)\} \tag{2.44}$$

The cross-covariance function is defined as

$$c_{xy}(t,\tau) = E\left\{\left[x(t) - m_x(t)\right]\right\}\left[y(t-\tau) - m_y(t-\tau)\right] \tag{2.45}$$

If the stochastic processes satisfy the following properties, they are wide-sense stationary (WSS) processes.

The mean function $m_x(t)$ of the process is a constant m_x for all t.

The autocorrelation function is independent of a time shift

2.2 Review of Optimization Methods

Optimization methods are widely used in BSS to estimate the coefficients of the separating system. Here, we review some typical optimization approaches including unconstrained and constrained methods.

2.2.1 Unconstrained Optimization Methods

The unconstrained optimization case is considered first. Assume the task is to minimize a cost function $J(\mathbf{w})$ with respect to a parameter matrix $\mathbf{w}$. The most classical approach for the unconstrained optimization problem is the gradient descent method [Fletcher 1987]. In the gradient descent optimization method, the cost function $J(\mathbf{w})$ is minimized iteratively by starting from some initial point $\mathbf{w}(0)$, computing the gradient of $J(\mathbf{w})$ at this point and then moving in the direction opposite to the gradient using a suitable step. The same procedure is repeated at the new point until convergence is achieved.

Assume the cost function is expressed as a function of the observed data as given by

$$J(\mathbf{w}) = E\{g(\mathbf{w},\mathbf{x})\} \tag{2.46}$$

The corresponding gradient descent algorithm is

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \mu(k)\frac{\partial}{\partial \mathbf{w}} E\{g(\mathbf{w},x(k))\}|_{\mathbf{w}=\mathbf{w}(k-1)} \tag{2.47}$$

The parameter $\mu(k)$ is the step-size or learning rate which controls the length of the step in the negative gradient direction and as such controls the convergence speed.

There are some disadvantages for the gradient descent method. The first one is that it can lead to a local minimum if the cost function is not quadratic and the update cannot escape from the local minimum. The second disadvantage is that its convergence speed is generally slow, especially when it approaches the minimum. Finally, it is very time consuming to

compute the mean values of the appropriate function at each iteration step, especially when new observations keep on coming in the course of the optimization.

In the stochastic gradient descent algorithm, the expectation operation in the learning rule is dropped and the new instantaneous learning rule is

$$\mathbf{w}(k) = \mathbf{w}(k-1) - \mu(k)\frac{\partial}{\partial \mathbf{w}} g(\mathbf{w}, x(k))|_{\mathbf{w}=\mathbf{w}(k-1)} \tag{2.48}$$

Although stochastic gradient adaptive algorithm leads to highly fluctuating directions of instantaneous gradients on every iteration step, the average direction is still the direction of the gradient descent algorithm. However, the convergence speed of the stochastic gradient descent is much slower than the corresponding steepest gradient descent algorithm in terms of coefficient updates or number of iterations. This is compensated by its very low computational cost per update.

In the gradient descent and stochastic gradient descent methods described above, it is assumed that the parameter space is in the Euclidean orthogonal coordinate space. However, in practice the parameter space is not always Euclidean but may have a Riemannian structure [Haykin 2000]. The Riemannian structure characterizes the intrinsic curvature of a particular manifold in N-dimensional space. Euclidean space, whose coordinate system is orthogonal, is only a special case of Riemannian space. For the general case, the Riemannian structure of the space can be incorporated in the optimization through the use of the natural gradient algorithm [Haykin 2000].

The natural gradient is based on differential geometry and employs knowledge of the Riemannian structure of the parameter space to adjust the gradient search direction. It can be used to overcome the poor convergence properties of the standard gradient adaptation.
In Riemannian geometry, the distance between two points $\mathbf{w}$ and $\mathbf{w}+\delta\mathbf{w}$ is

$$d_w(\mathbf{w}, \mathbf{w}+\delta\mathbf{w}) = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} \delta\mathbf{w}_i \delta\mathbf{w}_j g_{ij}(\mathbf{w})} = \sqrt{\delta\mathbf{w}^T \mathbf{G}(\mathbf{w})\delta\mathbf{w}} \tag{2.49}$$

where $\mathbf{G}(\mathbf{w})$ is the Riemannian metric tensor whose (i,j) th entry is $g_{ij}(\mathbf{w})$. In the Euclidean space $\mathbf{G}(\mathbf{w})=\mathbf{I}$ is the identity matrix.

For the cost function $J(\mathbf{w})$, The steepest-descent direction in the Riemannian space is defined as

$$\delta\mathbf{w} = -\mu\mathbf{G}^{-1}(\mathbf{w})\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \tag{2.50}$$

where μ is a positive constant. Thus, the natural gradient adaptation is defined as

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu(k)\mathbf{G}^{-1}(\mathbf{w}(k))\frac{\partial J(\mathbf{w}(k))}{\partial \mathbf{w}} \tag{2.51}$$

The natural gradient algorithm is widely used in blind signal separation algorithms to improve the convergence speed. In [Amari 1998], it is shown that the natural gradient algorithm for instantaneous blind source separation is

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta\mathbf{w}_{natural\_gradient} = \mathbf{w}(k) - \mu(k)\frac{\partial J(\mathbf{w}(k))}{\partial \mathbf{w}}\mathbf{W}^T(k)\mathbf{W}(k) \tag{2.52}$$

We will consider its performance later.

The classical literature has numerous other optimization approaches for unconstrained cases, we will not address them here. Interested readers can find more in [Fletcher 1987].

2.2.2 Constrained optimization methods [Fletcher 1987]

Generally, the task is the minimization or maximization of the cost function under some additional conditions. This scenario is addressed by constrained optimization methods.

Assuming the constraints are

$$H_i(\mathbf{w}) = 0, \quad i = 1,2,...,k \tag{2.53}$$

The most widely used constrained optimization method is the Lagrange multiplier method. In this method, a Lagrangian function is first formed based on the cost function and the constraints as follows:

$$L(\mathbf{w},\lambda_1,...,\lambda_k) = J(\mathbf{w}) + \sum_{i=1}^{k} \lambda_i H_i(\mathbf{w}) \tag{2.54}$$

$\lambda_1,...,\lambda_k$ are the Lagrange multipliers. The solution is the point where the gradient of Lagrangian function is zero with respect to the parameters $\mathbf{w}$ and all the multipliers λ_i.

When the constrained conditions are of the equality type and are simple, we can also use the projection optimization methods [Oja 1982]. In these methods, the minimization optimization problem is solved with the unconstrained learning rules as described above. However, in every iteration step, the resulting $\mathbf{w}$ is projected onto the constraint set to ensure it satisfies the constraints.

2.3 Review of Estimation Theory [Hyvarinen 2001] for BSS

The BSS problem can be viewed as the problem of estimating the quantities of interest from a given finite set of measurements. Thus, estimation theory provides an essential tool for BSS.

2.3.1 Problem Formulation

Assume there are N scalar measurements $x(1),x(2),...,x(N)$ containing information about m parameters $\theta_1,\theta_2,...,\theta_m$. The task is to estimate the m parameters from the N observations.

Represent the parameter vector as $\boldsymbol{\theta} = (\theta_1,\theta_2,...,\theta_m)^T$ and the observed data vector as $\mathbf{x} = (x(1),x(2),...,x(N))^T$, the estimated parameter vector can be expressed as

$$\hat{\boldsymbol{\theta}} = \mathbf{w}(\mathbf{x}) = \mathbf{w}\big(x(1), x(2), ..., x(N)\big) \tag{2.55}$$

2.3.2 Method of Moments [Hyvarinen 2001]

The method of moments is the simplest and oldest estimation method. In this method the probability distribution $p(x|\boldsymbol{\theta})$ of data samples is assumed known and identical. The basic idea of the method of moments is to form m equations for m unknown parameters by equating the first m theoretic moments to their respective values determined from the observed samples.

The theoretical moment α_j is defined as

$$\alpha_j = E\{x^j | \boldsymbol{\theta}\} = \int_{-\infty}^{\infty} x^j p(x|\boldsymbol{\theta}) dx, \qquad j = 1, 2, ... \tag{2.56}$$

The moments d_j as determined from the observed samples is obtained as

$$d_j = \frac{1}{N} \sum_{i=1}^{N} \big(x(i)\big)^j \tag{2.57}$$

The estimated parameters are obtained from the equations: $\alpha_j = d_j$. The method of moments is often inefficient and sometimes does not lead to an acceptable estimator.

2.3.3 Least-Squares Estimation [Hyvarinen 2001][Papoulis 1991]

In the basic linear least-squares (LS) method, the observed date vector is assumed to obey a linear model as:

$$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{v} \tag{2.58}$$

where $\mathbf{v}$ is the unknown measurement error vector and $\mathbf{H}$ is the observation matrix which is assumed known. The idea of least-squares method is to choose an estimator $\hat{\boldsymbol{\theta}}$ that

minimizes in some sense the effect of the errors. One basic choice is to have a least-squares cost function:

$$\varepsilon_{ls} = \frac{1}{2}\|\mathbf{v}\|^2 = \frac{1}{2}(\mathbf{x}-\mathbf{H}\boldsymbol{\theta})^T(\mathbf{x}-\mathbf{H}\boldsymbol{\theta}) \tag{2.59}$$

Minimization of this cost function has the following solution for $\hat{\boldsymbol{\theta}}_{LS}$:

$$\left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)\hat{\boldsymbol{\theta}}_{LS} = \mathbf{H}^T\mathbf{x} \tag{2.60}$$

$$\hat{\boldsymbol{\theta}}_{LS} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{x} \tag{2.61}$$

where $\left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1}$ is the pseudo inverse of $\left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)$.

The least-squares estimation method can be generalized in several ways such as weighted least-squares estimation, non-linear least-squares estimation [Papoulis 1991] etc.

2.3.4 Maximum Likelihood Method [Hyvarinen 2001][Papoulis 1991]

In maximum likelihood (ML) method, the parameter vector $\boldsymbol{\theta}$ is assumed fixed and no *a priori* information is available about it. The basic idea of ML estimation method is to choose $\boldsymbol{\theta}_{ML}$ that maximizes the likelihood function of the measurements which is

$$p(x|\boldsymbol{\theta}) = p\left(x(1), x(2), ..., x(N)|\boldsymbol{\theta}\right) \tag{2.62}$$

It means that the ML approach finds the parameter $\boldsymbol{\theta}_{ML}$ that makes $\mathbf{x}$ the most likely measurement. In practice, it is more convenient to use the log likelihood function $\ln p(\mathbf{x}|\boldsymbol{\theta})$. By maximizing it with respect to $\boldsymbol{\theta}$, we get the following equation:

$$\frac{\partial}{\partial\boldsymbol{\theta}}\ln p(x|\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{ML}} = 0 \tag{2.63}$$

which contains m scalar equations for the m parameters.

The disadvantages of this method are that the construction of the likelihood function can be very difficult, and the computational load in this method is also very demanding.

2.3.5 Bayesian Estimation [Hyvarinen 2001][Papoulis 1991]

In the Bayesian estimation method the parameter vector is assumed to be random vector which is modeled by a known probability density function $p_{\theta}(\theta)$.

The basic idea of Bayesian estimation method is to maximize the posterior density $p_{\theta|\mathbf{x}}(\theta|\mathbf{x})$ given the measurement $\mathbf{x}$. There are two most popular methods for this approach. One is the minimum mean-square error estimator, and the other one is to the maximum posterior density estimator.

In the minimum mean-square error method, the optimal vector $\hat{\theta}$ is chosen by minimizing the mean-square error ε_{MSE} with respect to $\hat{\theta}$ where

$$\varepsilon_{MSE} = E\left\{\left\|\theta - \hat{\theta}\right\|^2\right\} \tag{2.64}$$

In the method of maximum a posterior density, the posterior density $p_{\theta|\mathbf{x}}(\theta|\mathbf{x})$ is maximized with respect to $\hat{\theta}$. From equation (2.12), we can get

$$p_{\theta|\mathbf{x}}(\theta|\mathbf{x}) = \frac{p_{\theta,\mathbf{x}}(\theta,\mathbf{x})}{p_{\mathbf{x}}(\mathbf{x})} = \frac{p_{\mathbf{x}/\theta}(\mathbf{x}/\theta)\,p_{\theta}(\theta)}{p_{\mathbf{x}}(\mathbf{x})} \tag{2.65}$$

The denominator of the posterior density is $p_{\mathbf{x}}(\mathbf{x})$ which does not depend on the parameter $\hat{\theta}$, thus we can maximize the numerator with respect to $\hat{\theta}$ without worrying about the denominator.

2.4 Review of Information Theory for BSS [Hyvarinen 2001][Papoulis 1991]

Several very popular BSS approaches are based on information theory concepts and use the entropy of the source as an important measure. Entropy H for a discrete-valued random variable X is defined as

$$H(X) = -\sum_i P(X = a_i) \log\left(P(X = a_i)\right) \tag{2.66}$$

The entropy of a random variable is a measure of randomness of the variable. The more random a variable is, the larger its entropy.

Entropy H for continuous-valued random variable x with density $p_x(.)$ is defined as

$$H(x) = -\int_{-\infty}^{\infty} p_x(\xi) \log p_x(\xi) d\xi \tag{2.67}$$

which is often called differential entropy.

In BSS algorithms, we frequently encounter the calculation of the entropy of a transformation. Assuming $\mathbf{y} = \mathbf{g}(\mathbf{x})$ where $\mathbf{y}$ and $\mathbf{x}$ are random vectors and $\mathbf{g}$ is an invertible transformation function. The entropy $H(\mathbf{y})$ is calculated as

$$H(\mathbf{y}) = H(\mathbf{x}) + E\left\{\log\left|\det J\mathbf{g}(\mathbf{x})\right|\right\} \tag{2.68}$$

where $J\mathbf{g}(.)$ is the Jacobian matrix of the function $\mathbf{g}$, which is defined in Eq. (2.28).

For the special case of linear transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$, we obtain that

$$H(\mathbf{y}) = H(\mathbf{x}) + \log\left|\det \mathbf{A}\right| \tag{2.69}$$

2.4.1 Mutual Information

One concept often used in BSS algorithms is that of mutual information, which is a measure of the information that members of a set of random variables have about the other random variables in the set. It can be defined by two different points of view.

The mutual information I between n scalar random variables $x_i, i = 1,2,...,n$ can be defined by entropy as

$$I(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} H(x_i) - H(\mathbf{x}) \tag{2.70}$$

where $\mathbf{x}$ is the vector containing all the x_i.

The mutual information I between n scalar random variables $x_i, i=1,2,...,n$ can also be defined by Kullback-Leibler divergence as

$$I(x_1,x_2,...,x_n)=\int p(\mathbf{x})\log\frac{p(\mathbf{x})}{p(x_1)p(x_2)...p(x_n)}d\mathbf{x} \tag{2.71}$$

It is a kind of distance measure of independence between vector $\mathbf{x}$ and its components x_i. It is zero if and only if the component variables are independent.

2.4.2 Negentropy

Negentropy is defined as

$$J(\mathbf{x})=H(\mathbf{x}_{gauss})-H(\mathbf{x}) \tag{2.72}$$

where $\mathbf{x}_{gauss}$ is a Gaussian random vector with the same covariance matrix as $\mathbf{x}$.

One fundamental result in information theory is that a Gaussian variable has the largest entropy among all random variables with same variance matrix. That is why negentropy is obtained as a measure of nongaussianity.

2.5 Whitening of Signals [Hyvarinen 2001]

A zero mean random vector $\mathbf{y}=(y_1,y_2,...,y_n)^T$ is said to be white if its elements are uncorrelated and have unit variances i.e. $E\{y_iy_j\}=\delta_{ij}$.

Given a random vector $\mathbf{x}$ with n elements, we need to determine a linear transformation $\mathbf{V}$ to make the output $\mathbf{y}=\mathbf{Vx}$ white. For an off-line solution, we can first estimate the covariance matrix of vector $\mathbf{x}$ as $\mathbf{C}_x=E\{\mathbf{x}\mathbf{x}^T\}$. Let $\mathbf{E}$ be the matrix whose columns are the unit-norm eigenvectors of $\mathbf{C}_x$, $\mathbf{D}$ be the diagonal matrix of the eigenvalues of $\mathbf{C}_x$. Then the whiteing matrix $\mathbf{V}$ is

$$\mathbf{V} = \mathbf{U}\mathbf{D}^{-1/2}\mathbf{E}^T \tag{2.73}$$

where $\mathbf{U}$ is an arbitrary orthogonal matrix.

2.6 Principal Component Analysis (PCA) [Hyvarinen 2001]

Principal component analysis (PCA) is a widely used statistical tool in many applications such as feature extraction, data compression etc. The aim of PCA is to reduce data redundancy so as to represent data by a smaller set of variables. The redundancy in PCA is measured by correlation between the observed data samples.

2.6.1 Problem Formulation

Assume $\mathbf{x}$ is a random vector with n elements. T samples $\mathbf{x}(1),\mathbf{x}(2),...,\mathbf{x}(T)$ are available from this random vector. In practice, the elements of $\mathbf{x}$ are mutually correlated and have redundant information. By linearly transforming $\mathbf{x}$ to another vector $\mathbf{y}$ with fewer elements, we can try to remove the redundancy in vector $\mathbf{x}$. There are many methods in the literature to implement PCA and we introduce some of them here [Hyvarinen 2001] .

2.6.2 PCA by Variance Maximization[Hyvarinen 2001]

Denoting $\mathbf{w}_1 = \{w_{11}, w_{21},..., w_{n1}\}$, the linear combination of vector $\mathbf{x}$ and $\mathbf{w}_1$ is given by

$$y_1 = \mathbf{w}_1^T\mathbf{x} \tag{2.74}$$

If the variance of y_1 is maximized, y_1 is the first principal component of $\mathbf{x}$. The other principal components such as the m-th component can be obtained by maximizing the variance of y_m under the constraint that y_m is uncorrelated with all the previously found principal components.

2.6.3 *PCA by Minimum Mean-Square Error[Hyvarinen 2001]*

The idea of determining the PCA by minimum mean-square error method is to find a set of m orthonormal basis vectors, which span an m-dimensional subspace, such that the mean-square error between $\mathbf{x}$ and its projection on the subspace is minimal.

2.6.4 *PCA by On-Line Learning Rules[Hyvarinen 2001]*

The methods in 2.6.2 and 2.6.3 are used in batch processing where all samples of $\mathbf{x}$ are used in the estimation. However, in practice, we may not have all the samples of the random vector available or the random vector may not be stationary. We need to solve the PCA on-line.

The most straightforward on-line learning rule is the one based on stochastic gradient ascent. By taking the gradient of y_1^2 with respect to $\mathbf{w}_1$ and adding the constraint that $\|\mathbf{w}_1\| = 1$, the following learning rule is obtained:

$$\mathbf{w}_1(k+1) = \mathbf{w}_1(k) + \mu(k)\left(y_1(k)\mathbf{x}(k) - y_1^2(k)\mathbf{w}_1(k)\right) \tag{2.75}$$

where $y_1 = \mathbf{w}_1^T\mathbf{x}$ and $\mu(k)$ is the learning rate controlling the convergence speed.

The learning rule of other weights $\mathbf{w}_j$ can be obtained by taking the gradient of y_j^2 with respect to $\mathbf{w}_j$ and adding the normalization constraint of $\mathbf{w}_j$ and the constraint that y_m is uncorrelated with all the previously found principal components.

2.7 Conclusion

In this chapter, we review some fundamental knowledge necessary for understanding blind signal separation. Basic concepts in probability theory, statistical processes, optimization methods, estimation theory and information theory are reviewed with the focus being on concepts related to blind source separation. Principal component analysis is described in this chapter as a basic technique to perform source separation.

Chapter 3 Blind Source Separation Models and Estimation Methods

3.1 Introduction

Blind source separation is the process of separating desired source signals from a set of observed sensor signals. The general blind source separation problem can be formulated as follows. Assume there are N source signals $s_1, s_2, ..., s_N$ transmitting through a medium (such as air, cable, network etc). M sensors located in the medium capture the transmitted signals to get sensor signals $x_1, x_2, ..., x_M$. Representing the transmitting system (or mixing system) with operator $\mathbf{A}[.]$ and assuming this system is invertible, the separation is performed by estimating an unmixing system $\mathbf{W}[.]$ to invert the mixing operation. The general model for blind source separation is illustrated in Fig. 3.1. In this diagram, $v_1, v_2, ..., v_M$ are additive noises which are unavoidable in real environments.

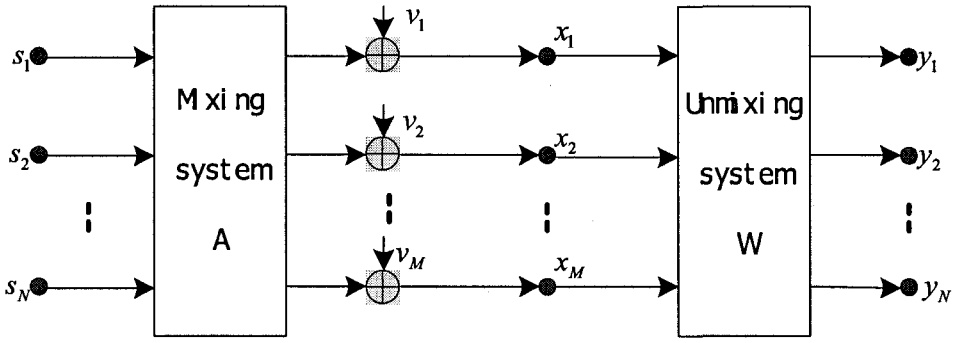


Figure 3-1: General blind source separation model

Representing the source signal vector as $\mathbf{s} = [s_1, s_2, ..., s_N]^T$, sensor signal vector as $\mathbf{x} = [x_1, x_2, ..., x_M]^T$, additive noise as $\mathbf{v}$, the mixing process can be expressed as

$$\mathbf{x} = \mathbf{A}[\mathbf{s}] + \mathbf{v} \tag{3.1}$$

and the unmixing process can be expressed as

$$\mathbf{y} = \mathbf{W}[\mathbf{x}] = \mathbf{W}[\mathbf{As} + \mathbf{v}] \approx \mathbf{s} \tag{3.2}$$

Blind source separation is a versatile tool in a wide range of applications since no prior knowledge of source signals and mixing system is needed in its separation algorithms. In this chapter, we review blind source separation models including the basic instantaneous BSS model and the convolutive BSS model along with their estimation methods. The instantaneous BSS model and its estimation methods are described in Section 3.2. The convolutive BSS model and its estimation methods are reviewed in Section 3.3. In Section 3.4, we review existing blind speech signal separation algorithms and describe the challenges for the blind speech signal separation problem. Conclusions are given in Section 3.5.

3.2 Instantaneous Blind Source Separation

3.2.1 Model Description

In instantaneous blind source separation, the mixing system is assumed to be instantaneous and linear, i.e. sensors capture instantaneous mixtures of source signals, with zero noise. The model is shown in Fig. 3.2.

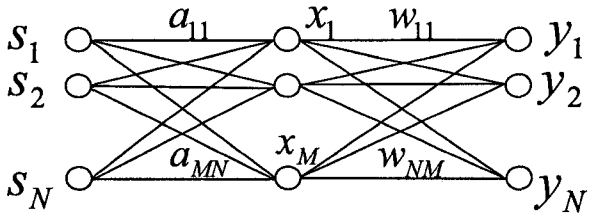


Figure 3-2: Instantaneous blind source separation model

The M observed signals $\mathbf{x} = [x_1, x_2, ..., x_M]^T$ are modeled as linear combinations of N source signals s_j as

$$x_i = a_{i1}s_1 + a_{i2}s_2 + ... + a_{iN}s_N \tag{3.3}$$

for all $i = 1,...,M$, where the $a_{ij}, i = 1,...,N, j = 1,...,M$ are the weight coefficients.

More conveniently, this model can be expressed by the vector-matrix notation as

$$\mathbf{x} = \mathbf{As} \tag{3.4}$$

where $\mathbf{A}$ is the $M \times N$ mixing scalar matrix as

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \cdots & \cdots & \cdots \\ a_{M1} & \cdots & a_{MN} \end{bmatrix} \tag{3.5}$$

In instantaneous BSS model, source signals are assumed to be statistically mutually independent. For simplicity, we assume that the number of independent source signals is equal to the number of the observed mixtures, i.e. $N = M$, and the mixing matrix is invertible. In addition, we assume there is no additive noise in the mixing system since BSS in the presence of noise is addressed as a special case.

The aim of BSS is to separate the mixtures and recover the sources by finding a matrix $\mathbf{W}$ and determining $\mathbf{y} = \mathbf{Wx}$ as the estimates of the source signals. Since we have no information about both the original source signals and the mixing system, there are some ambiguities in this model: the variances of the independent components cannot be determined and the order of the independent components cannot be determined. Thus, ideally, the recovered sources are the original ones up to permutation and scaling. That is

$$\mathbf{y} = \mathbf{Wx} = \mathbf{WAs} = \mathbf{PDs} \tag{3.6}$$

where $\mathbf{P}$ is a permutation matrix and $\mathbf{D}$ is a nonsingular diagonal matrix.

It is worth noting that the independence assumption about the source signals in the BSS model is stronger than assuming uncorrelatedness and whitening. This feature makes it different from other second-order based methods since it also exploits higher-order information.

Because the mixtures in the instantaneous BSS model are assumed instantaneous and the original source signals are assumed mutually independent, independent component analysis (ICA) methods are always used to deal with the basic BSS estimation problem. In many cases in the literature, instantaneous BSS and ICA terms are used interchangeably.

As mentioned above, the aim of BSS is to separate the mixtures and recover the sources by finding a separating matrix $\mathbf{W}$. Many methods have been proposed with all defining a cost function and an optimization algorithm for this cost function.

In the following section, we introduce some of the basic approaches for performing instantaneous BSS based on different cost functions. These cost functions can be divided into two categories, one exploits higher-order statistics implicitly through non-linear functions; the other exploits higher-order statistics directly by estimating them from available data. After constructing the cost function, we can use a suitable optimization method from these introduced in Chapter 2, (stochastic gradient, natural gradient etc.), to derive the corresponding BSS algorithm.

3.2.2 BSS by Maximization of Nongaussianity

A classical result in probability theory is the central limit theorem which says that the distribution of a sum of independent random variables tends towards a Gaussian distribution under certain conditions [Papoulis 1991]. Based on this theorem, even a sum of two independent random variables usually has a distribution that is closer to Gaussian than any of the two original random variables. This is a good starting point for BSS since the observed signals in BSS model are mixtures of independent components. Thus, the problem of BSS can be formulated as a search for components that are maximally Nonguassian.

Considering a linear combination y of the observed signals as

$$y = \mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{A}\mathbf{s} \tag{3.7}$$

where $\mathbf{w}$ is the corresponding weight vector. From the above equation, y is the weighted sum of independent source signals. If we can adjust the weight parameter vector $\mathbf{w}$ to make y as Nongaussian as possible, we obtain one of the independent source signals. Thus, the problem of estimating one of the independent source signals translates into maximizing the nongaussianity of y by adjusting the weight vector $\mathbf{w}$.

It is obvious that we need to measure the nongaussianity of y. Based on different methods for measuring nongaussianity, two different algorithms are reviewed below.

Measuring nongaussianity by kurtosis[Delfose 1995]

Kurtosis is a classic measure of nongaussianity and it can be used to estimate the independent components. Kurtosis is defined as the fourth-order cumulant of a random variable which is defined by equations (2.35) and (2.36) in Chapter 2. The value of kurtosis is zero for a Gaussian random variable and nonzero (negative or positive) for most nongaussian random variables. Typically we use the absolute value of kurtosis or the square of kurtosis as the measurement of the nongaussianity. The cost function based on the absolute value of kurtosis is

$$J(\mathbf{w}) = \left|kurt(y)\right| = \left|kurt(\mathbf{w}^T\mathbf{x})\right| \tag{3.8}$$

[Hyvarinen 1997] introduced a simple expression for computing the kurtosis and derived a fixed-point Newton-type algorithm for instantaneous BSS.

The advantage of using kurtosis as a measure of nongaussianity is its simplicity, both computational and theoretical. Computationally, kurtosis can be estimated simply by using the fourth moment of the sample data. Theoretically, performance analysis is simplified because of the linearity property of kurtosis. The linearity of kurtosis means that $kurt(x+y) = kurt(x) + kurt(y)$ when x and y are two statistically independent random variables.

However, using kurtosis as a measure of non-gaussianity has limitations. In practice, it needs to be estimated from measured samples and is very sensitive to outliers. This makes the kurtosis a non-robust measure of nongaussianity.

Measuring nongaussianity by negentropy [Delfose 1995]

The second important measure of nongaussianity is negentropy introduced in section 2.4.2, which is based on information theory. It is a robust but computationally complicated measure.

A fundamental principle of information theory is that a gaussian variable has the largest entropy among all random variables of equal variance matrix. This is why entropy can be used as a measure of nongaussianity. Entropy is defined by equation (2.72) in Chapter 2. Negentropy J of a random vector $\mathbf{y}$ is defined as follows

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \tag{3.9}$$

where $H(.)$ is the entropy function. $\mathbf{y}_{gauss}$ is a gaussian random vector with the same correlation and covariance matrix as $\mathbf{y}$. Negentropy is a kind of distance between an arbitrary random vector and its corresponding gaussian random vector. It is always nonnegative and it is zero if and only if $\mathbf{y}$ also has a gaussian distribution.

The problem in using negentropy as a measure of nongaussianity is the computation of negentropy. It is computationally very difficult using the fundamental definition in Eq. (3.9) since it requires an estimate of the pdf. Thus, we need to approximate the negentropy function by more computable functions.

Hyvarinen derived a fixed point algorithm by maximizing negentropy in [Hyvarinen 1999]. This algorithm is also called FastICA algorithm and is very suitable for off-line processing of BSS.

The methods described above consider the estimation of only one of the independent source signals. In practice, however, we may need to estimate more than one independent source signals. This can be done by running the algorithm many times and constraining the new weight vector to be orthogonal to all available weight vectors since the vectors $\mathbf{w}_i$ corresponding to different independent components must be orthogonal.

3.2.3 BSS by Maximum Likelihood Estimation [Cardoso 1997]

By determining the weight parameters in BSS that result in the highest probability for the observations, the maximum likelihood estimation method can be formulated based on statistical estimation theory. BSS algorithms maximizing likelihood estimation are derived in [Cardoso 1997] resulting in the update rule for ML estimation as follows.

$$\mathbf{W} = \mathbf{W} + \mu\Delta\mathbf{W} = \mathbf{W} + \mu\left(\mathbf{W}^{-T} - E\left\{\varphi(\mathbf{y})\mathbf{x}^T\right\}\right) \tag{3.10}$$

where μ is the learning rate and $\varphi(.)$ is a nonlinear function.

In [Amari 1996], Amari obtained the same result by minimizing Kullback-Leibler divergence in Eq. (2.71). He also implemented the natural gradient algorithm into the update equation. We will give detailed derivation of these algorithms in 3.2.6.

3.2.4 BSS by Minimization of Mutual Information[Common 1994]

In information theory, the mutual information is a measure of the dependence between random variables. Since the task in BSS is to recover the independent source signals, the mutual information can be used here as a cost function and the minimization of the mutual information can be used to recover the independent signals.

The mutual information is defined in equation (2.70) in Chapter 2. From the unmixing system model $\mathbf{y} = \mathbf{W}\mathbf{x}$, the mutual information between the outputs y_i is

$$I(y_1, y_2, ..., y_N) = \sum_{i=1}^{N} H(y_i) - H(\mathbf{y}) \tag{3.11}$$

By adjusting parameter matrix **W** to minimize the mutual information, the maximally independent components can be obtained.

Bell and Sejnowski in [Bell 1995] derived a BSS algorithm by minimizing the mutual information and obtained the same update equation as that of ML giving in Eq. (3.10).

3.2.5 BSS by Nonlinear Decorrelation[Hyvarinen 2001]

From Chapter 2, we noticed that a more practical definition of independent random variables is that they are nonlinearly uncorrelated, i.e. the random variables y_1 and y_2 are independent if and only if

$$E\{f(y_1)g(y_2)\} = E\{f(y_1)\}E\{g(y_2)\} \tag{3.12}$$

for any arbitrary continuous functions $f(.)$ and $g(.)$.

Thus nonlinear correlation can be used as a possible measure of independence. The nonlinear functions used in this kind of approach introduce higher-order statistics to help in BSS.

In [Herault], Herault and Jutten introduced feedback network and derived a BSS algorithm by nonlinear decorrelation. The computation load in this algorithm is heavy and the number of sources is limited to small number. In [Cichocki 1994], Cichocki et al proposed a feedforward network and also derived a BSS algorithm based on nonlinear decorrelation.

3.2.6 Instantaneous BSS Algorithm used in Our Thesis

As we know from above description of instantaneous BSS algorithms, there are different algorithms depending on how to measure independence. From [Bell 1995] [Amari 1996] and [Cardoso 1997], we know that the BSS algorithms derived from maximizing likelihood estimation and minimizing mutual information turn out to be the same. In the following, we

give the detail derivation from these directions to establish the basis of the BSS algorithm we will use. In the following, $\mathbf{s}=[s_1,...,s_N]^T$ is the vector of source signals, $\mathbf{x}=[x_1,...,x_M]^T$ is the vector of mixture signals, $\mathbf{y}=[y_1,...,y_N]^T$ is the vector of separated signals, $\mathbf{A}$ and $\mathbf{W}$ are instantaneous mixing and unmixing system and can be described as

$$\mathbf{A}=\begin{bmatrix} a_{11} & \cdots & a_{1N} \\ . & . & . \\ a_{M1} & \cdots & a_{MN} \end{bmatrix}; \quad \mathbf{W}=\begin{bmatrix} w_{11} & \cdots & w_{1M} \\ . & . & . \\ w_{N1} & \cdots & w_{NM} \end{bmatrix}.$$

BSS algorithm derived from Maximum Likelihood Estimation [Cardoso 1997]

Based on the mixing model in Fig. 3.2 and assuming we have T observations of $\mathbf{x}$, denoted by $\mathbf{x}(1),\mathbf{x}(2),...,\mathbf{x}(T)$, the likelihood to have an observation $\mathbf{x}(i), i=1,...,T$ given the parameter $\mathbf{W}$ is the product of the pdf $p_{\mathbf{x}}$ evaluated at the T points as defined in equation (2.62).

$$L=\prod_{t=1}^{T} p_{\mathbf{x}(t)|\mathbf{W}}\left(\mathbf{x}(t)\,|\,\mathbf{W}\right) \tag{3.13}$$

We will drop the dependence on $\mathbf{W}$ for simplicity. Based on the definition of probability for transformation $\mathbf{x}=\mathbf{As}$, as given by equation (2.29), we can show that

$$p_{\mathbf{x}(t)}\left(\mathbf{x}(t)\right)=\left|\det \mathbf{A}^{-1}\right| p_{\mathbf{s}}\left(\mathbf{s}\right) \tag{3.14}$$

Assuming $\mathbf{W}\approx\mathbf{A}^{-1}$ and $p_{\mathbf{y}}\left(\mathbf{y}\right)\approx p_{\mathbf{s}}\left(\mathbf{s}\right)$, (3.14) can be rewritten as

$$p_{\mathbf{x}(t)}\left(\mathbf{x}(t)\right)=\left|\det \mathbf{W}\right|\prod_{i=1}^{N} p_i\left(y_i(t)\right) \tag{3.15}$$

The likelihood can be written as

$$L=\prod_{t=1}^{T}\prod_{i=1}^{N}\left|\det \mathbf{W}\right| p_i\left(y_i(t)\right) \tag{3.16}$$

The task is to estimate the parameter $\mathbf{W}$ which maximizes the likelihood. In practice, it is more convenient to use the normalized log-likelihood as cost function since the

multiplication becomes addition operation in the log domain and both functions reach maximum at the same value for the parameter $\mathbf{W}$:

$$J(\mathbf{W}) = \frac{1}{T}\log L = \log|\det \mathbf{W}| + \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\log p_i(y_i(t)) \tag{3.17}$$

We can now estimate $\mathbf{W}$ by maximizing this likelihood expression with respect to $\mathbf{W}$. Using basic gradient descent approach, we can get

$$\Delta\mathbf{W} = \mu\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} = \mu\frac{\partial}{\partial \mathbf{W}}\left[\log|\det \mathbf{W}| + \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\log p_i(y_i(t))\right] \tag{3.18}$$

To simplify the right hand side, we note that the first term reduces to

$$\frac{\partial}{\partial \mathbf{W}}\log|\det(\mathbf{W})| = \mathbf{W}^{-T} \tag{3.19}$$

Now consider the second term:

$$\frac{\partial}{\partial \mathbf{W}}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\log p_i(y_i(t))\right] = E\left[\frac{\partial}{\partial \mathbf{W}}\sum_{i=1}^{N}\log p_i(y_i)\right] \tag{3.20}$$

Recall that

$$y_i = \sum_{j=1}^{M} w_{ij}x_j \tag{3.21}$$

And

$$\frac{\partial}{\partial w_{ij}}\sum_{i=1}^{N}\log p_i(y_i) = \frac{\dfrac{\partial p_i(y_i)}{\partial y_i}}{p_i(y_i)}\frac{\partial y_i}{\partial w_{ij}} = \frac{\dfrac{\partial p_i(y_i)}{\partial y_i}}{p_i(y_i)}x_j \tag{3.22}$$

Substituting (3.21) and (3.22) into (3.20), we get

$$\frac{\partial}{\partial \mathbf{W}}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\log p_i(y_i(t))\right] = -E\left[\varphi(\mathbf{y})\mathbf{x}^T\right] \tag{3.23}$$

where $E\{.\}$ is expectation operator, $\varphi(\mathbf{y}) = \left[\varphi_1(y_1), \varphi_2(y_2), ..., \varphi_N(y_N)\right]^T$ and

$$\phi_i(y_i) = -\frac{\partial}{\partial y_i}\log p_i(y_i) = -\frac{\dfrac{\partial p_i(y_i)}{\partial y_i}}{p_i(y_i)} \approx -\frac{1}{p_i(s_i)}\frac{\partial p_i(s_i)}{\partial s_i} \tag{3.24}$$

The update rule for ML estimation is then as follows:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu\Delta\mathbf{W} = \mathbf{W}_k + \mu(k)\left(\mathbf{W}_k^{-T} - E\left\{\varphi(\mathbf{y})\mathbf{x}^T\right\}\right) \tag{3.25}$$

where μ is the learning rate.

BSS algorithm derived from information maximization [Bell 1995]

In [Bell 1995], the structure in Fig. 3.3 is used for a two-input two-output separating system. In this diagram, $u_i = g(y_i); \mathbf{u} = \mathbf{g}(\mathbf{y})$. The purpose of this algorithm is to maximize the mutual information that the output $\mathbf{u}$ contains about its input $\mathbf{x}$.

$$H(\mathbf{u}) = \sum_{i=1}^{N} H(u_i) - I(u_1,...,u_N) \tag{3.26}$$

where $H(\mathbf{u})$ is joint entropy, $H(u_i)$ is marginal entropy and $I(u_1,...,u_N)$ is mutual information between u_i. To obtain the update equation for $\mathbf{W}$, we have to evaluate $\frac{\partial H}{\partial \mathbf{W}}$ as follows.

Differentiating Eq. (3.26), we get

$$\frac{\partial H(\mathbf{u})}{\partial \mathbf{W}} = \frac{\partial}{\partial \mathbf{W}}\left(-E\left[\log\left(p(\mathbf{u})\right)\right]\right) \tag{3.27}$$

where

$$p(\mathbf{u}) = \frac{p(\mathbf{x})}{|\mathbf{J}(\mathbf{x})|} \tag{3.28}$$

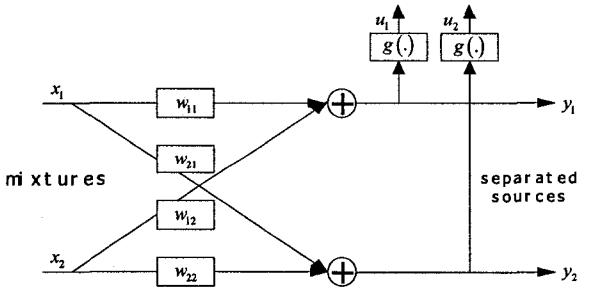


Figure 3-3: Two inputs and two outputs separating system

architecture based on information maximization from [Bell 1995]

$$\mathbf{J}(\mathbf{x}) = \det\begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \cdots & \frac{\partial u_1}{\partial x_N} \\ . & \cdots & . \\ \frac{\partial u_N}{\partial x_1} & \cdots & \frac{\partial u_N}{\partial x_N} \end{bmatrix} \tag{3.29}$$

$$\begin{aligned} \frac{\partial H(\mathbf{u})}{\partial \mathbf{W}} &= \frac{\partial}{\partial \mathbf{W}}\left(-E\left[\log\left(p(\mathbf{x})\right) - \log\left(\left|\mathbf{J}(\mathbf{x})\right|\right)\right]\right) \\ &= \frac{\partial}{\partial \mathbf{W}} E\left[\log\left(\left|\mathbf{J}(\mathbf{x})\right|\right)\right] = E\left[\frac{\partial}{\partial \mathbf{W}}\log\left(\left|\mathbf{J}(\mathbf{x})\right|\right)\right] \end{aligned} \tag{3.30}$$

Note that

$$\frac{\partial u_i}{\partial x_j} = \frac{\partial u_i}{\partial y_k}\frac{\partial y_k}{\partial x_j} = w_{ij}\frac{\partial u_i}{\partial y_k} \tag{3.31}$$

$$\frac{\partial u_i}{\partial y_k} = 0, \text{ for } i \neq k \tag{3.32}$$

Then Eq. (3.29) can be rewritten as

$$\mathbf{J}(\mathbf{x}) = \det\left(\begin{bmatrix} \frac{\partial u_1}{\partial y_1} & \cdots & 0 \\ . & \cdots & . \\ 0 & \cdots & \frac{\partial u_N}{\partial y_N} \end{bmatrix}\begin{bmatrix} w_{11} & \cdots & w_{1N} \\ . & \cdots & . \\ w_{N1} & \cdots & w_{NN} \end{bmatrix}\right) = \prod_{i=1}^{N}\frac{\partial u_i}{\partial y_i}\det(\mathbf{W}) \tag{3.33}$$

Substituting back in Eq. (3.30), we get

$$\begin{aligned} \frac{\partial H(\mathbf{u})}{\partial \mathbf{W}} &= \frac{\partial}{\partial \mathbf{W}}\log\left(\left|\prod_{i=1}^{N}\frac{\partial u_i}{\partial y_i}\det(\mathbf{W})\right|\right) \\ &= \frac{\partial}{\partial \mathbf{W}}\log\left(\left|\det(\mathbf{W})\right|\right) + \frac{\partial}{\partial \mathbf{W}}\log\left(\prod_{i=1}^{N}\left|\frac{\partial u_i}{\partial y_i}\right|\right) \\ &= \frac{\partial}{\partial \mathbf{W}}\log\left(\left|\det(\mathbf{W})\right|\right) + \frac{\partial}{\partial \mathbf{W}}\sum_{i=1}^{N}\log\left|\frac{\partial u_i}{\partial y_i}\right| \end{aligned} \tag{3.34}$$

Recall that

$$\frac{\partial}{\partial \mathbf{W}}\log\left(\left|\det(\mathbf{W})\right|\right) = \mathbf{W}^{-T} \tag{3.35}$$

$$\frac{\partial}{\partial w_{ij}}\sum_{i=1}^{N}\log\left|\frac{\partial u_i}{\partial y_i}\right| = \frac{1}{\dfrac{\partial u_i}{\partial y_i}}\frac{\partial}{\partial y_i}\left(\frac{\partial u_i}{\partial y_i}\right)x_j = \frac{1}{p(y_i)}\frac{\partial p(y_i)}{\partial y_i}x_j \tag{3.36}$$

where $p(y_i) = \left|\dfrac{\partial u_i}{\partial y_i}\right|$

$$\frac{\partial}{\partial \mathbf{W}}\sum_{i=1}^{N}\log\left|\frac{\partial u_i}{\partial y_i}\right| = \frac{1}{p(\mathbf{y})}\frac{\partial p(\mathbf{y})}{\partial \mathbf{y}}\mathbf{x}^T \tag{3.37}$$

Letting

$$\varphi(\mathbf{y}) = -\frac{1}{p(\mathbf{y})}\frac{\partial p(\mathbf{y})}{\partial \mathbf{y}} \tag{3.38}$$

We can now rewritten Eq. (3.30) as:

$$\frac{\partial H(\mathbf{u})}{\partial \mathbf{W}} = \mathbf{W}^{-T} - \varphi(\mathbf{y})\mathbf{x}^T$$

Thus, the update rule derived from information maximazation is then as follows:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu\Delta\mathbf{W} = \mathbf{W}_k + \mu(k)\left(\mathbf{W}_k^{-T} - E\left\{\varphi(\mathbf{y})\mathbf{x}^T\right\}\right) \tag{3.39}$$

where μ is the learning rate. As we can see, it is identical with (3.25).

BSS algorithm derived from minimizing Kullback-Leibler divergence [Amari 1996]

Based on the diagram in Fig. 3.2, Kullback-Leibler divergence of the output signal vector is

$$D(p(\mathbf{y})\|q(\mathbf{y})) = \int p(\mathbf{y})\log\frac{p(\mathbf{y})}{\prod_{i=1}^{N}p_i(y_i)}d\mathbf{y} \tag{3.40}$$

where $p(\mathbf{y})$ is the joint probability density of output signals, $p_i(y_i)$ is the probability density of output signal y_i, $q(\mathbf{y}) = \prod_{i=1}^{N}p_i(y_i)$.

$$D(p(\mathbf{y})\|q(\mathbf{y})) = \int p(\mathbf{y})\log p(\mathbf{y}) - \sum_{i=1}^{N}\left(\int p(\mathbf{y})\log p_i(y_i)\right) \tag{3.41}$$

If the outputs are independent, $p(\mathbf{y}) = \prod_{i=1}^{N}p_i(y_i)$.

$$D\left(p(\mathbf{y})\|q(\mathbf{y})\right) = -H(\mathbf{y}) + \sum_{i=1}^{N} H_i(y_i) \tag{3.42}$$

where $H(.)$ is the entropy operation given by:

$$\begin{aligned} H(\mathbf{y}) &= -E\left[\log\left(p(\mathbf{y})\right)\right] \\ &= -E\left[\log\left(\frac{p(x)}{\left|\det(\mathbf{W})\right|}\right)\right] \\ &= -E\left[\log\left(p(\mathbf{x})\right)\right] + \log\left(\left|\det(\mathbf{W})\right|\right) \\ &= H(\mathbf{x}) + \log\left(\left|\det(\mathbf{W})\right|\right) \end{aligned} \tag{3.43}$$

Then, substituting in Eq. (3.42), we get

$$D\left(p(\mathbf{y})\|q(\mathbf{y})\right) = -H(\mathbf{x}) - \log\left|\det(\mathbf{W})\right| - \sum_{i=1}^{N} E\left[\log\left(p_i(y_i)\right)\right] \tag{3.44}$$

Using standard gradient

$$\Delta D = \frac{\partial D}{\partial \mathbf{W}} = -\frac{\partial}{\partial \mathbf{W}} H(\mathbf{x}) - \frac{\partial}{\partial \mathbf{W}} \log\left(\left|\det(\mathbf{W})\right|\right) - \frac{\partial}{\partial \mathbf{W}} \sum_{i=1}^{N} E\left[\log\left(p_i(y_i)\right)\right] \tag{3.45}$$

Noting that

$$\frac{\partial H(\mathbf{x})}{\partial \mathbf{W}} = 0 \tag{3.46}$$

$$\frac{\partial}{\partial \mathbf{W}} \log\left(\left|\det(\mathbf{W})\right|\right) = \mathbf{W}^{-T} \tag{3.47}$$

and $y_i = w_{ij} x_j$. Elements in the last term of the right hand side in Eq. (3.45) can be expanded as

$$\frac{\partial}{\partial w_{ij}}\sum_{i=1}^{N}E\left[\log\left(p_i(y_i)\right)\right]=E\left[\frac{\partial}{\partial w_{ij}}\log\left(p_i(y_i)\right)\right]$$
$$=E\left[\frac{\frac{\partial p_i(y_i)}{\partial y_i}}{p_i(y_i)}\frac{\partial y_i}{\partial w_{ij}}\right] \tag{3.48}$$
$$=E\left[\frac{\frac{\partial p_i(y_i)}{\partial y_i}}{p_i(y_i)}x_j\right]$$

The last term on the right hand side of Eq. (3.45) becomes

$$\frac{\partial}{\partial \mathbf{W}}\sum_{i=1}^{N}E\left[\log\left(p_i(y_i)\right)\right]=-E\left[\varphi(\mathbf{y})\mathbf{x}^T\right]$$

where $\varphi(\mathbf{y})=\left[\frac{\frac{\partial p_1(y_1)}{\partial y_1}}{p_1(y_1)},\cdots,\frac{\frac{\partial p_N(y_N)}{\partial y_N}}{p_N(y_N)}\right]$ is a nonlinear function related to the probability density function of source signals.

Thus

$$\Delta D=\frac{\partial D}{\partial \mathbf{W}}=-\mathbf{W}^{-T}+E\left[\varphi(\mathbf{y})\mathbf{x}^T\right] \tag{3.49}$$

The coefficients $\mathbf{W}$ in the unmixing system are then updated as follows.

$$\mathbf{W}(k+1)=\mathbf{W}(k)+\Delta\mathbf{W} \tag{3.50}$$

$$\Delta\mathbf{W}_{\text{standard\_grad}}=-\mu\frac{\partial D}{\partial \mathbf{W}}=\mu\left(\mathbf{W}^{-T}-E\left[\varphi(\mathbf{y})\mathbf{x}^T\right]\right)$$

The update rule is then as follows:

$$\mathbf{W}_{k+1}=\mathbf{W}_k+\mu\Delta\mathbf{W}=\mathbf{W}_k+\mu(k)\left(\mathbf{W}_k^{-T}-E\left\{\varphi(\mathbf{y})\mathbf{x}^T\right\}\right) \tag{3.51}$$

where μ is the learning rate.

We can see that the update equation derived from minimizing Kullback-Leibler divergence in (3.51) is also identical with Eq. (3.25) (update equation derived from maximum likelihood estimation) and Eq. (3.39) (update equation derived from information maximization).

In [Amari 1996], Amari also implemented the natural gradient algorithm into the update equation in (3.51) to improve the convergence speed and resulting in the following update equation:

$$\Delta \mathbf{W}_{\text{natural\_grad}} = -\mu \frac{\partial D}{\partial \mathbf{W}} \mathbf{W}^T \mathbf{W} = \mu \left[\mathbf{I} - E\left(\varphi(\mathbf{y}) \mathbf{y}^T \right) \right] \mathbf{W} \tag{3.52}$$

And the update equation is

$$\mathbf{W} = \mathbf{W} + \mu \Delta \mathbf{W} = \mathbf{W} + \mu \left(\mathbf{I} - E\left\{ \phi(\mathbf{y}) \mathbf{y}^T \right\} \right) \mathbf{W} \tag{3.53}$$

In Chapter 4, we will compare the performance of standard gradient and natural gradient BSS algorithm.

3.3 Convolutive Blind Source Separation

The basic instantaneous BSS model and the algorithms to identify the unmixing matrix have some shortcomings that prevent their successful application to many real world situations. These include the effect of noise on successful learning of the separation solution, possibly unknown number of sources (especially noise sources), and the assumption that the source signals are stationary. Most notable is the assumption of the instantaneous mixing of the sources. In any real world recording, where the propagation of the signals through the medium is not instantaneous, there will be phase differences between the sources in the mixtures. In general, the sensor is not observing just a clean copy of the source, but a sum of multi-path copies distorted by the environment which can be modeled as convolutive mixtures. Thus, the application of the basic instantaneous BSS model is highly limited and we have to extend the basic BSS model to more realistic assumptions. In this section, we concentrate on convolutive BSS.

Convolutive blind signal separation has many applications in the real world environment, e.g. speech enhancement in the presence of multiple microphones, crosstalk removal in

multichannel communications, multipath channel identification and equalization, direction of arrival estimation in sensor arrays, improvement over beamforming microphones for audio and passive sonar, and identification of independent sources in various biological signals and others. Thus, in the following subsection, we will introduce the convolutive BSS model and methods for convolutive BSS.

3.3.1 Convolutive BSS Model

The convolutive BSS model is illustrated in Fig. 3.4. In the convolutive blind signal separation setup, N source signals $\{s_j(k)\}$; $1 \le j \le N$, pass through an unknown N-input, M-output linear time-invariant mixing system to yield the M mixed signals $\{x_i(k)\}$. Each source signal $s_i(k)$ is statistically independent of every other source signal $s_j(l)$ for $i \neq j$ and for all k and l.

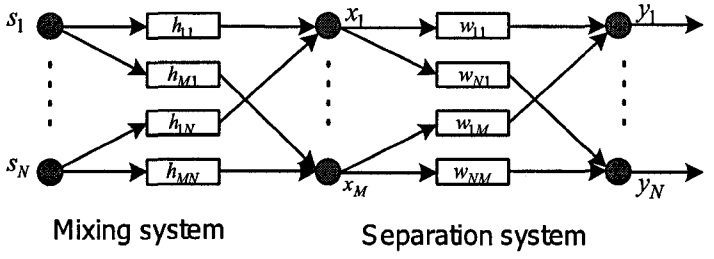


Figure 3-4: Convolutive BSS model

Defining the vectors $\mathbf{s}(k) = [s_1(k),...,s_N(k)]^T$ and $\mathbf{x}(k) = [x_1(k),...,x_M(k)]^T$, the i th sensor signal $x_i(t)$ is given by the noiseless linear convolutive mixing model as following:

$$x_i(t) = \sum_{j=1}^{N} \sum_{k=0}^{L-1} h_{ij}(k) s_j(t-k), \quad i = 1,2,...,M. \tag{3.54}$$

where t is the discrete-time index, $\{h_{ij}(k)\}$ is the impulse response characterizing the path from source j to sensor i, and $L-1$ defines the order of the FIR filters used to model the impulse response.

Equivalently, we can write

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_M \end{bmatrix} = \begin{bmatrix} \mathbf{h}_{11} & \cdots & \mathbf{h}_{1N} \\ \mathbf{h}_{21} & \cdots & \mathbf{h}_{2N} \\ \cdots & \cdots & \cdots \\ \mathbf{h}_{M1} & \cdots & \mathbf{h}_{MN} \end{bmatrix} * \begin{bmatrix} s_1 \\ s_2 \\ \cdots \\ s_N \end{bmatrix} \tag{3.55}$$

The z -transform $H_{ij}(z)$ of the system transfer function between the jth source and the ith sensor can be written as:

$$H_{ij}(z) = \sum_{t=0}^{L-1} h_{ij}(t) z^{-t} \tag{3.56}$$

And the convolutive BSS model in the z -domain given by:

$$X_i(z) = \sum_{j=1}^{N} H_{ij}(z) S_j(z), \quad i = 1,2,...,M. \tag{3.57}$$

where the convolution operation becomes simple multiplication.

The task of the convolutive BSS is to adaptively adjust an unmixing system with a causal FIR matrix $\{\mathbf{w}_{ij}\}$ such that the outputs of this system $\mathbf{y}(k) = [y_1(k),...,y_M k)]^T$ contain estimates of the N source signal sequences in $\{\mathbf{s}(k)\}$ without crosstalk. The i th output of the unmixing system is given as:

$$y_i(t) = \sum_{j=1}^{M} \sum_{k=0}^{l-1} w_{ij}(k) x_j(t-k), \quad i = 1,2,...,N. \tag{3.58}$$

Equivalently, we can also write it in vector form as

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ y_N \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{11} & \cdots & \mathbf{w}_{1M} \\ \mathbf{w}_{21} & \cdots & \mathbf{w}_{2M} \\ \cdot & \cdots & \cdot \\ \mathbf{w}_{N1} & \cdots & \mathbf{w}_{NM} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ x_M \end{bmatrix} \tag{3.59}$$

The unmixing model can also be rewritten in the z-domain as:

$$Y_i(z) = \sum_{j=1}^{M} W_{ij}(z) X_j(z), \quad i = 1,2,...,N. \tag{3.60}$$

where $W_{ij}(z)$ represents the unmixing or separation system.

From the model above, we can see that every element of the mixing matrix and the unmixing matrix in the convolutive BSS model is a ***filter*** instead of a ***scalar*** in the basic BSS model.

We will now review the three kinds of approaches to implement convolutive BSS algorithms.

3.3.2 Time Domain Convolutive BSS

The straightforward method for convolutive BSS problem is to extend existing instantaneous BSS algorithms to the convolutive situation in the time domain.

As a very first try, in [Torkkola 1996 A], Torkkola extended Bell and Sejnowski's informax algorithm [Bell 1995] to the convolutive situation where only delays between different source signals are considered and a feedback system architecture is used for this case. The separation system in [Torkkola 1996 A] is illustrated in Fig. 3.5. In this system, bias weights w_{01} and w_{02} are considered to account for possible non-zero average value. Since all signals are assumed to have zero average, these coefficients are generally not used in the later discussions.

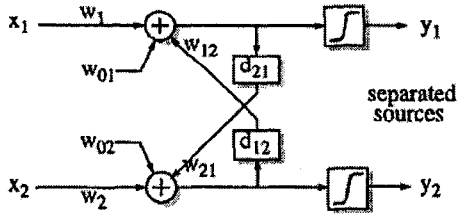


Figure 3-5: Torkkola's separation system for delays [Torkkola 1996 A]

The time domain adaptation rules for the weights $\mathbf{W}$ and the delays d_{ij} were derived based on the informax principle by stochastic gradient algorithm as follows.

$$\Delta \mathbf{W} \propto (\mathbf{I} - \mathbf{y}\mathbf{x}^T) + \mathbf{W}^{-T} \tag{3.61}$$

$$\Delta d_{ij} \propto -(1-2y_i) w_{ij} g\left(y\left(t - d_{ij}\right)\right) \tag{3.62}$$

where d_{ij} is delay between source signal i and j; $g(.)$ is a nonlinear function.

Although the approach for aligning delays in the above unmixing system is shown to be able to converge to correct values, it is very limited since only one delay with respect to one other source is considered.

In [Torkkola 1996 B], the real convolutive situation in which a matrix of filters replaces the matrix of scalars in basic BSS model was considered. A feedforward and a feedback network architecture are proposed to deal with convolutive mixtures. The system is illustrated in Fig. 3.6 where the w_{ij} boxes refer to FIR filters.

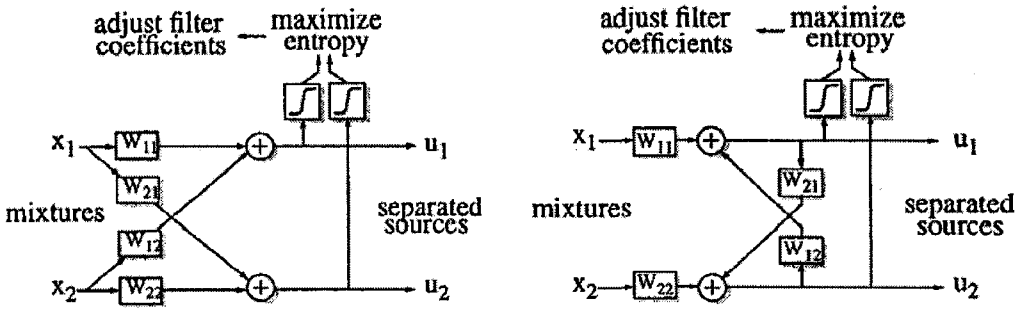


Figure 3-6: Torkkola's feedforward and feedback system for convolutive mixtures from [Torkkola 1996 B]

Again by maximizing the entropy at the output, the adaptation rules for the filter coefficients are easily derived in the time domain using stochastic gradient algorithm.

The above time domain algorithms exhibit very strong local minima. Another drawback of Torkkola's algorithm is the assumption that the filters in the system are causal. In general,

only minimum-phase filters have stable causal inverse. Thus, this method limits itself to applications of minimum-phase mixing system.

In [Lee 1997][Lee 1997 B], the feedforward filter network was studied and a more efficient time domain algorithm is derived using the natural gradient adaptation. Nonminimum-phase systems are considered by allowing acausal filters in the network since acausal filters can be realized by introducing an appropriate delay to the adaptation.

In conclusion, although there are ambiguities issues (scaling, sign, order) as in instantaneous BSS, the main problem for time domain convolutive BSS methods is its efficiency because of its high computational load and low convergence rate. Generally, time domain algorithms are only efficient for small mixing systems. For systems with long transfer function, their computational load is very heavy. In Chapter 8, we will approach this problem and propose novel time domain convolutive BSS algorithms with reduced computational complexity and better convergence rate.

3.3.3 Frequency Domain Convolutive BSS

The time domain convolutive BSS methods described above are very computationally demanding, and their convergence speed is slow, especially in real world applications such as audio signal separation. In such cases, the filters may need thousands of taps to properly invert the mixing matrix.

From the computational load point of view, it may be reasonable to move the algorithms to the frequency domain since convolution of long filters in the time domain becomes element-wise multiplication in the frequency domain as shown in (3.57) and (3.60). The significant advantage of frequency domain convolutive BSS methods is their reduced computational complexity. Assuming L is the length of the mixing filters, the computational complexity of frequency domain algorithm is $O(L \cdot \log L)$, whereas for time domain approaches it is $O(L^2)$, for long L, making the time domain algorithms inefficient. On the other hand,

frequency domain algorithms have better convergence speed since the filter parameters in the frequency domain methods lie in an orthogonal space.

Using short-time Fourier transform, the mixing system can be represented as

$$\mathbf{X}(f,t) = \mathbf{H}(f)\mathbf{S}(f,t) \tag{3.63}$$

where f is the frequency and t represents the time block of the short-time Fourier transformation, $\mathbf{S}(f,t) = [S_1(f,t),...,S_N(f,t)]^T$ is the transformation of the t th block of source signal vector, $\mathbf{X}(f,t) = [X_1(f,t),...,X_M(f,t)]^T$ is the transformation of the t th block of observed mixtures, $\mathbf{H}(f)$ is the mixing system matrix at frequency f.

The unmixing process can also be formulated in the frequency domain as:

$$\mathbf{Y}(f,t) = \mathbf{W}(f)\mathbf{X}(f,t) \tag{3.64}$$

where $\mathbf{Y}(f) = [Y_1(f,t),...,Y_N(f,t)]^T$ is the transformation of the t th block of estimated source signal vector, $\mathbf{W}(f)$ is the mixing system matrix at frequency f.

From Equations (3.63) and (3.64), we can see that the resulting mixtures at every frequency bin is an instantaneous mixing of the corresponding frequency bin of the original sources. This means that convolutive BSS problem is transformed as complex-valued instantaneous BSS at every frequency bin. Thus, any complex-valued instantaneous ICA algorithm can then be employed to deal with the separation in individual frequency bins. The updating equation (3.53) for instantaneous BSS is exploited for the learning rule of the unmixing matrix $\mathbf{W}$ at every frequency bin as follows.

$$\mathbf{W}_{i+1}(f) = \mathbf{W}_i(f) + \mu\left[\mathbf{I} - \varphi(\mathbf{y}(f,t))\mathbf{y}(f,t)^H\right]\mathbf{W}_i(f) \tag{3.65}$$

The advantage of frequency domain convolutive BSS lies in three factors. First the computational complexity is reduced since the convolution operations are transferred into multiplication operations by short-time FFT. Second, separation process can be performed in

parallel at all frequency bins. Finally any complex-valued instantaneous ICA algorithm can be employed to deal with the separation at each frequency bin. However, the permutation and scaling ambiguity in ICA algorithms as in Eq. (3.6), which is not a problem for instantaneous BSS, becomes a serious problem in frequency domain convolutive BSS.

This problem can be illustrated by Fig. 3.7. Frequency domain convolutive BSS is performed by instantaneous BSS at each frequency bin separately. As a result, the order and the scale of the unmixed signals are random because of the inherent indeterminacy of ICA algorithms. When we transform the separated signals back from frequency domain to time domain, the components at different frequency bin may not come from the same source signal and may not have consistent scale. Thus, we need to align the permutation and adjust the scale in each frequency bin so that a separated signal in time domain is obtained from frequency components of the same source signal and with consistent amplitude. This is well-known as the permuation and scaling problems of frequency domain convolutive BSS.

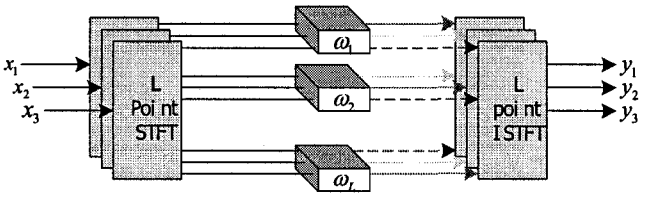


Figure 3-7: An illustration of the permutation problem in frequency domain convolutive BSS

Many approaches are proposed to deal with permutation and scaling problems in frequency domain convolutive BSS algorithms. In [Smaragdis 1998], the unmixing matrix is normalized to unity to ensure the unmixing matrix does not scale the data. In [Murata 2001], the scaling ambiguity is solved by calculating the inverse of a separation matrix. To deal with frequency permutation, in [Smaragdis 1998][Parra 2000], the unmixing matrix is smoothed by reducing the filter length or by averaging the separation matrices with adjacent frequencies. In [Kurita 2000][Saruwatari 2002][Ikram 2002], the directivity patterns obtained from the separation matrix are analyzed to search for the DOA of the source signal and these DOA information is used to resolve frequency permutation. In [Murata 2001], the inter-frequency correlation of the separated signal envolopes is used to resolve permutation problem. Clearly, all these approaches to fixing the permutation and scaling ambiguity

further add to the complexity and can result in performance deterioration when they do not perform perfectly well.

3.3.4 Time Frequency Domain Convolutive BSS

In this approach, only a single aspect or some aspects of the separation algorithm is done in the frequency domain; all other parts are done in the time domain. In [Joho 2003][Buchner 2004], the convolution computations in the time domain are speeded up by the overlap-save method in the frequency domain. In [Lambert 1997][Lee 1997], the separation criterion is applied in the time domain, a frequency domain representation of the separation filters is learned using FIR matrix algebra and the final time domain result is reconstructed using the overlap-save technique. The advantage of this method is that there is no permutation and scaling misalignment problem since independence criterion is implemented in the time domain. However, the main disadvantage of this method is that the estimated signals need to be moved from and to the frequency domain for every update. This reduces the computational savings obtained by going to the frequency domain.

3.4 Blind Speech Separation

Blind speech separation is a special case of BSS where the original signals and the separated signals are speech. Consider the scenario in which multiple speeches emitted from a number of different speakers or loudspeakers are perceived by us humans. Despite the fact that all speech signals arrive as a single waveform, it is possible for us to clearly distinguish between them and recognize those of particular interest. This phenomenon, referred to as the cocktail party effect, demonstrates the ability of humans to focus on signals of interest even in the presence of many competing sounds. BSS for speech signal is an attempt to mimic this human ability.

Blind separation of speech signal is much more difficult than one might expect. The complications we encounter in the real acoustical environment are as follows:

- The mixing is not instantaneous. Speech signals propagate very slowly and thus they arrive in the microphones at different times. Moreover, there are reverberant effects, especially if the recording is made in a large room. Thus the problem is more adequately modeled by a convolutive version of the BSS model.
- For speech signals in real acoustic envireonment, the filters used to model the mixing and unmixing channels are considerably long which makes it more difficult when using convolutive BSS algorithms.
- The system may be overcomplete since there may be more sources in the acoustic environment than the sensors.
- The speech signal is non-stationary and correlated between its samples in the same source. However, for the regular BSS algorithm, the standard assumption is that the source signals are i.i.d. signals.
- The mixing system may be nonstationary. In real acoustic environment, the speaker may be walking when speaking. Even the speaker is sitting in a place, he may move his head occasionally. This implies that the mixing matrix must be re-estimated quickly in a limited time frame, which also means a limited amount of data will be available.
- Noise is a big issue in real acoustic environment. It makes the estimation of the BSS model quite complicated, even in the basic instantaneous BSS case.

In this thesis, we will mainly focus on blind speech signal separation. We start by reviewing some existing approaches on blind speech signal separation.

3.4.1 Approaches Based on Convolutive BSS and Multichannel Blind Deconvolution

Convolutive BSS can be achieved by extending blind deconvolution approaches to the multichannel case, or by extending instantaneous blind source separation to the multipath case. Since blind deconvolution is a mature research field, it is possible to exploit some results in blind deconvolution to convolutive BSS algorithms. Researchers have applied multichannel blind deconvolution algorithms to convolutive blind speech signal separation directly and have achieved moderate success [Lambert 1996][Lambert 1997]. In [Lee 1997

A][Lee 1997 B], the multichannel blind deconvolution method is combined with the natural gradient adaptation algorithm to do convolutive BSS and some experimental results in real room recordings are obtained. In [Sun 2001] and [Kokkinakis 2003], convolutive BSS is achieved by combining the multichannel blind deconvolution, natural gradient adaptation and linear prediction analysis. Linear prediction analysis is used to whiten the mixtures of speech signal to make the input signal to the separating system satisfy the assumptions of multichannel blind deconvolution. The natural gradient adaptation method is used to accelerate the convergence rate. Some successful results for separating mixtures of multiple speech signals observed in a real acoustic environment are obtained. However, the problems are how to obtain accurate estimates of the linear prediction (LP) coefficients of the source signals and how to add the spectra information of the source signals to the extracted outputs. Another concern for these algorithms is the questionable benefit obtained from LP-based methods since there are no speech quality evaluation results given in these references.

3.4.2 Approaches Based on Convolutive BSS and Geometric Beamforming

Recently, some blind speech signal separation methods are proposed based on combined convolutive BSS and beamforming, something that human do automatically. Blind source separation and beamformimg have a similar goal—extracting specific source or sources while reducing undesired interferences. Both methods are concerned with the problem of optimizing a multichannel filter structure to improve the signals detected with a sensor array such a microphone array or an antenna array. However, in beamforming, the adaptation of multichannel filter exploits the source location information to form a spatial pattern with dominant response for the location of interest. The main drawback is the serious cross-talk or leakage problem, especially in a strong reverberant environment. On the other hand, from the preceding description of convolutive BSS algorithms, mainly the temporal and frequency information are used explicitly to separate the mixtures. The readily available information about the source locations is not used. In practice, however, desired and interfering signals often originate from different spatial locations and this information can be obtained easily. If this information can be used in the convolutive blind speech signal separation algorithms, their performance is expected to improve significantly.

In a series of papers [Parra 2001] [Parra 2002 A] [Parra 2002 B], a geometric source separation method was proposed to merge convolutive BSS with adaptive beamforming. The goal of this method is to resolve some of the ambiguities inherent in convolutive BSS algorithms by using geometric information such as sensor position or source location. In this paper, the geometric information used in adaptive beamforming is incorporated into the cross-power minimization algorithm of convolutive source separation in two ways. The first approach is the parameter initialization in which the unmixing filter matrix is initialized with a delay-sum beamformer pointing to the individual sources or initialized with beams that place zeros at all orientations of interfering sources. The second way is to incorporate the geometric information into the convolutive BSS algorithms by adding linear constraint conditions and using constrained optimization. The performance of this method is demonstrated for the separation of acoustic sources from multiple microphones in a reverberant environment.

From the above description of the geometric source separation method which was proposed to merge convolutive source separation with geometric beamforming, the benefit of this idea is that the readily available geometric information such as locations for source signals can be used to improve the performance of convolutive BSS algorithms in which only time and frequency information are used. However, in all these existing geometric source separation methods, the geometric constraints are incorporated into the convolutive BSS algorithms by parameter initialization or linear constraints. Although these approaches are reported to partially overcome some ambiguities inherent in convolutive BSS method and improve the performance of convolutive BSS, the performance of convolutive BSS algorithm and adaptive beamforming are constrained by one another because both are incorporated into one algorithm.

In [Saruwatari 2002][Saruwatari 2006], blind speech signal separation is performed by combing independent component analysis and beamforming. In this method, independent component analysis and beamforming are combined to deal with low convergence problem in convolutive BSS. First, ICA is used to perform blind source separation at every frequency

bin and the unmixing matrix can be obtained at each frequency bin. Accordingly, the directivity pattern at each frequency bin can be obtained from its unmixing matrix. Directions of arrival (DOA) of source signals are estimated from the directions of nulls at all frequency bins. In adaptation process, at each frequency bin, the direction null in the directivity pattern is compared with the estimated DOA of source signals, if it is steering to the proper direction, the unmixing matrix from ICA algorithm is used. If not, the null-steering beamformer constructed from the estimated DOA information is used to substitute unmixing matrix. By doing so, the unmixing matrix can be recovered from local minimum in the optimization procedure to improve its convergence speed.

3.4.3 Approaches Based on Convolutive BSS and Signal Nonstationarity

In the preceding convolutive BSS algorithms, it is assumed that the signals and the mixing system are stationary. However, in practical applications, such as acoustic signals recorded in a reverberant environment, the signals are not strictly stationary and the channels are slowly time variant.

In [Weinstein 1993], an alternative approach to the statistical independence condition is proposed to exploit the additional second-order information provided by non-stationary signals. It is shown that for non-stationary signals, a set of second order conditions can uniquely determine desired parameters, but no algorithm is provided there. In [Parra 2000], a multiple decorrelation approach was proposed for non-stationary signals based on the above principle. To reduce the computational load, the algorithm is implemented in the frequency domain and the least square adaptation approach is used to diagonalize the correlation matrix simultaneously at different time instants. Simulation results are obtained for real room recordings.

Other methods have been proposed for convolutive nonstationary BSS. In [Kawamoto], the cost function for instantaneous nonstationary BSS problem was extended to a convolutive mixture of nonstationary signals. The natural gradient adaptation is used to minimize the cost function. Simulation results show that this method can work for nonstationary artificial data

and nonminimum-phase systems. In [Jafari 2002], a wavelet domain algorithm was proposed for non-stationary convolutive mixtures. The natural gradient algorithm proposed in [Amari 1996] for BSS is extended to wavelet domain to reduce noise, further improve convergence speed of natural gradient algorithm and avoid permutation problem in Fourier transformation. The performance of the wavelet transform based algorithms is analyzed and reported to be superior to the frequency domain algorithm.

As we mentioned before, blind speech signal separation is a very complicated task. The prior information, independence and nongaussianity of the source signals, may be not enough to achieve good separation. To estimate the convolutive BSS model with a large number of parameters, and a rapid changing mixing matrix, more information on the signals and the mixing system may be required. First, one may need to combine the assumption of nongaussianity with the different time-structure assumptions. Speech signals have autocorrelations and nonstationarities, so this information could be used. Second, one may need to use some information on the mixing, for example, location of sources or configuration of sensors could be used. It is also possible that real-life speech signal separation requires sophisticated modeling of speech signals. Speech signals are highly structured, autocorrections and nonstationarity being just the simplest aspects of their time structure. These are potential research areas for blind speech signal separation.

3.5 Conclusion

In this chapter, we reviewed the instantaneous BSS model and its estimation methods from different points of view. The limitations of this basic BSS model are also analyzed. Then we introduced the more realistic BSS model – convolutive BSS model and its time domain and frequency domain estimation methods. Next, we considered blind speech signal separation and reviewed some existing approaches. The challenges for blind speech signal separation are also discussed in this chapter.

Chapter 4 Performance Comparison of Selected Existing BSS Algorithms

4.1 Introduction

In the previous chapter, we provided the basic understanding of blind source separation. BSS algorithms were classified into two classes: instantaneous BSS algorithms and convolutive BSS algorithms. The instantaneous BSS algorithms are used to separate the simpler case in which the mixtures are obtained by a mixing system expressed as a scalar matrix. The convolutive BSS algorithms are the extended version of instantaneous BSS algorithms, in which delays and multipath effects of real environments are considered. The mixing system of convolutive BSS can be expressed as a matrix whose elements are FIR filters. We also highlighted the special challenges when dealing with speech signals. Since convolutive BSS methods can be used to deal with more realistic environment, we will focus on them and on their applications to speech signal separation.

As a first step to understanding the properties and performance of established algorithms, we will apply these algorithms to different signals, run the simulations and discuss the results. Following these studies, we will propose new algorithms or implementations in the next chapters.

This chapter is organized as follows. In Section 4.2, we introduce the performance measures used in our simulations. In Section 4.3, we illustrate how to separate simultaneous mixtures by instantaneous BSS algorithms and compare performance of stochastic gradient and natural gradient BSS algorithms. In Section 4.4, we illustrate performance of time domain and frequency domain convolutive BSS algorithm for artificial source signals and speech signals. Finally we give our conclusions in Section 4.5.

4.2 Performance Evaluation

In our simulations, we need to define an objective performance index to measure the separation quality. In the following we introduce some performance evaluation tools commonly used in the literature. These tools will be used in our simulations in this chapter and the following chapters as well.

4.2.1 Performance Evaluation based on Intersymbol Interference

Representing the mixing system with $\mathbf{H}$ (normally we use $\mathbf{A}$ representing the mixing system in instantaneous BSS case and $\mathbf{H}$ for the mixing system in convolutive case, here we use $\mathbf{H}$ to represent both just for simplicity), the unmixing system with $\mathbf{W}$, we can express the global system as $\mathbf{P} = \mathbf{W} * \mathbf{H}$. In basic instantaneous BSS model, each element in ith row and jth column of $\mathbf{P}$ is a scalar p_{ij}. Ideally, $\mathbf{P}$ should be an identity matrix with possible permutations and scaling. Thus, only one p_{ij} is nonzero for a given i or a given j. In this situation, the intersymbol interference is defined either on a row or a column base [Amari 1998] as follows.

The row intersymbol interference (ISI) is defined as

$$ISI_{row} = \sum_i \left(\frac{\sum_j |p_{ij}|^2 - \max_j |p_{ij}|^2}{\max_j |p_{ij}|^2} \right) \tag{4.1}$$

The column intersymbol interference (ISI) is defined as

$$ISI_{column} = \sum_j \left(\frac{\sum_i |p_{ij}|^2 - \max_i |p_{ij}|^2}{\max_i |p_{ij}|^2} \right) \tag{4.2}$$

The overall intersymbol interference is defined as

$$ISI_{total} = ISI_{row} + ISI_{column} \tag{4.3}$$

If the sources have been separated perfectly, $\mathbf{P}$ becomes a permutation matrix. This means that in each of its rows and columns, only one of the elements equals unity while all the other elements are zero. Thus, the overall intersymbol interference reaches zero for an ideal separation. The larger the value of the overall intersymbol interference, the poorer the performance of the separation algorithm.

In the convolutive BSS model, each element in $\mathbf{P}$ is a vector $\mathbf{p}_{ij}$ and the intersymbol interference is defined as follows [Lambert 1996] by extending the above definition.

The row intersymbol interference (Row ISI) is defined as

$$ISI_{row} = \sum_i \left(\frac{\sum_i \sum_k \left| p_{ij}(k) \right|^2 - \max_{j,k} \left| p_{ij}(k) \right|^2}{\max_{j,k} \left| p_{ij}(k) \right|^2} \right) \tag{4.4}$$

The column intersymbol interference (Column ISI) is defined as

$$ISI_{column} = \sum_j \left(\frac{\sum_j \sum_k \left| p_{ij}(k) \right|^2 - \max_{j,k} \left| p_{ij}(k) \right|^2}{\max_{j,k} \left| p_{ij}(k) \right|^2} \right) \tag{4.5}$$

The overall intersymbol interference for convolutive BSS is also defined as the sum of ISI_{row} and ISI_{column}.

4.2.2 Performance Evaluation based on Signal-to-interference Ratio (SIR)

The performance of blind source separation system can also be evaluated by the signal to interference ration (SIR) [Makino 2005] which is defined as the power ratio between the target component and the interference components. The SIR of output i is obtained as

$$SIR_i = 10\log_{10}\frac{E\{p_{ii}s_i\}}{E\left\{\sum_{j\neq i} p_{ij}s_j\right\}}dB \tag{4.6}$$

for instantaneous BSS case and

$$SIR_i = 10\log_{10}\frac{E\{\mathbf{p}_{ii} * \mathbf{s}_i\}}{E\left\{\sum_{j\neq i} \mathbf{p}_{ij} * \mathbf{s}_j\right\}}dB \tag{4.7}$$

for convolutive BSS case, where * is the convolution operation and $E\{.\}$ is the expectation operation.

In all our simulations measuring SIR for speech signal, we cut off silence for more accurate evaluation.

4.2.3 Performance Evaluation based on PESQ Scores

Since the target signals in some of our simulations are speech signals, we use PESQ standard to measure the similarity between the recovered speech signal and the original speech signal to evaluate the performance of different separation approaches. The PESQ standard [ITU 2000] is described in the ITU-T P862 as a perceptual evaluation tool of speech quality. The key feature of the PESQ standard is that it uses a perceptual model to compare the original and degraded signal as processed by the human auditory system. The output of the PESQ is a measure of the subjective assessment quality of the degraded signal and is rated as a value between –0.5 and 4.5 which is known as the Mean Opinion Score (MOS). The higher the score, the better the speech quality.

4.3 Instantaneous Blind Source Separation Algorithms

A detailed review of instantaneous BSS algorithms has been provided in Chapter 3. Based on this review, it is clear that instantaneous BSS algorithms are relatively simple and cannot deal with complicated real world situations. Thus, the assessment provided here is given just

as a base for our understanding of convolutive BSS and is not our research focus. It should be noted that performance analysis and comparison for five instantaneous BSS algorithms has already been done in a good comparative study [Giannakopoulos 1998] [Giannakopoulos 1999].

4.3.1 Description of different gradient algorithm

In this section, we only assess Bell and Sejnowski's informax BSS algorithm [Bell 1995] and its corresponding natural gradient version [Amari 1996] to show how natural gradient adaptation algorithm improves convergence speed.

In [Bell 1995], the observations are transformed and processed through a nonlinear function $g(.)$ that approximates the cumulative density function of the sources. The system structure is illustrated in Fig. 4.1.

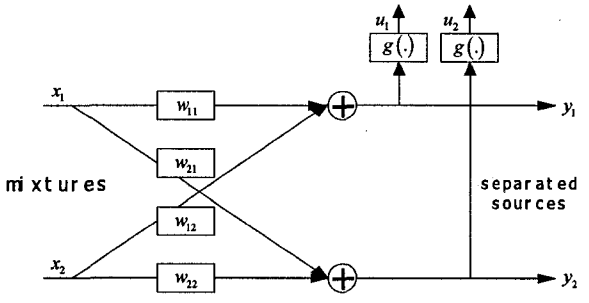


Figure 4-1: Two inputs and two outputs separating system architecture based on information maximization

In Fig. 4.1, $\mathbf{x}$ is the vector for received sensor signals and $\mathbf{W}$ is the unmixing system parameter matrix. The output vector is $\mathbf{y}$. To implement the information maximization algorithm, the output signal is processed by the nonlinear function $g(.)$ to get the nonlinear system output vector $\mathbf{u}$. The basic idea is that by maximizing the joint signal entropy defined by $Entropy(\mathbf{u})$, it is possible to minimize the mutual information between the outputs y_i of

the network. To maximize $Entropy(\mathbf{u})$ with respect to $\mathbf{W}$, a blind stochastic gradient adaptive algorithm is derived in [Bell 1995] as

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu(\mathbf{W}_k^{-T} - g(\mathbf{y})\mathbf{x}^T) \tag{4.8}$$

where μ is the step size or learning rate and k is the iteration number.

By using the natural gradient adaptive algorithm in [Amari 1996], a new separation matrix update equation is derived as given in equation (4.9) avoiding the calculation of $\mathbf{W}^{-1}$ in equation (4.8) and overcoming the poor convergence properties of stochastic gradient adaptation.

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu(\mathbf{I} - g(\mathbf{y})\mathbf{y}^T)\mathbf{W}_k \tag{4.9}$$

The detailed algorithm derivation can be found in Chapter 3.

4.3.2 Comparative performance of different gradient algorithm

In the first experiment, we use two generated Gamma signals as the source signals to exam the separation performance of the regular instantaneous BSS algorithm and natural gradient BSS algorithm described above.

The original signals are mixed by a scalar system to obtain the system outputs as the mixtures. In our simulation, the scalar system is given by

$$A=\begin{bmatrix} 0.4 & 0.7 \\ 0.6 & 0.5 \end{bmatrix} \tag{4.10}$$

Two BSS algorithms are used to separate the mixtures. One is the Bell and Sejnowski's informax algorithm described by Eq. (4.8) and the other one is natural gradient based BSS algorithm descried by Eq. (4.9). The step size used in this experiment is 0.002. The nonlinear function used is $g(y) = \tanh(y)$. Both algorithms work in on-line mode. The separation performance evaluated by ISI and SIR is illustrated in Fig. 4.2, Fig. 4.3 and Fig. 4.4.

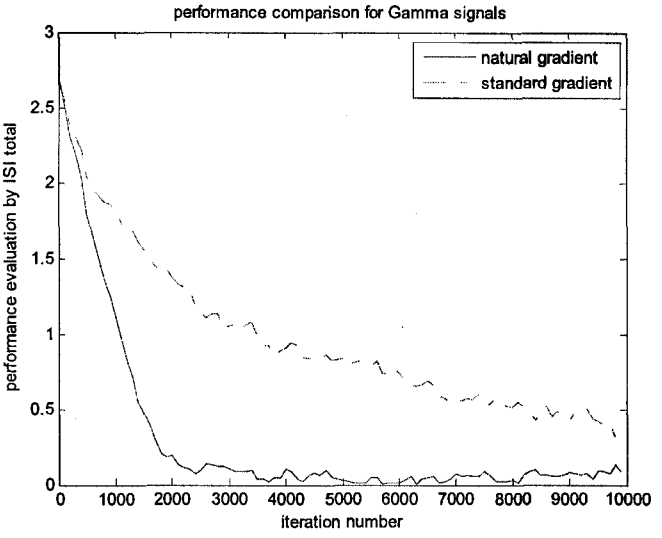


Figure 4-2: Separation performance evaluated by ISI for Gamma signal

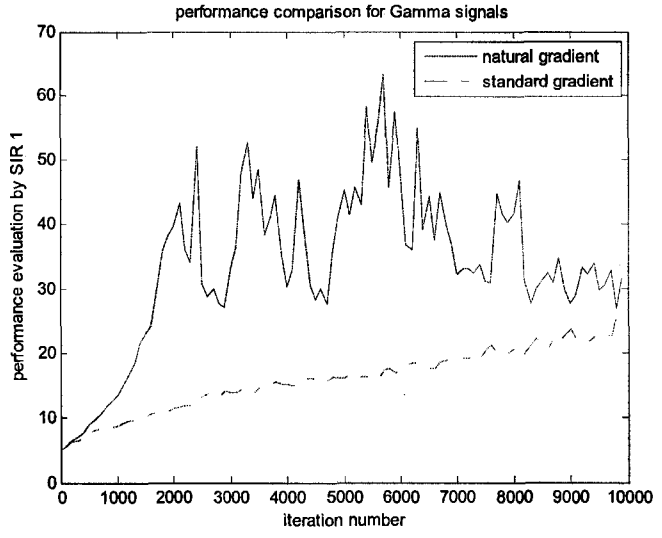


Figure 4-3: Separation performance of the first output evaluated by SIR for Gamma signal

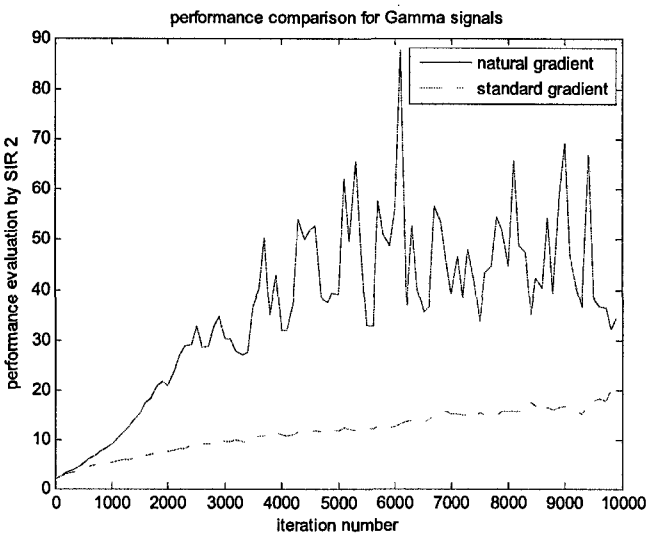


Figure 4-4: Separation performance of the second output evaluated by SIR for Gamma signal

In the second experiment, we use two speech signals as the source signals to exam the separation performance of the regular instantaneous BSS algorithm and the natural gradient BSS algorithm described above. The original signals are mixed by the same scalar system in (4.10) to obtain the system outputs as the mixtures. The step size used in this experiment is 0.002. The nonlinear function used is $g(y) = \tanh(y)$. The separation performance evaluated by ISI and SIR is illustrated in Fig. 4.5, Fig. 4.6 and Fig. 4.7.

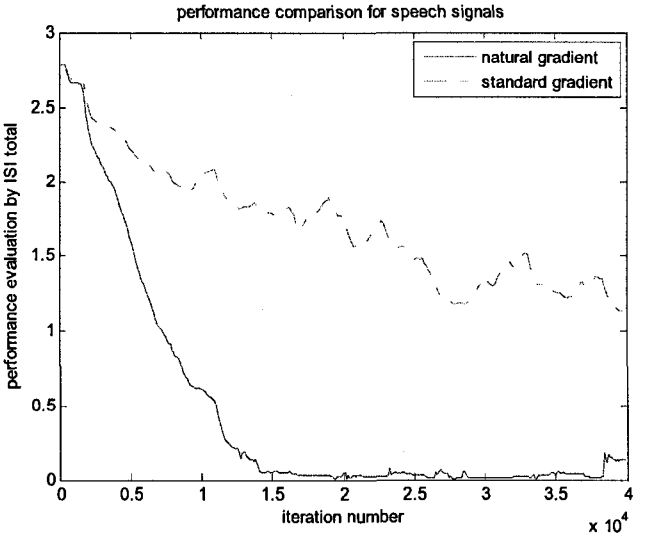


Figure 4-5: Separation performance evaluated by ISI for speech signal mixtures

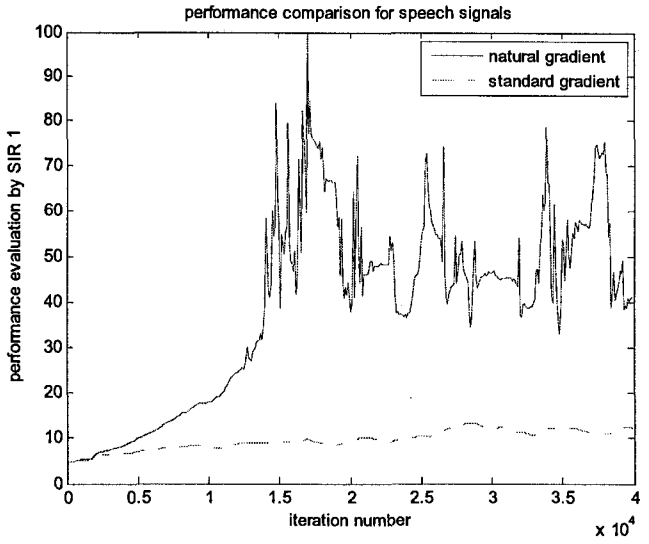


Figure 4-6: Separation performance of the first output evaluated by SIR for speech signal mixtures

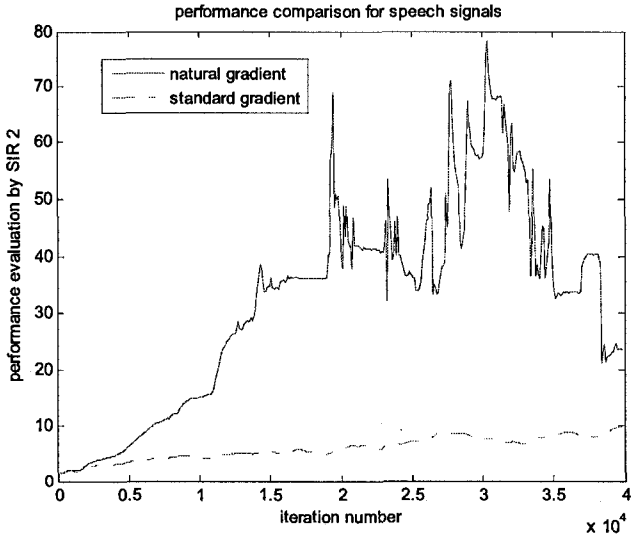


Figure 4-7: Separation performance of the second output evaluated by SIR for speech signal mixtures

It is clear from the above two experiments that the natural gradient provides superior convergence compared to the standard gradient. This is also confirmed by the PESQ scores in Table 4.1 and our informal listening experience.

PESQ	Mixture		Regular gradient BSS		Natural gradient BSS	
	mix1	mix2	out1	out2	out1	out2
s1	1.31	1.78	0.88	2.39	0.33	3.40
s2	2.10	1.73	2.56	1.16	4.18	0.64

Table 4-1: PESQ scores for mixtures and separated speech signals by regular and natural gradient BSS algorithms

From these simulation results, it is confirmed that the natural gradient adaptation algorithm has better convergence performance and better separation performance. The results are consistent by ISI, SIR as well as PESQ (for speech) assessment tools.

Based on the above simulation results and the results in [Giannakopoulos 1998][Giannakopoulos 1999], we conclude that:

Simultaneous mixtures of mutually independent sources can be separated successfully by instantaneous BSS algorithms.
Algorithms based on natural gradient adaptation converge faster and have better separation quality than algorithms based on stochastic gradient adaptation.

4.4 Convolutive Blind Source Separation

As we know from Chapter 3, the convolutive BSS model was proposed to deal with more realistic situations where signal delays and multi-path effects are taken into account, thus making it applicable to many real world applications. The convolutive BSS model has been described in detail in Chapter 3. In this section, we implement the convolutive BSS algorithm in time domain and frequency domain to better understand their performance.

4.4.1 Simulations

Experiment #1: Time domain BSS for convolutive mixture of Gamma signals

In this simulation, the mixing system is

$$H=\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}=\begin{bmatrix} 1 & 1 & 0.75; & 0.5 & 0.3 & 0.2 \\ -0.2 & 0.4 & 0.7; & 0.2 & 1 & 0 \end{bmatrix} \tag{4.12}$$

For this experiment, the filter length is L=64 and the step size is μ=0.000023.

The separation performance evaluated by ISI row and ISI column are illustrated in Fig. 4.8 and Fig. 4.9. The separation performances evaluated by SIR for two outputs are illustrated in Fig. 4.10 and Fig. 4.11.

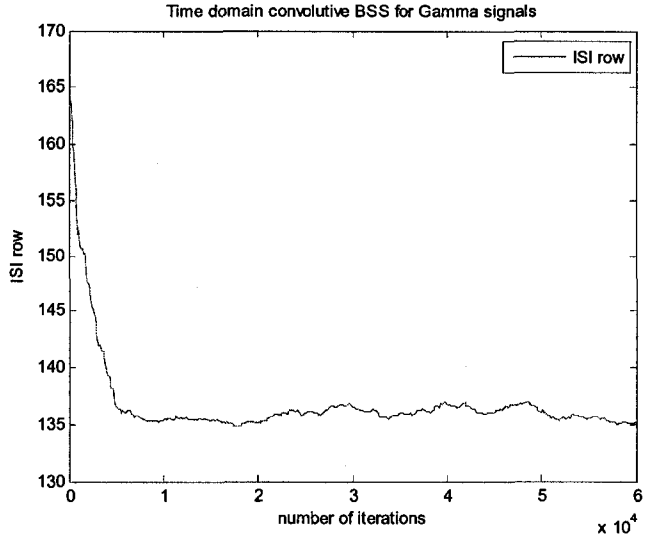


Figure 4-8: Separation performance evaluated by ISI row for convolutive Gamma mixture signals

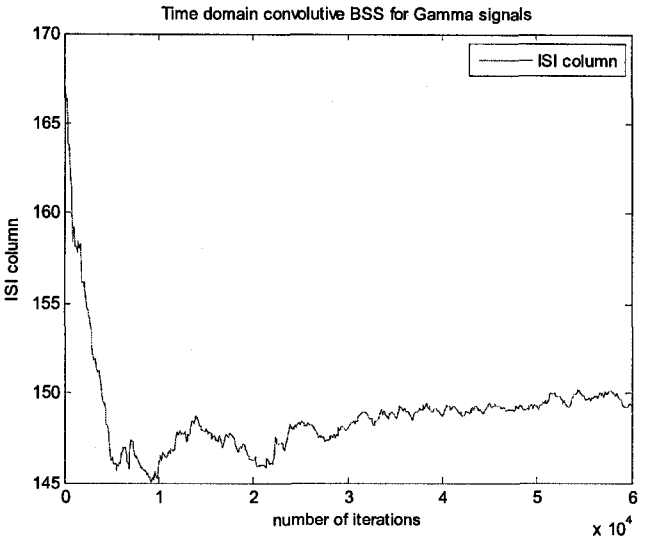


Figure 4-9: Separation performance evaluated by ISI column for convolutive Gamma mixture signals

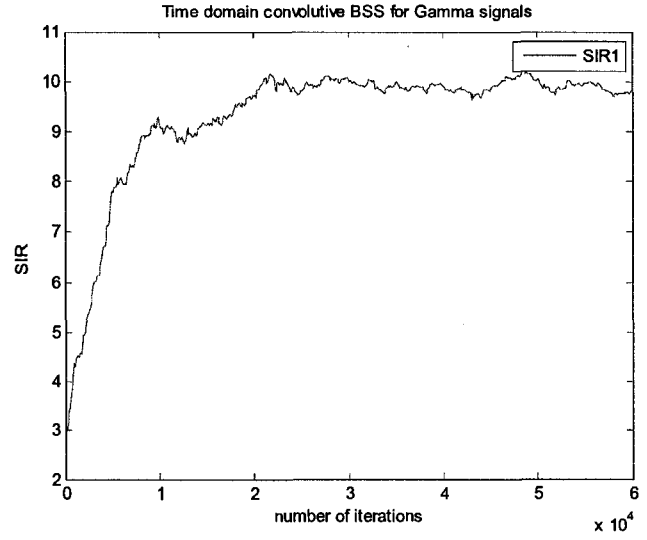


Figure 4-10: Separation performance of the first output evaluated by SIR for convolutive Gamma mixture signals

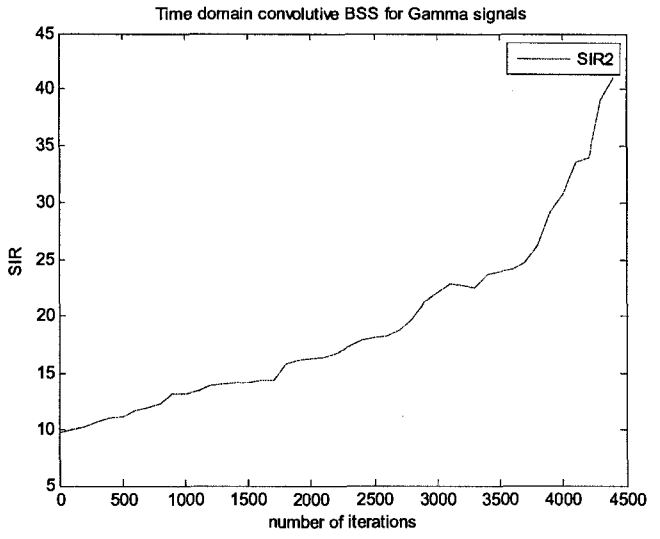
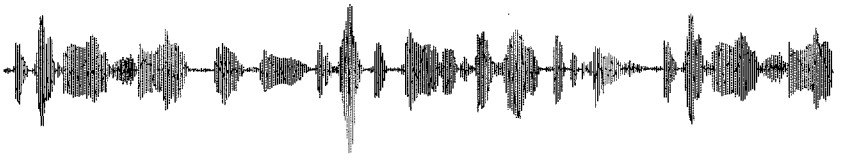


Figure 4-11: Separation performance of the second output evaluated by SIR for convolutive Gamma mixture signals

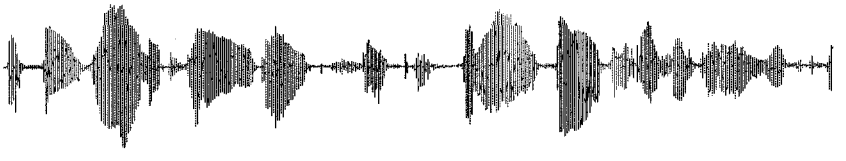
From above simulation results, we can see that time domain BSS algorithm can successfully separate convolutively mixed Gamma signals in terms of signal to interference ratio. In the following, we exam its separation performance for speech signals.

Experiment #2: Time domain BSS for convolutive mixture of speech signals

The input speech signals are illustrated as follows in Fig. 4.12. We use the same mixing system as used in Experiment #1 in (4.12). In this experiment, the filter length is =64 and the step size is μ=0.000023.



Source #1



Source #2

Figure 4-12: Source speech signals used in time domain convolutive BSS algorithm

The separation performance evaluated by ISI row and ISI column are illustrated in Fig. 4.13 and Fig. 4.14. The separation performances evaluated by SIR for two outputs are illustrated in Fig. 4.15 and Fig. 4.16.

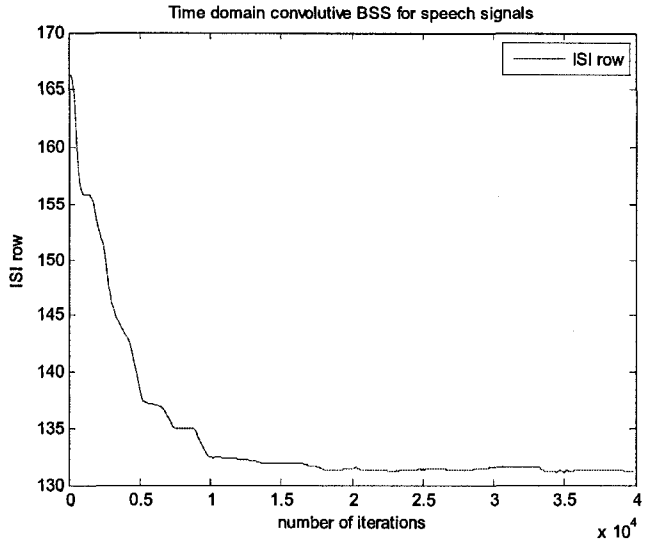


Figure 4-13: Separation performance evaluated by ISI row for convolutive speech mixture signals

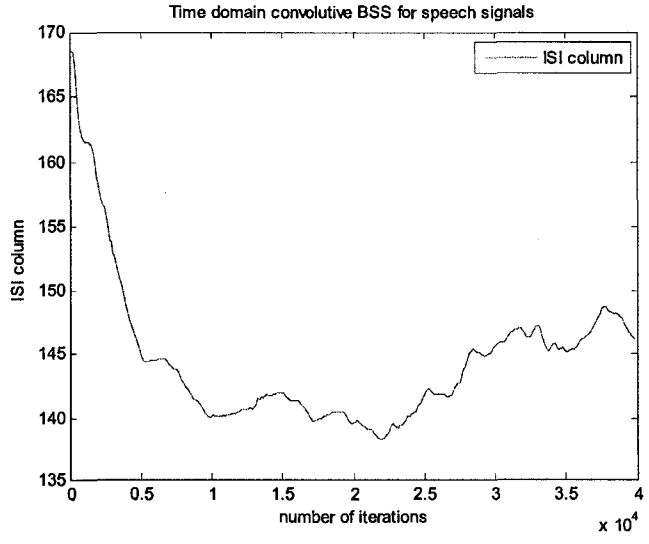


Figure 4-14: Separation performance evaluated by ISI column for convolutive Gamma mixture signals

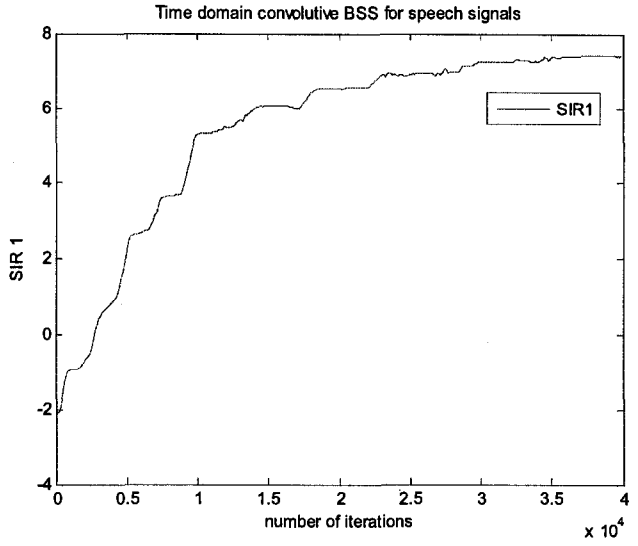


Figure 4-15: Separation performance of the first output evaluated by SIR for convolutive speech mixture signals

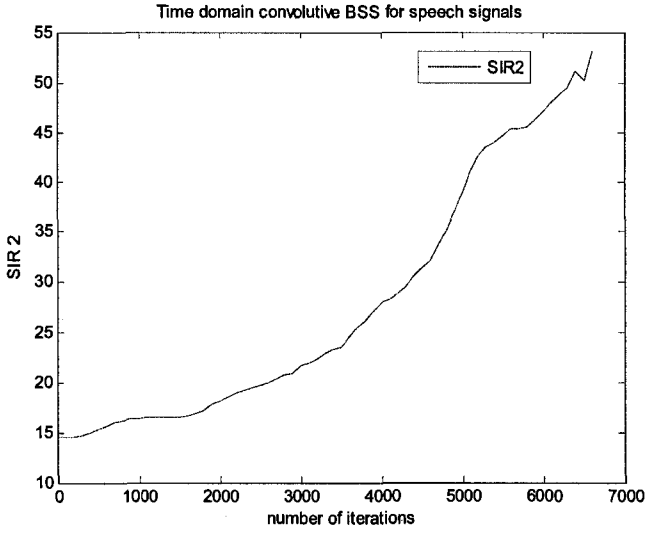
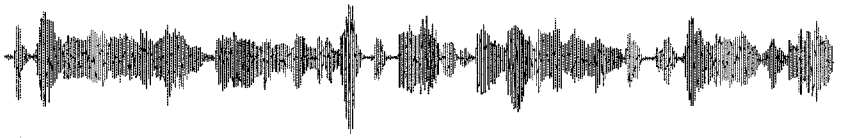
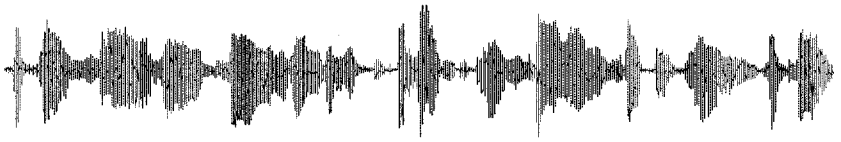


Figure 4-16: Separation performance of the second output evaluated by SIR for convolutive speech mixture signal

Since we are dealing with speech signals, we illustrate the waveforms of mixed speech signals in Fig. 4.17 and separated speech signals in Fig. 4.18. Their corresponding PESQ scores are shown in Table 4.2 which is consistent with our unofficial listening experience.

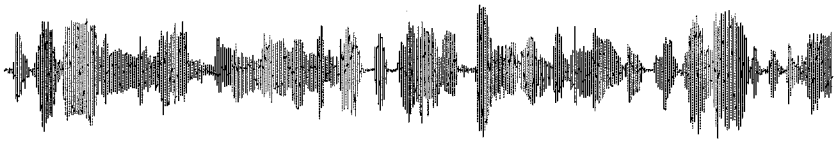


Mixture #1

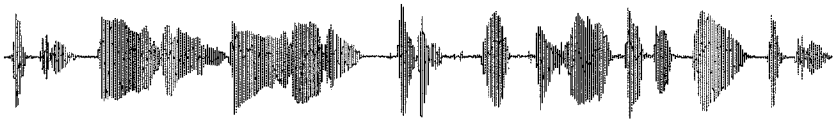


Mixture #2

Figure 4-17: Waveforms of mixed speech signals



Separated #1



Separated #2

Figure 4-18: Waveforms of separated speech signals

PESQ	Mixture		Separated	
	mix1	mix2	out1	out2
S1	1.99	1.03	2.40	0.30
S2	1.07	2.33	0.82	2.93

Table 4-2: PESQ scores of mixtures and separated signals in time domain convolutive BSS system

From the results, we can see that the algorithm converges successfully after going through all signal samples.

Under the same conditions, we repeat the experiments above using the frequency domain BSS algorithm.

Experiment #3: Frequency domain BSS for convolutive mixture of Gamma signals

The input signals are generated Gamma signals as in *Experiment #1*. We use the same mixing system as in (4.12). The parameters used in this experiment: the filter length L=64; the step size μ=0.0001.

The separation performance evaluated by ISI row and ISI column are illustrated in Fig. 4.19 and Fig. 4.20. The separation performances evaluated by SIR for two outputs are illustrated in Fig. 4.21 and Fig. 4.22.

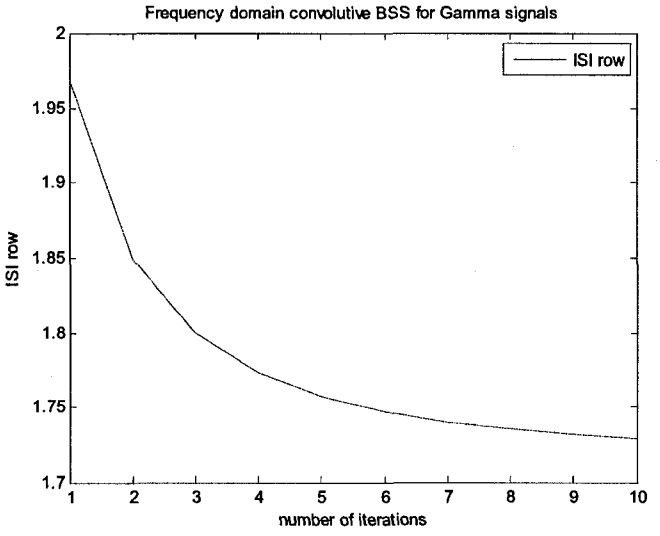


Figure 4-19: Separation performance evaluated by ISI row for convolutive Gamma mixture signals

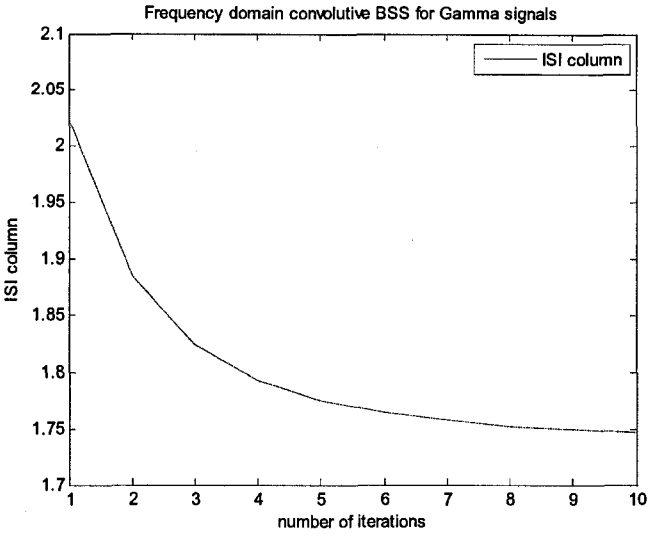


Figure 4-20: Separation performance evaluated by ISI column for convolutive Gamma mixture signals

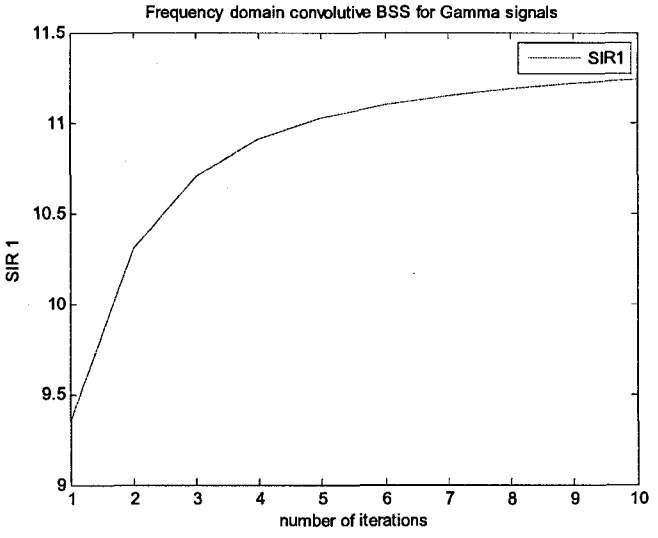


Figure 4-21: Separation performance of the first output evaluated by SIR for convolutive Gamma mixture signals

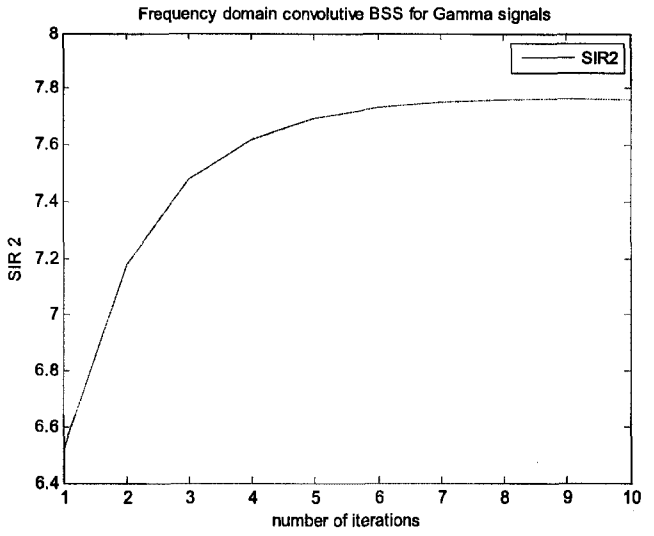


Figure 4-22: Separation performance of the second output evaluated by SIR for convolutive Gamma mixture signals

From above simulation results, we can see that frequency domain BSS algorithm works well for generated Gamma signal and the algorithm converges quickly after several iterations in terms of signal to interference ratio.

Experiment #4: Frequency domain BSS for convolutive mixture of speech signals

The input speech signals are the same as used in Experiment #2. We use the same mixing system as used in Experiment #1 in (4.12). The parameters used in this experiment: the filter length L=64; the step size μ=0.0001.

The separation performance evaluated by ISI row and ISI column are illustrated in Fig. 4.23 and Fig. 4.24. The separation performances evaluated by SIR for two outputs are illustrated in Fig. 4.25 and Fig. 4.26.

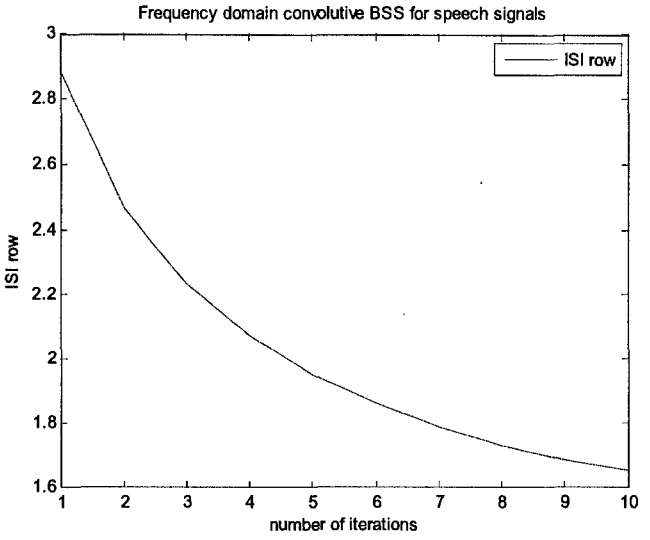


Figure 4-23: Separation performance evaluated by ISI row for convolutive speech mixture signals

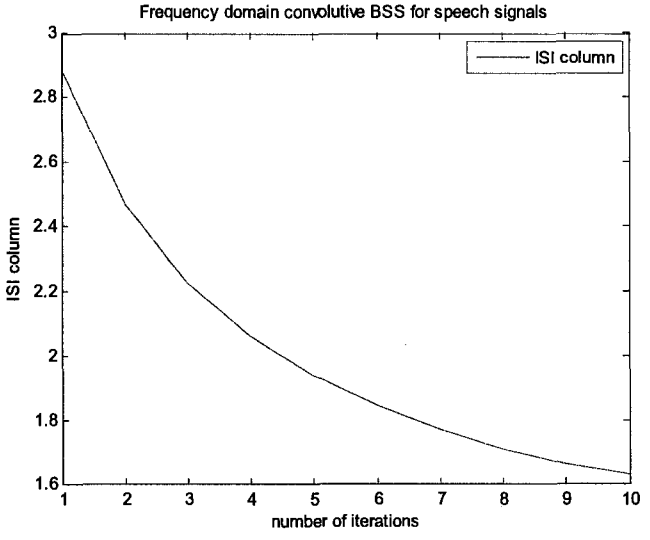


Figure 4-24: Separation performance evaluated by ISI column for convolutive speech mixture signals

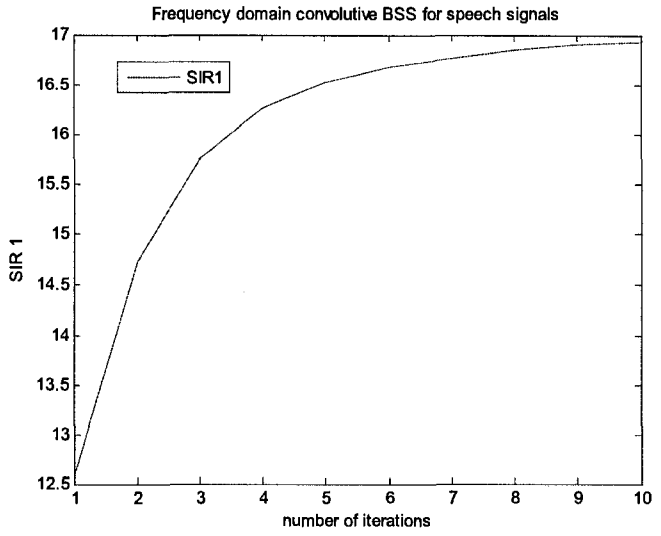


Figure 4-25: Separation performance of the first output evaluated by SIR for convolutive speech mixture signals

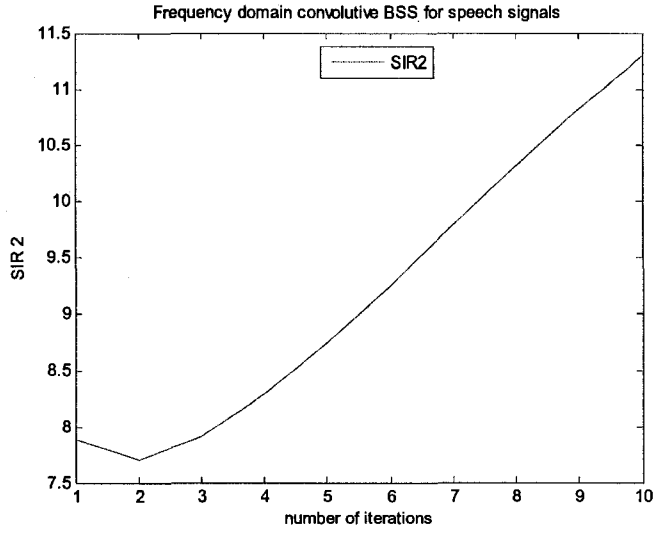
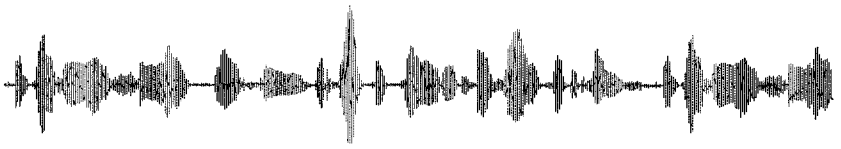
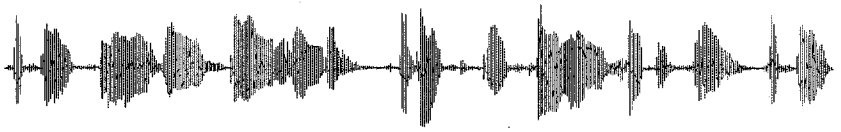


Figure 4-26: Separation performance of the second output evaluated by SIR for convolutive speech mixture signals

Their corresponding PESQ scores are shown in Table 4.3 which are consistent with our unofficial listening experience.



Separated signal #1



Separated signal #2

Figure 4-27: Separated speech signals by frequency domain BSS algorithm

PESQ	Mixture		Separated	
	mix1	mix2	out1	out2
S1	1.99	1.03	3.55	0.23
S2	1.07	2.33	0.90	3.56

Table 4-3: PESQ scores of mixtures and separated signals in frequency domain convolutive BSS system

From the results, we can see that the frequency domain algorithm converges successfully after several iterations and the quality of separated speech signals is much better than that separated by time domain BSS algorithm. Moreover, the computational complexity is also lower than its corresponding time domain approach.

4.5 Conclusion

From the simulation results shown above and our understanding of the literature, we have the following conclusions:

- Natural gradient based adaptation algorithms have much better convergence speed.
- Time domain convolutive BSS algorithm has slower convergence compared to frequency domain algorithm.

- For speech separation in reverberation environment, frequency domain convolutive BSS algorithm is in general a better choice than time domain algorithms.

Chapter 5 A Convolutive Blind Signal Separation Method for Joint Speech Signal Separation and Echo Cancellation

5.1 Introduction

Joint blind speech signal separation and cancellation arises in many applications such as teleconferencing and hand-free telephony. In these cases, what we need to do is to cancel the contributions of background signals and separate different local speakers so as to recover the desired individual speaker. Since the received signals in the above mentioned applications are also convolutive mixtures from source signals, we construct a unified BSS model for them and propose to use the convolutive blind signal separation (BSS) approach to deal with the problem of joint speech signal separation and echo cancellation. The validity of this approach is verified by our simulation results.

The convolutive BSS model used in our approach has been described in Chapter 3, thus we will not repeat it here. This chapter is organized as follows. In Section 5.2, we investigate the possibility of achieving network echo cancellation and acoustic echo cancellation using the convolutive BSS approach. We construct a unified BSS model for both network and acoustic echo cancellation. In Section 5.3, we describe a convolutive blind signal separation method based on natural gradient optimization and implicitly exploiting higher-order statistics to perform the joint speech signal separation and echo cancellation. Then in Section 5.4, we present some simulation results to verify the performance of our proposed approach. Finally we present our conclusions in Section 5.5.

5.2 Joint BSS-based Echo Cancellation Problem Formulation

In telephone communication networks, echoes are produced as a result of impedance mismatches in the network hybrids. The conventional network echo canceller is shown in Fig. 5.1.

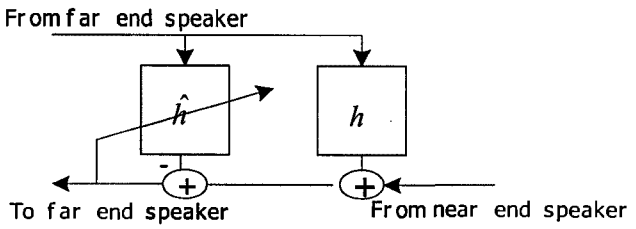


Figure 5-1: A simplified conventional network echo canceller

The basic idea in conventional network echo canceller is to build a model of the impulse response of echo path and then excite this model with the same input as the hybrid and local loop. The echo path model must have the ability to learn and adapt to the real echo path impulse response at the beginning of each call. To accomplish this task, the echo canceller uses an adaptive filter to construct the echo impulse response model. However, the performance of the adaptive filter degrades seriously when significant near-end signal is present. For this reason, practical echo cancellers have a near-end speech detector to terminate adaptation of the filter taps when significant near-end signal is present.

From the network echo canceller model described above, it is obvious that this problem can be approached from the perspective of blind signal separation. The signal from the near end speaker is the convolutive mixture of far-end speaker s_1 and near-end speaker s_2. If we can separate the mixtures (i.e. we can cancel the effect of the far-end speaker) and recover the near-end speaker signal, the echo cancellation task is done even in the presence of the near end speaker. Keeping this idea in mind, we construct the BSS model for network echo cancellation as follows noting that *we also have access to the far-end signal* s_1. The model is illustrated in Fig. 5.2.

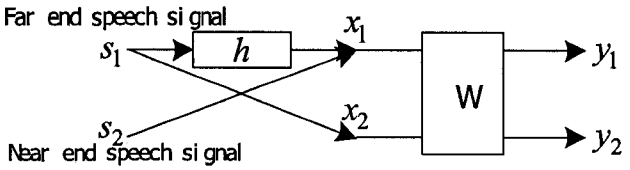


Figure 5-2: Convolution BSS model for network echo cancellation (only one sensor signal x_1 and one source signal x_2 is known here)

In Fig. 5.2, h is the impulse response of the actual echo path. x_1 and x_2 are the observed mixture signal which are obtained as

$$x_1 = s_1 * h + s_2 \tag{5.1}$$

$$x_2 = s_1 \tag{5.2}$$

From the above, we can see that the constructed model for the network echo cancellation in Fig. 5.2 is consistent with the model for convolutive blind signal separation in Fig. 3.3 in Chapter 3.

In acoustic echo cancellation, the problem becomes more complicated because there could be more than one local speaker, microphone and loudspeaker. A typical setup for a teleconferencing [Schobben 2002] is illustrated in Fig. 5.3. Generally both loudspeaker signals and reproduced far-end signals are picked up by the microphones. The task here is to cancel the contributions of echo signals from loudspeakers and separate the local speakers.

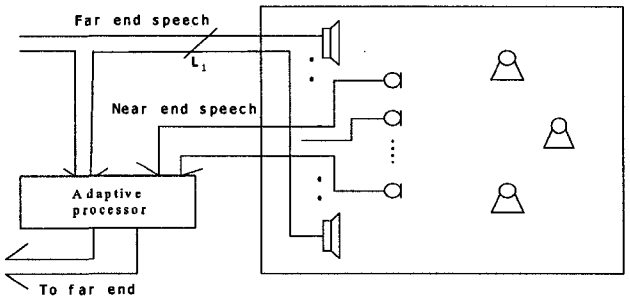


Figure 5-3: A typical setup for teleconferencing [Schobben 2002]

This is a more challenging task. But as with network echo cancellation, we can also construct a BSS model for this situation as in Fig. 5.4. Assuming there are N_1 local speakers and N_2 loudspeakers and N_1 microphones in the room. $s_1,...,s_{N_1}$ are the near end speech signals; $s_{N_1+1},...,s_{N_1+N_2}$ are the far end speech signals. $x_1,...,x_{N_1}$ are the observed mixture signals obtained from microphones and $x_{N_1+1},...,x_{N_1+N_2}$ are the observed signals from loudspeakers. The observations can be represented by the following equations.

$$x_i = \sum_{j=1}^{N_1+N_2} s_j * h_{ij}, \quad i = 1,...,N_1 \tag{5.3}$$

$$x_i = s_i, \quad i = N_1+1,...,N_1+N_2 \tag{5.4}$$

where h_{ij} is the room impulse response from j th speaker to i th microphone.

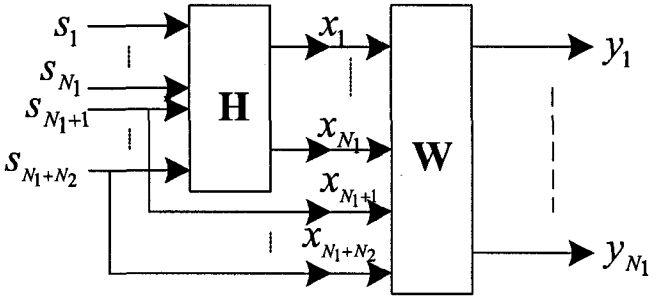


Figure 5-4: A convolutive BSS model for acoustic echo cancellation situation (there are N_1 sensor signals ($x_1,...,x_{N_1}$) and N_2 source signals ($s_{N_1+1},...,s_{N_1+N_2}$) are known here)

From Fig. 5.4, we see that the constructed model for acoustic echo cancellation is also consistent with convolutive BSS model. Thus, the multiple input and multiple output convolutive BSS method can be used to deal with the acoustic echo cancellation task. In convolutive BSS, an estimator measures the independence among all microphone signals and the far-end loudspeaker signals. This measure is used to update the separation system parameters in order to produce the separated local speakers.

From above constructed BSS model, we can see that using the convolutive BSS method to deal with the problem of joint blind speech signal separation and echo cancellation is feasible. Furthermore, there are some advantages in using the convolutive BSS method. For acoustic echo cancellation, convolutive BSS works well in the presence of active local speakers which seriously affects the performance of conventional acoustic echo canceller. For network echo cancellation, BSS also works well in the presence of the double talk. In essence, the presence of active local speakers or double talk was exploited in the convolutive BSS method to perform the cancellation and separation while in standard echo cancellation, they are problematic and require the adaptation.

5.3 Proposed Echo Cancellation Method Based on Statistical Independence

In [Schobben 1999][Schobben 2002], a frequency domain method based on decorrelation is proposed to perform joint acoustic echo cancellation and speech signal separation. The idea behind this method is to make all output signals mutually uncorrelated and uncorrelated with the far end signals so as to cancel the effect of the far end speaker signals and separate the local speakers.

However, if the signals come from independent sources, it is reasonable to assume that they are statistically independent; a stronger condition than uncorrelatedness if the signals are not jointly Gaussian processes. By imposing statistical independence between the reconstructed signals, we obtain additional constraints involving high-order cross-cumulants/spectras which cannot be obtained by second-order statistics. Thus, we can consider using algorithms based on signal independence to deal with joint acoustic echo cancellation and speech signal separation.

Moreover, only acoustic echo cancellation was considered in [Schobben 1999][Schobben 2002]. Based on our investigation in previous section, however, both the network echo cancellation and the acoustic echo cancellation models are consistent with convolutive blind source separation model. Thus, both of them can be dealt with by a unified convolutive BSS model.

From Chapter 3, we know that convolutive BSS algorithms exploit signal independence. In this chapter we use frequency domain convolutive BSS algorithm described as Eq. (3.65) to deal with our joint echo cancellation and speech separation. The algorithm is illustrated in Table 5.1.

Coefficient vector

$\underline{\mathbf{W}}_k = \begin{bmatrix} \underline{w}_{11}(k) & \underline{w}_{12}(k) \\ \underline{w}_{21}(k) & \underline{w}_{22}(k) \end{bmatrix}$; where $\underline{w}_{ij}(k)$ is the vector of coefficients in the frequency domain.

Input vector

$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2]$;

Initialization

$\underline{\mathbf{W}}_0$ is set to be unit matrix;

Main iteration

$\mathbf{X} = STFFT(\mathbf{x})$

$\mathbf{Y} = \underline{\mathbf{W}}\mathbf{X}$

$\underline{\mathbf{W}}_{k+1} = \underline{\mathbf{W}}_k + \mu(\underline{\mathbf{I}} - g(\underline{\mathbf{y}})\underline{\mathbf{y}}^T)\underline{\mathbf{W}}_k$

$g(\underline{\mathbf{y}}) = \tanh(\underline{\mathbf{y}})$

Table 5-1: Frequency domain convolutive BSS algorithm for 2 by 2 case

5.4 Experimental Results

The PESQ, a speech quality evaluation tool, described in Section 4.2.3 is used to measure the quality of the recovered speech signal compared to the original speech signal to evaluate the performance of different separation approaches. We compare the performance of our approach with the approach used in [Schobben 2002] and [Schobben 1999] in three scenarios: separation only, network echo cancellation and acoustic echo cancellation.

5.4.1 Performance Comparison for Separation only Case

In the first experiment, we verify the performance for the separation only case. A two- input and two-output mixing system, simulated by a 2 by 2 artificial impulse response matrix, is

implemented in this experiment. Two speech signals are used as source signals, one female and one male. The impulse response matrix is

$$h_{11} = [1.0 \quad 1.0 \quad -0.75] \qquad h_{12} = [0.5 \quad -0.3 \quad 0.2]$$
$$h_{21} = [-0.2 \quad 0.4 \quad 0.7] \qquad h_{22} = [0.2 \quad 1.0 \quad 0]$$

The mixture speeches are obtained as the convolution of source signals with the impulse response matrix. By adaptively adjusting the parameters in the separating system, we obtain separated speech signals as the estimated source signals.

The PESQ results for separated speech signals estimated by the suggested convolutive BSS approach and decorrelation approach in [Schobben 2002] are given in the Table 5.2. We can see that the proposed convolutive BSS approach achieved considerable improvement over the decorrelation approach.

PESQ Score	Separated speech #1 by decorrelation	Separated speech #1 by convolutive BSS
Original speech #1	1.85	3.79
PESQ Score	Separated speech #2 by decorrelation	Separated speech #2 by convolutive BSS
Original speech #2	1.4	3.92

Table 5-2: PESQ results for separating speech mixtures

5.4.2 Performance for Network Echo Cancellation Case

In the second experiment, we verify the performance for network echo cancellation case. We use an artificial impulse response for the echo path and two input speech signals, the far-end speech and the near-end speech, to simulate the network echo situation. The impulse response for the echo path in this experiment is

$h = [-0.2\ 0\ 0\ 0\ 0\ 0.1\ 0\ 0.05\ 0\ 0.2\ 0\ 0\ 0.1\ -0.2\ 0\ 0.1\ 0\ 0\ 0.4\ 0\ 0.7]$

which has 21 taps.

We estimate the near-end speech signal by the convolutive BSS method and the decorrelation method and results are given in the Table 5.3. It is seen that the performance of the proposed method is superior to the performance of the method based on the decorrelation. We also note that the PESQ score of the separated speech #2 by convolutive BSS is much higher than other scores. This is clear from Fig. 5.2. The mixture speech signal #2 is not contaminated since we directly get it from the far end speech port. However, we do not need to recover this signal. We only need to cancel its effect on the near end speech signal.

PESQ Score	Separated speech #1 by decorrelation	Separated speech #1 by convolutive BSS
Original speech #1	2.74	3.66
PESQ Score	Separated speech #2 by decorrelation	Separated speech #2 by convolutive BSS
Original speech #2	2.79	4.25

Table 5-3: PESQ results for network echo cancellation

5.4.3 Performance for Acoustic Echo Cancellation Case

In the third experiment, we verify the performance for acoustic echo cancellation. We use an artificial impulse response matrix simulating the room impulse responses and two near-end speaker signals and one far-end speaker signal to simulate the acoustic echo situation. The simulated room impulse response used in this experiment is given by (all have 10 taps):

h_{11}=[0 0.4 0.2 0 0 0 -0.1 0 0.02 0.03];
h_{21}=[0 0 0.1 0 -0.2 0 0 0.03 0.02 -0.2];
h_{12}=[0 0.5 0 0 0 -0.3 0 0 -0.1 0.1];
h_{22}=[0 0 0.2 0.4 0 0 0 -0.2 -0.3 -0.5];
h_{13}=[0 0.3 0.5 0 0 -0.5 -0.4 -0.05 0.1 0.3];
h_{23}=[0.2 0 0 0 0 -0.04 -0.08 -0.13 -0.07 -0.1];

We estimate the separated near-end speech signals by the convolutive BSS method and the decorrelation method. The PESQ scores are given in the Table 5.4. From the simulation results above, we can see that the performance of the proposed convolutive blind signal separation method also exceeds the decorrelation-based method in [Schobben 2002].

PESQ Score	Separated speech #1 by decorrelation	Separated speech #1 by convolutive BSS
Original speech #1	2.06	3.51
PESQ Score	Separated speech #2 by decorrelation	Separated speech #2 by convolutive BSS
Original speech #2	1.77	2.5

Table 5-4: PESQ compare result for acoustic echo cancellation

5.5 Conclusion

In this chapter, we propose to use a convolutive BSS algorithm which is implicitly based on higher-order statistics to perform blind speech signal separation and unrelated signal cancellation simultaneously. Simulation results confirm the validity of our approach and the fact that the proposed convolutive BSS approach to joint speech signal separation and echo cancellation exhibits a performance that is superior to the approach based on second-order statistics. The proposed approach continues to work well in double talk situations which adversely affect the performance of the conventional echo cancellation method. This makes the proposed approach suitable for network echo cancellation, teleconferencing and other applications.

Chapter 6 Perceptual Convolutive Blind Speech Signal Separation

6.1 Introduction

In speech separation, the target signal is speech and the end user is a human being. Thus, it is helpful to take into account the psychoacoustic characteristics of the human auditory system into the separation methods. In [Zwicker 1991], it is shown that the average human does not hear all frequencies the same way. The two main properties of the human auditory system that make up the psychoacoustic model are the absolute threshold of hearing and auditory masking. The absolute threshold of hearing means that in a quiet environment, our ears cannot hear a frequency component whose magnitude is below the threshold. Different frequencies have different thresholds. Human auditory masking has time domain masking properties and frequency domain masking properties. Frequency-domain masking means that our ear may not hear a frequency component of lower amplitude next to another of stronger amplitude. In the time domain, we may not hear an audio component which is very close, in time, to a strong audio component. All these psychoacoustic characteristics of the human auditory system have been widely adopted in various audio and speech coders to reduce the effects of quantization noise. Recently, masking properties have also been applied to speech enhancement. In [Virag 1999], it was shown that the utilization of properties of the human auditory system could attenuate audible noise without distortion. In [Huang 2002], a time domain forward masking nonlinear model for human auditory system was proposed and an algorithm for integrating the frequency domain simultaneous masking effect and the time domain forward masking effect was proposed. In [Ma 2004], the method proposed in [Johnston 1988] and [Virag 1999] was applied as a post-filter of a Kalman filter to enhance the speech quality leading to improved performance.

In this chapter, we investigate the feasibility of incorporating these psychoacoustic properties into our blind source separation algorithm to improve its performance for speech separation.

This chapter includes two parts. In the first part, we propose a two-stage post-filter based perceptual convolutive BSS system in which a frequency domain convolutive blind source separation system is concatenated with a post-filter system based on the masking properties of human auditory system. In the second part, a new perceptual convolutive blind separation algorithm is proposed based on the blind filtered-E LMS algorithm [Kuo 1994]. The error is weighted in the frequency domain by the function obtained from the absolute threshold to emphasize frequencies to which human ear is sensitive and de-emphasize frequencies inaudible to human ear.

The rest of this chapter is organized as follows. In Section 6.2, the masking properties of human auditory system are briefly introduced. In Section 6.3, the post-filter based convolutive blind speech separation method is proposed and its simulation results are presented. In Section 6.4, a new perceptual convolutive blind speech separation algorithm is proposed and its detailed derivation is provided, its corresponding simulation results are also provided in this section. Finally, we give our conclusions in Section 6.5.

6.2 Psychoacoustic Properties of the Human Auditory System

There are two main psychoacoustic properties for the human auditory system: absolute threshold of hearing and masking properties. Each of them provides a way to decide which portion of a signal is inaudible to the average human.

6.2.1 Absolute Threshold of Hearing

To determine the effect of frequency on hearing ability, a sinusoidal tone is played at a very low power. The power is slowly raised until the subject could hear the tone. This level is the threshold at which the tone could be heard and is called the absolute threshold for this frequency. Experiments show that the absolute threshold is different for different frequencies and it can be approximated by the following nonlinear function [Zwicker 1990]:

$$ATH(f) = 3.64\left(\frac{f}{1000}\right)^{-0.8} - 6.5e^{-0.6\left(\frac{f}{1000}-3.3\right)^2} + 10^{-3}\left(\frac{f}{1000}\right)^4 (dB \quad SPL) \qquad (6.1)$$

where f is the frequency in Hertz and SPL is sound pressure level.

6.2.2 Masking Properties of the Human Auditory System

Masking effects of the human auditory system is a phenomenon where one signal, the masker, can make other weaker signals, the maskee, inaudible if they are close enough to the masker in frequency or time. There are two kinds of masking effects for human ears which include time domain masking properties and frequency domain masking properties. The masking effect in frequency domain is simultaneous masking, where a lower-level signal component (maskee) is made inaudible by a simultaneously occurring stronger signal (masker). The masking threshold depends on the sound pressure level (SPL) of the masker and the frequency difference between the masker and the maskee. The masking effects in time domain include backward masking and forward masking. Backward masking masks signal components that occur before the masker. It can help to mask pre-echoes caused by the spreading of a large quantization error and its effective duration is short. Forward masking masks signal components that occur after the masker, and it has an effective duration ten times that of backward masking. Therefore, the forward masking effect can be more significant than backward masking since more signal components are masked. We only consider forward masking effect here.

6.2.3 Calculation of Frequency Domain Masking Threshold

The masking phenomenon in frequency domain is modeled by a noise masking threshold, below which all components are inaudible.

The calculation of the noise masking threshold for signal $s(n)$ is composed of the following steps [Johnston 1988]:

1) Critical band analysis of the signal: compute an N-point FFT of the signal $s(n)$; partition the signal from its frequency domain into critical bands, sum the energies in each critical band to get critical band spectrum.
2) Applying the spreading function to the critical band spectrum: convolve the critical band spectrum with a spreading function to take into account masking between different critical bands and obtain the spread critical band spectrum. The spreading function depends on the signal being noise-like or tone-like.
3) Computing the relative threshold offset.
4) Calculating the masking threshold. The masking threshold is obtained by subtracting the relative threshold offset from the spread critical band spectrum.

6.2.4 Calculation of Time Domain Forward Masking Threshold

The calculation of the forward masking threshold for signal $s(n)$ is composed of the following steps [Huang 2002]:

1) Critical band analysis of the signal: compute an N-point FFT of the signal $s(n)$; partition the signal from its frequency domain into critical band, sum the energies in each critical band to get critical band spectrum.
2) Applying the spreading function to the critical band spectrum: convolve critical band spectrum with a spreading function to take into account masking between different critical bands and obtain the spread critical band spectrum.
3) Transforming the signal physical energy given by the spread critical band spectrum to the internal loudness values.
4) Generating the output specific loudness by passing the internal loudness values through a low pass filter.
5) Calculating the total loudness as the sum of the output specific loudness in all critical bands and using it as the estimate of the forward masking level.

6.3 Post-filter Perceptual Convolutive Blind Source Separation Approach for Speech Signal

In this section, we first give the system structure for our proposed post-filter based convolutive BSS method. Then we give a description of the approach in our post filtering stage which integrates the time domain forward masking property and frequency domain simultaneous masking property.

6.3.1 Combining Convolutive BSS with the Masking Properties of the Human Auditory System

A new system for convolutive multichannel speech signal separation, which consists of a frequency domain convolutive BSS system and a post-filter to deal with the output from the BSS system is proposed and illustrated in Fig. 6.1.

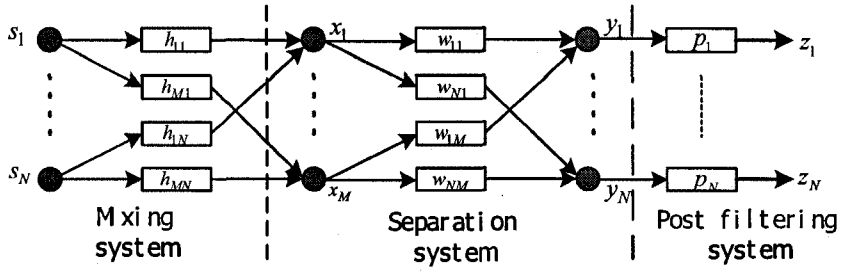


Figure 6-1: Post-filter based perceptual convolutive blind source separation system

In the convolutive BSS system, the mixtures are separated by a frequency domain convolutive BSS algorithm as described in Eq. (3.65). However, the separation cannot be perfect. The purpose of the post-filters is to reduce the residue reverberation and interference effects after an imperfect source separation by convolutive BSS system.

By adding the post filters, the demands on the convolutive BSS may be reduced, and consequently the order of the FIR filters in the convolutive BSS system may be reduced. For the convolutive BSS system with lower order filters, it can converge faster and its

computational complexity is reduced. Adding the post-filters, the performance of the whole system is expected to improve while the total computational complexity of the system is reduced.

6.3.2 Post-filter based on Masking Properties of Human Auditory Systems

In the post-filter, the output speech signal from the convolutive BSS system is processed on frame-by-frame basis. The approach in [Huang 2002] is used here to take into account both time domain forward masking effects and frequency domain simultaneous masking properties in the proposed system.

The time domain forward masking effects are modeled as a psychoacoustic specific loudness versus critical-band rate and time. The total loudness Q, defined as the sum of the output specific loudness in all critical bands, is used as an estimate of the time domain forward masking level [Huang 2002]. The total masking level is determined by integrating the frequency-domain simultaneous masking effect and the time-domain forward masking effect [Huang 2002] and the level depends on the current frame and the previous frames by the following equation.

$$M_t^k(i) = \max\left\{M_s^k(i), M_t^{k-1}(i)\cdot \exp^{-\Delta t/(\tau(i)\cdot Q)}\right\} \tag{6.2}$$

where $M_t^k(i)$ and $M_t^{k-1}(i)$ are the total masking levels of the ith critical band in the kth frame and the $(k-1)$th frame respectively. $M_s^k(i)$ is the simultaneous frequency domain masking level of the ith critical band in the kth frame, Δt is the time difference between two frames, $\tau(i)$ is the maximum decay time constant in each critical band and Q is the total loudness level. Both $\tau(i)$ and Q are using the same value as in [Huang 2002].

The post-filter is used here to shape the signal spectrum based on the computed total masking level. When the frequency component of the interfering speech signal at a certain frequency point is lower than the threshold, the interfering component at that frequency will be masked and we keep it unchanged at that frequency. When the energy of the interfering speech at a

certain frequency is greater than the masking threshold, the noise may not be masked. Then a factor computed from the threshold is used to decrease the amplitude and make the interfering component smaller.

6.3.3 Simulation Results

The PESQ score described in section 4.2.3 is used here to measure the similarity between the recovered speech signal and the original speech signal so that we can evaluate the performance of different separation approaches.

Two speech signals of 20 second duration each are used in our simulation. The sampling frequency is 8000Hz. The mixing system used in our simulation is

$$H=\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix};$$

$$h_{11}=[0\ \ 0\ \ 0\ \ 0.75\ \ -0.5\ \ -0.15\ \ 0.44\ \ -0.2\ \ -0.16\ \ 0.24\ \ -0.06\ \ -0.12\ \ 0.12\ \ 0.0\ \ 0.05\ \ 0.01];$$

$$h_{12}=[0\ \ 0\ \ 0\ \ 0\ \ 0.5\ \ 0\ \ 0\ \ -0.3\ \ 0\ \ 0\ \ 0\ \ 0.2\ \ 0\ \ 0\ \ 0\ \ 0];$$

$$h_{21}=[0\ \ 0\ \ 0\ \ 0\ \ -0.2\ \ 0\ \ 0\ \ 0.4\ \ 0\ \ 0\ \ 0\ \ 0.7\ \ 0\ \ 0\ \ 0\ \ 0];$$

$$h_{22}=[0\ \ 0\ \ 0\ \ 0.76\ \ 0.1\ \ -0.6\ \ 0.3\ \ 0.1\ \ -0.3\ \ 0.1\ \ 0.14\ \ -0.18\ \ 0.02\ \ 0.10\ \ -0.08\ \ -0.01]$$

Assume the two source speech signals are s_1 and s_2, we get two mixture signals x_1 and x_2 by the mixing system in $\mathbf{H}$. Then the mixtures are used as input to the convolutive BSS system. The outputs are y_1 and y_2. The signals y_1 and y_2 are then used as input to the post-filters P_1 and P_2 giving the outputs z_1 and z_2. The frame size used in the post-filters is 80.

By comparing the mixture signals x_1 and x_2, intermediate outputs y_1 and y_2 and outputs z_1 and z_2 with the source signals s_1 and s_2 and evaluating the speech quality using the PESQ scores, we get the following results in Table 6.1.

PESQ score	Mixture x1	y1 from BSS	z1 from post filter
Compared with source s1	1.46	3.09	3.12
PESQ score	Mixture x2	y2 from BSS	z2 from post filter
Compared with source s2	1.10	2.83	2.85

Table 6-1: Speech quality evaluation for output speech from different stage

From Table 6.1, we can see that the speech quality is only slightly improved by post-filtering system. The reason that the post filters can not work so well may be the method used in our post-filter system is for speech enhancement with additive noise, while in our situation, there are convolutive speech mixtures, the interference is also speech and they are mixed convolutively.

6.4 Proposed New Perceptual Convolutive Blind Speech Separation Algorithm

A new perceptual convolutive blind source separation algorithm is proposed in this part based on filtered-E LMS algorithm and the masking properties of human auditory system. This algorithm emphasizes separation at frequencies to which the human ear is sensitive and de-emphasizes separation at frequencies that are inaudible to the human ear thus incorporating the properties of human auditory system and making the algorithm more suitable for speech separation. Simulation results for blind speech signal separation show that the proposed algorithm can improve the separated speech quality.

First, we give our derivation of this new algorithm which is based on convolutive BSS model and the filtered-E algorithm. Then we give our simulation results and present our conclusion.

6.4.1 Filtered-E LMS Algorithm

The Filtered-E LMS algorithm [Kuo 1994] has its origins in the Filtered-X LMS algorithm and its structure is illustrated in Fig.6.3.

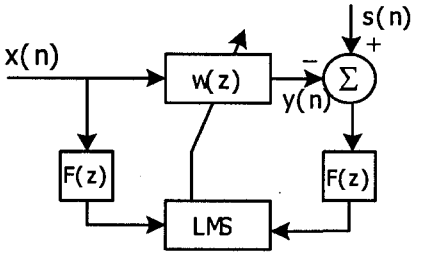


Figure 6-2: Structure for filtered-E LMS

The cost function is $J = E\left\{F\left(e(n)\right)^2\right\}$, where $e(n) = y(n) - s(n)$.

By using stochastic gradient adaptation, the weight update rule is

$$w(n+1) = w(n) - \frac{\mu}{2}\frac{\partial J}{\partial w} \tag{6.3}$$

Since $\frac{\partial J}{\partial w} = 2F\left(e(n)\right)F\left(x(n)\right)$, the adaptation rule becomes

$$w(n+1) = w(n) - \mu F\left(e(n)\right)F\left(x(n)\right) \tag{6.4}$$

6.4.2 Single Channel Blind LMS Algorithm (source signal s is not available)

The standard single channel conventional LMS algorithm defines the cost function as

$$J = E\left\{\left|e(n)\right|^2\right\} \tag{6.5}$$

where error $e(n) = y(n) - s(n)$.

The adaptation rule is given by

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) - \mu e(n)\mathbf{x}(n) \\ &= \mathbf{w}(n) - \mu\big(y(n) - s(n)\big)\mathbf{x}(n)\end{aligned} \tag{6.6}$$

where μ is the step size.

We now extend this LMS algorithm to the blind case using Bussgang nonlinearity as is done for blind deconvolution in the unsupervised adaptive filter algorithm [Lambert 1996]. This algorithm, referred to as the blind LMS algorithm, is described as

$$w(n+1) = w(n) - \mu e(n)x(n) \tag{6.7}$$

where the error signal here is defined as

$$e(n) = y(n) - g\big(y(n)\big) \tag{6.8}$$

and the function $g(.)$ is the zero-memory Bussgang non-linearity given by:

$$g(y) = -E|y|^2 \frac{\partial \log p_y(y)}{p_y(y)} \tag{6.9}$$

where $p_y(y)$ is the pdf of y. The output of this function is treated as the desired response in the conventional LMS algorithm.

6.4.3 Proposed Single Channel Blind Filtered-E LMS Algorithm

Now, we extend the blind LMS algorithm to blind Filtered-E LMS algorithm using a procedure similar to conventional Filtered-E LMS algorithm.

The error signal in (6.8) is weighted by a filter function $F(\cdot)$ giving the cost function:

$$J = E\left\{F\big(e(n)\big)^2\right\} \tag{6.10}$$

The weight vector update can be expressed as

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2}\frac{\partial J}{\partial \mathbf{w}} \tag{6.11}$$

From (6.10), we can get the adaptation rule as

$$\mathbf{w}(n+1)=\mathbf{w}(n)-\mu F\big(y(n)-g\big(y(n)\big)\big)F\big(\mathbf{x}(n)\big) \tag{6.12}$$

6.4.4 Multichannel Blind Filtered-E LMS Algorithm

By using the FIR matrix tools in [Lambert 1996], we can extend the above single channel blind Filtered-E LMS algorithm to multichannel to deal with the convolutive BSS system described in Chapter 3. The extended algorithm is

$$\mathbf{W}(n+1)=\mathbf{W}(n)-\mu F\big(\mathbf{y}(n)-g\big(\mathbf{y}(n)\big)\big)F\big(\mathbf{x}(n)\big)^{H} \tag{6.13}$$

where $\mathbf{W}$ is a matrix whose every component is an FIR filter and $(\ )^{H}$ means conjugate transpose.

To reduce the computational complexity of the algorithm, we move to the frequency domain where the algorithm is described as follows.

$\mathbf{x}\xrightarrow{FFT}\mathbf{X}$; $\mathbf{y}\xrightarrow{FFT}\mathbf{Y}$

$\mathbf{Y}=\underline{\mathbf{W}}\mathbf{X}$, and $\underline{\mathbf{W}}$ is a matrix where every component is a polynomial.

Now error $FFT\_e=\mathbf{Y}-fft\big(g(y)\big)$ and the adaptation rule is

$$\underline{\mathbf{W}}_{k+1}=\underline{\mathbf{W}}_{k}-\mu F\big(\mathbf{Y}-fft\big(g(y)\big)\big)F(\mathbf{X})^{H} \tag{6.14}$$

6.4.5 Perceptual Frequency Domain Convolutive Blind Speech Signal Separation Algorithm

For our speech signal separation, since the target signal is speech and the end user is human being, it is reasonable to take into account the properties of human auditory system. In standard adaptive algorithms, all signal spectrum components are treated equally. However, the psychoacoustic model shows that the average human does not hear all frequencies equally. To take advantage of the masking properties of the human hearing system, in our proposed algorithm, we form the filter function in (6.14) by exploiting the properties of

human auditory system. The advantage of this approach is that we can emphasize the frequencies to which the human ear is sensitive and de-emphasize the frequencies to which the human ear is not sensitive. By doing so, we expect to improve the separation performance for emphasized frequencies. The degraded performance for the de-emphasized frequencies does not affect the overall performance since the human ear cannot discern them. Here we only use the absolute threshold to form the filter function.

6.4.6 Simulation Results

Two speech signals are used in our simulation, as used in 6.3.3, each has a 20 second duration. The sampling frequency is 8000Hz. The error weighting filter function used in our simulation is illustrated in Fig. 6.3. It is based on the absolute threshold value in Eq. (6.1). The weights in different frequency bins are obtained be normalizing the absolute threshold value to 0~1 to emphasis frequencies sensitive to human ears.

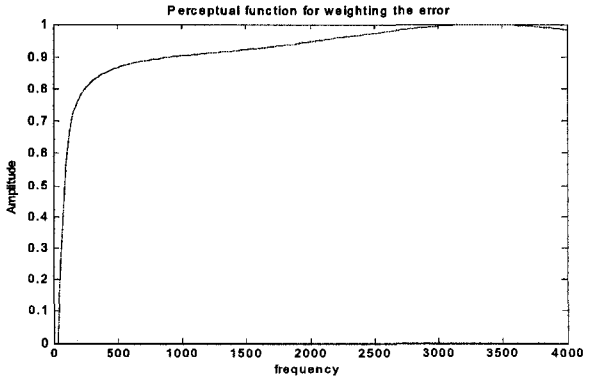


Figure 6-3: Filter function used in our simulation

The PESQ scores for the separated speech signal with the perceptual blind LMS and without perceptual blind LMS are illustrated in Table 6.2.

PESQ score	Separated signal without perceptual weighting	Separated signal with perceptual weighting
Compare with source s1	2.54	2.82

PESQ score	Separated signal without perceptual weighting	Separated signal with perceptual weighting
Compare with source s2	2.38	2.79

Table 6-2: Speech quality evaluation for separated speech

The above comparison is under the same mixing condition for separation with and without perceptual weighting. From the simulation results, we can see that the perceptual blind LMS convolutive source separation algorithm has resulted in higher quality for the separated signals.

6.5 Conclusions

In this chapter, we propose two perceptual convolutive blind speech signal separation approaches. In the first approach, a two-stage post-filter based perceptual convolutive BSS system, a frequency domain convolutive blind source separation system concatenated with post filter system based on masking propertied of human auditory system, is proposed. Both time domain masking properties and frequency domain masking properties are taken into account in the post filters to shape the speech spectrum in order to attenuate the residual interference and reverberation effects remaining after imperfect speech separation. Simulation results show that the proposed approach slightly improves the performance of convolutive blind source separation system at no additional computational cost. In the second approach, we propose a new perceptual convolutive blind source separation algorithm to deal with speech signal separation. By exploiting the properties of the human auditory system, the proposed algorithm has been shown to improve the quality of the separated signal at the cost of very little additional complexity.

Chapter 7 Blind Speech Signal Separation Combining Independent Component Analysis and Beamforming

7.1 Introduction

In many applications, there is a definite need to recover signals that have been mixed together. Teleconferencing is one such application where we need to separate different speakers whose speech is picked up by any given microphone. Such a problem has been tackled using Blind Source Separation algorithms (utilizing time and frequency domain information) [Haykin 2000][Cichocki 2000] and beamforming (utilizing spatial information) [Veen 1988] from different point of views. In this chapter, we investigate how to combine these two approaches to benefit from both for better overall system performance.

Blind source separation (BSS) [Haykin 2000][Cichocki 2000] is a technique for estimating the original source signals based on information about observed mixed signals at the sensors, i.e. the estimation is performed without exploiting information about either the source signals or the mixing system. Independent component analysis (ICA) [Hyvarinen 2001] serves as a major statistical tool for solving the separation problem. Recently, convolutive blind source separation has been widely used in audio source separation and many successful applications have been reported [Lambert 1997][Lee 1997]. In convolutive blind source separation, separation is performed using the assumption that the source signals are independent with no information about the geometry of the auditory scene (such as direction of arrival of source signals, microphone array configuration etc.). Theoretically, the sensors in blind source separation framework can be randomly arranged in a room, even in the same location provided they satisfy the independence condition. Only time/frequency information of sensor signals is utilized in separation algorithms. However, some aspects limit further applications of BSS in real-world acoustic environments. These include low convergence rate and high computational requirements in time domain methods, frequency permutation and arbitrary

amplitude scaling in frequency domain methods and performance degradation in heavy reverberant environments.

On the other hand, a relatively well-established research topic –beamforming for acoustic signals – approaches this problem from a spatial point of view. In beamforming [Veen 1988], a structured array of sensors is used to steer the overall gain pattern of the array sensors to form a spatial filter which can extract the signal from a specific direction and reduce signals from other directions. This enhances the receiver's performance with regards to source identifiability, direction tracking and enhanced reception. Thus, compared with blind source separation, the advantage of beamfoming is that the spatial information about the mixing system and/or source signals is utilized. However, blind source separation exploits a strong statistical condition -- independence -- between source signals, which may also be suitable for beamforming since source signals coming from different locations in a beamforming scenario are likely to be independent as well.

Since convolutive blind source separation and beamforming have similar goals while exploiting different information to perform separation, it is worthwhile to explore the possibilities of combining their advantages. Recently, the relationship between convolutive blind source separation and beamforming has been investigated in [Araki 2003] and some interesting results have been obtained. Based on these results, some combinations of convolutive blind source separation and beamforming have been proposed to solve problems in blind source separation and obtained improved separation results. In this chapter, we review current combination approaches of convolutive blind source separation and beamforming, and propose our new combination methods with reduced complexity for blind speech separation in real acoustic environments.

The rest of this chapter is organized as follows. In Section 7.2, we review basic knowledge of beamforming. We discuss the similarities and differences of convolutive blind source separation and beamforming in detail in Section 7.3. We review existing combination approaches of convolutive blind source separation and beamforming in Section 7.4. In Section 7.5, we propose our first combination approach and present the simulation results for

this approach. In this approach, some knowledge about the room impulse responses is needed, thus, this method is not completely blind. In Section 7.6, we propose a new completely blind speech signal separation by combining independent component analysis and beamforming and present the corresponding simulation results. Finally we give our conclusions in Section 7.7.

7.2 Introduction to Beamforming

A beamformer [Veen 1988] is a spatial filter which uses an array of sensors to collect spatial samples of propagation signals and to estimate the signal from a desired direction in the presence of noise and interference signals. As we already know, usually a temporal filter is used to separate signals with different frequency components by collecting temporal samples of signals. Similarly, a spatial filter is used to separate signals occupying the same temporal frequency band by collecting spatial samples of signals. Typically, a beamformer linearly combines the spatially sampled time series from each sensor to obtain a scalar output time series, in the same manner as an FIR filter linearly combines temporally sampled data. In Fig. 7.1 and Fig. 7.2, two kinds of beamformers are illustrated. The first one is the basic one and it only exploits the spatial sample information of signals and is suitable for narrowband signal processing. The second one exploits both spatial and temporal sample information and can be used to process broadband signals.

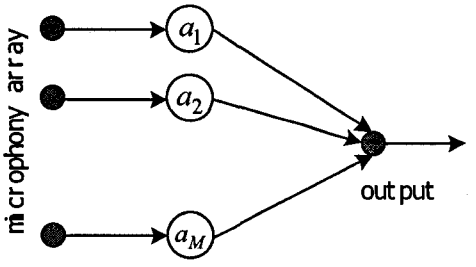


Figure 7-1: Beamformer with its sensor outputs multiplied by scalar weights

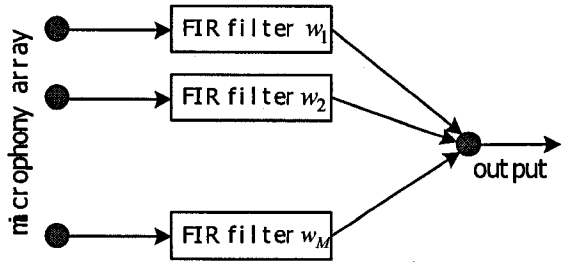


Figure 7-2: Beamformer with its senor outputs convolved by FIR filters

The main contribution of beamforming is its spatial selectivity. In beamforming, the filter coefficients are optimized to produce a spatial pattern with a dominant response in the direction of interest while the response for the positions of interfering signals is minimized. By its spatial pattern, it can receive signals from a specified direction. Thus, it can greatly reduce part of the effects of reverberation and interference. However, in multipath or reverberant environments, the selected direction may include signals that originate from different sources that end up arriving in the same directions.

7.3 Comparison of Convolutive Blind Source Separation and Beamforming

In the following, we analyze the similarities and differences between convolutive BSS and beamforming.

7.3.1 Similarities

The Goal

The basic goal for convolutive blind source separation and beamforming is similar; they both attempt to extract selected sources while reducing undesired interferences. In convolutive BSS, multiple desired signals are extracted simultaneously to achieve mutually independent outputs. In beamforming, generally only one desired signal from a specified direction is of interest and filters are adjusted to extract it.

The Structure

Both convolutive blind source separation and beamforming deal with signals received by sensor arrays. Both of them use a multichannel filter structure to simulate the unmixing process and to estimate desired signal or signals by adjusting the parameters in the filter array.

Unmixing matrix and Beamformers

In frequency domain blind source separation, at a given frequency bin, the unmixing matrix can be interpreted as a null steering beamformer that uses a blind algorithm to place nulls on the interfering sources. In multiple input and multiple output convolutive BSS, every output attempts to recover only one of source signals. The unmixing system for every output can be viewed as an adaptive beamformer which forms specific beam pattern to extract the signal from the direction of selected source signal. This can be illustrated in Fig. 7.3 by a 2 by 2 convolutive separation system.

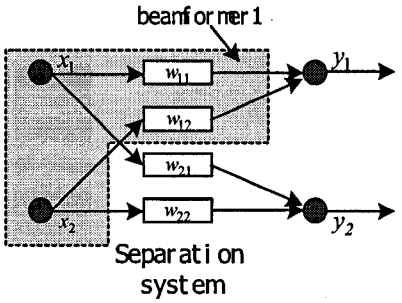


Figure 7-3: Convolutive BSS system can be viewed as multiple beamformers

7.3.2 Differences

Information Utilized

In convolutive blind source separation, time and frequency information of received signals is used to perform separation to the exclusion of the spatial setup. In beamforming, spatial

information such as location of source signals and configuration of sensor arrays is used to perform separation.

Approach

In convolutive blind source separation, the objective is to make the output signals as independent as possible. It emphasizes the frequency selectivity of the filter array in the unmixing system. In beamforming, the criterion is to form a good directivity pattern to extract the signal from a specific direction and reduce signals from other directions. It emphasizes the spatial selectivity of the filter array in unmixing system.

Limitations

The main limitation of beamforming is its cross-talk. Since beamforming produces a spatial pattern to extract a given signal, it cannot do anything about the cross talk signals in a multipath and reverberant environment. The main limitation of blind source separation is its ambiguities because of its independence criterion. The signals in BSS can only be recovered up to arbitrary scaling and permutation. The ambiguities cause a big problem for frequency domain convolutive blind source separation since arbitrary scaling and permutation exist at every frequency bin.

7.4 Review of Combined of BSS and Beamforming for Signal Separation

From Section 7.3, we can see that convolutive blind source separation and beamforming solve the signal separation problem from different points of view using different information. Each approach has its advantages and weaknesses. In order to achieve better separation performance, it is worth investigating possible approaches combining the advantages from both perspectives.

There are three kinds of approaches for combining convolutive blind source separation and beamforming in the literature.

7.4.1 Incorporation of Geometrical Information into Convolutive Blind Source Separation Algorithm

As we know, geometric information such as location of source (or direction of arrival) and sensor configuration is used in beamforming to align the beam pattern to specific direction. This kind of combination approach basically incorporates this geometrical information into the convolutive BSS algorithms.

The significant problem in frequency domain convolutive BSS is frequency permutation problem. In [Parra 2001][Parra 2002 A][Parra 2002 B], the geometric information used in adaptive beamforming is incorporated in frequency domain convolutive blind source separation algorithm as linear constraints or as the initial adaptation condition. These additional constraints inevitably reduce existing degrees of freedom so as to resolve some of the ambiguities in convolutive blind source separation algorithm. Two constrained methods were proposed; one is the geometrically initialized source separation in which the BSS filter coefficients are geometrically initialized with the filter parameters corresponding to delay-sum beamformers pointing to the individual sources. The other one is the geometrically constrained source separation in which geometric information is incorporated through a penalty term and linear constrained optimization algorithm is used for optimizing the cost function. Simulation results show that the new geometric source separation system obtained better separation performance than both the system with conventional beamformer only and the system with BSS only.

An accurate steering direction is assumed to be available in [Parra 2001][Parra 2002 A][Parra 2002 B]. This assumption is not always true. A new geometrically constrained BSS algorithm is proposed in [Knaak 2003] without this assumption. This algorithm is based on FastICA algorithm [Hyvarinen 1999] and roughly estimated geometric information. The advantage of this algorithm is that it attempts to solve the permutation problem by incorporating geometrical information. The performance of this algorithm is not sensitive to

the precision of the estimated geometrical constraint resulting in robustness of the algorithm in reverberant acoustical environment.

Besides incorporating geometrical information into frequency domain convolutive BSS for solving permutation problem, in [Aichner 2002], a new time domain convolutive BSS algorithm is proposed by utilizing geometric information such as sensor positions and assumed locations of sources. In this new algorithm, a null beamformer is constructed based on the available geometric information. The parameters in the null beamformer are exploited as initial conditions in a time domain convolutive BSS algorithm to speed its convergence rate and improve separation performance since the convergence and result of separation of gradient-based algorithms are influenced significantly by the initial conditions. Simulation results show that the separation performance is superior than conventional time domain convolutive BSS algorithm and can even work well in long reverberant environment.

7.4.2 Formulation of Convolutive Blind Source Separation as Multiple Sets of Adaptive Beamforming to Resolve Ambiguities in BSS

From section 7.3.1 and Fig. 7.3, convolutive blind source separation system can be viewed as multiple sets of adaptive beamforming, which means the separation filter array for every output can be viewed as a beamformer. Thus, we can utilize methods for analyzing beamforming to analyze BSS. Directivity pattern is a good example.

In [Kurita 2000], the idea described above is used in blind signal separation algorithm to deal with frequency permutation and arbitrary scaling problem in frequency domain convolutive BSS. First, the unmixing matrix for every frequency bin is obtained from instantaneous BSS method. Since the filter array connected with the same output is viewed as a beamformer, its corresponding directivity pattern can be calculated by beamforming approach. Thus, the null direction for every output at each frequency bin can be obtained from the directivity pattern. By swapping the output order of every frequency bin to make the output signals from frequency components with consistent null direction, the frequency permutation ambiguity can be resolved.

The directivity patterns used in [Kurita 2000] have grating lobes at high frequencies, which affect the accuracy of estimated direction of sources. In [Ikram 2002], the directivity patterns at different frequencies are investigated and a new approach is proposed by estimating the source location from the lower band of frequencies where no grating lobes appear. The frequency permutation is aligned by looking for nulls in the neighborhood of the estimated DOA.

Besides dealing with frequency permutation problem in [Kurita 2000], the directivity patterns obtained from unmixing matrix are also used to improve the convergence speed of convolutive BSS algorithm. In a series papers [Saruwatari 2002][Saruwatari 2006] , independent component analysis and beamforming are combined to deal with low convergence problem in convolutive BSS. First, ICA is used to perform blind source separation at every frequency bin and the unmixing matrix can be obtained at each frequency bin. Accordingly, the directivity pattern at each frequency bin can be calculated from its unmixing matrix as in [Kurita 2000]. Directions of arrival (DOA) of source signals are estimated from the directions of nulls at all frequency bins. In adaptation process, at each frequency bin, the direction null in the directivity pattern is compared with the estimated DOA of source signals, to verify if it is steering to the proper direction, the unmixing matrix from ICA algorithm is used. If not, the null-steering beamformer constructed from the estimated DOA information is used to substitute for unmixing matrix. By doing so, the unmixing matrix can be recovered from local minima in the optimization procedure to improve its convergence speed.

In [Mitianoudis 2003], the properties of using beamforming to address the permutation problem in frequency domain convoultive BSS algorithm was investigated in detail. From the observations obtained, frequency permutation alignment in high frequencies by beamforming is difficult because of spatial aliasing. Even the approaches described above [Saruwatari 2002][Saruwatari 2006] may not work well since the nulls corresponding to the both sources are really close.

The approach proposed in [Kurita 2000] plots the directivity pattern for every frequency bin, which is very time consuming. Moreover, for situations with more than two sources, it is difficult to estimate DOA of source signals from null directions since the directivity pattern becomes too complicated. In [Sawada 2004 B], a new method dealing with permutation problem in situations with more than two sources is proposed. In this approach, the unmixing matrix resulted from BSS stage is still viewed as beamformer. However, an approach is proposed to directly calculate direction of sources from the unmixing matrix at each frequency bin. By sorting the obtained directions of sources, a permutation matrix can be constructed to resolve the frequency permutation problem.

The problem of direction estimation from unmixing matrix is that directions of arrival cannot be estimated accurately at some frequencies, especially at low frequencies and high frequencies. In [Sawada 2004 A], a new robust and precise method for solving frequency permutation in frequency domain convolutive BSS is proposed by integrating direction of arrival approach and interfrequency correlation approach [Murata 2001]. Interfrequency correlation approach for frequency permutation alignment is based on the idea that signal envelops have high correlations at neighboring frequencies if separated signals are from the same source signal. However, the correlation approach is not robust since a misalignment at a frequency can cause misalignments in consequent frequencies. In this new method, for the frequencies where the direction of arrival can be estimated accurately, direction of arrival approach is used to align the frequency permutation, for other frequencies, correlation approach is used to do the alignment based on neighbouring correlation.

7.4.3 Utilization of the Beamforming Structure and the ICA Cost Function

In [Baumann 2003], a new convolutive blind source separation algorithm is proposed based on a beamforming structure and the ICA cost function. In this method, the unmixing system is constructed as multiple sets of beamformers as in Fig. 7.3. Besides steering the nulls towards interfering signal as in the conventional beamforming, the null-directions are also adjusted to make output signals as independent as possible. This means that the multiple set of beamformers are not adjusted separately, however, they are adjusted dependently to obtain

mutually independent outputs. Thus, the independence criterion, which includes higher-order statistics, and geometric information both are exploited in this algorithm. Simulation results show that it can achieve reasonable good SNR improvement for speech separation in real room recording.

7.5 Proposed Combined Adaptive Beamforming and Frequency Domain Convolutive Blind Source Separation

As we mentioned before, the significant advantage that adaptive beamforming provides is that it can exploit the spatial information of the sensor array. On the other hand, the strength of convolutive blind source separation is that it utilizes a very strong statistical property of source signals—mutual independence. Our purpose is to combine both for better separation performance.

We note that our hearing system provides a perfect example for combining beamforming and blind source separation. In a cocktail party environment, when listening to someone, we first direct ours ears towards the sound of a specific person, and then concentrate on separating the audio signals of different speakers in that direction. This means that for separating multichannel audio signals, our ears first form a beamformer to concentrate on signals from selected directions and ignore signals from other interfering directions. Then our complete auditory system acts as a blind source separation unit to separate the received signals from ours ears.

Our proposed system attempts to mimic the performance of human ears in a cocktail party environment. First, adaptive beamforming is used to isolate signals from specific directions (ear function), and then blind source separation is used to separate signal from different sources aiming in that direction (full auditory system function). The proposed two-stage separation system is shown in Fig. 7.4.

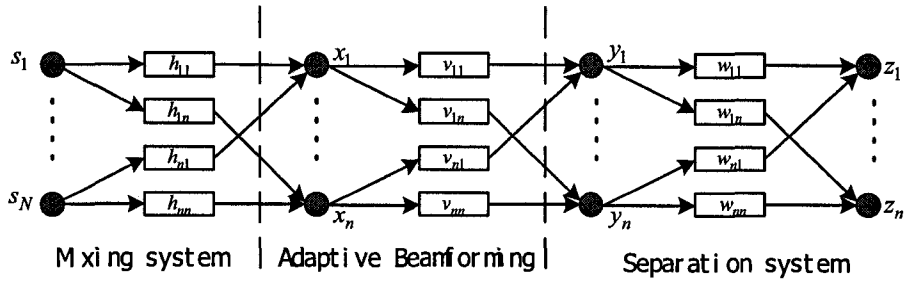


Figure 7-4: Proposed system architecture for combining beamforming with convolutive BSS

The mixing system in Fig. 7.4 is the same as the system described in Fig. 3.3. We describe the adaptive beamforming stage and blind separation system in the following.

7.5.1 Adaptive Beamforming Stage

The sound source localization and separation system proposed in [Asano 2001] is implemented here as our adaptive beamforming stage to zoom in the directions of selected speakers. First the range and direction of the speakers are estimated by an extended spatial spectrum estimator, MUSIC [Schmidt 1986]. Then the minimum variance beamformer is constructed based on the estimated location information. The detailed algorithm description is as follows:

1. Transfer time domain mixture signals to frequency domain.

$$\mathbf{X}(k,t) = \left[X_1(k,t),...,X_M(k,t)\right]^T = FFT\left(\left[x_1,x_2,...,x_M\right]\right)^T \tag{7.1}$$

where k is the index for frequency, t is the index for the time frame and x_i is the mixture signals.

2. Calculate the spatial correlation matrix from frequency domain mixture signal.

$$\mathbf{R}_k = E\left\{\mathbf{X}(k,t)\mathbf{X}^H(k,t)\right\} \tag{7.2}$$

where $E\{.\}$ is the expectation.

3. Compute eigenvalue decomposition of $\mathbf{R}_k$.

$$\mathbf{R}_k = \mathbf{E}_k \mathbf{\Lambda}_k \mathbf{E}_k^{-1} \tag{7.3}$$

The eigenvectors are divided into two parts: $\mathbf{E}_k^s$ and $\mathbf{E}_k^n$. $\mathbf{E}_k^s$ is corresponding to the N dominant eigenvalues and $\mathbf{E}_k^n$ corresponding to the rest of the eigenvalues.

4. Calculate the MUSIC spatial spectrum [Schmidt 1986].

$$P(r,\theta,k) = \frac{1}{\left|\tilde{a}_k(r,\theta)\mathbf{E}_k^n\right|^2} \tag{7.4}$$

where $\tilde{a}_k(r,\theta)$ is the normalized location vector for the scanning point (r,θ) as

$$\tilde{a}_k(r,\theta) = \frac{a_k(r,\theta)}{\left\|a_k(r,\theta)\right\|} \tag{7.5}$$

5. Average the MUSIC spatial spectrum over frequencies k_L to k_H to obtain the final spatial spectrum.

$$\bar{P}(r,\theta,k) = \frac{1}{K}\sum_{k=k_L}^{k_H} P(r,\theta,k) \tag{7.6}$$

where $K = k_H - K_L$.

6. Identify the peaks of the averaged spatial spectrum to determine the location of the sources as $(r_1,\theta_1),...,(r_N,\theta_N)$.

7. Construct corresponding beamformers according to estimated location information. The beamformer coefficients for the n th source is obtained from the following equation:

$$\mathbf{V}_n = \frac{\mathbf{R}_k^{-1}\mathbf{a}_k(r_n,\theta_n)}{\mathbf{a}_k(r_n,\theta_n)\mathbf{R}_k^{-1}\mathbf{a}_k(r_n,\theta_n)} \tag{7.7}$$

8. The spectrum of the n th beamformer output is

$$Y_n(k,t) = \mathbf{V}_n^H(k)\mathbf{X}(k,t) \tag{7.8}$$

7.5.2 Convolutive BSS Stage

In convolutive blind source separation stage, an unmixing system $\mathbf{W}$ is adaptively adjusted to make the outputs as independent as possible to recover the independent source signals. The update equation for estimating the FIR separating system $\underline{\mathbf{W}}$ is the algorithm used in [Lambert 1997][Lee 1997] as follows.

$$\underline{\mathbf{W}}_{k+1} = \underline{\mathbf{W}}_k + \mu(\underline{\mathbf{I}} - g(\underline{\mathbf{z}})\underline{\mathbf{z}}^T)\underline{\mathbf{W}}_k \tag{7.9}$$

where $\mathbf{z}$ is output vector from convolutive BSS stage and $g(\cdot)$ is a non-linear function.

The advantage of this proposed approach lies in the following aspects.

1. The two-stage setup allows the advantages of both beamforming and convolutive BSS to be implemented entirely.
2. By dividing a complex task into subtasks, we reduce the complexity of the overall system and make it more flexible for implementation.
3. Through the beamforming stage, the reverberation effects are greatly reduced, thus we can use shorter FIR filters in the subsequent BSS stage, which reduce the overall system complexity.
4. We can select different convolutive BSS algorithms, such as time domain, frequency domain, higher-order statistics based algorithms etc., in the second stage to get best separation result.
5. This approach mimics the way human ears separate mixed audio signals.

7.5.3 Simulation Results

In our simulation, two source speech signals are used. The sampling frequency is 8000Hz. The microphone array with 8 sensors is used to receive the mixed speech signals. Since the end users are humans, we use PESQ score [ITU 2000] to measure the similarity between the

recovered speech signal and the original speech signal in order to evaluate system performance at each stage. The PESQ standard is described in the ITU-T P862 as a perceptual evaluation tool of speech quality. The key process of the PESQ operation is to use a perceptual model analogous to the psychological representation of the original and degraded signal in the human auditory system. The output of the PESQ is a measure of the subjective assessment quality of the degraded signal and is rated as a value between –0.5 to 4.5.The bigger the score, the better the speech quality.

In the mixing system, we use the measured room impulse response [Asano 2001] to generate the 8 mixed signal $x_1, x_2, ..., x_8$ from two source signals s_1 and s_2. In the adaptive beamforming stage, the two speaker locations are estimated from the mixtures and two beamformers are constructed to get signals y_1 and y_2 from these two specific locations. In the convolutive BSS stage, the coefficients of the unmixing system are adaptively adjusted to further cancel the remaining effects of cross-talk and get the estimated source signals z_1 and z_2.

Different speech mixing combinations are used in our experiments to verify the performance of our proposed system, including female and male speech mixtures, female and female speech mixtures, male and male speech mixtures. For each case, we have three different speech signals to repeat the experiments.

The PESQ scores for the sensors' 8 mixed speech signals in a variety of male and male mixtures cases compared with two source signals are shown in Tables 7.1 to 7.4.

PESQ		x1	x2	x3	x4	x5	x6	x7	x8
female1/female2	s1	1.69	1.65	1.63	1.59	1.44	1.52	1.43	1.48
	s2	1.58	1.25	1.36	1.33	1.47	1.52	1.77	1.61
female1/female3	s1	1.86	1.90	1.85	1.75	1.68	1.69	1.54	1.80
	s2	1.56	1.40	1.44	1.46	1.36	1.73	1.79	1.51
female2/female3	s1	1.95	1.88	1.91	1.90	1.68	1.78	1.70	1.93
	s2	1.26	1.14	1.12	1.22	1.20	1.43	1.43	1.34

Table 7-1: PESQ for Female and female mixture case

PESQ		x1	x2	x3	x4	x5	x6	x7	x8
male1/male2	s1	2.07	2.03	2.11	2.07	1.95	1.94	1.87	2.06
	s2	1.65	1.57	1.64	1.71	1.78	1.80	1.87	1.72
male1/male3	s1	2.11	2.07	2.15	2.12	1.96	1.94	1.87	2.10
	s2	1.75	1.63	1.67	1.75	1.83	1.90	1.93	1.80
male2/male3	s1	2.07	2.04	2.14	2.12	1.94	1.92	1.83	2.06
	s2	1.72	1.61	1.66	1.70	1.54	1.85	1.89	1.77

Table 7-2: PESQ for Male and male mixture case

PESQ		x1	x2	x3	x4	x5	x6	x7	x8
female1/male1	s1	1.84	1.83	1.81	1.74	1.59	1.65	1.54	1.76
	s2	1.44	1.37	1.46	1.63	1.70	1.69	1.73	1.39
female1/male2	s1	1.78	1.76	1.79	1.74	1.58	1.62	1.52	1.71
	s2	1.38	1.25	1.40	1.38	1.56	1.54	1.78	1.40
female1/male3	s1	1.83	1.84	1.82	1.73	1.59	1.66	1.50	1.78
	s2	1.61	1.52	1.60	1.61	1.80	1.79	1.88	1.68

Table 7-3: PESQ for Female and male mixture case

PESQ		x1	x2	x3	x4	x5	x6	x7	x8
female/female	s1	1.83	1.81	1.80	1.75	1.60	1.66	1.56	1.74
	s2	1.46	1.26	1.31	1.34	1.34	1.56	1.66	1.49
male/male	s1	2.08	2.04	2.13	2.10	1.95	1.93	1.86	2.07
	s2	1.71	1.60	1.65	1.72	1.72	1.85	1.90	1.76
female/male	s1	1.82	1.81	1.80	1.74	1.59	1.64	1.52	1.75
	s2	1.48	1.38	1.49	1.54	1.68	1.67	1.80	1.49

Table 7-4: Averaged PESQ scores for mixed speech signals in three different mixing combination

From the PESQ scores in the above tables, we can see that the mixed speech signals are equally similar to both sources, i.e. both source signals are heard in the mixture.

In the adaptive beamforming stage, two outputs y_1 and y_2 are obtained as the estimated signals from the selected directions. From these output signals, we can hear that the source signals are the dominant part of corresponding outputs and lower multipath reflections can also be heard. The PESQ scores for the two outputs y_1 and y_2 in different mixing cases are shown in Tables 7.5 to 7.8.

PESQ	female1/female2		female1/female3		female2/female3	
	y1	y2	y1	y2	y1	y2
s1	2.27	0.70	2.39	1.07	2.37	0.81
s2	0.60	2.19	0.75	2.19	0.65	2.08

Table 7-5: PESQ scores for outputs from adaptive beamforming stage in female and female mixture case

PESQ	male1/male2		male1/male3		male2/male3	
	y1	y2	y1	y2	y1	y2
s1	2.50	1.68	2.52	1.50	2.46	1.25
s2	1.52	2.32	0.87	2.31	0.89	2.27

Table 7-6: PESQ scores for outputs from adaptive beamforming stage in male and male mixture case

PESQ	female1/male1		female1/male2		female2/female3	
	y1	y2	y1	y2	y1	y2
s1	2.29	0.48	2.27	0.72	2.28	0.89
s2	0.88	2.39	0.81	2.33	0.64	2.30

Table 7-7: PESQ scores for outputs from adaptive beamforming stage in female and male mixture case

PESQ	female/female		male/male		female/male	
	y1	y2	y1	y2	y1	y2
s1	2.34	0.86	2.49	1.48	2.28	0.69
s2	0.67	2.15	1.09	2.30	0.77	2.34

Table 7-8: Average PESQ scores for outputs from adaptive beamforming stage

The PESQ scores from Table 7.5 to Table 7.8 show that the outputs from the adaptive beamforming stage are more biased to one source signal and away from the other source signal. Personal listening confirms the improved signal quality.

In the convolutive BSS stage, the outputs from the adaptive beamforming stage are further processed to reduce the effects of cross-talk. Since most of the reverberation effects have already been removed by the adaptive beamforming stage, we found that we only need to use very short FIR filters to complete the speech separation in the selected direction. In this simulation, the length of FIR filter is only 32 and the BSS algorithm easily converges to acceptable results. Similar separation quality cannot be obtained even by filters with 1024 taps when there is no adaptive beamforming stage as pre-processor. Thus, the computation complexity for BSS is greatly reduced.

The PESQ scores for the two final outputs z_1 and z_2 in the mixing cases considered are shown in Tables 7.9 to 7.12.

PESQ	female1/female2		female1/female3		female2/female3	
	z1	z2	z1	z2	z1	z2
s1	2.33	0.46	2.44	0.82	2.47	0.76
s2	0.56	2.34	0.48	2.28	0.41	2.11

Table 7-9: PESQ scores for outputs from convolutive BSS stage in female and female mixture case

PESQ	male1/male2		male1/male3		male2/male3	
	z1	z2	z1	z2	z1	z2
s1	2.67	1.44	2.72	0.96	2.63	0.72
s2	0.95	2.51	0.66	2.48	0.55	2.43

Table 7-10: PESQ scores for outputs from convolutive BSS stage in male and male mixture case

PESQ	female1/male1		female1/male2		female1/male3	
	z1	z2	z1	z2	z1	z2
s1	2.47	0.45	2.45	0.78	2.47	0.38
s2	0.85	2.55	0.68	2.43	0.44	2.41

Table 7-11: PESQ scores for outputs from convolutive BSS stage in female and male mixture case

PESQ	female/female		male/male		female/male	
	z1	z2	z1	z2	z1	z2
s1	2.41	0.68	2.67	1.04	2.46	0.54
s2	0.48	2.24	0.72	2.47	0.66	2.46

Table 7-12: Average PESQ scores for outputs from convolutive BSS stage

The PESQ scores from Table 7.9 to Table 7.12 show that the combined beamforming convolutive BSS algorithm further improves the quality of separation. This was also confirmed by informal listening experiments.

On the negative side, the recovered speech signals are distorted in some sense because of their flatter spectrum. The separation performance is expected to be enhanced if this distortion is addressed.

7.5.4 Conclusion for this proposed approach

In this section, we investigate approaches combining spatial information used in beamforming with time/frequency processing used in convolutive blind source separation aiming for better separation performance given the increased information used. Carefully comparing similarities and differences of BSS and adaptive beamforming and reviewing existing combination approaches, we present our proposed new combined method which mimics the way our hearing system used to separate audio signal in acoustic environments. Simulation results confirm our expectations and show that our system works well in a real room environment.

7.6 Proposed Truly Blind Combined Independent Component Analysis and Beamforming Speech Separation Algorithm

As we can see from Section 7.5, our proposed system structure in which BSS system cascades with beamforming system works very well as we expected. However, the problem in this system is that we used the MUSIC algorithm to search the DOA of the source speech. This approach has two problems. One is that we need more microphones than speakers. The second one is that we need some prior information about the room; this makes the whole method not completely blind.

Recently the relationship between the ICA based frequency domain BSS and null beamformer was studied by H. Saruwatari in a series of papers [Saruwatari 2002][Saruwatari 2006]. It is shown that the unmixing matrix in frequency domain BSS system has directivity patterns similar to the null beamformers after the BSS system converges by ICA based update algorithm. The DOA information of sources can be estimated from the directivity patterns.

In this section, based on the ideas used in [Saruwatari 2002][Saruwatari 2006] and our own investigation, we propose a new approach combining independent component analysis and beamforming for blind speech signal separation in real acoustic environment. By mimicking human hearing system, our separation system is again constructed as a beamforming system cascaded with a blind source separation (BSS) system. In the beamforming stage, the DOAs

of selected sources are estimated blindly; then beamformers are constructed to extract signals from these directions. In the BSS stage, frequency domain convolutive algorithm is utilized to further reduce the interference in the given direction and improve the separation performance. Based on detailed study of the system performance at different frequencies, we propose modification to significantly reduce the overall complexity. Compared with existing systems, our approach significantly reduces the computational complexity while keeping similar separation performance.

7.6.1 Review of combination of ICA and beamforming for fast convergence

In [Saruwatari 2002][Saruwatari 2006], independent component analysis and beamforming are combined to deal with low convergence problem in convolutive BSS. First, ICA is used to perform blind source separation at every frequency bin and the unmixing matrix is determined at each frequency bin. Accordingly, the directivity pattern at each frequency bin is obtained from its unmixing matrix. Directions of arrival (DOA) of source signals are estimated from the directions of nulls at all frequency bins. In adaptation process, at each frequency bin, the null in the directivity pattern is compared with the estimated DOA of source signals. If it is steering to the proper direction, the unmixing matrix from ICA algorithm is used. If not, the null-steering beamformer constructed from the estimated DOA information is used to substitute unmixing matrix. By doing so, the unmixing matrix can be recovered from local minimum in the optimization procedure to improve its convergence speed. This algorithm is reviewed in detail as follows.

1) Initialization: set the initial $\mathbf{W}_0(f)$ to an arbitrary value.

2) Coefficients update: the coefficients are updated by ICA-based algorithm at each frequency bin independently as follows.

$$\mathbf{W}_{i+1}^{(ICA)}(f) = \mathbf{W}_i(f) + \eta\left[diag\left(\left\langle \boldsymbol{\varphi}(\mathbf{y}(f,t))\mathbf{y}^H(f,t)\right\rangle_t\right) - \left\langle\left\langle \boldsymbol{\varphi}(\mathbf{y}(f,t))\mathbf{y}^H(f,t)\right\rangle_t\right\rangle\right]\mathbf{W}_i(f) \quad (7.10)$$

In this equation, the superscript ICA means the unmixing matrix is obtained from ICA update algorithm. However, $\mathbf{W}_i(f)$ in the right side of the equation can be obtained either from ICA or from beamforming depends on the criterion in step 5.

3) DOA estimation: the directivity pattern for l th source at frequency f is calculated as $F_l(f,\theta)$ by multiplying the unmixing matrix at frequency f with a steering vector $\mathbf{e}(f,\theta)$ as follows.

$$\left[F_1(f,\theta),F_2(f,\theta)\right]^T = \mathbf{W}^{(ICA)}(f)\mathbf{e}(f,\theta) \tag{7.11}$$

$$\mathbf{e}(f,\theta)=\left[\exp(j2\pi f d_1 \sin\theta/c),\exp(j2\pi f d_2 \sin\theta/c)\right]^T \tag{7.12}$$

By searching the null in the directivity patterns and averaging the null directions over the whole frequency band, the DOAs of the sources are estimated as $\hat{\theta}_l$.

4) Beamforming: construct the beamformers based on the null beamforming technique from the obtained DOA estimates.

$$\mathbf{W}^{(BF)}(f)=\left[\mathbf{e}\left(f,\hat{\theta}_1\right),\mathbf{e}\left(f,\hat{\theta}_2\right)\right]^{-1} \tag{7.13}$$

5) Selection of unmixing matrix: calculate the estimated coherence function once for $\mathbf{W}^{(ICA)}(f)$ and once for $\mathbf{W}^{(BF)}(f)$ as given by

$$C(\mathbf{W}(f))=\frac{\left|\left\langle Y_1(f,t)Y_2^*(f,t)\right\rangle_t\right|}{\sqrt{\left\langle|Y_1(f,t)|^2\right\rangle_t\left\langle|Y_2(f,t)|^2\right\rangle_t}} \tag{7.14}$$

where $\left[Y_1(f,t),Y_2(f,t)\right]^T=\mathbf{W}(f)\left[X_1(f,t),X_2(f,t)\right]^T$.

The better unmixing matrix is selected based on the following criterion.

$$\mathbf{W}(f)=\begin{cases}\mathbf{W}^{(ICA)}(f) & C\left(\mathbf{W}^{(ICA)}(f)\right)\le C\left(\mathbf{W}^{(BF)}(f)\right)\\ \mathbf{W}^{(BF)}(f) & C\left(\mathbf{W}^{(ICA)}(f)\right)> C\left(\mathbf{W}^{(BF)}(f)\right)\end{cases} \tag{7.15}$$

6) Permutation and scaling: the permutation problem is resolved by sorting the outputs from the directivity information in step 3. The inconsistent scaling problem is resolved by normalizing the directivity patterns according to the gain in each source direction.

The limitation of this approach is its huge computational complexity. From the above description, we can see that *at every iteration and at each frequency*, the ICA algorithm is used to update the weight coefficients; then the DOA information at this frequency is obtained by searching for the null from the directivity pattern of unmixing matrix; the beamformer is formed for every frequency and a comparison is conducted between ICA and beamformer directions. All these operations are very time consuming and very complicated.

Another limitation of this method is that its separation performance is very sensitive to the frame length of frequency domain BSS algorithm. In the same simulation environment later used in Section 6, we test the separation performance of the method in [Saruwatari 2002][Saruwatari 2006] under different frame lengths. The separation quality is shown in Table 7.13 to Table 7.17, as evaluated by PESQ [ITU 2000]. In Table 7.13, we show the PESQ scores of mixture signals compared with original source signals. The PESQ scores of separated signals compared with original source signals when the frame length is 256, 512, 1024 and 2048 are shown in Table 7.14 to Table 7.17 respectively. From these tables, we can see that a minimum frame length of 1024 is needed in this approach so that the mixed speech is separated with acceptable quality. This further contributes to the high computational complexity of the algorithm.

PESQ	x1	x2
s1	1.62	1.60
s2	1.47	1.50

Table 7-13: PESQ for the mixtures

PESQ	out1	out2
s1	1.88	1.19
s2	1.29	1.43

Table 7-14: PESQ for the separated signal when frame length = 256

PESQ	out1	out2
s1	2.11	0.61
s2	1.05	1.70

Table 7-15: PESQ for the separated signal when frame length = 512

PESQ	out1	out2
s1	2.62	0.70
s2	0.26	2.49

Table 7-16: PESQ for the separated signal when frame length = 1024

PESQ	out1	out2
s1	2.58	0.63
s2	0.22	2.70

Table 7-17: PESQ for the separated signal when frame length = 2048

7.6.2 Analysis of Unmixing Filter Matrix in FD-BSS

The relationship between the ICA-based frequency domain BSS and null beamformer was studied in [10][11][12]. It is shown that after convergence, the unmixing matrix in ICA-based frequency domain BSS system has directivity patterns similar to the null beamformers. In this section, we further study this property through simulation with particular attention to the impact of the frame length used on the accuracy of estimation of the DOA. We also investigate relationship of the estimated DOAs and the frequency bands used in this estimation to further understand what affects the quality of the resulting estimates and determines the necessary complexity.

In our simulation, two mixture signals from [Sawada] are used in our frequency domain BSS system. The room setup is shown in Fig. 7.5.

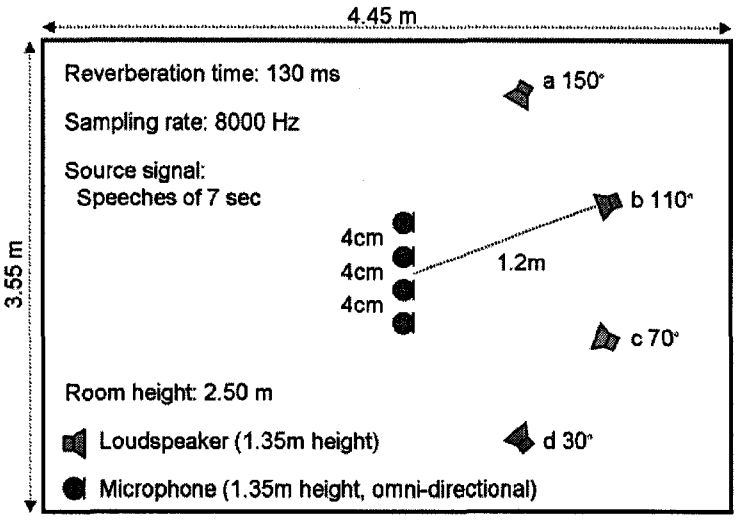


Figure 7-5: Experiment environment description from [Sawada].

Experiment #1

The simulation environment for this experiment is set up as follows:

Source signal: speech of 6s; Distance between 2 microphones: 4cm; Sampling frequency: 8000Hz; Frame size of FFT: 1024; Frame shift: 16; Step size: 0.0001

In Fig. 7.6, we show the directivity pattern for one source after 100 iterations. From this diagram, we can see that the unmixing matrix obtained from frequency domain BSS algorithm shows a directivity pattern similar to the null beamformers after the algorithm converges. The DOA of source signals can be estimated from this directivity pattern.

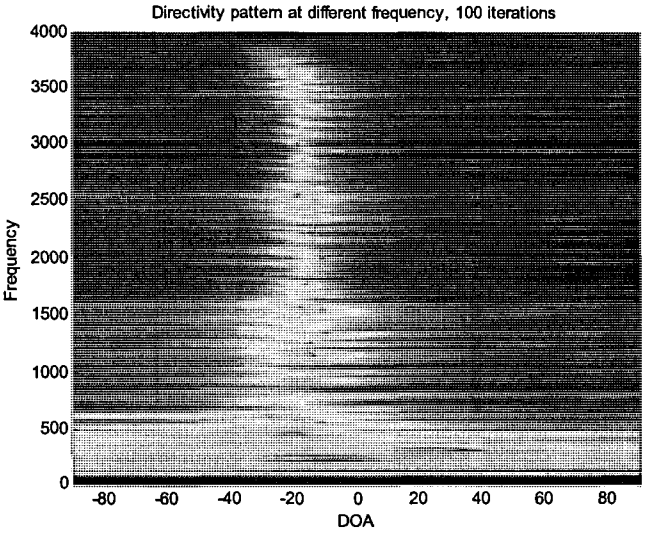


Figure 7-6: Directivity pattern after 100 iterations for one source

However, it is clear from Fig. 7-6 that the reliability of the directivity pattern is a function of the frequency. Figures 7-7, 7-8, 7-9 and 7-10 show the directivity patterns obtained based on different frequencies. The sharpness of the minimum affects the accuracy of determining the DOA. Clearly, the DOA can be accurately determined at f = 3156 Hz (Fig. 7-7), but not at f = 781 Hz (Fig. 7-8). The DOA cannot be determined with any reliability at f = 78 Hz (Fig. 7-9) and f = 3992 Hz (Fig. 7-10).

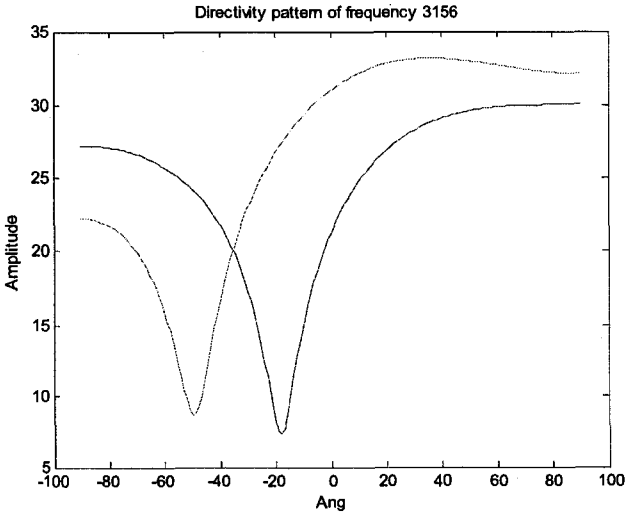


Figure 7-7: Directivity pattern for frequency 3156 Hz

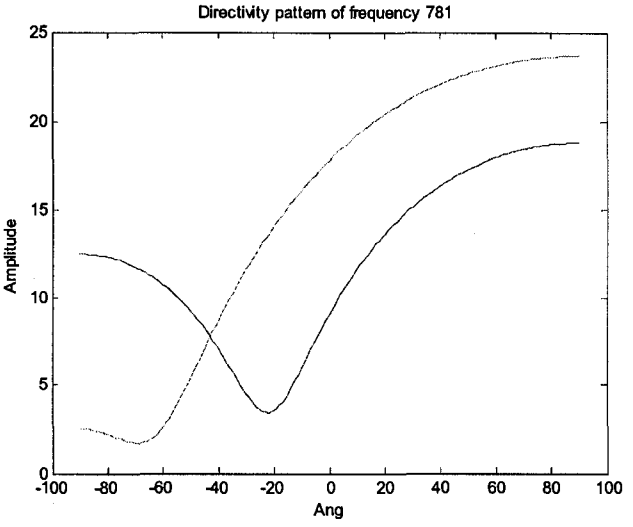


Figure 7-8: Directivity pattern for frequency 781 Hz

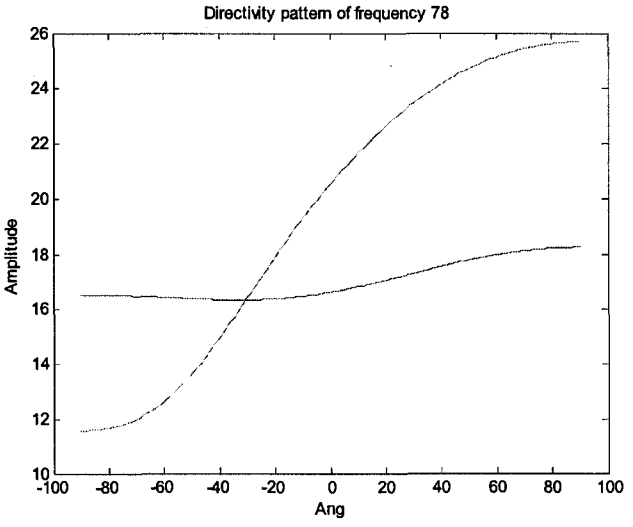


Figure 7-9: Directivity pattern for frequency 78 Hz

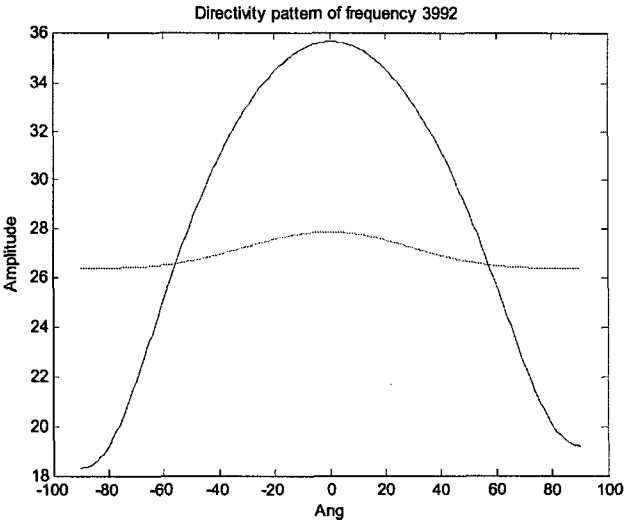


Figure 7-10: Directivity pattern for frequency 3992 Hz

Experiment #2

In this experiment, we test the effect of frame length on the estimation of DOAs of source signals. Different frame lengths are used in this experiment: 128, 256, 512, 1024 and 2048. When estimating the DOAs of the source signals, we average the obtained DOAs from each frequency over different frequency bands: over all frequencies (0 to 4000), band 1 (frequencies 0 to 500), band 2 (frequencies 500 to 1000), band 3 (frequencies 1000-1500), band 4 (frequencies 1500-2000), band 5 (frequencies 2000-2500), band 6 (frequencies 2500-3000), band 7 (frequencies 3000-3500), band 8 (frequencies 3500-4000).

Case #1: Frame length = 128

Fig. 7.11 to Fig. 7.15 show the estimated DOA for two sources at different frequencies after 20, 40, 60, 80 and 100 iterations when the frame length is 128. In Fig. 7.16, we show the averaged DOA estimations over different frequency bands and at different iterations.

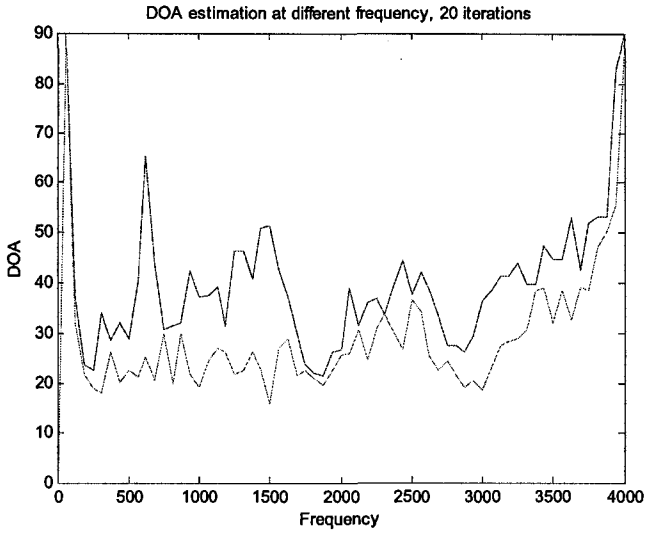


Figure 7-11: DOA estimation at different frequency after 20 iterations when frame length = 128

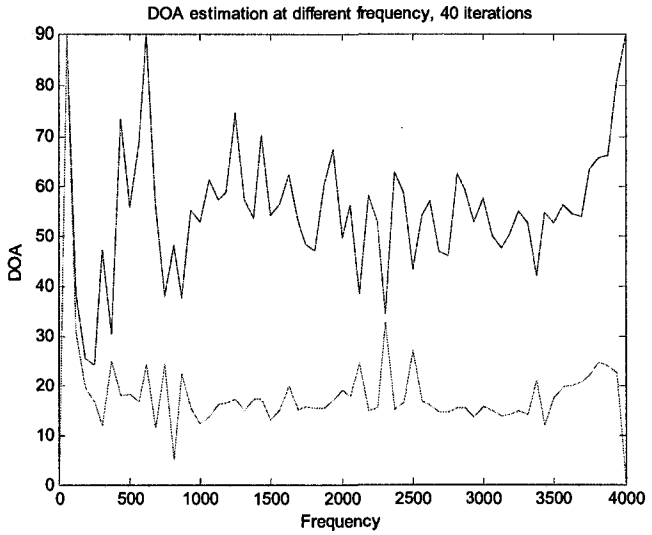


Figure 7-12: DOA estimation at different frequency after 40 iterations when frame length = 128

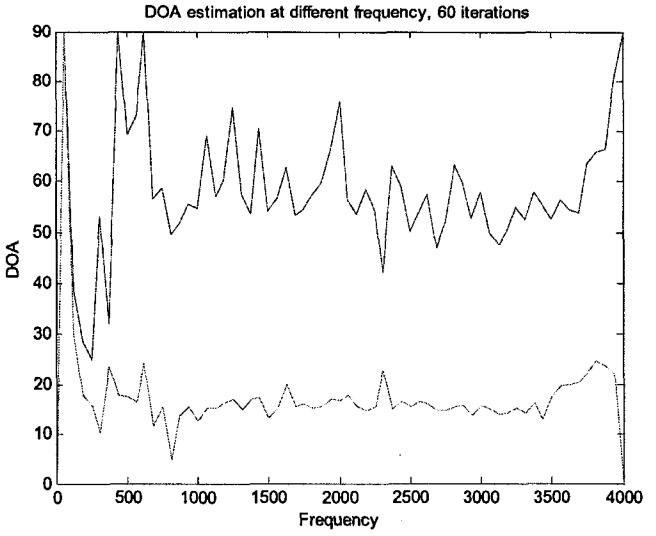


Figure 7-13: DOA estimation at different frequency after 60 iterations when frame length = 128

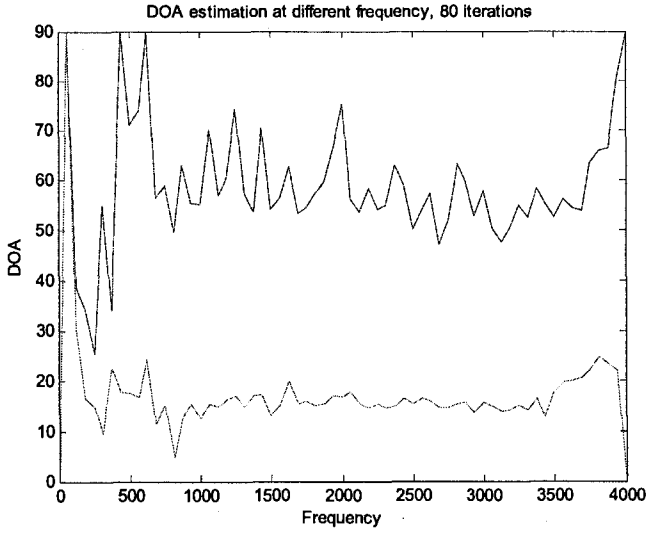


Figure 7-14: DOA estimation at different frequency after 80 iterations when frame length = 128

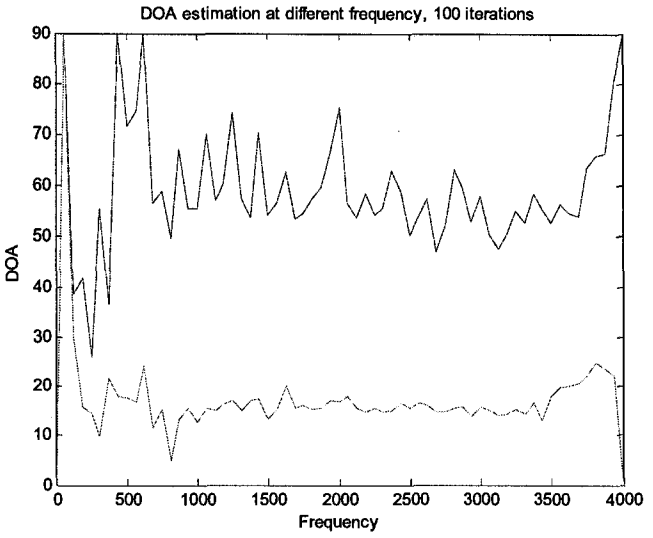


Figure 7-15: DOA estimation at different frequency after 100 iterations when frame length = 128

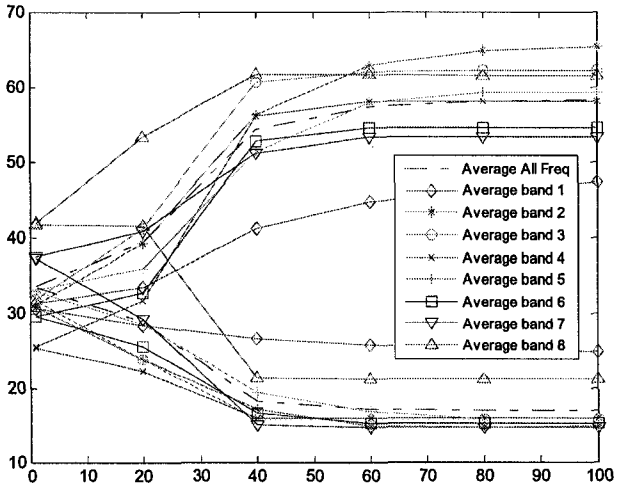


Figure 7-16: DOA estimation over different frequency band and from different iterations when frame length = 128

Case #2: Frame length = 256

Fig. 7.17 to Fig. 7.21 show the estimated DOA for two sources at different frequencies after 20, 40, 60, 80 and 100 iterations when the frame length is 256. In Fig. 7.22, we show the averaged DOA estimations over different frequency bands and at different iterations.

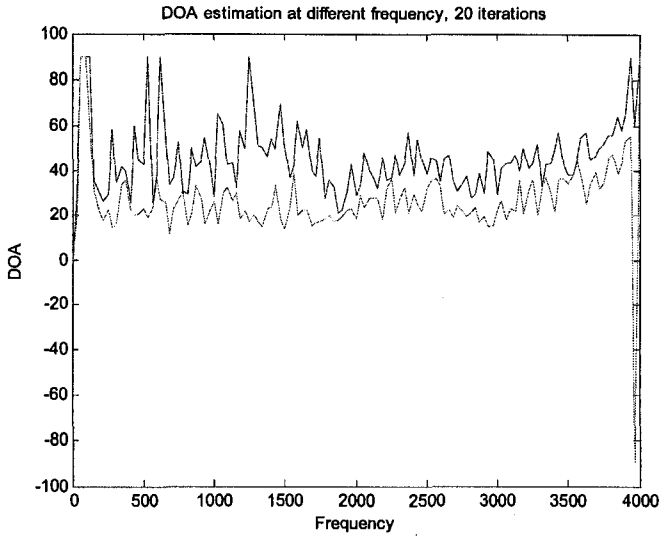


Figure 7-17: DOA estimation at different frequency after 20 iterations when frame length = 256

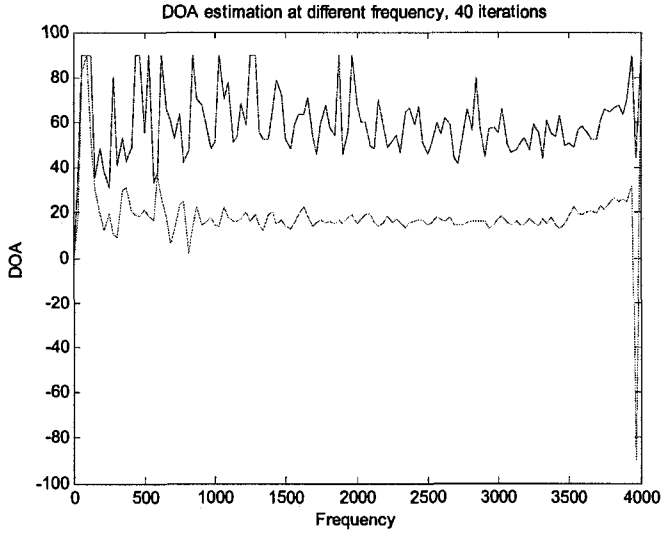


Figure 7-18: DOA estimation at different frequency after 40 iterations when frame length = 256

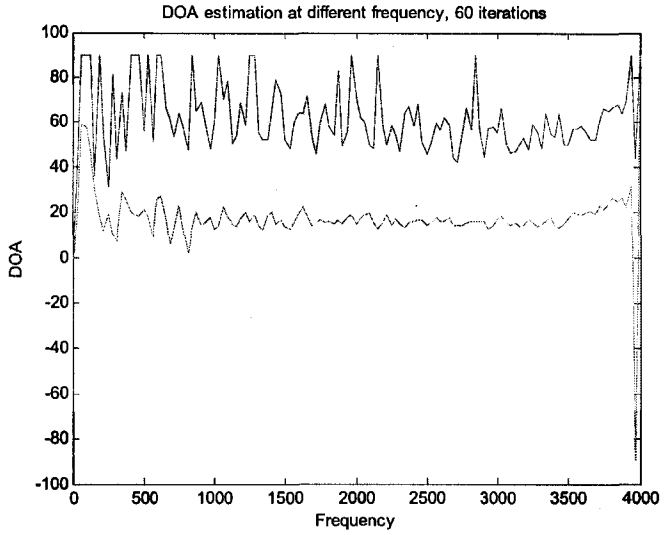


Figure 7-19: DOA estimation at different frequency after 60 iterations when frame length = 256

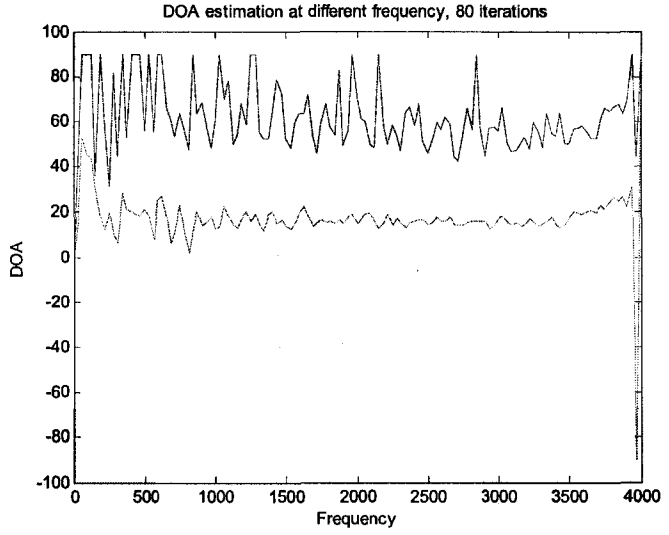


Figure 7-20: DOA estimation at different frequency after 80 iterations when frame length = 256

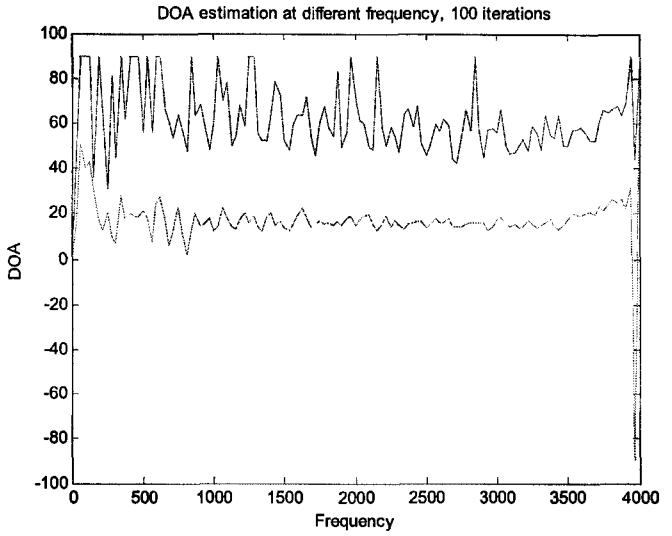


Figure 7-21: DOA estimation at different frequency after 100 iterations when frame length = 256

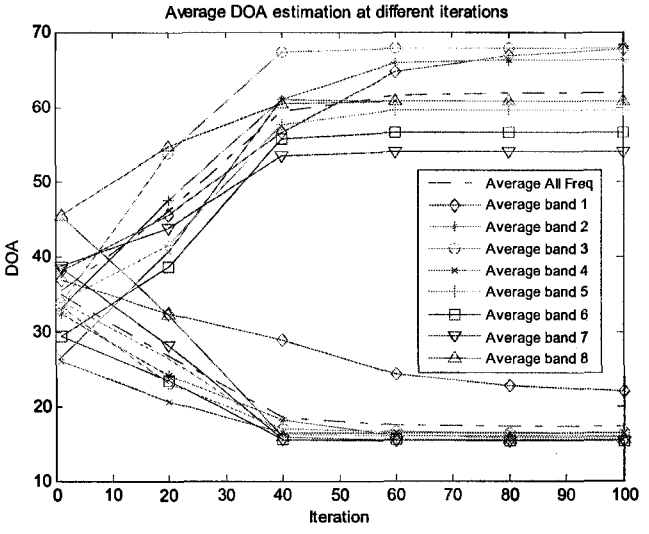


Figure 7-22: DOA estimation over different frequency band and from different iterations when frame length = 256

Case #3: Frame length = 512

Fig. 7.23 to Fig. 7.27 show the estimated DOA for two sources at different frequencies after 20, 40, 60, 80 and 100 iterations when the frame length is 256. In Fig. 7.28, we show the averaged DOA estimations over different frequency bands and at different iterations.

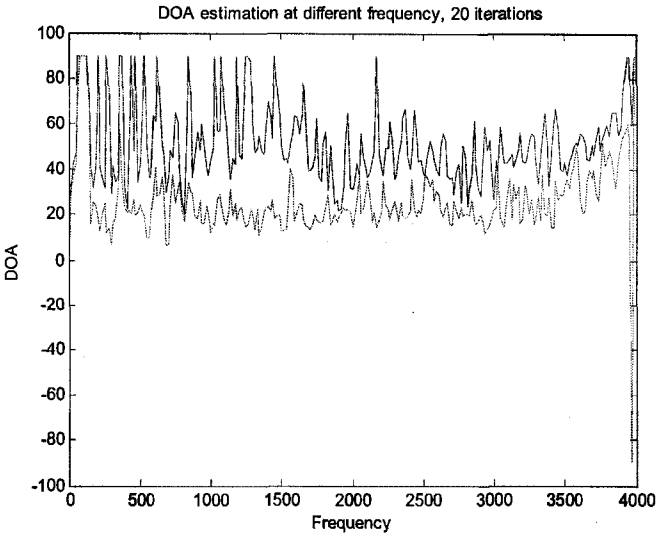


Figure 7-23: DOA estimation at different frequency after 20 iterations when frame length = 512

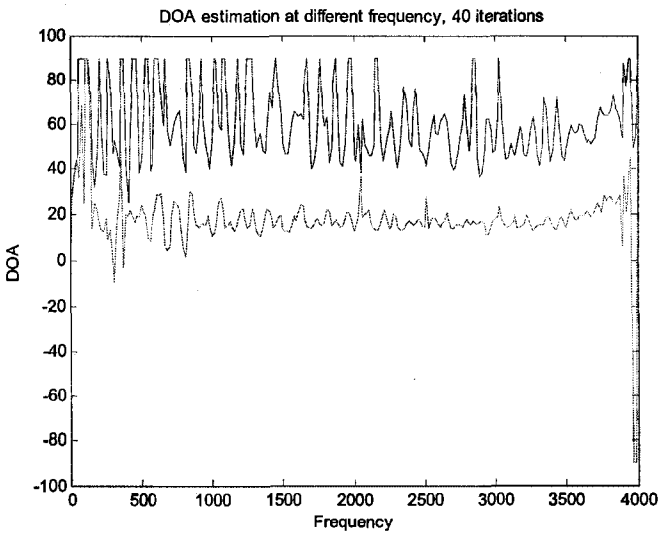


Figure 7-24: DOA estimation at different frequency after 40 iterations when frame length = 512

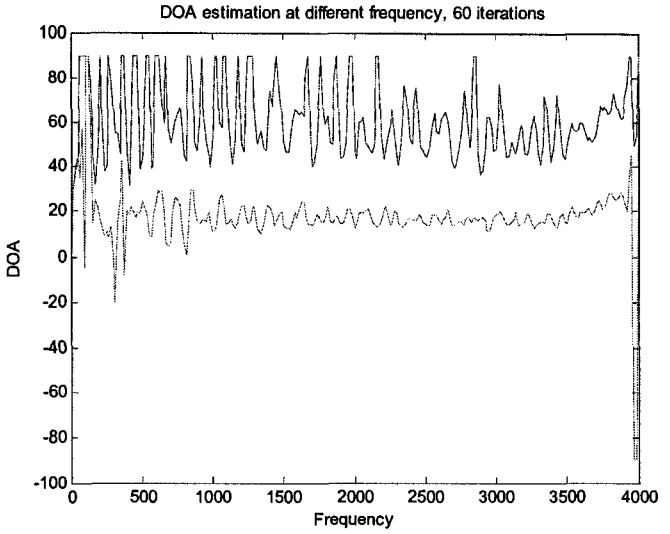


Figure 7-25: DOA estimation at different frequency after 60 iterations when frame length = 512

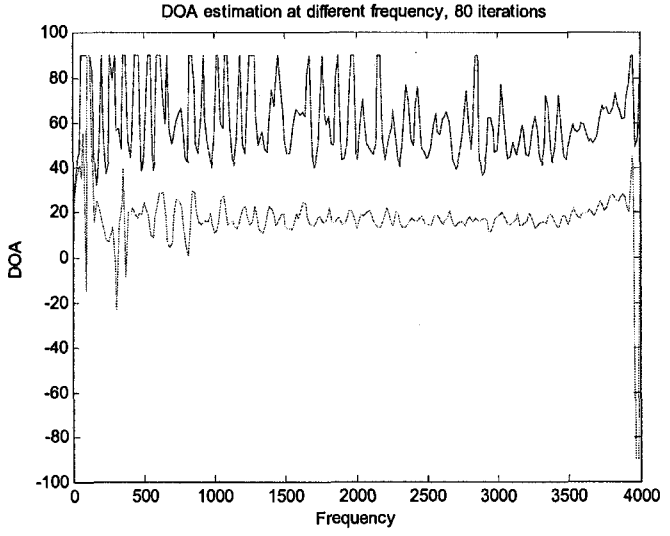


Figure 7-26: DOA estimation at different frequency after 80 iterations when frame length = 512

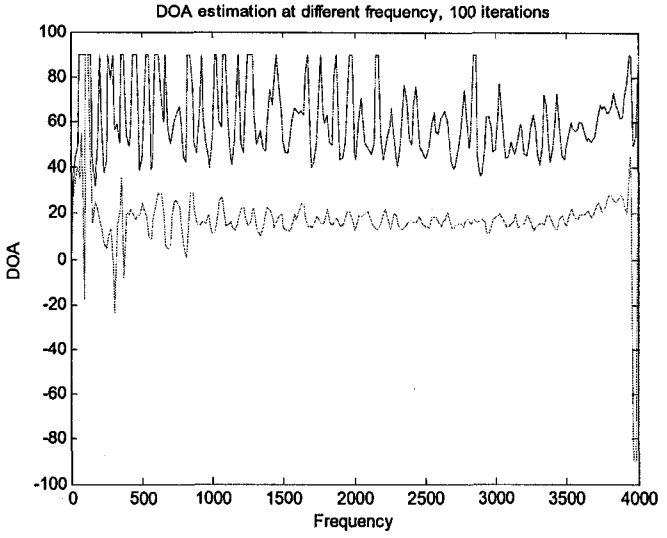


Figure 7-27: DOA estimation at different frequency after 100 iterations when frame length = 512

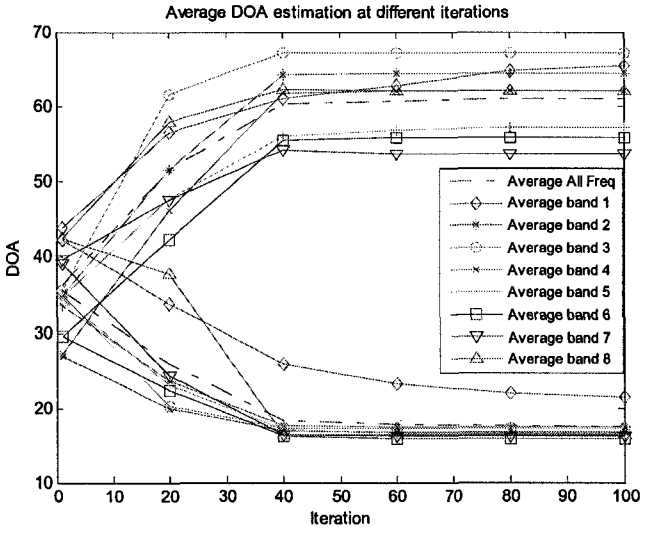


Figure 7-28: DOA estimation over different frequency band and from different iterations when frame length = 512

Case #4: Frame length = 1024

Fig. 7.29 to Fig. 7.33 show the estimated DOA for two sources at different frequencies after 20, 40, 60, 80 and 100 iterations when the frame length is 256. In Fig. 7.34, we show the averaged DOA estimations over different frequency bands and at different iterations.

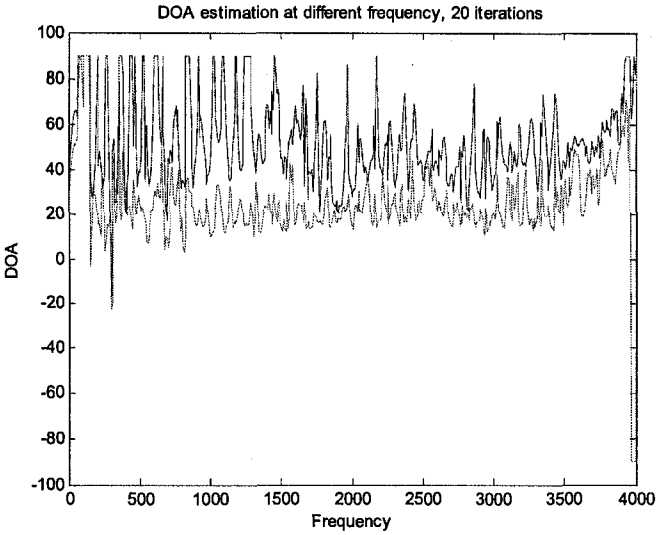


Figure 7-29: DOA estimation at different frequency after 20 iterations when frame length = 1024

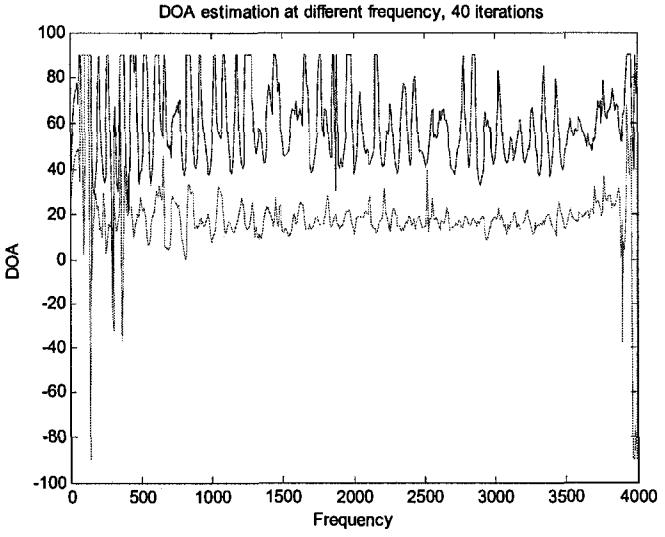


Figure 7-30: DOA estimation at different frequency after 40 iterations when frame length = 1024

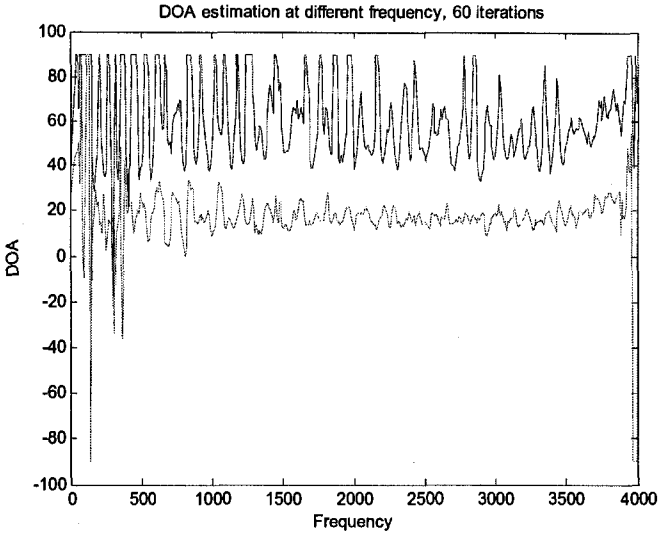


Figure 7-31: DOA estimation at different frequency after 60 iterations when frame length = 1024

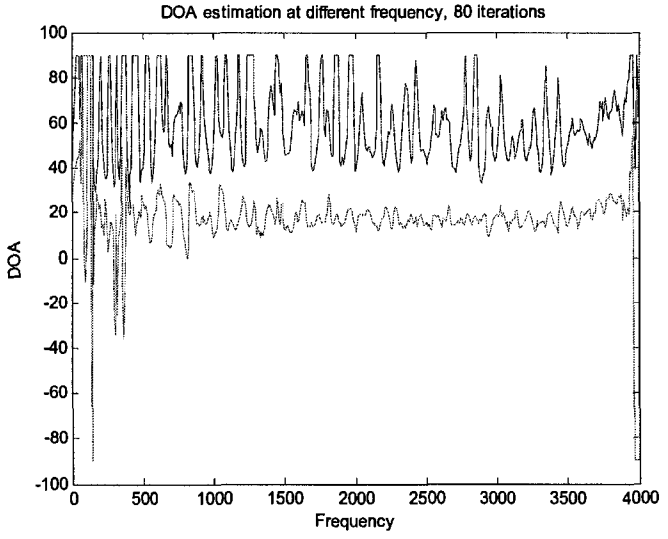


Figure 7-32: DOA estimation at different frequency after 80 iterations when frame length = 1024

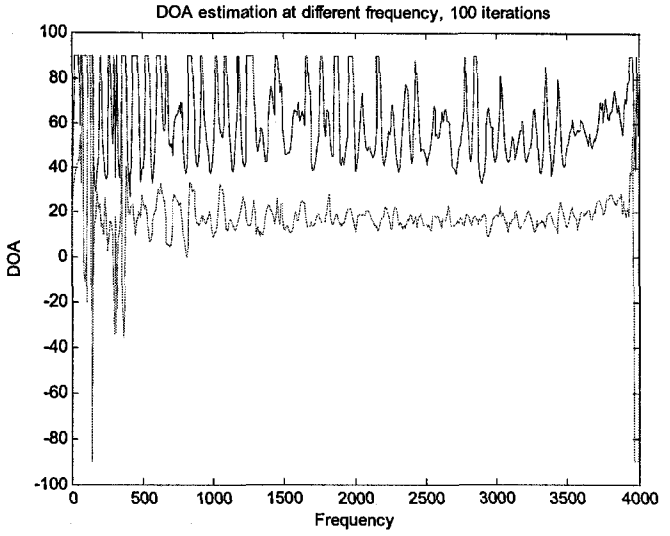


Figure 7-33: DOA estimation at different frequency after 100 iterations when frame length = 1024

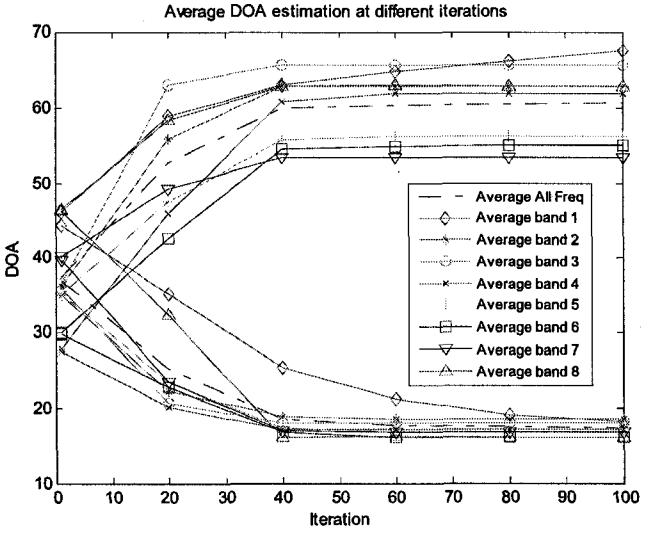


Figure 7-34: DOA estimation over different frequency band and from different iterations when frame length = 1024

Case #5: Frame length = 2048

Fig. 7.35 to Fig. 7.39 show the estimated DOA for two sources at different frequencies after 20, 40, 60, 80 and 100 iterations when the frame length is 256. In Fig. 7.40, we show the averaged DOA estimations over different frequency bands and at different iterations.

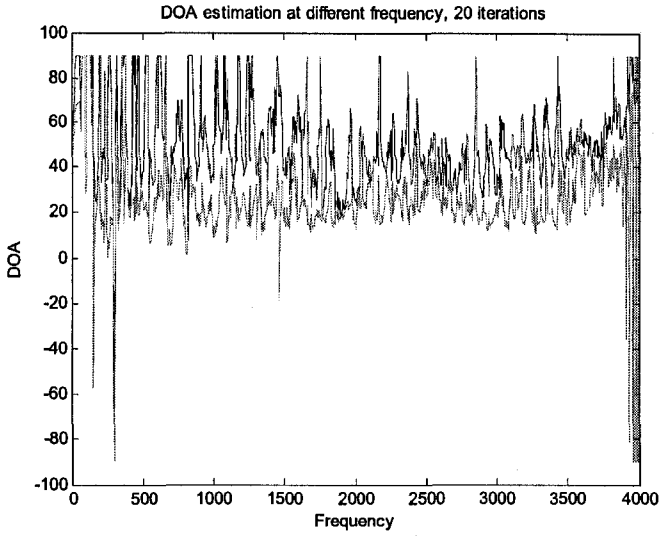


Figure 7-35: DOA estimation at different frequency after 20 iterations when frame length = 2048

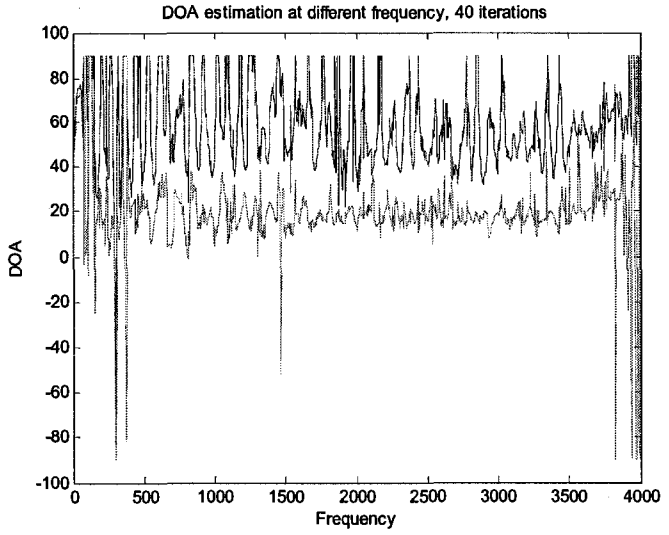


Figure 7-36: DOA estimation at different frequency after 40 iterations when frame length = 2048

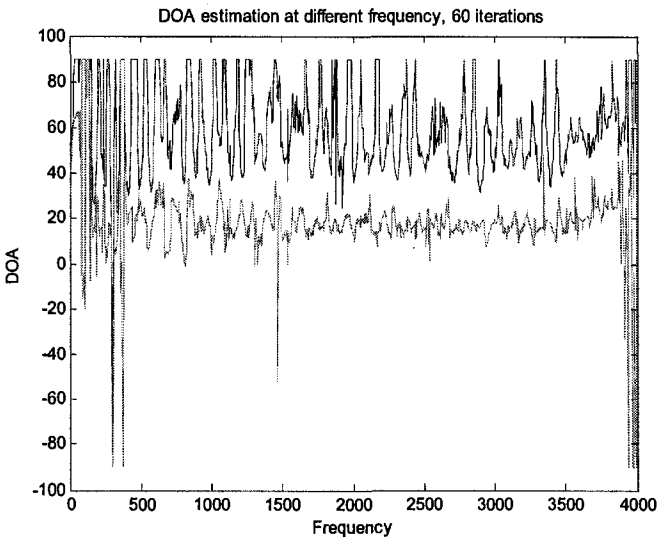


Figure 7-37: DOA estimation at different frequency after 60 iterations when frame length = 2048

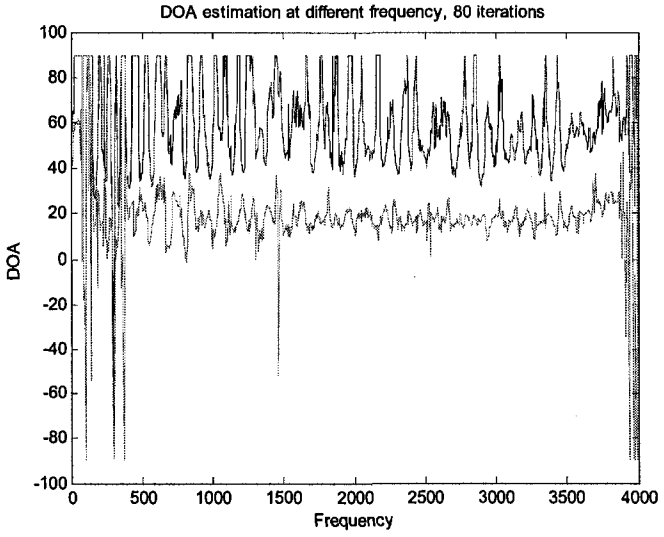


Figure 7-38: DOA estimation at different frequency after 80 iterations when frame length = 2048

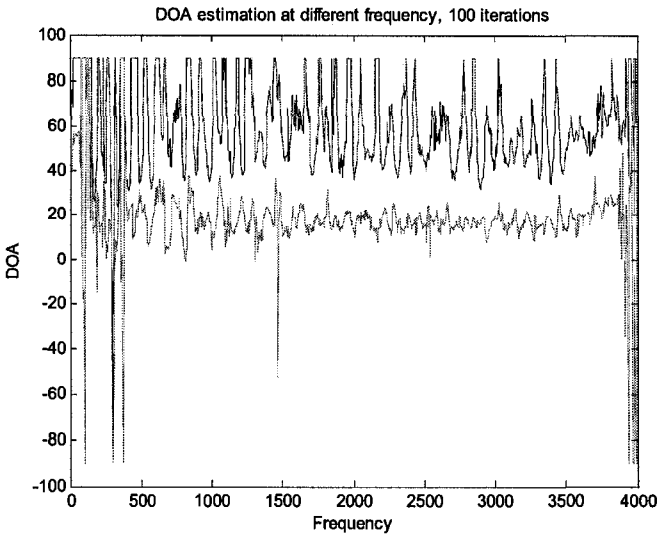


Figure 7-39: DOA estimation at different frequency after 100 iterations when frame length = 2048

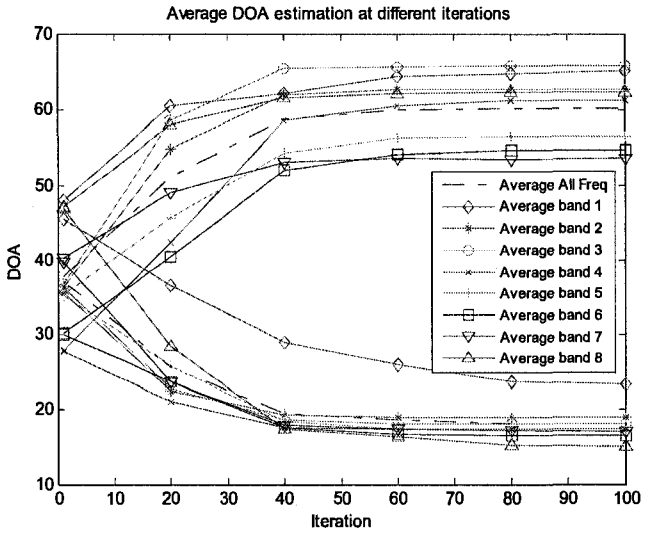


Figure 7-40: DOA estimation over different frequency band and from different iterations when frame length = 2048

From these simulation results, we can see that the algorithm converges after 40 iterations under different frame length conditions. It is also clear that the DOA estimated over a small frequency band, excluding the lowest and highest bands, is similar to that estimated over the whole frequency bins. Band 4 in particular provides DOA estimation very close to that

estimated over all bands. In the following, we compare the estimated DOA under different frame length conditions in Fig. 7.41.

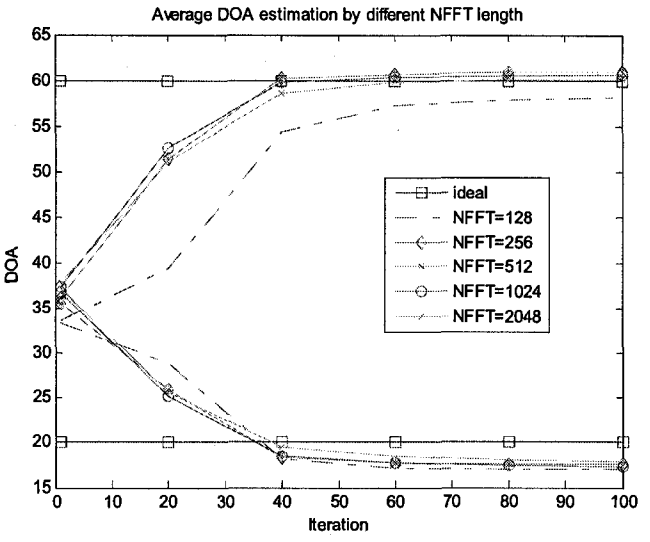


Figure 7-41: Average DOA estimation from the whole band under different frame length

Fig. 7.41 shows that the average DOA estimation from the whole frequency band using different frame lengths is independent of the frame length when the frame length is 256 or larger.

Since the DOA estimation from band 4 is very close to that estimated from the whole band, in the following, we show the DOA estimation from the whole frequency band and band 4 in detail under different frame length conditions in Fig. 7.42 to Fig. 7.46.

Next, to verify the need for estimation of DOA over all frequency bands, we compare the DOA estimation using the whole frequency bands to the DOA estimation using different frame lengths.

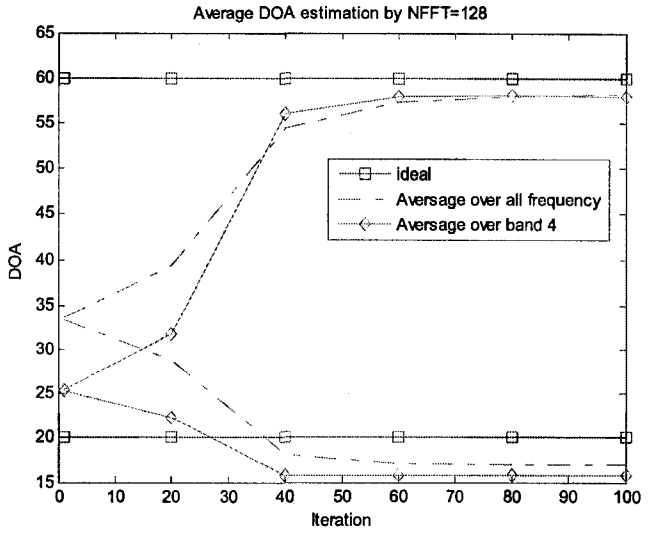


Figure 7-42: Average DOA estimation from the whole band and band 4 when frame length = 128

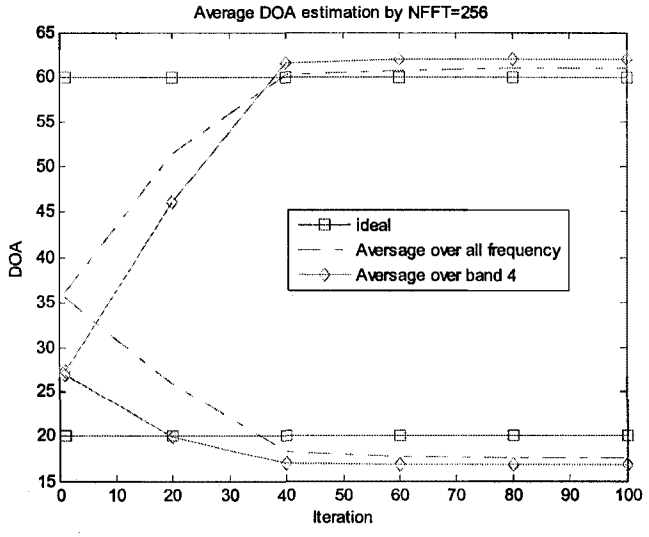


Figure 7-43: Average DOA estimation from the whole band and band 4 when frame length = 256

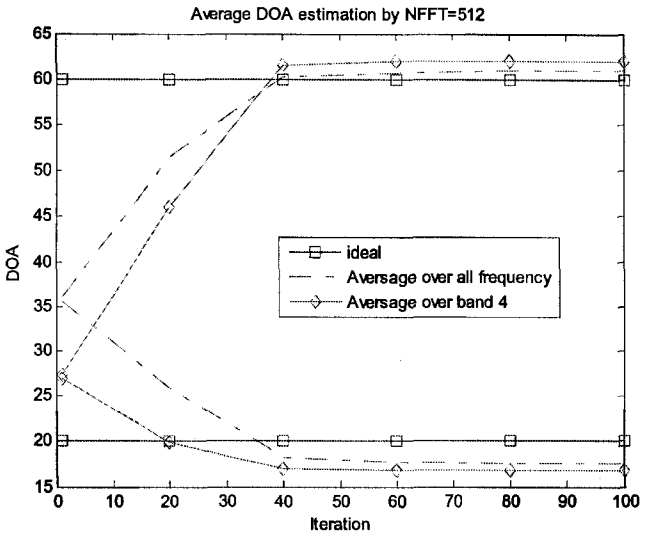


Figure 7-44: Average DOA estimation from the whole band and band 4 when frame length = 512

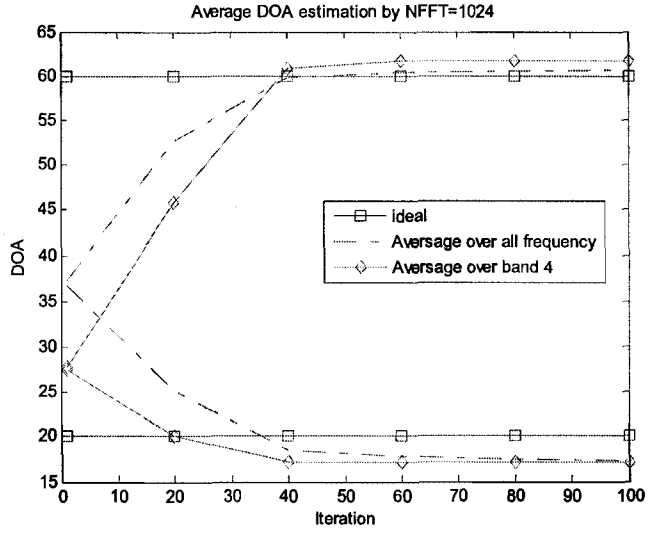


Figure 7-45: Average DOA estimation from the whole band and band 4 when frame length = 1024

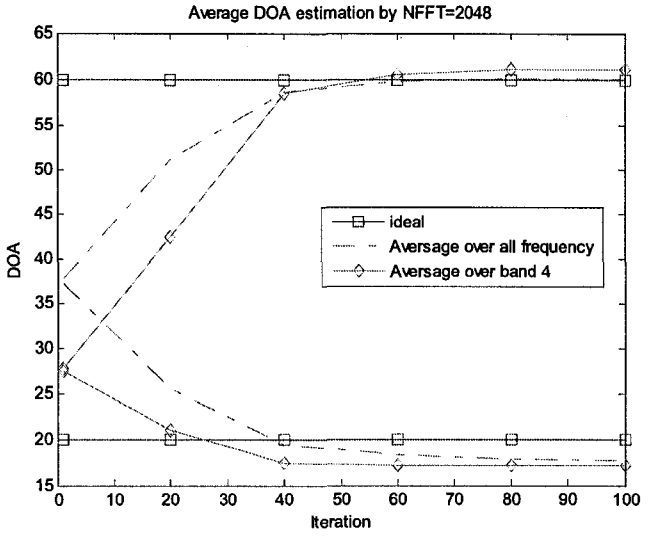


Figure 7-46: Average DOA estimation from the whole band and band 4 when frame length = 2048

From Fig. 7.42 to Fig. 7.46, we can see that band 4 provides DOA estimation that is very close to that estimated from the whole frequency band.

Conclusion from Simulation Results

The above simulations confirm the known facts that the unmixing matrix based on frequency domain BSS algorithm has the same directivity pattern as null beamformers after the algorithm converges and does indeed provide a good blind DOA estimation. The simuations also show that:

- Since DOA estimation is conducted in every frequency bin independently, it is not necessary to determine the unmixing matrix at every frequency bin to obtain DOA estimation. A subset of frequency bands should be enough.
- Not every frequency bin can offer good DOA estimations for source signals. Low frequency bands and high frequency bands do not provide good estimations.
- DOA estimation is not sensitive to frame length; the accuracy of estimated DOAs is very close under different frame length conditions.

7.6.3 Proposed new combined of beamforming and BSS system

7.6.3.1 Motivation

The performance of blind speech signal separation based on ICA is very limited in real acoustic environments with long reverberation time. Moreover, long FIR filters are needed for the unmixing system in frequency domain BSS in long reverberant environments, which results in high computational load. Beamforming provides a good choice for reducing reverberations effect in speech signal separation. The computational complexity of beamforming is low when the DOAs of source signals are available making it an attractive option.

In an attempt to achieve better separation, we reconsider how the human hearing system approaches this problem. In a cocktail party environment, when listening to someone, we first direct our ears towards the sound of a specific person, and then concentrate on separating the audio signals of different speakers in that direction. This implies that for separating multichannel audio signals, our ears first form a beamformer to concentrate on signals from a selected direction and ignore signals from other interfering directions. Then the overall hearing system acts as a blind source separation unit to separate the desired signal from other received signals in the same direction.

Our proposed system attempts to mimic the performance of a human system in a cocktail party environment. The previous Beamforming and BSS system has one serious limitation in that it is not truly blind. The resulting system also has significant complexity. In this section, we propose a new lower complexity truly blind Beamforming and BSS system. First, beamforming is used to isolate signals from specific directions (outer ear function), and then blind source separation is used to separate signals from different sources aiming in that direction (overall hearing system).

The first challenge of using beamforming in the first stage is how to blindly obtain the DOA information of sources. From our investigation in Section 4, we can see that frequency

domain BSS provides a good approach for blindly DOA estimation, thus, we will use it in the first stage to blindly estimate a rough DOAs of source signals.

7.6.3.2 The proposed two-stage combined beamforming-BSS separation system

This system includes two stages, in the first stage blind beamforming is used to obtain signal from estimated source directions and reduce reverberation effects. In the second stage, frequency domain blind source separation algorithm is exploited to further separate residual interferences and improve the separation performance.

Beamforming stage

In this stage, we first blindly estimate DOAs from the unmixing FIR matrix of a BSS system. Then, we construct beamformer based on the estimated DOA information to produce signals from the specific directions.

In [Saruwatari 2006], the DOAs of the sources are estimated from the directivity pattern of the unmixing matrix in the BSS system. The directivity pattern is obtained from the beamforming point of view and is calculated as follows.

$$[F_1(f,\theta), F_2(f,\theta)]^T = \mathbf{W}(f)e(f,\theta) \tag{7.16}$$

where $F_i(f,\theta)$ is the directivity pattern and $e(f,\theta)$ is the steering vector defined as

$$\mathrm{e}(f,\theta) = \left[\exp(j2\pi f d_1 \sin\theta / c), \exp(j2\pi f d_2 \sin\theta / c)\right]^T \tag{7.17}$$

In (7.17), d_1, d_2 are positions of the microphones and c is the propagation velocity of sound. The DOA of the i^{th} speech source at the m^{th} frequency bin is obtained by searching the null of the directivity pattern at the m^{th} frequency bin. Then the DOA of the i^{th} speech is estimated by averaging the DOA of the i^{th} speech source through all frequency bins.

There are some drawbacks for DOA estimation with this approach. First, searching the null directions from the directivity pattern is very time consuming and introduces significant computational load, especially for cases with more than two source signals. Second, the quality of DOAs from the directivity pattern at some frequencies is poor, especially at low frequency bins and at high frequency bins (the phase difference at low frequencies between microphones is very small and spatial aliasing may occur at high frequencies). Our proposed system starts out with the system in [Saruwatari 2006] and then introduces modifications to reduce its complexity without sacrificing its performance.

First, we introduce a new DOA estimation algorithm [Sawada 2004 A] into our beamforming stage for its simplicity. In [Sawada 2004 A], a closed-form formula estimating DOAs was proposed to solve the high computational load problem of the previous algorithm. The formula for estimating θ_k is

$$\theta_k = \arcsin \frac{\arg\left(\frac{\left[\mathbf{W}^{-1}\right]_{jk}}{\left[\mathbf{W}^{-1}\right]_{j'k}} \right)}{2\pi f c^{-1}\left(d_j - d_{j'}\right)} \tag{7.18}$$

From (7.18), we do not need to plot the directivity pattern for each frequency bin and then search a directivity null as the DOA estimation for that frequency bin. We simply calculate the DOA from the unmixing matrix of that frequency bin. This approach for estimating DOAs of sources offers significantly lower computational cost than the previous approach and it was proven in [Sawada 2004 A] that the θ_k calculated by this closed form formula is the same as the null direction in the directivity pattern.

Next, we obtain DOA estimation based on a subset of frequency bins (as opposed to the full frequency band). From the results in Section 4, we concluded that the quality of DOA estimation depends on the frequency bin being used. Low frequency bands and high frequency bands do not provide good estimation. Since the DOA estimation can be conducted in every frequency bin independently, we propose to estimate the DOAs based on the mid-frequency band excluding both lower and higher bands. By doing this we can

significantly reduce the computational load and improve the quality of the overall estimate. For example if we only use 1/8th of the full frequency band to get the DOA estimation, we only have 1/8 of computational load compared with the original full band estimation approach. Moreover, by using the closed-form DOA calculation instead of null directivity searching in [Sawada 2004 A] as explained earlier, the complexity is further reduced.

Finally, since the quality of DOA estimation is not sensitive to the frame length, we can use small frame length in this stage to get DOA estimations while reducing computational complexity.

After we obtained our estimated DOAs of source signal, we construct two null beamformers to get signals from these specific directions as follows.

$$W(f)=\begin{bmatrix} \exp(j2\pi f d_1 \sin\theta_1 / c) & \exp(j2\pi f d_2 \sin\theta_1 / c) \\ \exp(j2\pi f d_1 \sin\theta_2 / c) & \exp(j2\pi f d_2 \sin\theta_2 / c) \end{bmatrix}^{-1} \tag{7.19}$$

Convolutive blind source separation stage

In the convolutive blind source separation stage, an unmixing system **W** is adaptively adjusted to make the outputs as independent as possible to recover the independent source signals. The blind source separation algorithm is described in detail in Chapter 3 and will not be repeated here.

This proposed approach shared the advantage with the approach proposed in Section 7.5. However, tt is worth noting that in our previous method, we used the MUSIC algorithm to search the DOA of the source speech. This approach has two limitations. One is that it requires more microphones than speakers. The other limitation is that some prior information about the room is required. This makes the whole method not completely blind. In the current system, we use as many speakers as microphones and the DOAs of source signals are estimated completely blindly from the unmixing matrix of BSS system.

Comparing the proposed system with the combination of ICA and BF reported in existing algorithms, we note that while the purpose of the combination of ICA and BF in [Saruwatari 2002] [Saruwatari 2006] is to increase the convergence speed of BSS algorithm by temporally substituting for the BSS unmixing matrix with the matrix based on null beamforming through iterative optimization, the aim of our system is to blindly separate speech mixtures with significantly lower computational complexity.

7.6.4 Simulation Results

In our simulation, we use our proposed system to separate speech signals in a reverberant environment and show the performance of the system in this section. As before, we use PESQ scores to evaluate the separation performance.

7.6.4.1 Mixing system

The mixed signals are generated by convolving speech signals with the measured real room impulse responses. One signal is located at a DOA of 20 degree and the other one is at a DOA of 60 degree. The PESQ scores for the mixed signals compared with the original signals are shown in Table 7.18. From the Table, we can see that each mixed speech signal is almost equally similar to both sources, i.e. both source signals are heard in the mixtures. This was confirmed by informal listening test where it was difficult to identify the original speech signal.

PESQ	x1	x2
s1	1.62	1.60
s2	1.47	1.50

Table 7-18: PESQ scores for the mixtures

7.6.4.2 Beamformers as front stage of blind speech separation system

Based on the description in Section 7.6.3, we first blindly estimate the DOA information of source signals. From the estimated DOA information, we can construct a beamformer system as the front stage of our blind speech signal separation system.

In the following, we show the DOA estimations based on the approach we presented in Section 7.6.3 over different frequency bands and at different iterations with different frame lengths. Tables 7.19, 7.21, 7.23, 7.25 and 7.27 show the DOA estimation for the first source using frame lengths varying from 128 to 2048 and for different number of iterations. Tables 7.20, 7.22, 7.24, 7.26 and 7.28 provide the same results for the second source. From these tables, it is confirmed that the estimated DOAs are similar for different frame lengths and at different iteration points after the BSS algorithm converges. Also we can see that band 4 provides the closest DOA estimations to that estimated from the whole frequency band.

Iteration	All frequency	Band #1	Band #2	Band #3	Band #4	Band #5	Band #6	Band #7	Band #8
20	39.31	33.43	39.30	41.09	31.71	35.85	32.70	40.87	53.21
40	54.38	41.07	56.19	60.66	56.01	51.26	52.71	51.08	61.62
60	57.32	44.64	62.91	62.00	57.95	57.73	54.49	53.20	61.59
80	57.92	45.86	64.75	62.14	57.99	59.21	54.51	53.29	61.57
100	58.18	47.27	65.41	62.17	57.98	59.25	54.52	53.29	61.57

Table 7-19: Estimation of DOA for first source signal when frame length = 128

Iteration	All frequency	Band #1	Band #2	Band #3	Band #4	Band #5	Band #6	Band #7	Band #8
20	28.83	28.40	23.78	23.74	22.27	28.51	25.50	29.27	41.56
40	18.16	26.51	17.18	15.62	15.80	19.44	16.66	15.01	21.33
60	17.18	25.67	14.84	15.63	15.82	16.76	15.16	14.57	21.17
80	16.99	25.19	14.71	15.65	15.83	15.70	15.17	14.59	21.17
100	16.96	24.89	14.76	15.66	15.82	15.71	15.18	14.59	21.17

Table 7-20: Estimation of DOA for second source signal when frame length = 128

Iteration	All frequency	Band #1	Band #2	Band #3	Band #4	Band #5	Band #6	Band #7	Band #8
20	46.05	45.44	47.46	53.79	40.59	41.43	38.46	43.73	54.73
40	59.37	56.64	60.90	67.24	61.00	57.68	55.70	53.46	60.40
60	61.48	64.64	65.94	67.76	60.81	59.63	56.51	53.96	60.83
80	61.78	66.78	66.15	67.75	60.85	59.62	56.50	53.98	60.83
100	61.89	67.71	66.17	67.75	60.86	59.60	56.50	53.97	60.84

Table 7-21: Estimation of DOA for first source signal when frame length = 256

Iteration	All frequency	Band #1	Band #2	Band #3	Band #4	Band #5	Band #6	Band #7	Band #8
20	26.74	32.38	24.05	22.99	20.60	26.32	23.44	28.10	32.11
40	18.49	28.90	18.19	16.88	16.39	16.28	15.47	15.46	15.85
60	17.53	24.37	16.02	16.51	16.41	16.08	15.44	15.41	15.49
80	17.30	22.68	15.86	16.48	16.41	16.07	15.44	15.41	15.49
100	17.20	21.93	15.86	16.47	16.42	16.06	15.43	15.41	15.50

Table 7-22: Estimation of DOA for second source signal when frame length = 256

Iteration	All frequency	Band #1	Band #2	Band #3	Band #4	Band #5	Band #6	Band #7	Band #8
20	51.46	56.37	51.53	61.52	46.10	47.57	42.18	47.42	57.84
40	60.30	60.90	64.19	67.15	61.59	55.96	55.35	54.04	62.26
60	60.64	62.64	64.33	67.16	61.93	56.76	55.67	53.63	62.08
80	60.93	64.64	64.34	67.17	61.94	57.06	55.68	53.62	62.08
100	61.03	65.42	64.34	67.17	61.94	57.05	55.68	53.62	62.08

Table 7-23: Estimation of DOA for first source signal when frame length = 512

Iteration	All frequency	Band #1	Band #2	Band #3	Band #4	Band #5	Band #6	Band #7	Band #8
20	25.87	33.81	23.68	20.23	19.89	22.99	22.42	24.38	37.56
40	18.29	25.87	17.55	17.36	17.01	17.18	16.29	16.38	16.40
60	17.76	23.18	17.47	17.33	16.82	16.55	15.82	16.26	16.34
80	17.61	22.06	17.47	17.32	16.82	16.51	15.83	16.26	16.34
100	17.54	21.49	17.47	17.30	16.82	16.51	15.83	16.26	16.34

Table 7-24: Estimation of DOA for second source signal when frame length = 512

Iteration	All frequency	Band #1	Band #2	Band #3	Band #4	Band #5	Band #6	Band #7	Band #8
20	52.66	58.85	55.80	62.92	45.81	47.41	42.47	49.15	58.28
40	59.83	62.84	62.64	65.53	60.85	55.75	54.45	53.44	62.65
60	60.33	64.64	62.72	65.54	61.78	56.30	54.91	53.42	62.89
80	60.49	65.99	62.72	65.54	61.78	56.31	54.92	53.42	62.74
100	60.65	67.43	62.72	65.54	61.78	56.31	54.92	53.42	62.64

Table 7-25: Estimation of DOA for first source signal when frame length = 1024

Iteration	All frequency	Band #1	Band #2	Band #3	Band #4	Band #5	Band #6	Band #7	Band #8
20	25.09	35.03	22.15	20.59	20.07	23.50	22.65	23.49	32.27
40	18.44	25.42	18.84	17.92	17.18	17.34	16.74	16.90	16.01
60	17.71	21.18	18.49	17.92	17.10	16.78	16.04	16.78	16.28
80	17.42	19.10	18.48	17.91	17.10	16.78	16.05	16.78	16.03
100	17.30	18.12	18.48	17.91	17.10	16.78	16.05	16.78	16.06

Table 7-26: Estimation of DOA for second source signal when frame length = 1024

Iteration	All frequency	Band #1	Band #2	Band #3	Band #4	Band #5	Band #6	Band #7	Band #8
20	51.21	60.52	54.83	58.49	42.36	45.77	40.48	49.00	57.92
40	58.62	62.02	61.93	65.31	58.51	54.32	52.10	53.05	61.50
60	59.87	64.22	62.47	65.58	60.49	56.22	54.17	53.54	62.03
80	60.10	64.68	62.48	65.64	61.12	56.43	54.59	53.44	62.17
100	60.16	65.01	62.48	65.63	61.13	56.45	54.61	53.54	62.21

Table 7-27: Estimation of DOA for first source signal when frame length = 2048

Iteration	All frequency	Band #1	Band #2	Band #3	Band #4	Band #5	Band #6	Band #7	Band #8
20	25.54	36.50	22.43	22.81	21.00	25.64	23.53	23.62	28.31
40	19.45	28.77	19.14	18.56	17.51	18.35	17.53	17.86	17.30
60	18.49	25.95	18.94	17.92	17.28	17.26	16.53	17.24	16.21
80	18.00	23.65	18.94	17.93	17.33	17.03	16.37	17.03	15.14
100	17.90	23.27	18.94	17.92	17.34	16.99	16.35	16.89	14.93

Table 7-28: Estimation of DOA for second source signal when frame length = 2048

This confirms that we can reduce the computational complexity of the beamforming system by using a smaller frame length and getting DOA estimation over a subset of the whole frequency band while keeping the speech quality of output signals from the beamforming system.

After obtaining the estimated DOA of the source signals, we can construct the corresponding beamformers to extract signals from these directions. In the following, we use PESQ scores to evaluate the quality of output speech signals from the beamforming system based on the DOA estimations from different iteration points under different frame length conditions. Since band 4 provides closest DOA estimation to that estimated using the whole frequency band, we compare the results for both band #4 and full band estimates. Tables 7.29, 7.31, 7.33, 7.35 and 7.37 provide PESQ scores for outputs from a beamforming system constructed based on the DOA estimation over all frequencies at different iterations with frame length = 128, 256, 512, 1024 and 2048 respectively. . Tables 7.30, 7.32, 7.34, 7.36 and 7.38 provide PESQ scores for outputs from a beamforming system constructed based on the DOA estimation over band #4 at different iterations with frame length = 128, 256, 512, 1024 and 2048 respectively.

PESQ	40 iterations		60 iterations		80 iterations		100 iterations	
	out1	out2	out1	out2	out1	out2	out1	out2
s1	2.16	0.89	2.18	0.86	2.18	0.87	2.18	0.86
s2	0.62	1.76	0.64	1.78	0.65	1.78	0.66	1.78

Table 7-29: PESQ scores for outputs from beamforming system which is constructed based on the DOA estimation over all frequencies at different iterations with frame length = 128

PESQ	40 iterations		60 iterations		80 iterations		100 iterations	
	out1	out2	out1	out2	out1	out2	out1	out2
s1	2.20	0.89	2.20	0.85	2.20	0.82	2.20	0.85
s2	0.62	1.77	0.62	1.78	0.62	1.78	0.62	1.78

Table 7-30: PESQ scores for outputs from beamforming system which is constructed based on the DOA estimation over band 4 at different iterations with frame length = 128

PESQ	40 iterations		60 iterations		80 iterations		100 iterations	
	out1	out2	out1	out2	out1	out2	out1	out2
s1	2.17	0.96	2.18	0.86	2.19	0.82	2.19	0.87
s2	0.65	1.77	0.65	1.77	0.66	1.77	0.65	1.78

Table 7-31: PESQ scores for outputs from beamforming system which is constructed based on the DOA estimation over all frequencies at different iterations with frame length = 256

PESQ	40 iterations		60 iterations		80 iterations		100 iterations	
	out1	out2	out1	out2	out1	out2	out1	out2
s1	2.20	0.87	2.20	0.86	2.20	0.85	2.20	0.85
s2	0.60	1.77	0.62	1.77	0.62	1.77	0.62	1.77

Table 7-32: PESQ scores for outputs from beamforming system which is constructed based on the DOA estimation over band 4 at different iterations with frame length = 256

PESQ	40 iterations		60 iterations		80 iterations		100 iterations	
	out1	out2	out1	out2	out1	out2	out1	out2
s1	2.17	0.87	2.18	0.85	2.18	0.84	2.18	0.88
s2	0.66	1.76	0.64	1.76	0.63	1.76	0.68	1.76

Table 7-33: PESQ scores for outputs from beamforming system which is constructed based on the DOA estimation over all frequencies at different iterations with frame length = 512

PESQ	40 iterations		60 iterations		80 iterations		100 iterations	
	out1	out2	out1	out2	out1	out2	out1	out2
s1	2.19	0.85	2.20	0.86	2.20	0.86	2.20	0.86
s2	0.60	1.76	0.63	1.76	0.63	1.76	0.63	1.76

Table 7-34: PESQ scores for outputs from beamforming system which is constructed based on the DOA estimation over band 4 at different iterations with frame length = 512

PESQ	40 iterations		60 iterations		80 iterations		100 iterations	
	out1	out2	out1	out2	out1	out2	out1	out2
s1	2.17	0.84	2.18	0.87	2.19	0.85	2.19	0.85
s2	0.64	1.75	0.65	1.76	0.64	1.76	0.67	1.76

Table 7-35: PESQ scores for outputs from beamforming system which is constructed based on the DOA estimation over all frequencies at different iterations with frame length = 1024

PESQ	40 iterations		60 iterations		80 iterations		100 iterations	
	out1	out2	out1	out2	out1	out2	out1	out2
s1	2.19	0.82	2.19	0.85	2.19	0.85	2.19	0.85
s2	0.64	1.76	0.68	1.76	0.68	1.76	0.68	1.76

Table 7-36: PESQ scores for outputs from beamforming system which is constructed based on the DOA estimation over band 4 at different iterations with frame length = 1024

PESQ	40 iterations		60 iterations		80 iterations		100 iterations	
	out1	out2	out1	out2	out1	out2	out1	out2
s1	2.15	0.81	2.17	0.84	2.18	0.79	2.18	0.82
s2	0.71	1.75	0.64	1.75	0.63	1.76	0.63	1.76

Table 7-37: PESQ scores for outputs from beamforming system which is constructed based on the DOA estimation over all frequencies at different iterations with frame length = 2048

PESQ	40 iterations		60 iterations		80 iterations		100 iterations	
	out1	out2	out1	out2	out1	out2	out1	out2
s1	2.19	0.85	2.19	0.83	2.19	0.89	2.19	0.90
s2	0.62	1.75	0.68	1.76	0.69	1.76	0.69	1.76

Table 7-38: PESQ scores for outputs from beamforming system which is constructed based on the DOA estimation over band 4 at different iterations with frame length = 2048

From these tables, we confirm that the speech quality are very similar under different frame length whether band 4 or the full frequency band is used.

This confirms that we can reduce the computational complexity of the beamforming system by using a smaller frame length and getting DOA estimation over a subset of the whole frequency band while keeping the speech quality of output signals from the beamforming system.

To see the effect of DOA estimation on the performance of our proposed system, we run simulations for several scenarios of DOA estimation varying from a perfect estimate (20, 60) to others with varying errors: (16, 52), (17, 62), (22, 57), some being worse than that we get from over estimation algorithm. The names of the resulting speech files are described in Table 7.39 and their PESQ scores compared with source signals are shown in Table 7.40. Comparing with the PESQ of the mixed signals as given in Table 7.18, we can see that the beamforming stage did provide improvement in all cases. Still, it is clear that the improvement varies with the DOA used. Note that, the signal output from this stage will be further processed by the BSS stage.

Group No.	Ang (degree)	output file
1	20	bf11
	60	bf12
2	16	bf21
	52	bf22
3	17	bf31
	62	bf32
4	22	bf41
	57	bf42

Table 7-39: Speech files description from 4 scenarios for beamforming stage

PESQ	bf11	bf12	bf21	bf22	bf31	bf32	bf41	bf42
s1	1.75	0.64	1.69	0.56	1.75	0.62	1.73	0.68
s2	0.80	2.14	1.27	2.20	0.90	2.19	1.31	2.09

Table 7-40: PESQ scores of beamformer output signals for 4 scenarios

7.6.4.3 BSS as the second stage of blind speech signal separation system

In this stage, we use the following parameters for our simulations.

Input signal: outputs from beamforming stage;
Sampling frequency: 8000Hz;
Frame size of FFT: 128;
Frame shift: 16;
Step size: 0.0001

It should be noted that the frame size of FFT is much smaller than that used in [Saruwatari 2006]. That is because the reverberant effects have been already reduced by the beamforming stage. From this point of view, the computational complexity is reduced again.

The output from the beamforming stage is used as the input to the BSS stage. The names of the output speech files are described in Table 7.41 and their PESQ scores compared with source signals are shown in Table 7.42.

Group No.	input file	output file
1	bf11	fbs_11
	bf12	fbs_12
2	bf21	fbs_21
	bf22	fbs_22
3	bf31	fbs_31
	bf32	fbs_32
4	bf41	fbs_41
	bf42	fbs_42

Table 7-41: Speech files description from 4 scenarios for BSS stage

PESQ	fbs_11	fbs_12	fbs_21	fbs_22	fbs_31	fbs_32	fbs_41	fbs_42
s1	2.12	0.16	2.09	0.22	2.13	0.35	2.11	0.26
s2	0.60	2.31	0.56	2.34	0.58	2.30	0.61	2.33

Table 7-42: PESQ scores of BSS output signals for 4 scenarios

From Table 7.42, we can see that the BSS stage further improves the separated speech signal quality by making the signal more biased to one source signal and away from the another signal.

To quantify the separation improvement in these two stages, we use the improvement in PESQ as an evaluation tool. Since the goal of the separation process is to force the output signal away from one source signal and close to the other source signal, we define the PESQ improvement as the sum of the PESQ score away from one source and PESQ score bias to the other source. For example, consider a signal x as the estimation of source s_1. Let its PESQ score compared to source signal s_1 be px_1 and compared to source signal s_2 be px_2. After processing, we get a new signal y, and its PESQ score compared to source signal s_1 is py_1, with source signal s_2 is py_2. Thus, the PESQ improvement from signal x to y compared to source signal s_1 is $py_1 - px_1$; compared to source signal s_2 is $px_2 - py_2$. The total PESQ improvement is $(py_1 - px_1) + (px_2 - py_2)$.

We now quantify the improvement provided by each stage independently. Based on Table 7.40, the PESQ improvement obtained from outputs of the beamforming stage is given in Table 7.43.

PESQ imp	bf11	bf12	bf21	bf22	bf31	bf32	bf41	bf42
s1	0.28	0.86	0.22	0.94	0.28	0.88	0.26	0.82
s2	0.82	0.54	0.34	0.61	0.72	0.60	0.31	0.49
Total	1.09	1.41	0.56	1.55	1.00	1.47	0.57	1.31

Table 7-43: PESQ improvement for beamforming stage for 4 scenarios

In the BSS stage, we process the outputs from beamforming stage and obtained the separated signals after BSS algorithm. Compared with the outputs of beamforming stage, the PESQ improvement obtained from outputs of the BSS stage is as follows in Table 7.44.

PESQ_imp	fbs_11	fbs_12	fbs_21	fbs_22	fbs_31	fbs_32	fbs_41	fbs_42
s1	0.37	0.47	0.40	0.34	0.37	0.27	0.38	0.42
s2	0.20	0.16	0.72	0.14	0.32	0.11	0.70	0.23
Total	0.58	0.64	1.12	0.48	0.69	0.38	1.07	0.66

Table 7-44: PESQ improvement for BSS stage for 4 scenarios

From the simulation results, we can see that the proposed system works very well for speech separation in the real acoustic environment with reduced computational complexity and flexible system structure. And they are also consistent with our informal listening test.

Discussion:

Compared with the system in [Saruwatari 2002][Saruwatari 2006], the proposed system divides the task of separating mixed speech signals in heavy reverberant environments into two sub-tasks. The first task is to roughly reduce the reverberant effects. This is achieved by beamforming through a rough estimation of source directions. In the second task, the BSS system removes the interface signals in that direction. Based on the de-reverberant mixture signals, the proposed BSS stage can work well even with much smaller filter lengths to

perform its separation task. Furthermore, the frequency permutation problem becomes easier to deal with under lighter reverberant input signals. In [Saruwatari 2002] [Saruwatari 2006], the estimation of directions of sources, separation and frequency permutation are all combined inside one system, so the requirement for the accuracy of DOA estimation is high. Since the frequency permutation is worse under high reverberant environment, errors in frequency permutation have significant impact when using smaller frame length. That it why a large frame length is necessary for this system and thus increasing its computational complexity.

7.6.4.4 Conclusion for this section

In this Section, we investigate the properties of the unmixing matrix in frequency domain BSS in detail. Based on its properties, we propose a new approach combining independent component analysis and beamforming for blind speech signal separation in real acoustic environment. By mimicking human hearing system, our separation system is constructed as beamformers cascaded with a BSS system. The beamformers are used in the front stage to orient the system to adjust our system to relevant source direction followed by a BSS system to further reduce the interference in that direction and improve the separation performance. Compared with existing systems, the proposed approach significantly reduces the computational complexity and maintains the separation performance.

7.7 Conclusion

In this chapter we investigate approaches combining spatial information used in beamforming with time/frequency processing used in convolutive blind source separation aiming for better separation performance given the increased information used. We first carefully compare similarities and differences of BSS and beamforming and reviewing existing combination approaches in the literature. Then we present our proposed first combination method which includes an adaptive beamforming in the front cascading a BSS system to mimic the way our ears use to separate audio signal in acoustic environments. Simulation results confirm our expectations and show that our system

works pretty well in real room environment. The challenge for this combined approach is that some prior room information is needed for DOA estimation. By investigating the properties of the unmixing matrix in frequency domain BSS in detail, we proposed a new approach combining independent component analysis and beamforming for blind speech signal separation in real acoustic environment. In the beamforming stage, the DOAs of selected sources are estimated blindly; then beamformers are constructed to extract signals from these directions. In the BSS stage, frequency domain convolutive algorithm is utilized to further reduce the interference in the given direction and improve the separation performance. Compared with existing systems, our approach significantly reduces the computational complexity while keeping similar separation performance.

Chapter 8 Time Domain Convolutive Blind Source Separation Employing Selective-tap Adaptive Algorithms

In this chapter, we investigate novel algorithms to improve the convergence and reduce the complexity of time domain convolutive BSS algorithm. First, we propose the application of MMax partial update algorithm [Aboulnasr 1999] to the time domain convolutive BSS (MMax BSS). We demonstrate that the partial update scheme applied in the MMax LMS algorithm for single channel can be extended to multichannel time domain convolutive BSS with little deterioration in performance and possible computation complexity saving. Next, we propose exclusive maximum selective-tap time domain convolutive BSS algorithm (XMax BSS) that reduces the interchannel coherence of the tap-input vectors and improves the conditioning of the autocorrelation matrix resulting in improved convergence rate and reduced misalignment. Moreover, the computational complexity is reduced since only half tap inputs are selected for updating. Simulation results have shown a significant improvement in convergence rate compared to existing techniques.

8.1 Introduction

Blind source separation (BSS) [Haykin 2000][Cichocki 2000] is an established area of work estimating source signals based on information about observed mixed signals at sensors, i.e., the estimation is performed without exploiting information about either the source signals or the mixing system. Independent Component Analysis (ICA) [Hyvarinen 2001] is the main statistical tool for dealing with the BSS problem with the assumption that the source signals are mutually independent. In the instantaneous BSS case, signals are mixed instantaneously and ICA algorithms can be directly employed to separate the mixtures. However, in a realistic environment, signals are always mixed in convolutive manner because of propagation delay and reverberation effects. Therefore, much research deals with convolutive blind source separation based on extending instantaneous blind source separation or independent component analysis to convolutive case.

The straightforward choice in time domain convolutive blind source separation is based on directly extending instantaneous BSS to the convolutive case [Amari 1997][Douglas 2003]. This approach sounds great theoretically and achieves good separation results once the algorithm converges. However, time domain convolutive blind source separation suffers from high computational complexity and low convergence rate, especially for systems requiring long FIR filters for the separation.

Frequency domain convolutive BSS [Smaragdis 1998][Parra 2000] was proposed to deal with the expensive computational complexity problem of time domain BSS. In frequency domain BSS, complex-valued ICA for instantaneous BSS is employed in every frequency bin independently. The advantage of this approach is that any existing complex-value instantaneous BSS algorithm can be used and the computational complexity is reduced by exploiting the FFT for convolution computation. That is the basis of popularity of frequency domain approaches. However, the permutation and scaling ambiguity in ICA algorithm, which is not a problem for instantaneous BSS, becomes a serious problem in frequency domain convolutive BSS. Since frequency domain convolutive BSS is performed by instantaneous BSS at each frequency bin separately, the order and the scale of the unmixed signals are random because of the inherent ambiguity of ICA algorithms. When we transform the separated signals back from frequency domain to time domain, the components at different frequency bins may not come from the same source signal and may not have a consistent scale factor. Thus, we need to align these components and adjust the scale in each frequency bin so that a separated signal in time domain is obtained from frequency components of the same source signal and with consistent amplitude. This is well-known as the permutation and scaling problem of frequency domain convolutive BSS. Many approaches were proposed in the literature to address the complex permutation problem of frequency domain BSS [Sawada 2004][Ikram 2002]. These built-in problems in frequency domain approaches make it worthwhile to reconsider ways of reducing the complexity of time domain approaches and improving their convergence rates.

In recent years, several partial update adaptive algorithms were proposed to model single channel systems with reduced overall system complexity by only updating a part of

coefficients. Within these partial update algorithm, the MMax NLMS in [Aboulnasr 1999] was reported to have the closest performance to the full update case for any given number of coefficients to be updated. In [Khong 2006], the MMax selective-tap strategy was extended to two channel case to exclusively select maximum coefficients as a means to reduce interchannel coherence in stereophonic acoustic echo cancellation rather than as a way to reduce complexity. Simulation results for this exclusive maximum adaptive algorithm showed that it can significantly improve the convergence rate compared with existing stereophonic echo cancellation techniques.

In this chapter, we propose using these approaches for BSS in the time domain to address complexity and slow convergence problems. First, we propose MMax natural gradient-based partial update time domain convolutive BSS algorithm (MMax BSS). In this algorithm, only a subset of coefficients in the separation system gets updated at every iteration. We demonstrate that the partial update scheme applied in the MMax LMS algorithm for a single channel can be extended to the multichannel time domain convolutive BSS with little deterioration in performance and possible computational complexity saving. By employing selective tap strategies used for stereophonic acoustic echo cancellation [Khong 2006], we propose an exclusive maximum selective-tap time domain convolutive BSS algorithm (XMax BSS). The exclusive tap-selection update procedure reduces the interchannel coherence of the tap-input vectors and improves the conditioning of the autocorrelation matrix so as to accelerate convergence rate and reduce the misalignment. The computational complexity is reduced as well since only half of the input taps are selected for updating (note that little overhead is needed to select the set to be updated). Simulation results have shown a significant improvement in convergence rate compared with existing techniques. As far as we know, the application of partial update and selective-tap update schemes to time domain natural gradient based BSS algorithms is in itself novel.

The rest of this chapter is organized as follows. In Section 8.2, we review the single channel MMax partial update adaptive algorithm for linear filters and nonlinear filters. In Section 8.3, we review exclusive maximum selective-tap adaptive algorithm for stereophonic echo cancellation. We propose our MMax partial update time domain convolutive BSS algorithm

in Section 8.4 and the exclusive maximum update time domain convolutive BSS algorithm in Section 8.5. We present simulation results of the proposed algorithms for generated Gamma signals and speech signals in Section 8.6. In Section 8.7, we draw our conclusions from our work.

8.2 Partial Update Adaptive Algorithms

In this section, we review partial update adaptive algorithms with special emphasis on data-dependant algorithms. As a side point, we then demonstrate that the same approach applied in the MMax LMS partial update algorithm for linear adaptive filters [Aboulnasr 1999] can be extended to the class of nonlinear filters known as Volterra filters. The impact of the fact that the input vector is no longer a set of delayed input values on the complexity reduction due to the partial update is noted. Simulation results show that, as for linear filters, considerable saving is possible with little deterioration in performance.

8.2.1 Introduction

Adaptive filters have been used routinely to model unknown, possibly time-varying systems. In many cases, the number of parameters used is prohibitive limiting the practical application of powerful algorithms because of the complexity of updating a large number of coefficients at the same time. Partial update algorithms attempt to address this issue by limiting the number of coefficients being updated in a given iteration. The selection of which coefficients to update is critical in determining the performance of the resulting algorithm. Initially, this selection was done on a preset, rotating basis [Douglas 1997]. These algorithms are simple to implement but invariably lead to significantly lower convergence rates given the arbitrary nature of choosing the subset of coefficients to update.

In this section, we will concentrate on algorithms that select the subset of coefficients to be updated based on some criterion so as to reduce the performance deterioration due to slower update of coefficients. This implies a dynamic determination of the coefficients to update

and allows for selecting these coefficients to minimize the impact of not updating the full set of system parameters.

In Section 8.3.2, we review the fundamentals of partial update algorithms. In Section 8.3.3, we consider algorithms where the set of coefficients to be updated is dynamically determined at every iteration based on the received data. In Section 8.3.4, we highlight some of the variants of the data-dependent partial update algorithms. Section 8.3.5 presents possible extensions of the partial update concept to nonlinear Volterra filters with preliminary results confirming good performance for partial update.

8.2.2 Fundamental Partial Update algorithms

Consider the standard adaptive filter set-up where $x(n)$ is the input, $y(n)$ is the output and $d(n)$ is the desired output, all at instant n. The output error $e(n)$ is given by

$$e(n) = d(n) - y(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n) \tag{8.1}$$

where $\mathbf{w}(n)$ is the $L\times 1$ column vector of the filter coefficients and $\mathbf{x}(n)$ is the $L\times 1$ column vector of the current and past inputs to the filter, both at instant n. The i^{th} element of $\mathbf{w}(n)$ is $w_i(n)$ and it multiplies the i^{th} delayed input $\mathbf{x}(n)$, $i = 0,\cdots,L-1$.

The basic NLMS algorithm is known for its extreme simplicity providing for coefficient update as given by:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\frac{\mathbf{x}(n)}{\|\mathbf{x}(n)\|^2} \tag{8.2}$$

where μ is the step size determining the speed of convergence and the steady state error. The complexity of implementing such an adaptive filter is effectively $2L$ multiply/add with L

operations needed for the update of the L coefficients and another L operations needed for the calculation of the output $y(n)$.

The basic idea of partial update adaptive filtering is to allow for the use of filters with a number of coefficients L large enough to model the unknown system while reducing the overall complexity by updating only M coefficients at a time. This results in considerable savings for $M << L$. Invariably, there are penalties for this partial update, the most obvious of which being reduced convergence rate. The question then becomes which coefficients should we update and how do we minimize the impact of the partial update on the overall filter performance.

Early attempts at partial update of the coefficients simply divided the coefficients into sets that were selected either sequentially by dividing the coefficient vector into blocks of length M and updating one block every iteration in a sequential form or by updating every M^{th} coefficient, again in order [Douglas 1997]. There is minimal additional overhead in implementing this selective update and the savings are proportional to the ratio of M/L. Since each set of coefficients will be updated every M/L iterations, the more the savings, the lower the performance of the algorithm. It is inevitable that the performance deteriorates considerably since the available information about the system dynamics are not used at all in identifying which coefficients can result in the most error reduction and as such need to be updated.

In [Godavarti 2000], it was proposed to update the blocks in a random order (as opposed to sequentially). It was shown that while this setup has performance comparable to the periodic partial update, it does result in faster convergence for some deterministic signals in the context of adaptive beamforming.

8.2.3 Data-Dependant Partial Update algorithms

While the above algorithms reduce the complexity, the price paid in convergence rate may not be tolerated, particularly for LMS algorithms in acoustic environments where the

convergence speed is not fast to start with. This led to other approaches where the set of coefficients to be updated is not predetermined, rather is selected to maximize the performance of the system in some sense.

In [Douglas 1994], the max-NLMS presented an algorithm were only one coefficient is updated in every iteration (using a slightly modified update equation). This coefficient is selected as the one multiplying the input with the largest absolute value. While this algorithm provided considerable saving in complexity, it was shown to diverge for some data sets.

In [Aboulnasr 1999], the set of M coefficients to be updated is selected as the one that provides the maximum reduction in error. It is shown that this criterion reduces to the set of coefficients multiplying inputs $x(n-i)$ with the largest magnitude using the standard NLMS update equation. This selective-tap updating can be expressed as

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu\mathbf{Q}(n)e(n)\frac{\mathbf{x}(n)}{\|\mathbf{x}(n)\|^2} \tag{8.3}$$

where $\mathbf{Q}(n)$ is the tap-selection matrix as

$$\mathbf{Q}(n)=diag\{\mathbf{q}(n)\} \tag{8.4}$$

$$q_i(n)=\begin{cases}1, & |x_i(n)|\in\{M \text{ maxima of } |\mathbf{x}(n)|\ \}\\ 0, & \text{otherwise}\end{cases} \tag{8.5}$$

An analysis of the mean square error convergence is provided in [Aboulnasr 1999] based on matrix formulation of data-dependent partial updates. Based on the analysis, it was shown that the MMax algorithm provides the closest performance to the full update case for any given number of coefficients to be updated. This was confirmed in [Werner 2004].

Theoretically, the determination of this set of coefficients needs to be done every iteration through a sorting algorithm. However, the complexity is not significant given the fact that the input vector $\mathbf{x}(n)$ is a time series. Once the full set of input samples is sorted as the

samples arrive one after the other, a new iteration results in dropping the oldest sample and deciding where to insert the newest sample in the already-sorted set.

In [Naylor 2003], it was proposed that a "short-sort" algorithm be used to further reduce the overhead of the M-Max algorithm needed for sorting. The impulse response is divided into two regions. For the first region where the bulk of the energy of the response exists, all coefficients are updated in each iteration. For the second region where the coefficients are likely to be very small, the M-Max partial update algorithm is used. Given the significantly smaller size of this set of coefficients, sorting overhead is reduced.

In [Doğançay 2001], the partial update algorithm is formulated as a constrained optimization problem along the lines of the NLMS leading to a common framework incorporating several existing algorithms.

8.2.4 Variations of data-Dependent partial Update Algorithms

Even when the set of coefficients to be updated is predetermined, improved performance with reduced coefficient update can still be achieved if prior information regarding the nature of the response is utilized. For example, in [Abousaada 1992] the system response is decomposed into two stages. The first stage representing an arbitrary main response receives full update. The second stage is an up-sampled adaptive filter, where nonzero coefficients are updated every iteration, followed by a fixed lowpass filter to perform the interpolation between the samples. This can be seen as a variant on the concept of partial update.

In [Deng 2004], prior knowledge of the fact that the system response is sparse with large non-zero samples concentrated in the same region was used to speed up initial convergence by weighting the input vector with an estimate of the channel response. The coefficients are all updated initially to enable the system to differentiate between large and small values. Following this initial convergence, large coefficients are updated every iteration to speed up their convergence while small coefficients (who will likely stay small) are updated based on some partial update algorithm. Finally, once convergence is achieved, partial update is used

for all coefficients. It was shown that the performance of this selective update algorithm is roughly equivalent to that of the full update at reduced complexity.

The concept of partial update has also been applied to domains other than the time domain and to algorithms other than the LMS. In [Mayyas 2004], the M-Max algorithm was used in a decomposed transform domain to reduce the overall complexity showing very good performance. In [Attallah 2001], it was used in the DCT domain resulting in performance comparable to the full update case. In [Doğançay 2002], selective update was proposed in the subband domain showing strong performance with speech signals while [Doğançay 2001] and [Werner 2001] successfully applied partial update to the Affine Projection algorithm.

In [Werner 2004], the savings of partial update of LMS were integrated in set membership filters where complexity is reduced by allowing coefficients to vary within a feasible set providing further reduction in complexity at minimal deterioration in performance.

8.2.5 Partial Update algorithms for nonlinear Volterra filters [Aboulnasr 2005]

Volterra filters provide a mathematically tractable model for nonlinear systems to which much of the literature developed for linear systems has been extended [Mathews 2001]. The output is expressed in polynomial form as the sum of a linear component and higher order products of the input.

Consider the output of a third order Volterra system given by:

$$y(n)=\sum_{k=0}^{N_1-1} h_1(k;n)x(n-k)+\sum_{k_1=0}^{N_2-1}\sum_{k_2=0}^{N_2-1-k_1} h_2(k_1,k_2;n)x(n-k_1)x(n-k_2) + \sum_{k_1=0}^{N_3-1}\sum_{k_2=0}^{N_3-1-k_1}\sum_{k_3=0}^{N_3-1-k_1-k_2} h_3(k_1,k_2,k_3;n)x(n-k_1)x(n-k_2)x(n-k_3) \tag{8.6}$$

where N_1 is the filter length for the linear part, N_2 is the memory depth for the second-order part and N_3 is the memory depth for the third-order part; $h_1(k,n), h_2(k_1,k_2;n)$ and

$h_3(k_1,k_2,k_3;n)$ are the linear, second-order and third-order coefficients of the adaptive filter at time n respectively. There are N_1 coefficients $h_1(k,n)$, $N_2(N_2+1)/2$ coefficients $h_2(k_1,k_2;n)$ and $N_3(N_3+1)(N_3+2)/6$ coefficients $h_3(k_1,k_2,k_3;n)$. In the following, we represent $h_1(k,n), h_2(k_1,k_2;n)$ and $h_3(k_1,k_2,k_3;n)$ by the vectors

$$H_1(n)=\left[h_1(0;n),\cdots,h_1(N_1-1;n)\right] \quad ;$$

$$H_2(n)=\left[h_2(0;n),\cdots,h_2\left(N_2(N_2+1)/2-1;n\right)\right]$$

$$H_3(n)=\left[h_3(0;n),\cdots,h_3\left(N_3(N_3+1)(N_3+2)/6-1;n\right)\right].$$

The LMS algorithm for the Volterra filter is described as follows.

Coefficient vector: $[H_1(n);H_2(n);H_3(n)]$

Input vector:

$$X_1(n)=\left[x(n),x(n-1),\cdots,x(n-N_1+1)\right]$$

$$X_2(n)=\left[x^2(n),x(n)x(n-1),\cdots,x^2(n-N_2+1)\right]$$

$$X_3(n)=\left[x^3(n),x^2(n)x(n-1),\cdots,x^3(n-N_3+1)\right]$$

Initialization: Arbitrarily choose $H_1(n);H_2(n);H_3(n)$

Main iteration:

$$e(n)=d(n)-\sum_{i=1}^{3}H_i(n)X_i^T(n)$$

$$H_i(n+1)=H_i(n)+\mu_i e(n)X_i(n);\ \ i=1,2,3$$

In this section, we consider the extension of the MMax algorithm [Aboulnasr 1999] to the class of Volterra filters. Thus, only the coefficients multiplying the largest P% input values will be updated, where P=L/N*100%. It should be noted that in this case, "input values" refers to the elements of the vector [X] = [X1, X2, X3]. The main challenge compared to the linear case is that the input vector is no longer a set of shifted values. As such, to determine the P% largest values, we will need a full sorting of the elements of X in every iteration. To reduce this additional complexity, sorted time series of sub-lists X1, X2, X3 are maintained separately through merging of even smaller sorted lists. To avoid resorting the whole list, there are two alternatives to selecting the coefficient set to update. In the first approach (option 1), these sorted sub-lists are merged in every iteration into one large sorted set (at some complexity lower than a full resort) and the coefficients multiplying the P% largest values are updated. In option 2, the P% largest coefficients of every sorted sub-list X1, X2,

X3 are updated in every iteration. Detailed complexity and performance analysis for partial update Volterra filters is being conducted.

Simulation Results

In our simulation, we set N_1=10, N_2=4 and N_3=3. The coefficients for the unknown system are set as follows.

$$H_1 = [0.85, 0.8, 0.9, 0.7, 0.6, 0.75, 0.65, 0.8, 0.6, 0.8]$$
$$H_2 = [0.5, 0.3, 0.2, 0.2, 0.3, 0.1, 0.15, 0.25, 0.05, 0.25]$$
$$H_3 = [0.1, 0.3, 0.3, 0.2, 0.4, 0.3, 0.2, 0.4, 0.2, 0.1]$$

For each option, we compare the system performance for P=100%, 70% or 50%. The input signal is white noise. Simulation results are given in Figures [8.1-8.3].

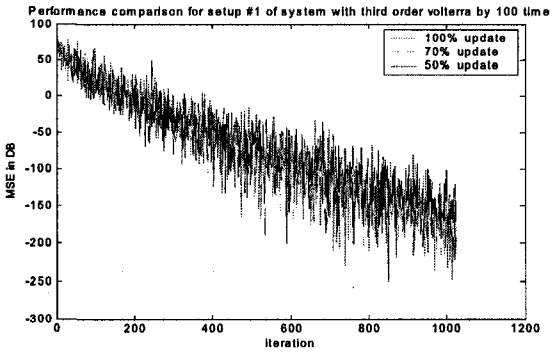


Figure 8-1: System performance for option 1 with 100%, 70% and 50% coefficient update

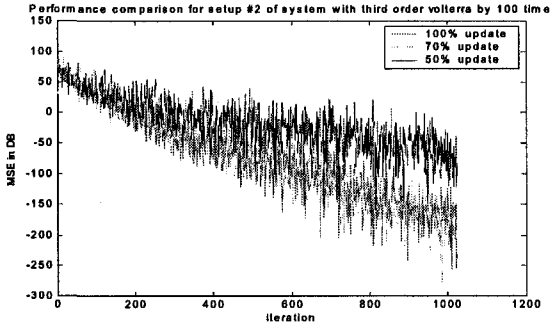


Figure 8-2: System performance for option 2 with 100%, 70% and 50% coefficient update

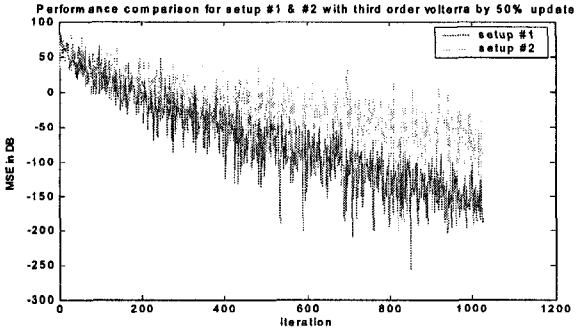


Figure 8-3: System performance for 50% coefficient update for options 1 and 2.

It is clear that option 1 provides the best performance with minimal deterioration even for a 50% coefficient update (at the cost of additional complexity for merging the sub-lists). However, the deterioration for option 2 is not significant as can be seen from Figure 8.1.

8.3 Exclusive Maximum Selective-tap Adaptive Algorithm

Recently, an exclusive maximum (XM) partial update algorithm was proposed in [Khong 2006] to deal with stereophonic echo cancellation. The XM algorithm was motivated by MMax partial update scheme [Aboulnasr 1999] as both select a subset of coefficients for updating in every adaptative iteration. However, in the XM partial update, the goal is not to reduce computational complexity. Rather the exclusive maximum tap selection strategy was proposed to reduce interchannel coherence in a two channel stereo system and improve the conditioning of the input vector autocorrelation matrix. We now review the algorithm in [Khong 2006] here since it forms the basis of our proposed XMax time domain convolutive BSS algorithm.

In stereophonic acoustic environment, the stereophonic signals $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$ are transmitted to louderspeakers in the receiving room and coupled to the microphones in this room by the room impulse responses. In stereophonic acoustic echo cancellation, these coupled acoustic echoes have to be cancelled. Let the receiving room impulse responses for $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$ be $\mathbf{h}_1(n)$ and $\mathbf{h}_2(n)$ respectively. Two adaptive filters $\hat{\mathbf{h}}_1(n)$ and

$\hat{\mathbf{h}}_2(n)$ of length L in stereophonic acoustic echo canceller are updated to estimate $\mathbf{h}_1(n)$ and $\mathbf{h}_2(n)$. The desired signal for the adaptive filters is

$$d(n)=\sum_{j=1}^{2}\mathbf{h}_j^T(n)\mathbf{x}_j(n) \tag{8.7}$$

where $\mathbf{h}_j(n)=\left[h_{j,0}(n),h_{j,1}(n),...,h_{j,L-1}(n)\right]^T$ and

$\mathbf{x}_j(n)=\left[x_j(n),x_j(n-1),...,x_j(n-L+1)\right]^T$.

Thus, the error signal is

$$e(n)=d(n)-\sum_{j=1}^{2}\hat{\mathbf{h}}_j^T(n)\mathbf{x}_j(n) \tag{8.8}$$

Adaptive algorithms such as LMS, NLMS, RLS and Affine Projection (AP) can be used to update these two adaptive filters $\hat{\mathbf{h}}_1(n)$ and $\hat{\mathbf{h}}_2(n)$. The exclusive maximum tap selection scheme is outlined in the following steps.

1) At each iteration, calculate the interchannel tap-input magnitude difference vector as $\mathbf{p}=|\mathbf{x}_1|-|\mathbf{x}_2|$

2) Sort $\mathbf{p}$ in descending order as $\breve{\mathbf{p}}=\left[\breve{p}_1,...,\breve{p}_L\right]^T$, $\breve{p}_1>\breve{p}_2>...>\breve{p}_L$.

3) Order $\mathbf{x}_1$ and $\mathbf{x}_2$ according to the sorting of $\breve{\mathbf{p}}$ as $\breve{\mathbf{x}}_1=\left[\breve{x}_{1,1},\breve{x}_{1,2},...,\breve{x}_{1,L-1}\right]^T$ and $\breve{\mathbf{x}}_2=\left[\breve{x}_{2,1},\breve{x}_{2,2},...,\breve{x}_{2,L-1}\right]^T$

4) The first channel coefficients corresponding to the M largest elements of $\mathbf{p}$ get updated and the second channel coefficients corresponding to M smallest elements of $\mathbf{p}$ get updated.

8.4 Proposed MMax Partial Update Time Domain Convolutive BSS Algorithm

From the description of MMax partial update in Section 3, we know that the principle of MMax partial update algorithm for single channel is to update the subset of coefficients which has the most impact on $\Delta\mathbf{w}$. Our proposed MMax partial update convolutive BSS algorithm is based on the same principle.

In the MMax LMS algorithm [Aboulnasr 1999], given $\Delta \mathbf{w}(n) = e(n)\mathbf{x}(n)$, the $e(n)$ is common to all elements of $\Delta \mathbf{w}(n)$, then the larger the $|x(n-i)|$, the larger its impact on error. Thus, in MMax LMS algorithm, the coefficients corresponding to M largest values in $|\mathbf{x}(n)|$ are updated.

However, in time domain natural gradient based convolutive BSS, $\Delta \underline{\mathbf{W}}$ is as follows.

$$\Delta \underline{\mathbf{W}} = -\mu \frac{\partial D}{\partial \underline{\mathbf{W}}} \underline{\mathbf{W}}^T \underline{\mathbf{W}} = \mu \left[\mathbf{I} - E\left(\varphi(\mathbf{y}) \mathbf{y}^T \right) \right] \underline{\mathbf{W}} \tag{8.9}$$

every component of $\underline{\mathbf{W}}$ is a FIR filter and there is no common value for $\Delta \mathbf{W}$. Based on MMax partial update principle, the coefficients with the largest values in $\Delta \mathbf{W}_{ij}$ are the ones to be updated.

Assuming the order of FIR filters in unmixing system is L, for every FIR filter $\mathbf{w}_{ij}$ in $\underline{\mathbf{W}}$, coefficients corresponding to M largest values in $\Delta \mathbf{W}_{ij}$ get updated. We show this algorithm using a 2 by 2 system as an example in Table 8.1.

From the algorithm description, the challenge compared to the MMax LMS algorithm [Aboulnasr 1999] is that we need to sort the elements in $\Delta \mathbf{W}_{ij}$ in every iteration, which is the additional complexity. However, we only need to update the selected coefficients, which results in some savings.

1. Initialize $\mathbf{W} = \begin{bmatrix} \mathbf{w}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{21} & \mathbf{w}_{22} \end{bmatrix}$.

2. Iteration k

$$\mathbf{x}_1 = \{x_1(k), x_1(k-1), ..., x_1(k-L)\};$$
$$\mathbf{x}_2 = \{x_2(k), x_2(k-1), ..., x_2(k-L)\};$$
$$y_1 = \mathbf{w}_{11} * \mathbf{x}_1^T + \mathbf{w}_{12} * \mathbf{x}_2^T;$$
$$y_1 = \mathbf{w}_{21} * \mathbf{x}_1^T + \mathbf{w}_{22} * \mathbf{x}_2^T;$$
$$u_1 = \tanh(y_1);$$
$$u_2 = \tanh(y_2);$$
$$\Delta\mathbf{W} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} * \begin{bmatrix} y_1 & y_2 \end{bmatrix} \right\} * \mathbf{W};$$
$$\Delta\mathbf{W}_{new} = \begin{bmatrix} \mathbf{Q}_{11} * \Delta\mathbf{w}_{11} & \mathbf{Q}_{12} * \Delta\mathbf{w}_{12} \\ \mathbf{Q}_{21} * \Delta\mathbf{w}_{21} & \mathbf{Q}_{22} * \Delta\mathbf{w}_{22} \end{bmatrix};$$
$$\mathbf{Q}_{ij} = diag\left\{\mathbf{q}_{ij}^T\right\},\ i, j = 1, 2;$$
$$\mathbf{q}_{ij}(m) = \begin{cases} 1 & \Delta\mathbf{w}_{ij}(m) \in \{M \text{ maxima of } \Delta\mathbf{w}_{ij}\} \\ 0 & \text{otherwise} \end{cases};$$
$$\mathbf{W} = \mathbf{W} + \eta * \Delta\mathbf{W}_{new};$$
$$k = k + 1;$$

3. Go to step 2 to start a new iteration.

Table 8-1: MMax partial update convolutive BSS algorithm

8.5 Proposed Exclusive Maximum Selective-tap Time Domain Convolutive BSS Algorithm

As we already know from Section 8.4, exclusive maximum tap selection can reduce interchannel correlation and improve the conditioning of the input autocorrelation matrix. In this section, we examine the effect of tap-selection on interchannel cohenrence and extend this idea to our multichannel blind source separation case.

8.5.1 Interchannel Decorrelation by Tap Selection

The squared coherence function is defined as

$$C_{\mathbf{x}_1\mathbf{x}_2}(f)=\frac{\left|P_{\mathbf{x}_1\mathbf{x}_2}(f)\right|^2}{P_{\mathbf{x}_1\mathbf{x}_1}(f)P_{\mathbf{x}_2\mathbf{x}_2}(f)} \tag{8.10}$$

where $P_{\mathbf{x}_1\mathbf{x}_2}(f)$ is the cross power spectrum between the two channels and f is the normalized frequency [Khong 2006].

A two-input two-output system is considered in this section. The mixing system used in the simulation is as follows.

$$\mathbf{H}=\begin{bmatrix}\mathbf{h}_{11} & \mathbf{h}_{12}\\ \mathbf{h}_{21} & \mathbf{h}_{22}\end{bmatrix}$$

$$\mathbf{h}_{11}=[1\ 0.8\ -0.2\ 0.78\ 0.4\ -0.2\ 0.1];$$

$$\mathbf{h}_{22}=[0.8\ 0.6\ 0.1\ -0.1\ 0.3\ -0.2\ 0.1];$$

$$\mathbf{h}_{12}=\gamma\mathbf{h}_{11}+(1-\gamma)\mathbf{b}$$

$$\mathbf{h}_{21}=\gamma\mathbf{h}_{22}+(1-\gamma)\mathbf{b}$$

where $\mathbf{b}$ is an independent white Gaussian noise with zero mean.

In the simulation, we set $\gamma=0.9$ to reflect the high interchannel correlation found in practice. The two tap input signals $\mathbf{s}_1$ and $\mathbf{s}_2$ are generated as zero mean, unit variance Gamma signals. The mixtures $\mathbf{x}_1$ and $\mathbf{x}_2$ are obtained from following equations.

$$\mathbf{x}_1=\mathbf{s}_1*\mathbf{h}_{11}+\mathbf{s}_2*\mathbf{h}_{12}$$

$$\mathbf{x}_2=\mathbf{s}_1*\mathbf{h}_{21}+\mathbf{s}_2*\mathbf{h}_{22}$$

where * is convolution operation.

The squared coherence for the $\mathbf{x}_1$ and $\mathbf{x}_2$ with full taps selected is shown in Fig 8.4. In Fig 8.5, the squared coherence for inputs with taps selected according to the MMax selection criterion as described in Section 8.4 is shown. We can see that the correlation is reduced, but

not significantly. Fig 8.6 shows the squared coherence for signals with exclusive tap selected, i.e., the selection of the same tap-index in both channel is not permitted. We can see that the correlation is reduced significantly. This confirms that exclusive tap selection strategy does indeed reduce interchannel coherence and as such improves the conditioning of the input autocorrelation matrix even in the mixing environment of blind source separation case.

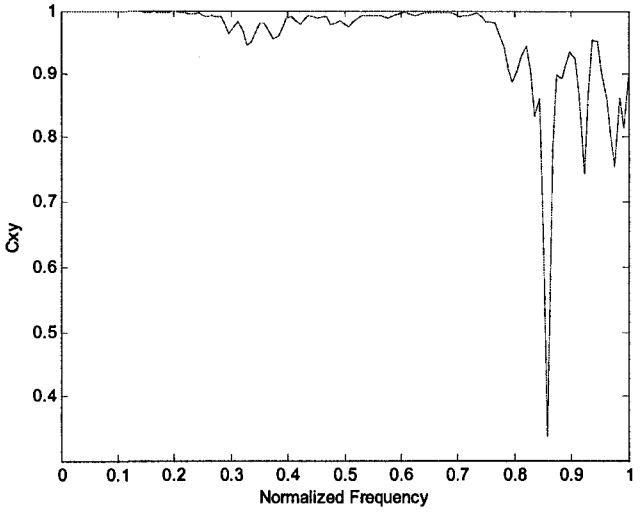


Figure 8-4: Squared coherence for x_1 and x_2 with full tap inputs selected.

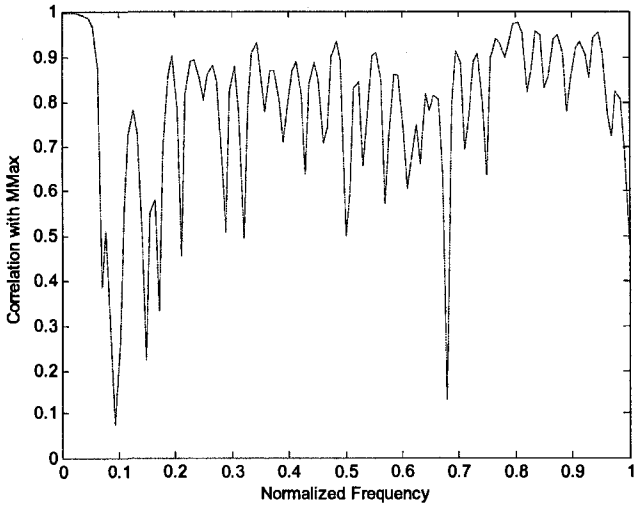


Figure 8-5: Squared coherence for x_1 and x_2 with 50% MMax tap inputs selected.

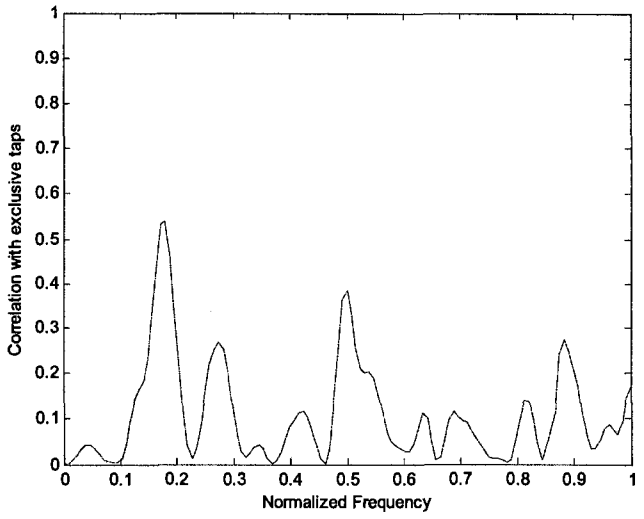


Figure 8-6: Squared coherence for $\mathbf{x}_1$ and $\mathbf{x}_2$ with exclusive maximum tap inputs selected.

8.5.2 Proposed XMax update algorithm for time domain convolutive BSS

As a result of improved conditioning of input autocorrelation matrix, we expect improved convergence rate in time domain convolutive BSS when using this update algorithm for a two by two blind source separation system.

Based on the exclusive maximum tap selection scheme proposed in [Khong 2006], we propose the exclusive maximum time domain convolutive BSS algorithm (XMax BSS) as follows.

Define $\mathbf{p}$ as the interchannel tap input magnitude difference vector at time n as

$$\mathbf{p} = |\mathbf{x}_1| - |\mathbf{x}_2| \tag{8.11}$$

Sort $\mathbf{p}$ in descending order as

$$\breve{\mathbf{p}} = [\breve{p}_1, \ldots, \breve{p}_L]^T, \breve{p}_1 > \breve{p}_2 > \ldots > \breve{p}_L \tag{8.12}$$

Order $\mathbf{x}_1$ and $\mathbf{x}_2$ according to the sorting of $\breve{\mathbf{p}}$ such that $\breve{x}_{1,i}$ and $\breve{x}_{2,i}$ corresponding to $\breve{p}_i = |\breve{x}_{1,i}| - |\breve{x}_{2,i}|$.

Taps corresponding to the $M = 0.5L$ largest elements of the input magnitude difference vector $\mathbf{p}$ in the first channel and the M smallest elements of $\mathbf{p}$ in the second channel are selected for updating of the output signal y_1; Taps corresponding to the $M = 0.5L$ largest elements of the input magnitude difference vector $\mathbf{p}$ in the second channel and the M smallest elements of $\mathbf{p}$ in the first channel are selected for updating of the output signal y_2. The detailed algorithm is shown in Table 8.2.

8.5.3 Computational Complexity of the Proposed Algorithm

The complexity is defined as the total number of multiplications and comparisons per sample period for each channel. In XMax convolutive BSS algorithm, we need to sort the interchannel tap input magnitude difference vector. For an unmixing system with filter length L, we require at most $2+2\log_2 L$ comparisons per sample period by the SORTLINE procedure [Pitas 1989]. However, the number of multiplications required for computing convolution per sample period is reduced from $4L$ to $2L$ for a two by two BSS system. Thus, the overall computational complexity is still reduced provided $L > 2$, which is always satisfied for convolutive BSS case.

1. Initialize $\mathbf{W} = \begin{bmatrix} \mathbf{w}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{21} & \mathbf{w}_{22} \end{bmatrix}$.

2. Iteration k

$\mathbf{x}_1 = \{x_1(k), x_1(k-1), ..., x_1(k-L)\}$;

$\mathbf{x}_2 = \{x_2(k), x_2(k-1), ..., x_2(k-L)\}$;

$\mathbf{p} = |\mathbf{x}_1| - |\mathbf{x}_2|$;

$\overline{\mathbf{x}}_{11} = \mathbf{Q}_{11} * \mathbf{x}_1$; $\overline{\mathbf{x}}_{21} = \mathbf{Q}_{21} * \mathbf{x}_1$;

$\overline{\mathbf{x}}_{12} = \mathbf{Q}_{12} * \mathbf{x}_2$; $\overline{\mathbf{x}}_{22} = \mathbf{Q}_{22} * \mathbf{x}_2$;

$\mathbf{Q}_{11} = diag\left\{\mathbf{q}_{11}^T\right\}$;

$\mathbf{q}_{11}(m) = \begin{cases} 1 & \mathbf{p}(m) \in \{M \text{ maxima of } \mathbf{p}\} \\ 0 & \text{otherwise} \end{cases}$;

$\mathbf{Q}_{12} = diag\left\{\mathbf{q}_{12}^T\right\}$;

$\mathbf{q}_{12}(m) = \begin{cases} 1 & \mathbf{p}(m) \in \{M \text{ minimum of } \mathbf{p}\} \\ 0 & \text{otherwise} \end{cases}$;

$\mathbf{Q}_{21} = diag\left\{\mathbf{q}_{21}^T\right\}$;

$\mathbf{q}_{21}(m) = \begin{cases} 1 & \mathbf{p}(m) \in \{M \text{ minimum of } \mathbf{p}\} \\ 0 & \text{otherwise} \end{cases}$;

$\mathbf{Q}_{22} = diag\left\{\mathbf{q}_{22}^T\right\}$;

$\mathbf{q}_{22}(m) = \begin{cases} 1 & \mathbf{p}(m) \in \{M \text{ maxima of } \mathbf{p}\} \\ 0 & \text{otherwise} \end{cases}$;

$\overline{y}_1 = \mathbf{w}_{11} * \overline{\mathbf{x}}_{11}^T + \mathbf{w}_{12} * \overline{\mathbf{x}}_{12}^T$;

$\overline{y}_2 = \mathbf{w}_{21} * \overline{\mathbf{x}}_{21}^T + \mathbf{w}_{22} * \overline{\mathbf{x}}_{22}^T$;

$u_1 = \tanh(\overline{y}_1)$;

$u_2 = \tanh(\overline{y}_2)$;

$\Delta\mathbf{W} = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} * \begin{bmatrix} \overline{y}_1 & \overline{y}_2 \end{bmatrix} \right\} * \mathbf{W}$;

$\mathbf{W} = \mathbf{W} + \eta * \Delta\mathbf{W}$;

$k = k + 1$;

3. Go to 2 to start another iteration.

4. calculate separated signals as

$y_1 = \mathbf{w}_{11} * \mathbf{x}_{11}^T + \mathbf{w}_{12} * \mathbf{x}_{12}^T$;

$y_2 = \mathbf{w}_{21} * \mathbf{x}_{21}^T + \mathbf{w}_{22} * \mathbf{x}_{22}^T$;

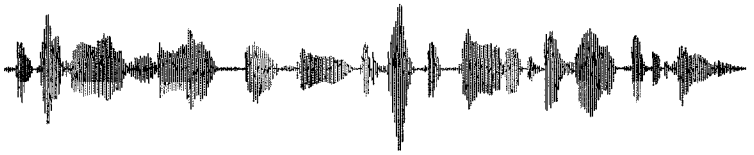
Table 8-2: XMax convolutive BSS algorithm

8.6 Simulations

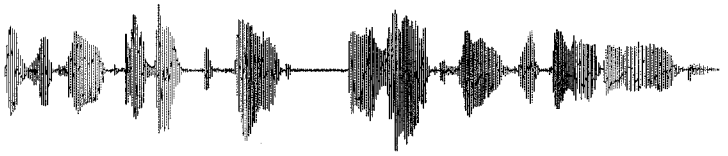
8.6.1 Experiment Setup

In the following simulations, our source signals $\mathbf{s}_1$ and $\mathbf{s}_2$ are generated Gamma signals or speech signals. The Gamma signals are generated with zero mean, unit variance. The speech signals used in our simulations include 3 female speeches and 3 male speeches with sample rate 8000 Hz to form 9 combinations. These speech signals are from ITU-T supplement database and are described as follows.

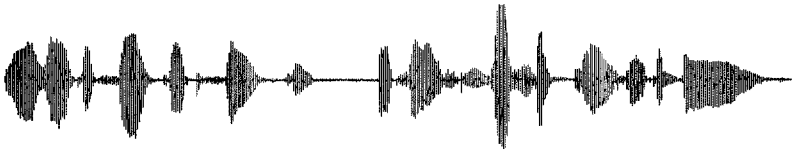
Female speech #1 (f1) "He broken new shoelace that day the coffee stand is too high for the couch"



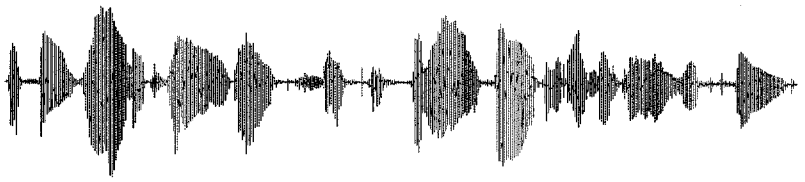
Female speech #2 (f2) "Would you please give us the fax he arrived home every other night"



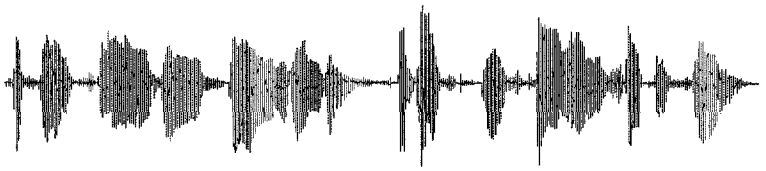
Female speech #3 (f3) "you were the perfect hostess he punched deliciously at a ball"



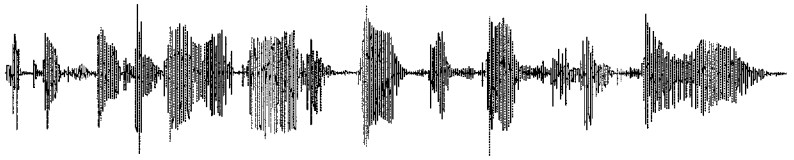
Male speech #1 (m1) "the fan whirled its round blades softly, the line where the edges join was clean"



Male speech #2 (m2) "the stale smell of old beer lingers the desk was firm on the sky floor"



Male speech #3 (m3) "it takes heat to ling out the door beef is scarcer than some"



A simple mixing system is used in our simulations to demonstrate and compare separation performance. The mixing system is given by

$$\mathbf{H} = \begin{bmatrix} 1.0 & 1.0 & -0.75; & -0.2 & 0.4 & 0.7 \\ 0.2 & 1.0 & 0.0; & 0.5 & -0.3 & 0.2 \end{bmatrix}$$

The mixture signals are obtained by convolving the source signals with the mixing system. The filter length in the separation system is set at 64.

The separation performance is evaluated by intersymbol interference (ISI), signal to interference ratio (SIR) and perceptual evaluation of speech quality (PESQ) which is described in Chapter 3.

In the following, we will compare the separation performance of the regular convolutive BSS algorithm, MMax partial update BSS algorithm and XMax selective-tap BSS algorithm.

8.6.2 MMax Partial Update Time Domain BSS for Convolutive Mixture

In this simulation, we test the performance of MMax partial update time domain BSS algorithm for convolutive mixtures. In the following diagram, 'reg' means full update time domain BSS algorithm; 'par56' means MMax partial update time domain BSS algorithm with M=56; 'par48' means MMax partial update time domain BSS algorithm with M=48; 'par32' means MMax partial update time domain BSS algorithm with M=32, where M is the number of coefficients updated each iteration in a given channel.

In the first experiment, we use generated Gamma signals as the original signals and use the convolutive BSS model in Chapter 3 to get the mixture signals. The performance of full update time domain convolutive BSS algorithm, MMax partial update convolutive BSS algorithm evaluated by the SIR measure defined in Eq. (4.7) are shown in Fig 8.7 and Fig 8.8. The performance of regular time domain convolutive BSS algorithm, MMax partial update convolutive BSS algorithm evaluated by the ISI measure defined in Eq. (4.4) and (4.5) are shown in Fig 8.9 and Fig 8.10.

From Fig 8.7 and Fig 8.8, we can see that as expected, the MMax partial update convolutive BSS algorithm converges slightly slower than the regular BSS algorithm while only a subset of coefficients get updated. However, it converges to similar SIR values. On the other hand, from Fig 8.9 and Fig 8.10, it seems that the performance of MMax partial update BSS algorithm is even better than the regular BSS algorithm, which is not our expected result and is also not consistent with the results in Fig 8.7 and Fig 8.8. This also happens for other separation cases of different Gamma signals and different speech signals. By analyzing the definition of ISI, we believe that the ISI measure, originally developed for the instantaneous case, is not reliable for convolutive BSS case. In instantaneous BSS model, p_{ij} is a scalar and the ISI measure for separation performance is quite meaningful. However, there are some drawbacks for extending this measure to the convolutive case since $\mathbf{p}_{ij}$ is an FIR filter now. In the definition of ISI, $\mathbf{p}_{ij}$ is treated as an unrelated component rather than FIR filter which can produce target signals or introduce interference. Moreover, it does not take into

account the effect from source signals. Thus, in our following simulations, we will only present results evaluated by SIR and PESQ.

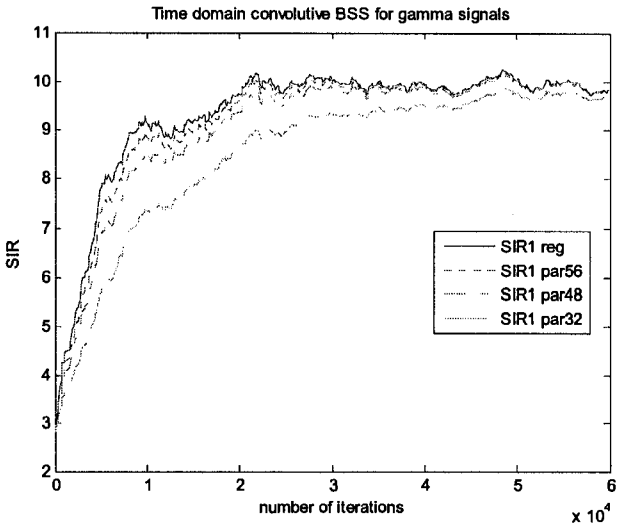


Figure 8-7: Separation performance of time domain regular convolutive BSS and MMax partial update BSS for Gamma signal measured by SIR for the first output

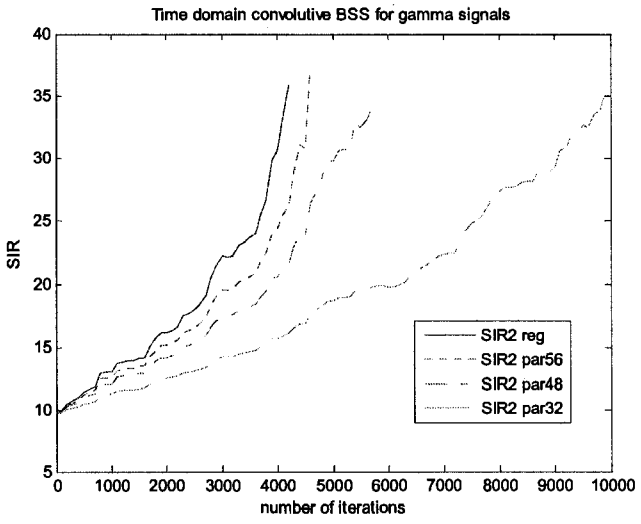


Figure 8-8: Separation performance of time domain regular convolutive BSS and MMax partial update BSS for Gamma signal measured by SIR for the second output

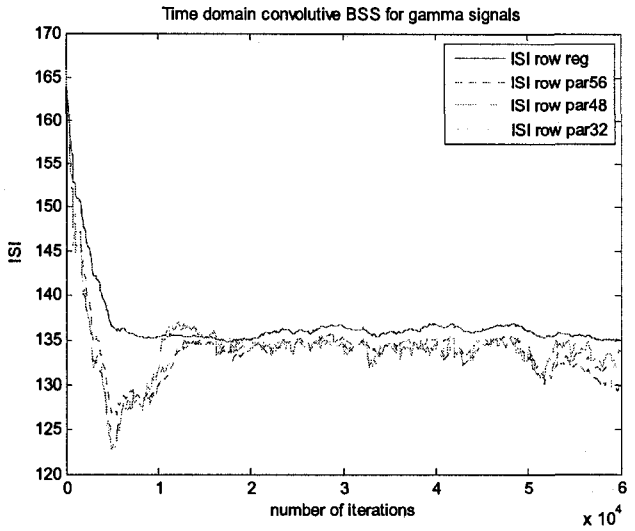


Figure 8-9: Separation performance of time domain regular convolutive BSS and MMax partial update BSS for Gamma signal measured by ISI_row.

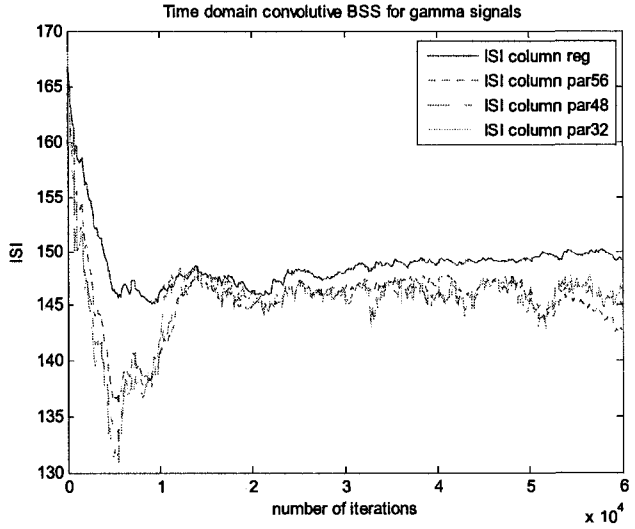


Figure 8-10: Separation performance of time domain regular convolutive BSS and MMax partial update BSS for Gamma signal measured by ISI_column.

In the second experiment, we use speech signals as the original signals and use the same mixing system to get the mixture signals. In Fig 8.11 and Fig 8.12, we show the performance of regular time domain convolutive BSS algorithm, MMax partial update BSS convolutive algorithm for one combination of speech signals, the separation performance is evaluated by SIR. The performance for other combinations of speech signals is similar to that shown in Fig 8.11 and Fig 8.12.

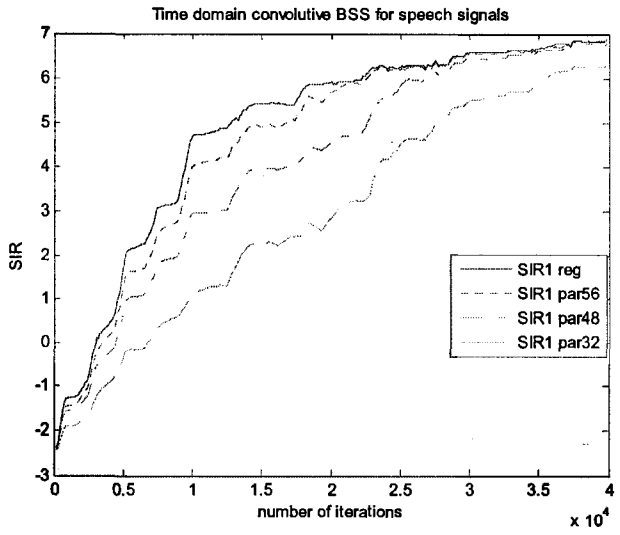


Figure 8-11: Separation performance of time domain regular convolutive BSS and MMax partial update BSS for speech signal measured by SIR.

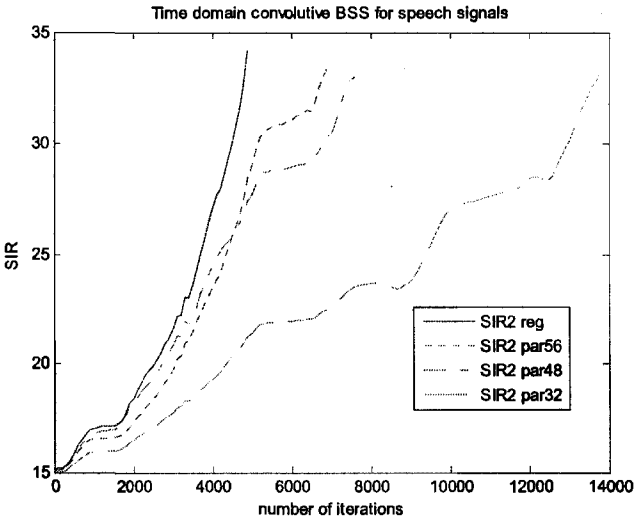


Figure 8-12: Separation performance of time domain regular convolutive BSS and MMax partial update BSS for speech signal measured by SIR.

Since we used speech signals in the second experiment, we also use PESQ to evaluate the separation performance. In the following, we evaluate the similarity between the mixtures, the separated signals from regular and MMax BSS algorithms with the original source signals by PESQ score. Table 8.3-8.11 show the PESQ evaluation results for different combinations of female and male speech signals, Table 8.12 shows their average PESQ

evaluation results. In these Tables, (S1, S2) present the original source signals; (mix1,mix2) present the mixture signals; (regular out1, regular out2) present separated signals from full update BSS algorithm; (partial M=56 out1, partial M=56 out2) present separated signals from MMax BSS algorithm with M=56; (partial M=48 out1, partial M=48 out2) present separated signals from MMax BSS algorithm with M=48; (partial M=32 out1, partial M=32 out2) present separated signals from MMax BSS algorithm with M=32.

PESQ	Mixture		Regular		Partial M=56		Partial M=48		Partial M=32	
	mix1	mix2	out1	out2	out1	out2	out1	out2	out1	out2
S1	2.11	0.65	2.42	0.68	2.41	0.60	2.42	0.59	2.39	0.51
S2	1.30	2.36	0.81	2.68	0.76	2.64	0.79	2.54	0.71	2.56

Table 8-3: PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm for combination f1 and m1

PESQ	Mixture		Regular		Partial M=56		Partial M=48		Partial M=32	
	mix1	mix2	out1	out2	out1	out2	out1	out2	out1	out2
S1	1.99	1.03	2.40	0.30	2.39	0.53	2.40	0.45	2.38	0.33
S2	1.07	2.33	0.82	2.92	0.97	2.70	1.32	2.67	0.68	2.70

Table 8-4: PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm for combination f1 and m2

PESQ	Mixture		Regular		Partial M=56		Partial M=48		Partial M=32	
	mix1	mix2	out1	out2	out1	out2	out1	out2	out1	out2
S1	2.04	0.90	2.36	0.60	2.35	0.31	2.36	0.58	2.33	0.48
S2	1.34	2.29	0.98	2.74	0.93	2.70	0.79	2.57	0.95	2.49

Table 8-5: PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm for combination f1 and m3

PESQ	Mixture		Regular		Partial M=56		Partial M=48		Partial M=32	
	mix1	mix2	out1	out2	out1	out2	out1	out2	out1	out2
S1	2.18	0.91	2.56	0.56	2.51	0.58	2.58	0.56	2.52	0.56
S2	1.42	2.36	1.03	2.65	1.20	2.67	0.96	2.62	1.02	2.59

Table 8-6: PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm for combination f2 and m1

PESQ	Mixture		Regular		Partial M=56		Partial M=48		Partial M=32	
	mix1	mix2	out1	out2	out1	out2	out1	out2	out1	out2
S1	2.10	0.99	2.34	0.57	2.29	0.57	2.28	0.57	2.26	0.53
S2	1.58	2.42	1.33	2.90	1.50	2.83	1.54	2.81	1.32	2.83

Table 8-7: PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm for combination f2 and m2

PESQ	Mixture		Regular		Partial M=56		Partial M=48		Partial M=32	
	mix1	mix2	out1	out2	out1	out2	out1	out2	out1	out2
S1	2.19	1.14	2.35	0.39	2.39	0.40	2.31	0.39	2.32	0.32
S2	1.30	2.21	0.86	2.75	0.86	2.71	0.87	2.79	0.88	2.74

Table 8-8: PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm for combination f2 and m3

PESQ	Mixture		Regular		Partial M=56		Partial M=48		Partial M=32	
	mix1	mix2	out1	out2	out1	out2	out1	out2	out1	out2
S1	2.23	0.92	2.39	0.79	2.34	0.89	2.33	0.80	2.34	0.91
S2	1.07	2.53	1.33	2.66	1.18	2.54	1.51	2.49	1.32	2.37

Table 8-9: PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm for combination f3 and m1

PESQ	Mixture		Regular		Partial M=56		Partial M=48		Partial M=32	
	mix1	mix2	out1	out2	out1	out2	out1	out2	out1	out2
S1	2.11	1.18	2.33	1.14	2.35	1.23	2.25	1.03	2.29	1.38
S2	1.67	2.43	1.30	2.91	1.32	2.86	1.26	2.76	1.34	2.71

Table 8-10: PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm for combination f3 and m2

PESQ	Mixture		Regular		Partial M=56		Partial M=48		Partial M=32	
	mix1	mix2	out1	out2	out1	out2	out1	out2	out1	out2
S1	2.14	1.12	2.27	0.49	2.27	0.40	2.23	0.44	2.25	0.40
S2	1.52	2.45	1.21	2.72	1.23	2.67	1.28	2.68	1.05	2.64

Table 8-11: PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm for combination f3 and m3

PESQ	Mixture		Regular		Partial M=56		Partial M=48		Partial M=32	
	mix1	mix2	out1	out2	out1	out2	out1	out2	out1	out2
S1	2.12	0.98	2.38	0.61	2.37	0.61	2.35	0.60	2.34	0.60
S2	1.36	2.37	1.08	2.77	1.10	2.70	1.15	2.66	1.03	2.62

Table 8-12: Average PESQ scores for mixtures and separated signals from regular BSS algorithm and MMax BSS algorithm

From these Tables, we can see that the separation performance evaluated by PESQ is consistent with the SIR results. The separation algorithms make the separated signals more biasd to one source signal and away from other source signal. The separation performance as evaluated by PESQ and SIR is also consistent with our informal listening tests.

From the above simulation results, we can see that similar to MMax NLMS algorithm for single channel linear filters or nonlinear filters, there is a slight deterioration in performance of the proposed MMax partial update time domain convolutive BSS algorithm as the number of updated coefficients is reduced. However, the performance at 50% coefficients updated is still quite acceptable.

8.6.3 Time Domain Exclusive Maximum selective tap BSS for Convolutive Mixture

In this simulation, we test the performance of XMax selective tap time domain BSS algorithm for convolutive mixtures.

In the first experiment, we use generated Gamma signals as the original signals and use the convolutive BSS model in Chapter 3 to get the mixture signals. The performance of regular time domain convolutive BSS algorithm, XMax selective tap convolutive BSS algorithm evaluated by SIR is shown in Fig 8.13 and Fig 8.14.

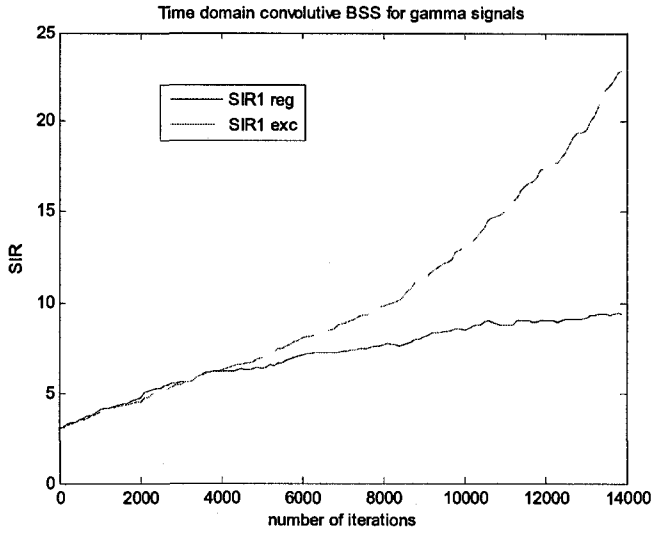


Figure 8-13: Separation performance of time domain regular convolutive BSS and XMax selective tap BSS for Gamma signal measured by SIR for the first output

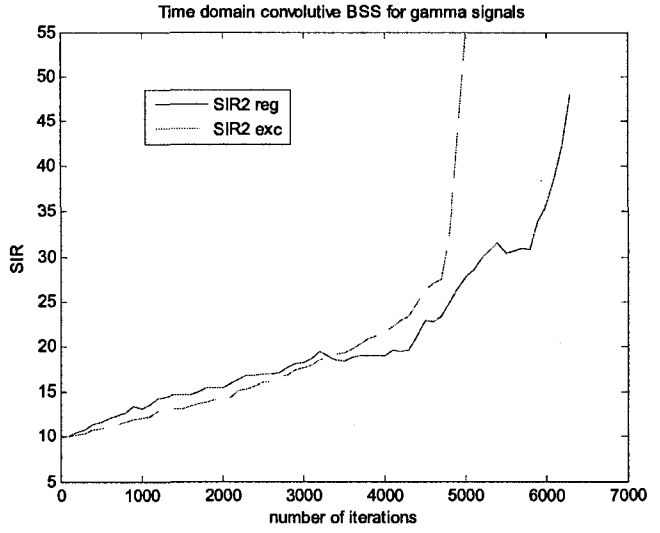


Figure 8-14: Separation performance of time domain regular convolutive BSS and XMax selective tap BSS for Gamma signal measured by SIR for the second output

From Fig 8.13 and Fig 8.14, we can see that XMax BSS algorithm has much better convergence rate compared with regular BSS algorithm for generated Gamma signals.

In the second experiment, we use speech signals as the original signals and use the same mixing system to get the mixture signals. In Fig 8.15 and Fig 8.16, we show the performance

of regular time domain convolutive BSS algorithm, XMax selective tap BSS convolutive algorithm for one combination of speech signals, the separation performance is evaluated by SIR. The performance for other combinations of speech signals is similar to that shown in Fig 8.15 and Fig 8.16.

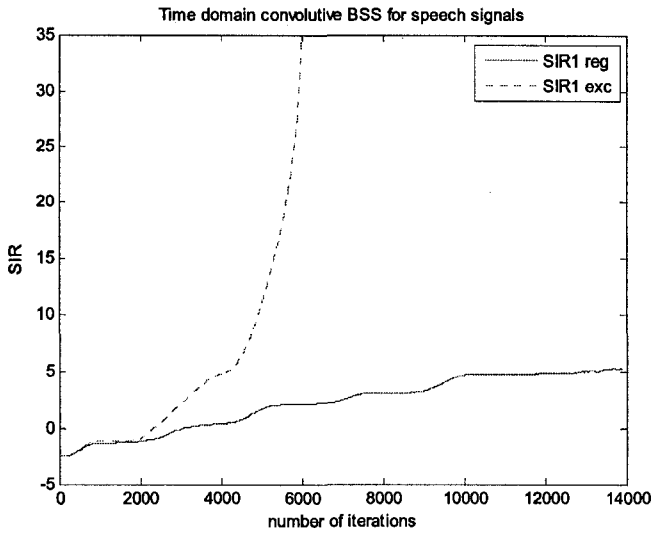


Figure 8-15: Separation performance of time domain regular convolutive BSS and XMax selective tap BSS for speech signal measured by SIR for the first output

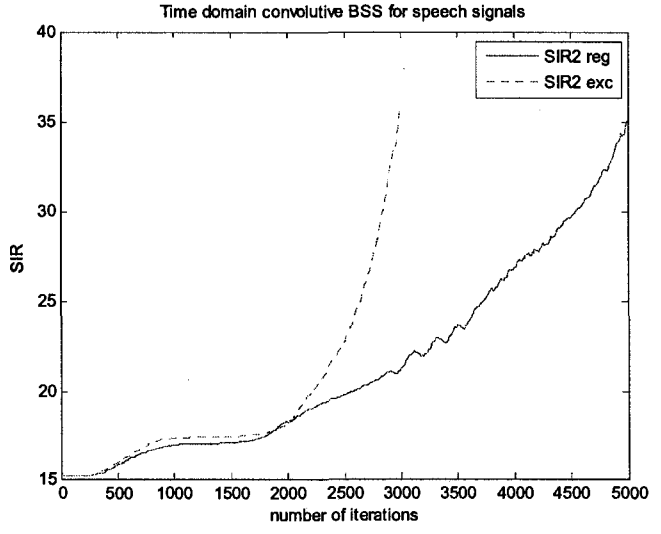


Figure 8-16: Separation performance of time domain regular convolutive BSS and XMax selective tap BSS for speech signal measured by SIR for the second output

From the plots, we can see that XMax BSS algorithm has much better convergence rate compared with regular BSS algorithm for both generated Gamma signals and speech signals.

Since we used speech signals in the second experiment, we also use PESQ to evaluate the separation performance. In the following, we evaluate the similarity between the mixtures, the separated signals from regular and XMax BSS algorithms with the original source signals by PESQ score. Table 8.13-8.21 show the PESQ evaluation results for different combinations of female and male speech signals, Table 8.22 shows their average PESQ evaluation results. In these Tables, (S1, S2) present the original source signals; (mix1,mix2) present the mixture signals; (regular BSS out1, out2) present separated signals from regular BSS algorithm; (XMax BSS out1, out2) present separated signals from XMax BSS. The performance evaluation by PESQ is consistent with that measured by SIR. The separation performance evaluated by PESQ and SIR is also consistent with our informal listening tests.

PESQ	Mixture		Regular BSS		Xmax BSS	
	mix1	mix2	out1	out2	out1	out2
s1	2.08	1.10	2.35	0.56	3.14	0.52
s2	1.60	2.57	1.13	2.74	0.67	2.74

Table 8-13: PESQ scores for mixture signals, separated signals by regular BSS algorithm and XMax selective tap BSS algorithm for combination f1 and m1

PESQ	Mixture		Regular BSS		Xmax BSS	
	mix1	mix2	out1	out2	out1	out2
s1	2.02	1.16	2.32	0.11	3.14	0.11
s2	1.48	2.33	0.80	2.93	0.55	2.93

Table 8-14: PESQ scores for mixture signals, separated signals by regular BSS algorithm and XMax selective tap BSS algorithm for combination f1 and m2

PESQ	Mixture		Regular BSS		Xmax BSS	
	mix1	mix2	out1	out2	out1	out2
s1	2.05	0.96	2.36	0.33	3.14	0.44
s2	1.36	2.29	0.95	2.80	0.72	2.80

Table 8-15: PESQ scores for mixture signals, separated signals by regular BSS algorithm and XMax selective tap BSS algorithm for combination f1 and m3

PESQ	Mixture		Regular BSS		Xmax BSS	
	mix1	mix2	out1	out2	out1	out2
s1	2.13	0.73	2.41	0.63	3.00	0.63
s2	1.43	2.41	0.93	2.74	0.58	2.74

Table 8-16: PESQ scores for mixture signals, separated signals by regular BSS algorithm and XMax selective tap BSS algorithm for combination f2 and m1

PESQ	Mixture		Regular BSS		Xmax BSS	
	mix1	mix2	out1	out2	out1	out2
s1	2.05	0.83	2.21	1.04	3.00	0.72
s2	1.44	2.32	1.12	2.93	0.18	2.93

Table 8-17: PESQ scores for mixture signals, separated signals by regular BSS algorithm and XMax selective tap BSS algorithm for combination f2 and m2

PESQ	Mixture		Regular BSS		Xmax BSS	
	mix1	mix2	out1	out2	out1	out2
s1	2.17	1.02	2.29	0.98	3.00	0.97
s2	1.26	2.21	1.00	2.80	0.95	2.80

Table 8-18: PESQ scores for mixture signals, separated signals by regular BSS algorithm and XMax selective tap BSS algorithm for combination f2 and m3

PESQ	Mixture		Regular BSS		Xmax BSS	
	mix1	mix2	out1	out2	out1	out2
s1	2.18	1.12	2.31	0.34	2.83	0.39
s2	1.68	2.60	1.37	2.74	1.72	2.74

Table 8-19: PESQ scores for mixture signals, separated signals by regular BSS algorithm and XMax selective tap BSS algorithm for combination f3 and m1

PESQ	Mixture		Regular BSS		Xmax BSS	
	mix1	mix2	out1	out2	out1	out2
s1	2.09	1.43	2.14	1.34	2.80	0.50
s2	1.67	2.44	1.15	2.92	1.26	2.93

Table 8-20: PESQ scores for mixture signals, separated signals by regular BSS algorithm and XMax selective tap BSS algorithm for combination f3 and m2

PESQ	Mixture		Regular BSS		Xmax BSS	
	mix1	mix2	out1	out2	out1	out2
s1	2.12	1.14	2.31	0.33	2.87	0.33
s2	1.63	2.47	1.09	2.80	0.45	2.80

Table 8-21: PESQ scores for mixture signals, separated signals by regular BSS algorithm and XMax selective tap BSS algorithm for combination f3 and m3

PESQ	Mixture		Regular BSS		Xmax BSS	
	mix1	mix2	out1	out2	out1	out2
s1	1.87	0.95	2.04	0.59	2.64	0.46
s2	1.58	2.26	1.22	2.55	1.06	2.56

Table 8-22: Average PESQ scores for mixture signals, separated signals by regular BSS algorithm and XMax selective tap BSS algorithm

Based on the above simulations, we can see that XMax BSS algorithm significantly improves the convergence rate compared with regular time domain convolutive BSS algorithm. This improved convergence is not achieved at a cost of increased complexity given the discussion in Section 8.5.3.

8.7 Conclusion

In this chapter, we investigated time domain convolutive BSS algorithm and propose two novel algorithms to address the slow convergence rate and high computational complexity problem in time domain BSS. In the proposed MMax partial update time domain convolutive BSS algorithm (MMax BSS), only a subset of coefficients in the separation system gets updated at every iteration. We showed that the partial update scheme applied in the MMax LMS algorithm for single channel can be extended to multichannel natural gradient based time domain convolutive BSS with little deterioration in performance and possible computation complexity saving. In the proposed exclusive maximum selective-tap time domain convolutive BSS algorithm (XMax BSS), the exclusive tap-selection update procedure reduces the interchannel coherence of the tap-input vectors and improves the conditioning of the autocorrelation matrix so as to accelerate convergence rate and reduce the

misalignment. Moreover, the computational complexity is reduced as well since only half tap inputs are selected for updating. Simulation results have shown a significant improvement in convergence rate compared with existing techniques. The extension of the proposed XMax BSS algorithm to more than two channels is still an open problem.

Chapter 9 Conclusions and Future Work

9.1 Summary and Conclusions

Speech signal separation is the problem of decomposing mixtures of speeches into its components. It is a natural task for humans since it is done by our ears and brain automatically. However, many issues need to be solved when we want to implement a similar system using microphones and computers. There are many approaches addressing these separation problems. In this thesis, we have covered some aspects of blind speech signal separation. We study the principles behind the Blind Source Separation and search for better systems to improve the performance of blind speech signal separation.

In the first stage of our work, we studied principles for conducting instantaneous blind source separation and convolutive blind source separation. Simulations are performed on instantaneous and convolutive blind separation of Gamma mixture signals and mixed speech signals for better understanding of BSS. Then we applied convolutive blind source separation algorithm to deal with the problem of joint blind speech signal separation and cancellation, such as network echo cancellation and acoustic echo cancellation. Since the end users for speech signal separation are humans, we proposed perceptual convolutive speech signal separation by taking advantage of human hearing system properties. As a first try, we proposed a post-filter based perceptual convolutve blind speech separation system: a frequency domain convolutive BSS system cascaded with a post filter system. This new system can slightly improve the overall performance without much computation increase. Then we proposed a new perceptual convolutive speech separation algorithm based on filtered-E LMS algorithm and convolutive BSS algorithm. In this new algorithm, the error is weighted by a function obtained from the absolute hearing threshold to emphasize frequencies to which human ear is sensitive and de-emphasize frequencies inaudible to human ear. The proposed new algorithm has been shown to improve the quality of the separated signal and exhibit improved convergence.

The main contributions of this thesis lie in the following two topics. One is the combined beamforming and BSS system for blind speech signal separation in real acoustic room environmemts. The other one is the partial update time domain BSS algorithms.

Many existing blind source separation approaches achieve good performance only in artificial situations. When applying these BSS algorithms to a real acoustical environment, e.g. a number of people talking in a room, the performance is greatly compromised by the effect of the room reflections or reverberations and ambient noise. Thus, a two-stage system is proposed to separate speech signals in real room environments by combining beamforming with convolutive blind source separation. In the first stage, the adaptive beamforming is implemented to separate signals from selected directions based on estimated source location information. In the second stage, the convolutive blind source separation algorithm is used to further separate the remaining cross-talk from the outputs of adaptive beamforming stage. By doing so, we combine the spatial information used in beamforming with time/frequency processing used in convolutive BSS aiming for better separation performance given the increased information used. In our first system, we use an adaptive beamformer in the first stage and a frequency domain based BSS system in the second stage. Simulation results for speech separation in a real room environment show that beamforming greatly reduces the reverberation effects while the subsequent convolutive BSS converges more easily with significantly reduced complexity. However, the problem in this system is that we use the MUSIC algorithm in the adaptive beamforming stage. More microphones than speakers need to be used in this system and we need some prior information about the room to conduct the adaptive beamforming. Thus, in our next system, we propose a completely blind beamforming and BSS combination for speech signal separation in real acoustic room environment. By investigating the properties of the unmixing matrix in frequency domain BSS in detail, we proposed to blindly estimate the DOA with significantly reduced complexity. Some strategies are also used to reduce the overall system complexity. Compared with existing systems, our second combined system significantly reduces the computational complexity and maintains the separation performance.

Even though frequency domain convolutive BSS algorithms are very popular in the literature, its inherent permuatation and scaling ambiguity problems limit its applications. Thus, in Chapter 8 we come back to time domain and propose novel algorithms to improve the convergence and reduce the complexity of time domain convolutive BSS algorithm. First, we propose the application of MMax partial update algorithm to the time domain convolutive BSS (MMax BSS). We demonstrate that the partial update scheme applied in the MMax LMS algorithm for single channel can be extended to multichannel time domain convolutive BSS with little deterioration in performance and possible computation complexity saving. Next, we propose exclusive maximum selective-tap time domain convolutive BSS algorithm (XMax BSS) that reduces the interchannel coherence of the tap-input vectors and improves the conditioning of the autocorrelation matrix resulting in improved convergence rate and reduced misalignment. Moreover, the computational complexity is reduced since only half the input taps are selected for updating. Simulation results have shown a significant improvement in convergence rate compared to existing techniques.

9.2 Future work

Blind speech signal separation can be performed in time and frequency domain. While frequency domain algorithms are so popular in the literature, their inherent frequency permutation problem is still difficult to resolve completely, especially in high reverberant environments.

For time domain BSS algorithms, it is well-known that their low convergence and high computational complexity limit their applications. As we can see from Chapter 8, the proposed exclusive maximum selective-tap time domain convolutive BSS algorithm (XMax BSS) has shown a significant improvement in convergence rate compared to existing techniques. Moreover, the computational complexity is reduced as well since only half tap inputs are selected for updating. However, the extension of the proposed XMax BSS algorithm to more than two channels is still an open problem.

Other open problems for blind speech signal separation include following listed, but not limited to these.

- Blind speech signal separation in sub-band
- Blind speech signal separation in additive noise environments
- Blind speech separation for more sources than sensors in convolutive environments
- Real time implementation of blind speech signal separation

Reference:

[Aboulnasr 1999] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," IEEE Transactions on Signal Processing, vol. 47, no. 5, pp. 1421-1424, May 1999.

[Abousaada 1992] A. Abousaada, T. Aboulnasr, W. Steenaart, "An echo tail canceler based on adaptive interpolated FIR filtering," IEEE Transactions on Circuits and Systems, vol. CAS-41, pp. 409-415, July 1992.

[Aboulnasr 2005] T. Aboulnasr and Q. Pan, "Data Dependant Partial Update Adaptive Algorithms for Linear and Nonlinear Systems", invited paper, EUSIPCO 2005.

[Aichner 2002] R. Aichner, S. Araki and S. Makino, "Time-domain blind source separation of non-stationary convolved signals by utilizing geometric beamforming," 12th IEEE Workshop on Neural Networks for signal processing, Sept. 2002, pp.445-454.

[Amari 1996] S. Amari and A. Cichocki, "A new learning algorithm for blind signal separation," In advances in neural information processing systems 8, pp.757-763, MIT press, 1996.

[Amari 1997] S. Amari, S. Douglas, A. Cichocki and H. Yang, "Multichannel blind deconvolution and equalization using the natural gradient," Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications, pp.101-104, April 1997.

[Amari 1998] S. Amari, "Natural gradient works efficiently in learning," Neural Computation, Vol. 10, pp.251-276. Feb., 1998.

[Araki 2003] S. Araki, S. Makino, Y. Hinamoto, R. Mukai, T. Nishikawa, and H. Saruwatari, "Equivalence between frequency domain blind source separation and frequency

domain adaptive beamforming for convolutive mixtures, " EURASIP Journal on Applied Signal Processing, vol. 2003, no. 11, pp. 1157-1166, Nov. 2003.

[Asano 2001] F. Asano, M. Goto, K. Itou and H. Asoh, "Real-time sound source localization and separation system and its application to automatic speech recognition," Proc. Eurospeech2001, pp.1013-1016.

[Attallah 2001] S. Attallah and S.W. Liaw, "Analysis of DCTLMS algorithm with a selective coefficient updating," IEEE Transactions on circuits and Systems-II, Vol 48, No.6, pp.628-632, June 2001.

[Baumann 2003] W. Baumann, D. Kolossa and R. Orglmeister, "Beamforming-based convolutive source separation," IEEE International Conference Acoustics, Speech, and Signal Processing (ICASSP'03), Vol. 5, 2003 pp:357-360.

[Bell 1995] A.J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," Neural Computation, Vol.7, PP. 1129-1159, 1995.

[Buchner 2004] H. Buchner, R. Aichner and W. Kellermann, "Blind source separation for convolutive mixtures: A unified treatment," in Audio Signal Processing for Next-generation Multimedia Communication Systems, Eds. Y. Huang and J. Benesty, Kluwer Academic Publishers, pp. 255 -293, 2004.

[Cardoso 1997] J. F. Cardoso, "Infomax and maximum likelihood for source separation," IEEE Letters and Signal Processing, Vol. 4, pp.112-114, 1997.

[Cichocki 1994] A. Cichocki, R. Unbehauen, L. Moszczynski and E. Rummert, "A new on-line adaptive algorithm for blind separation of source signals," In Proc. Int. Symposium on Artificial Neural Networks, ISANN-94, pp.406-411,1994.

[Cichocki 2000] A. Cichocki and S. Amari, Adaptive Blind Signal and Image Processing, John Wiley & Sons, 2000.

[Common 1994] P. Common, "Independent component analysis, a new concept?," Signal Processing, 36:287-314, April 1994.

[Delfose 1995] N. Delfose and P. loubaton, "Adaptive blind separation of independent sources: a deflation approach," Signal Processing, 45:59-83, 1995.

[Deng 2004] H. Deng and M. Doroslovački, "New sparse adaptive filters with partial update," Proceedings of Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP 2004, vol. 2, pp. 845-848.

[Doğançay 2001] K. Doğançay and O. Tanrikulu, "Adaptive filtering algorithms with selective partial updates," IEEE Transactions on Circuits And Systems-II, vol. 48, no. 8, pp. 762-769, August 2001.

[Doğançay 2002] K. Doğançay and O. Tanrikulu, "Generalized subband decomposition LMS algorithm employing selective partial updates," Proc. of the IEEE Conf. on Acoustics, Speech and Signal Processing, ICASSP 2002, vol. 2, pp 1377-1380.

[Douglas 1994] S. C. Douglas, "A family of normalized LMS algorithms," IEEE Signal Processing Letters, vol. 1, no. 3, pp.49-51, March 1994.

[Douglas 1997] S. Douglas, "Adaptive filters Employing Partial Updates," IEEE Trans. on Circuits and Systems, CAS-II, Vol. 44, No 3., pp. 209-216, March 1997.

[Douglas 2003] S. C. Douglas and X. Sun, "Convolutive blind separation of speech mixtures using the natural gradient," Speech Commun., vol.39, pp65-78, 2003.

[Fletcher 1987] R. Fletcher, "Practical Method of Optimization," Wiley, 2nd Edition,1987.

[Gaeta 1990] M. Gaeta and J. L. Lacoume, "Source separation without priori knowledge: the maximum likelihood solution," In Proc. EUSIPCO'90, pp.621-624,1990.

[Giannakopoulos 1998] X. Giannakopoulos, "Comparison of adaptive independent component analysis algorithms," Master Thesis, Helsinki University of Technology, Finland, March, 1998.

[Giannakopoulos 1999] X. Giannakopoulos, J. Karhunen and E. Oja, "An experimental comparison of neural algorithms for independent component analysis and blind separation," Int. Journal of Neural Systems, Vol.9, No.2, April, 1999, pp.99-114.

[Godavarti 2000] M. Godavarti and A. Hero, "Stochastic partial update LMS algorithm for adaptive arrays," Proc. of the IEEE Sensor Array and Multichannel Signal Processing Workshop, pp.322-326, 2000.

[Haykin 2000] S. Haykin, ed., Unsupervised adaptive filtering (Volume I: Blind Source Separation), John Wiley & Sons, 2000.

[Herault] J. Herault, C. Jutten and B. Ans, "Dection de grandeurs primitives dans un message composite par une architecture de calul neuromimetique un apprentissage non supervise," Proc. GRETSI, France.

[Huang 2002] Y. Huang and T. Chiueh, "A new audio coding scheme using a forward masking model and perceptually weighted vector quantization," IEEE Trans. On speech and audio processing, Vol. 10, No. 5, July 2002, pp. 325-335.

[Hyvarinen 1997] A. Hyvarinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," Neural Computation, Vol. 9, No.7, pp.1483-1492, 1997.

[Hyvarinen 1999] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," IEEE Trans. on Neural Networks, Vol. 10, No.3, pp.626-634, 1999.

[Hyvarinen 2001] A. Hyvarinen, J. Karhunen, and E. Oja, Independent Component Analysis, John Wiley & Sons, 2001.

[Ikram 2002] M. Z. Ikram and D. R. Morgan, "A beamforming approach to permutation alighment for multichannel frequency domain blind speech separation," Proc. ICASSP 2002, pp. 881-884, May 2002.

[ITU 2000] ITU-T Recommend P.862, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to end speech quality assessment of narrowband telephone network and speech codecs," May 2000.

[Jafari 2002] M. G. Jafari and J. A. Chambers, "Wavelet domain natural gradient algorithm for blind source separation of non-stationary sources," Electronics Letters, 4th July 2002, Vol. 38, No. 14, pp. 759-761.

[Johnston 1988] J. D. Johnston, "Transform Coding of Audio Signals Using Perceptual Noise Criteria", IEEE J. Selected Areas in Communications, Vol. 6, No. 2, pp.314-323, Feb. 1988.

[Joho 2003] M. Joho and P. Schnitter, "Frequency domain realization of a multichannel blind deconvolution algorithm based on the natural gradient," Proc. ICA2003, pp.543-548, April 2003.

[Kawamoto] M. Kawamoto, A K. Barros, A. Mansour, K. Matsuoka and N. Ohnishi, "Blind separation for convolutive mixtures of non-stationary signals", Proc. of International Symposium on Nonlinear Theory and its Applications, pp. 1001-1004, Hawaii, 1997.

[Khong 2006] A. W. H. Khong and P. A. Laylor, "Stereophonic Acoustic Echo Cancellation Employing Selective-Tap Adaptive Algorithms," IEEE Trans. on Speech and Audio Processing.

[Knaak 2003] M. Knaak, S. Araki and S. Makino, "Geometrically constraint ICA for convolutive mixtures of sound," in Proc. IEEE ICASSP 2003, pp.725-729.

[Kokkinakis 2003] Kokkinakis, V. Zarzoso and A.K. Nandi, "Blind separation of acoustic mixtures based on linear prediction analysis," 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), April 2003, Japan.

[Kuo 1994] S. M. Kuo and J. Tsai, "Residual noise shaping technique for active noise control system," Journal of the Acoustical Society of America, Vol.95, No. 3, March 1994, pp. 1665-1668.

[Kurita 2000] S. Kurita, H. Saruwatari, S. Kajita, K. Takeda and F. Itakura, "Evaluation of blind signal separation method using directivity pattern under reverberant conditions," in Proc. IEEE ICASSP, 2000, pp.3140-3143.

[Lambert 1996] R. H. Lambert, "Multichannel blind deconvolution: FIR matrix algebra and separation of multipath mixtures," Ph.D. Thesis, University of Southern California, Department of Electrical Engineering, May 1996.

[Lambert 1997] R. H. Lambert and A. J. Bell, "Blind separation of multiple speakers in a multipath environment," IEEE international conference on acoustic, speech and signal processing, ICASSP-97, Vol.1, Apr. 1997, PP. 423-426.

[Lee 1997] T. Lee, A. J. Bell and R. Orglmeister, "Blind source separation of real world signals," International conference on neural network, Vol.4, Jun. 1997, PP. 2129-2134.

[Lee 1997 B] T. Lee, A. J. Bell and R. H. Lambert, "Blind separation of delayed and convolved sources," In Advances in Neural Information Processing System 9, MIT press, Cambridge, 1997, pp.758-764.

[Ma 2004] N. Ma, M. Bouchard and R. Goubran, "Perceptual Kalman filtering for speech enhancement in colored noise", Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2004, vol. 1, pp.717-720, Montreal, May 2004.

[Makino 2005] S. Makino, H. Sawada, R. Mukai and S. Araki, "Blind source separation of convolutive mixtures of speech in frequency domain," IEICE Trans. Fundamentals, Vol. E88-A, No.7, July 2005.

[Mathews 2001] V.J. Mathews and G.L. Sicuranza, Polynomial Signal Processing, John Wiley &Sons, New York, 2001.

[Mayyas 2004] K. Mayyas and T. Aboulnasr, "Reduced complexity Transform-domain adaptive algorithm with selective coefficient update," IEEE Transactions on Circuits and Systems-II, vol. 51, no 1, pp. 136-142, March 2004 .

[Mitianoudis 2003] N. Mitianoudis and M. E. Davies, "Using beamforming in the audio source separation problem," 7th International Symposium on Signal Processing and its Applications, Paris, July, 2003.

[Murata 2001] N. Murata, S. Ikeda and A. Ziehe, "An approach to blind source separation based on temporal structure of speech signals," Neurocomputing, Vol. 41, No. 1-2, pp. 1-24, Oct.2001.

[Naylor 2003] P. Naylor and W. Sherliker, "A short-sort M-Max NLMS partial update adaptive filter with applications to echo cancellation," Proceedings of Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP 2003, Vol. 5, pp. 373-376.

[Papoulis 1991] A. Papoulis, "Probability, random variables, and stochastic processes," McGraw-Hill, 3rd edition, 1991.

[Parra 2000] L. Parra and C. Spence, "Convolutive blind separation of nonstationary sources," IEEE Trans. Speech Audio Process., Vol.8, No. 3, pp.320-327, May 2000.

[Parra 2001] L. Parra and C. Alvino, "Geometric source separation: merging convolutive source separation with geometric beamforming," Proceedings of the 2001 IEEE signal processing society workshop, Neural Networks for signal processing, Sep. 2001, pp.273-282.

[Parra 2002 A] L. Parra and C. Fancourt, "An adaptive beamforming perspective on convolutive blind source separation," Book chapter in "Noise Reduction in Speech Applications", Ed. Gillian Davis, CRC Press LLC, 2002.

[Parra 2002 B] L. Parra and C. Alvino, "Geometric source separation: merging convolutive source separation with geometric beamforming," IEEE Trans. on Speech and Audio Processing, Vol. 10, No. 6, Sep. 2002, pp352-362.

[Pham 1992] D.T. Pham, P. Garrat and J. Jutten, "Separation of a mixture of independent sources through a maximum likelihood approach," In Proc. EUSIPCO, PP. 771-774, 1992.

[Pitas 1989] I. Pitas, "Fast algorithms for running ordering and max/min calculation," IEEE Trans. Circuits Syst. , Vol. 36, pp. 795-804, Jun. 1989.

[Saruwatari 2002] H. Saruwatari, T. Kawanura, K. Sawai, A. Kaminuma and M. Sakata, "Blind source separation based on fast-convergence algorithm using ICA and beamforming for real convolutive mixture," IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002, (ICASSP '02), Vol.1, pp.921 -924.

[Saruwatari 2006] H. Saruwatari, T. Kawamura, T. Nishikawa, A. Lee and K. Shikano, "Blind source separation based on a fast-convergence algorithm combining ICA and beamforming," IEEE Trans. Speech Audio Processing, Mar. 2006.

[Sawada] http:// www.kecl.ntt.co.jp/icl/signal/sawada.

[Sawada 2004 A] H. Sawada, R. Mukai, S. Araki and S. Makino, "A robust and precise method for solving the permutation problem of frequency domain blind source separation," IEEE Trans. Speech Audio Process., Vol. 12, pp. 530-538, Sept. 2004.

[Sawada 2004 B] H. Sawada, R. Mukai, S. Araki, and S. Makino, "Convolutive blind source separation for more than two sources in the frequency domain, " In Proc. ICASSP2004, vol. III, pp. 885-888, May 2004.

[Schmidt 1986] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," IEEE Trans. Antennas Propag., Vol. SP-34, No. 3, pp.276-280, March 1986.

[Schobben 1999] D. W. E. Schobben and P. C. W. Sommen, "A new algorithm for joint blind signal separation and acoustic echo canceling," Fifth international Symposium on signal processing and its applications, ISSPA'99, Brisbane, Australia, August 1999.

[Schobben 2002] D.W.E. Schobben and P.C.W. Sommen, "A Frequency domain blind signal separation method based on decorrelation," IEEE Transactions on signal processing, Vol.50, No. 8, August 2002.

[Smaragdis 1998] P. Smaragdis, "Blind separation of convolved sound mixtures in the frequency domain," Proc. Int. Workshop on Independence and Artificial Neural Networks, Tenerife, Spain, Feb. 1998.

[Sun 2001] X. Sun and S.C. Douglas, "A natural gradient convolutive blind source separation algorithm for speech mixtures," In Proc. ICA, San Diego, CA, Dec., 2001, pp.59-64.

[Torkkola 1996 A] K. Torkkola, "Blind separation of delayed sources based on information maximization," IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-96. Volume: 6 , 7-10 May 1996 Page(s): 3509 –3512.

[Torkkola 1996 B] K. Torkkola, "Blind separation of convolved sources based on information maximization," IEEE Workshop on Neural Networks for Signal Processing, Japan,1996, pp. 423 –432.

[Veen 1988] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," IEEE ASSP Mag., Apr. 1988, pp.4-24.

[Virag 1999] N.. Virag, "Single Channel Speech Enhancement Based on Masking Properties of the Human Auditory System", IEEE Trans. Speech Audio Processing, Vol. 7, No. 2, pp.126-137, Mar. 1999.

[Weinstein 1993] E. Weinstein, M. Feder and A. Oppenheim, "Multi-channel signal separation by decorrelation," IEEE Trans. Speech and Audio Processing, Vol. 1, No.4, Oct. 1993.

[Werner 2001] S. Werner, J. Apolinário and M.L.R. de Campos, "The data-selective constrained Affine Projection algorithm," Proc. of the IEEE Conf. on Acoustics, Speech and Signal Processing, ICASSP 2001, vol. 6, pp. 3745-3748.

[Werner 2004] S. Werner, M. L.R. de Campos and Paulo Diniz, "Partial update NLMS algorithms with data-selective partial updating," IEEE Transactions on Signal Processing, vol. 52, no. 4, pp. 938-949, April 2004.

[Zwicker 1990] Zwicker E and Fastl H, "Psychoacoustics," Springer-Verlag, Berlin, Germany, 1990

[Zwicker 1991] E. Zwicker and U.T. Zwicker, "Audio engineering and psychoacoustic: matching signals to the final receiver, the human auditory system." Journal of the Audio Engineering Society, Vol. 39, No.3, Mar, 1991, pp. 115-126.