A COMPARISON OF CONVOLUTIVE BLIND SOURCE SEPARATION ALGORITHMS APPLIED TO SPEECH SIGNALS

by

ANDREW JOHN PATTERSON, B.S.E.E., B.S.C.S., M.S.E.E.

A DISSERTATION

IN

ELECTRICAL ENGINEERING

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

DOCTOR OF PHILOSOPHY

Approved

Tanja Karp
Chairperson of the Committee

Sunanda Mitra

Brian Nutter

Peter Westfall

Accepted

John Borrelli
Dean of the Graduate School

May, 2006

ACKNOWLEDGEMENTS

I would first like to thank my committee members for their patience and comments throughout writing this dissertation. I would especially like to thank Dr. Tanja Karp for her guidance and willingness to always discuss any questions I had.

I would also like to thank my wonderful parents whose love and support has been constant and unwavering. My in-laws have also been great to me throughout this dissertation. Thank you for all the encouragement, but most of all thank you for not coming close to my computer while I finished!!

I thank God for giving me the wonderful opportunity and experience I had at Texas Tech, and for providing wisdom and encouragement when I needed it most.

Lastly and most importantly I would like to thank my loving wife. Her support and encouragement was crucial in me finishing this dissertation. Her patience with me while I spent the late nights was unbelievable. Thank you. I owe you one!!

CONTENTS

ABSTRACT

Blind source separation aims at estimating a number of unobserved source signals from several observed mixtures of those source signals. In the case of acoustic applications, the sources are people speaking, and the mixtures are microphone recordings. Many different algorithms have been proposed to solve this problem. There is, however, a need for a more involved comparison of the performance of the different algorithms. This dissertation examines the performance of several blind source separation algorithms applied to speech signals.

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

Imagine being at a loud party where one is trying to hold a conversation with a friend. Despite the fact that your ears receive a complex mixture of your friend's voice with the other sounds coming from the room, you are capable of focusing on only what your friend is saying. It is also possible to eavesdrop on other's conversations or even listen to the music. A computer, however, loses the ability to track an individual talker when other sounds are picked up by a microphone.

One of the more popular approaches that attempts to solve the cocktail party problem is called *blind source separation*. Blind source separation is defined as the process of estimating N_s zero-mean, statistically independent sources, $\mathbf{s}(n) = [s_1(n), s_2(n), \ldots, s_{N_s}(n)]^\mathsf{T}$ from N_x observed signals, $\mathbf{x}(n) = [x_1(n), x_2(n), \ldots, x_{N_x}(n)]^\mathsf{T}$. In audio applications, the source signals are people speaking, and the sensors are microphones. Blind in this context stresses that the original sources $\mathbf{s}(n)$ are unobservable and little to no prior information is known about them or the mixing process.

A block diagram of a generic BSS system is shown in Figure 1.1. As depicted in Figure 1.1, the measured signals are assumed to be generated by an unknown mixing of the original source signals. The goal of blind source separation is then to invert this mixing process, resulting in estimates of the sources.

There are two basic approaches to model the mixing process. The first is the instantaneous case, in which the mixing process is modeled as a matrix. In this situation,each microphone records a sum of the original source signals.

The second situation of BSS is that of convolutive mixing, in which the mixing process is modeled as filters. In this case each microphone records scaled and delayed versions of every source signal. The delayed versions of the signals are due to reflections of the signals from the room acoustics.

Blind source separation is sometimes used interchangeably with Independent Component Analysis (ICA). Blind source separation refers to the process of blindly estimating source signals solely from an observation of a mixture of those signals. Independent component analysis is the underlying principle generally utilized in order to solve the blind source separation problem [23]. Independent component analysis is an extension of Principle Components Analysis(PCA). PCA imposes only second-order independence between components, while constraining the directions of the components to be orthogonal. ICA, on the other hand, imposes statistical independence between the components, but has no orthogonality constraint. Despite these small differences, BSS and ICA refer to essentially the same process. Thus this dissertation will use the terms BSS and ICA interchangeably, just as is done in current literature about these topics.

ICA will be formally introduced in Chapter II.

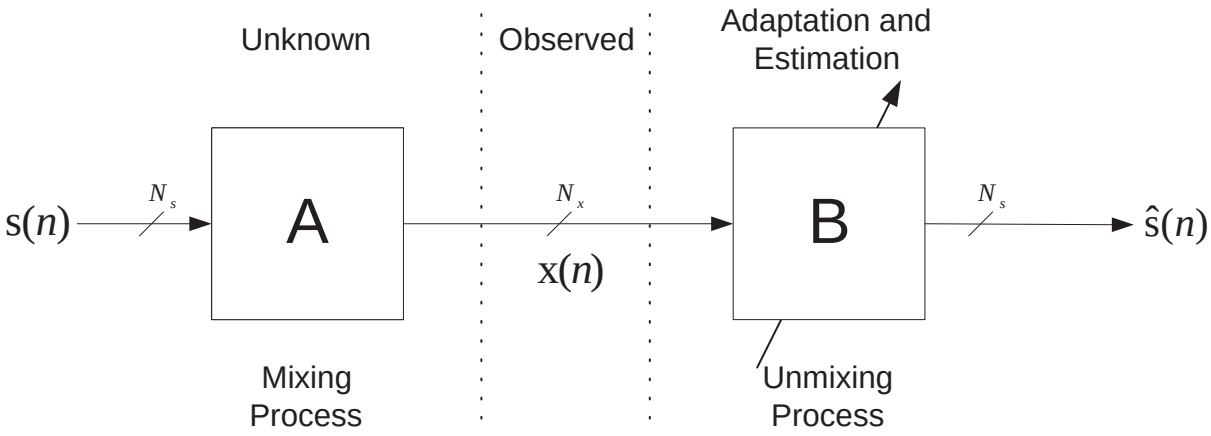


Figure 1.1: Block diagram of a generic BSS model.

1.1 Applications

Applications of BSS and ICA are described in detail in [26] and in Chapter 6 of [23]. A brief overview of some of the applications of ICA and BSS will be presented here.

One application is the separation of EEG signals. EEG data signals consist of voltage measurements of brain activity. Many different sensors are placed on the head of a patient. The captured voltages are assumed to be generated by some mixture of brain activity. ICA is used to gain insight on the brain activity by extracting these individual signals. Since the signals of interest are recorded at each of the electrodes, but have a different weight at each electrode, ICA can be used to find these weights, thus allowing for the extraction of the signals of interest.

Yet another application of ICA is telecommunications. In CDMA systems, multiple users share the same bandwidth of a channel and transmit simultaneously. The users are discriminated by a spreading code. The spreading code for each user is orthogonal, thus allowing for separation at the receiver. The problem, however, is that due to multipath fading, the signals at the receiver may no longer be orthogonal. ICA can be used in this case to help improve the SNR at the receiver.

Convolutive blind source separation has also be applied to music. Many aspiring musicians may want to learn a song that is recorded and mixed to a CD. It would be advantageous to be able to extract only a specific instrument such as the guitar, voice, etc. In [16] Douglas applied convolutive blind source separation in order to separate two voices from an *a cappella* recording. Vincent used ICA to separate different musical instruments of audio soundtracks to be able to interactively modify musical recordings [48]. This can be used for learning to play a song or for karaoke. Plumbley et al. applied audio source separation for the automatic transcription of music in order to write the notes of recorded music [38].

1.2 Problem Description

Many algorithms are available today that attempt to solve the blind source separation problem. Despite this fact, there has been little literature that has actually rigorously tested these algorithms. In fact, many of the algorithms seem to have only been evaluated for synthetic scenarios, e.g. no "real-world" data. This dissertation

examines five different convolutive blind source separation algorithm's ability to separate mixed speech signals. The algorithms examined were chosen such that they give a fair representation of the various algorithms currently available. Most algorithms today can be categorized as either a time-domain (TD) or frequency-domain (FD) algorithm, and within these main categories, they can then again be subdivided into information-theoretic (IT) or non-information-theoretic (NIT) algorithms. Given this classification scheme, a popular algorithm within each class was chosen in order to evaluate the performance of the algorithm in several different scenarios:

1. Ability of the algorithm to deal with simple instantaneous or delayed mixing.

2. Ability of the algorithm to deal with convolutive mixing.

3. Ability of the algorithm to deal with similarity between the original sources. Specifically, this dissertation addresses the scenarios in which the speakers have similar spectral content or similar temporal characteristics. To test these scenarios, simulations were performed to test the ability of the different algorithms to deal with the same speaker talking at the same time (although this has little physical meaning unless one is trying to separate the voices of twins) and with different speakers saying the same sentence.

4. Ability of the algorithm to deal with "real world" mixing. Again, many of the algorithms present today have had limited evaluation on actual recordings. This dissertation examined the performance of each of the algorithms for several different recording situations.

5. Any increase in performance by having more microphones than speakers.

The set of chosen algorithms evaluated in this dissertation is given in Table 1.1. The way these algorithms were chosen will be described in Chapter IV.

Table 1.1: Set of chosen algorithms for evaluation.

Method	**Section Covered**	**Category**
INFOMAX [44]	4.1.1	TD-IT
NHBSS [19]	4.2.1	TD-IT
GENSOS [11]	4.2.2	TD-NIT
BSU [31]	4.3.1.1	FD-IT
JBD [34]	4.3.2	FD-NIT

1.3 Outline

The remainder of this dissertation will be organized in the following manner. An introduction to blind source separation will be given by beginning with instantaneous mixtures in Chapter II. The model will then be extended in order to incorporate a more realistic model in Chapter III by introducing convolutive mixing. Due to the strong relationship between blind source separation and blind deconvolution, this chapter will also briefly cover blind deconvolution methods. In Chapter IV, a categorization of blind source separation algorithms chosen to be evaluated will be done. An introduction and mathematical description of these algorithms will also be given in this chapter. In Chapter V, the results of this dissertation will be presented along with a final comparison between the different selected algorithms. Finally we will conclude this dissertation in Chapter VI along with give some directions for future work.

CHAPTER II
INSTANTANEOUS MIXTURES

In the simplest blind source separation situation, each mixture is modeled as a linear combination of the original source signals. This is the so-called instantaneous mixing model as shown in Figure 2.1. In this case, the observed mixtures can be written as

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) + \mathbf{n}(n) \tag{2.1}$$

where $\mathbf{A}$ is the a mixing matrix and $\mathbf{n}(n)$ is additive noise. The additive noise is still

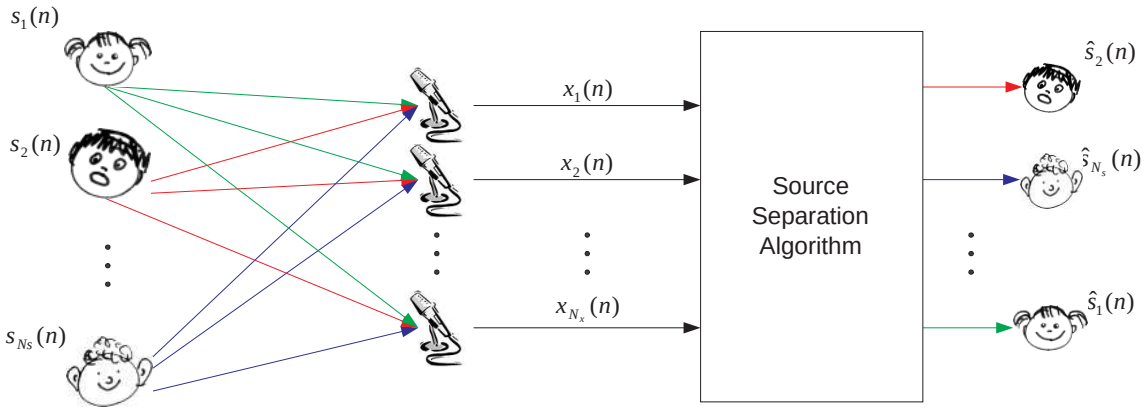


Figure 2.1: Block diagram of instantaneous BSS model.

a very difficult problem to solve. In cases where there are more microphones than sources, i.e. $N_x > N_s$, it is possible to reduce the noise via Principal Component Analysis (PCA) [23]. However, there are not always more microphones than sources, so it will be assumed throughout this paper that the additive noise is negligible, i.e. the SNR is high. It will also be assumed that the number of sources equals the number of microphones, i.e., $N_s = N_x$. In this instantaneous mixture case, the goal of blind source separation is to estimate an unmixing matrix $\mathbf{B} \approx \mathbf{A}^{-1}$. In other words, blind

source separation attempts to find a matrix $\mathbf{B}$ such that

$$\hat{\mathbf{s}}(n) = \mathbf{B}\mathbf{x}(n) \tag{2.2}$$

where $\hat{\mathbf{s}}(n)$ is an estimate of the original source signals $\mathbf{s}(n)$.

Because the problem is solved blindly, there are a few ambiguities that will hold [26]. Since $\hat{\mathbf{s}}(n)$ and $\mathbf{A}$ are both unknown, we are unable to determine the energies (variances) of the original signals. The reason is that any source signal multiplied by a scalar could always be canceled by dividing the corresponding column of the mixing matrix $\mathbf{A}$. Because of this indeterminacy, the variance of each source signal is assumed to be 1, i.e. $E\{s_i(n)^2\} = 1$. Another ambiguity is the order of the estimated source signals. Without *a priori* information, it is impossible to determine the order of the original source signals. Defining a permutation matrix $\mathbf{P}$, where each column of $\mathbf{P}$ has only one non-zero entry, then a re-ordering of the source signals can be written as

$$\mathbf{s}'(n) = \mathbf{P}\mathbf{s}(n). \tag{2.3}$$

The mixed signals are then written as

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}'(n) = \mathbf{A}\mathbf{P}^{-1}\mathbf{P}\mathbf{s}(n) \tag{2.4}$$

where $\mathbf{A}\mathbf{P}^{-1}$ is just a new unknown mixing matrix.

These two indeterminacies are generally rewritten into one formal statement:

$$\hat{\mathbf{s}}(n) = \mathbf{B}\mathbf{A}\mathbf{s}(n) = \mathbf{\Delta}\mathbf{P}\mathbf{s}(n) \tag{2.5}$$

where $\mathbf{\Delta}$ is a nonsingular diagonal matrix. In other words the goal of blind source separation is to find an estimate of the original source signals up to a scaling and permutation.

One method, perhaps the most widely used, that attempts to solve this problem is called *Independent Component Analysis* (ICA). ICA exploits the statistical independence between the original source signals in order to separate them from the observed

mixtures. The various methods used to perform ICA basically differ in the way that they measure the statistical independence of the measured signals. A gradient method is used to maximize this statistical measure. Some of the most popular solutions fall into several different categories [26]. The first is ICA by the maximization of non-gaussianity [14, 26, 27]. In this method, the elements of the unmixing matrix $\mathbf{B}$ are updated so that the "gaussianity" of $\mathbf{x}(n)$ is minimized. The motivation behind this method is that the mixed signals are assumed to be more Gaussian than the original signals due to the central limit theorem. Thus maximizing the "nongaussianity" of the mixed signals is in fact separating them. The measure of nongaussianity is usually calculated by either the kurtosis [27] or by negentropy [26].

Another popular approach of ICA is by maximum likelihood (ML) [5, 36] estimation or the very popular infomax principle [8]. In ML, the parameters of $\mathbf{B}$ are found by using the ML principle, whereas the infomax aims to maximize the information between the input and the output. Cardoso showed in [13] that maximum likelihood is equivalent to maximizing the information transferred in the network.

Yet another approach is ICA by the minimization of mutual information. It is well known that mutual information is a measure of dependence between random variables. ICA via the minimization of mutual information consists of finding $\mathbf{B}$ such that the mutual information between the individual source signals $s_i(n)$ is minimized. In practice there is need to estimate the mutual information between the source signals. Many of these estimates of the mutual information lead to algorithms that are similar if not equal to the ones previously mentioned [26], thus they are intimately connected to those that use nongaussianity, ML, or the infomax principle.

Finally, other approaches estimate the desired source signals by using cumulant tensors. This approach separates the signals by forcing fourth-order cumulant tensors to be zero, i.e., higher-order decorrelation. The idea is that for statistically independent signals, all of the cross-cumulants between the sources should be zero. Therefore, forcing the fourth order cumulants to be approximately zero, is in fact forcing the

observed signals to be as independent as possible, thus achieving separation. The most popular algorithm in this family is Cardoso's JADE algorithm proposed in [15].

Instantaneous blind source separation has been around for about 20 years. Many solutions have been found to the blind source separation problem, and comparisons of these algorithms are readily available. Some comparisons can be found in [26, 35, 40, 46].

CHAPTER III
CONVOLUTIVE MIXTURES

Although well documented and the results are promising, the instantaneous mixture model is rather simple and does not hold for most real world applications, especially for real room acoustic environments. Suppose instantaneous blind source separation is to be utilized in order to solve the cocktail party problem, as discussed in Chapter I, by placing a microphone array in the room. The instantaneous mixture model fails to hold in this scenario because each microphone not only picks up the sound signals, but also delayed and attenuated versions of the same signals, due to reflections from the floor, walls, and the acoustic environment. In fact, each microphone receives filtered versions of the different signals, as shown in Figure 3.1. Thus a better model is the convolutive model:

$$\mathbf{x}(n) = \sum_{p=0}^{P} \mathbf{A}_p \mathbf{s}(n-p) + \mathbf{n}(n). \tag{3.1}$$

where $\mathbf{A}_p$ is a mixing matrix at time lag p. The goal of blind source separation is to find an FIR inverse model[1]

$$\hat{\mathbf{s}}(n) = \sum_{p=0}^{L} \mathbf{W}_p \mathbf{x}(n-p) \tag{3.2}$$

so that $\hat{\mathbf{s}}(n)$ is an estimate of the original source signals.

A clarification is needed here of what is meant by an estimate of the original source signals. In the case of stationary, independent and identically distributed (i.i.d). signals, it is possible not only to separate the signals but also remove the reverberation due to the room acoustics. In other words, the estimated signals are simply a scaled, permuted and delayed version of the original sources. Speech signals, however, are not stationary and derevereberation is difficult if not impossible by

[1]Throughout the literature, it is common to use $\mathbf{B}$ for the instantaneous model and $\mathbf{W}$ for the convolutive model.

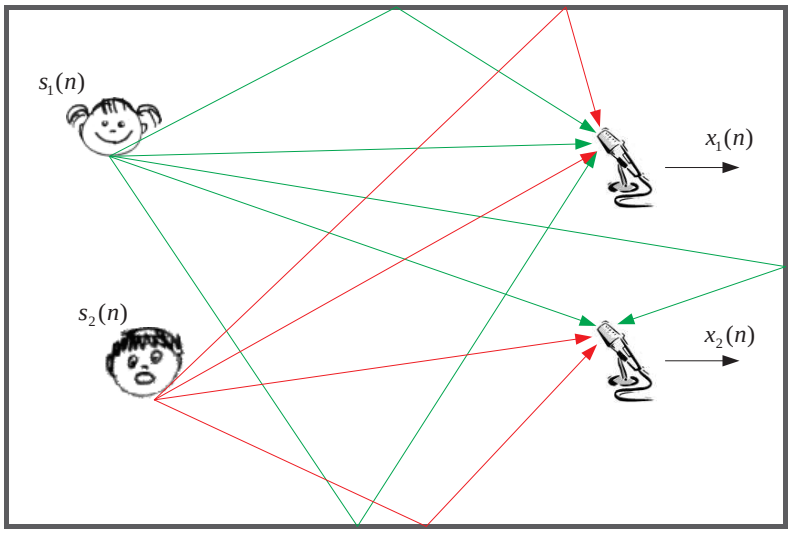


Figure 3.1: An example of the multipath problem in convolutive mixing. Each microphone not only picks up the original source signals, but possibly also delayed and scaled version of them as well.

current methods. Since this dissertation deals with speech signals, the statement "a good estimate of the original source signals" refers to separation quality, meaning there is only one source present in each separated signal. So "a good estimate" includes a filtered version of the original source signals.

Current blind source separation techniques for convolutive mixtures basically fall into two main categories: time-domain and frequency-domain algorithms. The algorithm will be classified into one of these categories depending on where the coefficient updates for the unmixing matrix occur. Time-domain methods are thus those in which $\mathbf{W}_p$ is updated in the time-domain. Time-domain methods were the first efforts in tackling convolved mixtures and were inspired by the already present blind deconvolution methods that will be discussed in section 3.1.

One problem with time-domain methods is that they tend to be complex computationally due to the relationship of filter coefficients with each other [42]. Time-domain methods generally aim to update the unmixing matrix by information theoretic constraints [37]. By optimizing information theoretic cost functions, the spatial depen-

dencies between the estimated signals are removed, but the temporal dependencies within a signal is also removed [42]. This gives rise to spectral flattening of the signals, which is usually undesirable, especially when dealing with speech. These problems can be overcome by moving to the frequency-domain, where the problem is transformed to multiple independent instantaneous blind source separation problems.

In frequency-domain approaches, the filter coefficients are adapted in the frequency-domain. In this approach the estimated source signals may be found in either the time-domain or the frequency-domain. Despite this fact, they will still be considered as frequency-domain approaches because the optimization is done in the frequency-domain. Frequency-domain approaches can be less computationally expensive, but they pay for this by introducing a new dilemma: an inherent permutation problem. The permutation problem of blind source separation has already been described for the case of instantaneous mixtures. Permutations pose a larger problem when moving to the frequency-domain. By transforming to the frequency-domain, the blind source separation problem can be treated as multiple instantaneous problems, one for each frequency bin. These problems are independent between frequency bins, but now there may be a different arbitrary permutation in each bin. Thus upon solving the multiple instantaneous problems in the frequency-domain, the reconstructed signals are simply remixed unless all the permutations are properly aligned. Many current frequency-domain approaches simply use existing instantaneous algorithms and aim to solve the permutation problem [32, 42, 45].

Another drawback of frequency-domain methods appears when there is insufficient data in order to cover long mixing filter lengths. Generally the blind source separation problem is transformed to the frequency-domain via the short-time Fourier transform (STFT). The longer the mixing filter, the longer the frame size of the STFT must be, but little overlap between frames is generally desired. As the frame increases, the number of data points in each frequency bin decreases, resulting in insufficient amounts of data to train the unmixing filters $\mathbf{W}_p$.

Before we introduce some of the current convolutive blind source separation algorithms, we will begin with single input blind deconvolution.

3.1 Blind Deconvolution

It is often advantageous to introduce convolutive blind source separation by beginning with blind deconvolution. Blind deconvolution, often called blind equalization in communication areas [9, 22], consists of only one input signal and one observed signal. The input signal is unobserved and mixed with scaled and delayed versions of itself. Blind deconvolution aims to estimate the original source from the observed signal only, without knowing the convolving system. In this case, an estimate is the original source signal, possibly scaled and delayed, with no filtering effects.

In noiseless blind deconvolution [9,16,23,24], an observed discrete-time signal $x(n)$ is assumed to be generated by passing an unknown i.i.d. source signal through some unknown, causal, linear time-invariant filtering process, i.e.

$$x(n) = \sum_{p=0}^{\infty} a_p s(n-p) \tag{3.3}$$

where a_p is the coefficient of the impulse response at time lag p. Equalization aims to retrieve the original source signal by means of filtering $x(n)$ with another discrete-time filter,

$$\hat{s}(n) = \sum_{p=-\infty}^{\infty} w_p x(n-p) \tag{3.4}$$

where $w_p(n)$ is the impulse response of the equalizer. In practice $w(n)$ is usually chosen to be a causal Finite Impulse Response (FIR) filter of length $L+1$, i.e.

$$\hat{s}(n) = \sum_{p=0}^{L} w_p x(n-p). \tag{3.5}$$

Probably the most widely used adaptation method is an Least Mean Square (LMS)-type gradient approach given by

$$w(n+1) = w(n) \pm \mu \frac{\partial J}{\partial w}. \tag{3.6}$$

Applying the chain rule to Eq. (3.6) leads to

$$w(n+1) = w(n) \pm \mu \frac{\partial J}{\partial \hat{s}} \frac{\partial \hat{s}}{\partial w}. \tag{3.7}$$

The minimum mean squared error (MMSE) estimate for w_p is found by optimizing the cost function

$$J_{MMSE} = E\left\{|s(n) - x(n)|^2\right\}. \tag{3.8}$$

MMSE estimators are optimal when the source signal is Gaussian [29]. However, if prior knowledge of the probability distribution function (pdf) of the source signal is known, then one can make use of Bussgang algorithms. The Bussgang property states that [25]

$$E\left\{\hat{s}(n)\hat{s}(n-p)\right\} = E\left\{\hat{s}(n)g(\hat{s}(n-p))\right\} \tag{3.9}$$

where $g(\hat{s})$ is a suitable zero-memory nonlinearity. Applying the Bussgang property to the MMSE cost function gives rise to the non-Gaussian MMSE (MMSE-NG) [29] cost function

$$J_{MMSE-NG} = E\left\{|\hat{s}(n) - g(s(n))|^2\right\}. \tag{3.10}$$

However this is the supervised version and requires a training sequence. If this is to be a blind algorithm, the source signal $s(n)$ is replaced with its estimate $\hat{s}(n)$, i.e.

$$J_{BLMS} = E\left\{|\hat{s}(n) - g(\hat{s}(n))|^2\right\} \tag{3.11}$$

which will be called the blind LMS cost function.

In Bussgang type algorithms, the unmixing filter is updated by

$$w_p(n+1) = w_p(n) - \mu(n)\frac{\partial J(\hat{s}(n))}{\partial w_p} = w_p(n) + \mu(n)g(\hat{s}(n))x^*(n-p) \tag{3.12}$$

where $g(\hat{s}) = -\frac{\partial J(\hat{s})}{\partial \hat{s}}$, $\mu(n)$ is a series of adaptation steps, and * represents complex conjugation. The function $g(\hat{s})$ is chosen depending on the source signal statistics. The optimal choice of $g(\hat{s})$ is [9]

$$g(\hat{s}) = \frac{-p_s'(s)}{p_s(s)} \tag{3.13}$$

where $p_s(s)$ is the pdf of the source signal. One of the best performing Bussgang type algorithms, especially for subgaussian communication signals (e.g. M-ary QAM), is the Godard algorithm [22]. Godard used the cost function

$$J(\hat{s}(n)) = E\left\{\left[|\hat{s}(n)|^p - R_p\right]^2\right\} \tag{3.14}$$

where p is some positive integer, and R_p is chosen so that the gradient of the cost function J is zero for perfect equalization. Godard showed that this value can be calculated as

$$R_p = \frac{E\left\{|s(n)|^{2p}\right\}}{E\left\{|s(n)|^p\right\}}. \tag{3.15}$$

This makes

$$g(\hat{s}) = \frac{\partial J(\hat{s})}{\partial \hat{s}} = 2E\left\{(|\hat{s}(n)|^p - R_p)\frac{\partial(|\hat{s}(n)|^p - R_p)}{\partial \hat{s}}\right\} \tag{3.16}$$

$$= 2pE\left\{(|\hat{s}(n)|^p - R_p)|\hat{s}(n)|^{p-2}\hat{s}(n)\right\}. \tag{3.17}$$

The constant-modulus algorithm (CMA) is a member of the family of Godard algorithms that is best for M-ary QAM signals. The CMA algorithm is obtained by setting $p = 2$ in Eq. (3.14) and (3.15).

3.1.1 Extension Of Blind Deconvolution To BSS

Blind deconvolution provides a good starting point in which to begin discussing blind source separation of convolutive mixtures. Obviously, the blind deconvolution algorithms presented in the above section are not applicable to systems in which there are multiple signals and observations. Another problem of blind deconvolution algorithms is the assumption of an independent and identically distributed (i.i.d.) source signal. Although communication signals may fit this assumption, speech definitely does not.

It is obvious to see that blind deconvolution and blind source separation are similar tasks which both aim to blindly invert a linear system. In fact it, is possible to translate blind source separation algorithms to blind deconvolution tasks. Douglas and Haykin describe this process in [16, 18], where they first relate blind source

separation under a circulant mixing matrix to deconvolution. They then apply a procedure to translate the blind source separation task to a true deconvolution process. This results in the blind deconvolution algorithm

$$w_p(n+1) = w_p(n) + \mu[w_p(n) - g(\hat{s}(n-L))u^*(n-p)] \tag{3.18}$$

where $g(\hat{s})$ is suitable nonlinearity that depends on the source signal's statistics, * denotes complex conjugation, and $u(n)$ is found by

$$u(n) = \sum_{i=0}^{L} b^*_{L-i}\hat{s}(n-i). \tag{3.19}$$

3.1.2 Multichannel Blind Deconvolution

In [29,30], Lambert effectively extends Bussgang single channel blind equalization techniques to a multichannel cost function. Remember that, for the single channel case, the blind LMS cost function, given in Eq. (3.11), was $J_{BLMS} = E\{|\hat{s}(n) - g(\hat{s}(n))|^2\}$. Lambert notes that for multichannel blind equalization, the cost function can be formed as the sum of all the cost functions for each channel, i.e.

$$\mathbf{J} = \sum_{i=1}^{N_x} J_i(x_i) \tag{3.20}$$

resulting in the multichannel blind LMS (MBLMS) cost function of

$$\mathbf{J}_{MBLMS} = \text{tr}\left[E\left\{[\hat{\mathbf{s}}(n) - g(\hat{\mathbf{s}}(n))][\hat{\mathbf{s}}(n) - g(\hat{\mathbf{s}}(n))]^H\right\}\right] \tag{3.21}$$

where tr [] denotes the trace of the argument, and g is the Bussgang nonlinearity. The corresponding update for the MBLMS is

$$\underline{\mathbf{W}}(n+1) = \underline{\mathbf{W}}(n) + \mu(\hat{\mathbf{s}}(n) - g(\hat{\mathbf{s}}(n)))\mathbf{x}^*(n) \tag{3.22}$$

where $\underline{\mathbf{W}}$ is an FIR matrix, i.e.

$$\underline{\mathbf{W}} = \begin{bmatrix} w_{11} & \cdots & w_{1,N_s} \\ \vdots & & \vdots \\ w_{N_s,1} & \cdots & w_{N_s,N_s} \end{bmatrix} \tag{3.23}$$

and w_{ij} is the channel impulse response between source j and sensor i.

3.1.3 Multichannel Blind Deconvolution Using the Natural Gradient

Amari et al. [7] make use of the connection between the blind source separation task and the blind deconvolution task to extend the algorithms for instantaneous mixtures into multichannel blind deconvolution algorithms [16]. Restating the discrete-time source and observed signals as $\mathbf{s}(n) = [s_1(n), s_2(n), \ldots, s_{N_s}(n)]^\mathsf{T}$ and $\mathbf{x}(n) = [x_1(n), x_2(n), \ldots, x_{N_s}(n)]^\mathsf{T}$ respectively, they model the convolutive mixing as

$$\mathbf{x}(n) = \sum_{p=0}^{\infty} \mathbf{A}_p \mathbf{s}(n-p) \tag{3.24}$$

where

$$\mathbf{A}_p = \begin{bmatrix} a_{11p} & \cdots & a_{1,N_s,p} \\ \vdots & & \vdots \\ a_{N_s,1,p} & \cdots & a_{N_s,N_s,p} \end{bmatrix}. \tag{3.25}$$

is an instantaneous mixing matrix at lag p. The goal is then to find an unmixing matrix $\mathbf{W}_p$ so that

$$\hat{\mathbf{s}}(n) = \sum_{p=0}^{L} \mathbf{W}_p(n) \mathbf{x}(n-p) \tag{3.26}$$

is an estimate of the original source signals.

This can be more conveniently represented in operator form by the use of z-transforms. Equations (3.24) and (3.25) then become

$$\begin{aligned} \mathbf{x}(n) &= \mathbf{A}(z)\mathbf{s}(n) \\ \hat{\mathbf{s}}(n) &= \mathbf{W}(z,n)\mathbf{x}(n) = \mathbf{C}(z,n)\mathbf{s}(n), \end{aligned}$$

where $\mathbf{W}(z,n) = \sum_{p=-\infty}^{\infty} \mathbf{W}_p(n) z^{-p}$, $\mathbf{A}(z)$ and $\mathbf{C}(z,n) = \mathbf{W}(z,n)\mathbf{A}(z)$ are the z-transforms of the channel, demixing filter, and combined system, respectively. The delay operator z^{-1}, such that $s_i(n)z^{-p} = s_i(n-p)$.

Amari et al. derive the updates for $\mathbf{W}_p$ using a natural gradient search method [4] with the cost function

$$J(\mathbf{W}(z,n)) = -\frac{1}{2\pi j} \oint \log |\det \mathbf{W}(z,n)| z^{-1} \, \mathrm{d}z - \sum_{i=1}^{N_s} \log p_i(\hat{s}_i(n)) \tag{3.27}$$

where $p_{s_i}(s_i)$ is the pdf of the i^{th} source signal and $j = \sqrt{-1}$. To find the natural gradient algorithm that minimizes the expected value of the cost function $J(\mathbf{W}(z,n))$, the total differential in $\mathrm{d}J(\mathbf{W}(z,n))$ when $J(\mathbf{W}(z,n))$ undergoes an infinitesimal change $\mathrm{d}\left(J\mathbf{W}(z,n)\right)$ [6] must be determined. First define

$$g_i(\hat{s}_i) = -\frac{\mathrm{d}\log p_i(\hat{s}_i)}{\mathrm{d}\hat{s}_i} \tag{3.28}$$

$$\mathbf{g}(\hat{\mathbf{s}}(n)) = [g_i(\hat{s}_i), \ldots, g_{N_s}(\hat{s}_{N_s})]. \tag{3.29}$$

Then

$$\mathrm{d}\left(-\sum_{i=1}^{N_s} \log p_i(\hat{s}_i(n))\right) = \mathbf{g}^{\mathsf{T}}(\hat{\mathbf{s}}(n))\,\mathrm{d}\hat{\mathbf{s}}(n) \tag{3.30}$$

where

$$\mathrm{d}\hat{\mathbf{s}}(n) = \mathrm{d}\mathbf{W}(z,n)\mathbf{x}(n) = \mathrm{d}\mathbf{W}(z,n)\mathbf{W}^{-1}(z,n)\hat{\mathbf{s}}(n). \tag{3.31}$$

Defining

$$\mathrm{d}\mathbf{X}(z,n) = \mathrm{d}\mathbf{W}(z,n)\mathbf{W}^{-1}(z,n) \tag{3.32}$$

$$= \sum_{p=-\infty}^{\infty} \mathrm{d}\mathbf{X}_p(n) z^{-p} \tag{3.33}$$

and combining (3.31) and (3.32) leads to

$$\mathrm{d}\left(-\sum_{i=1}^{N_s} \log p_i(\hat{s}_i(n))\right) = \mathbf{g}^{\mathsf{T}}(\hat{\mathbf{s}}(n))\,\mathrm{d}\mathbf{X}(z,n)\hat{\mathbf{s}}(k). \tag{3.34}$$

Similarly, Amari et al. show that

$$\mathrm{d}\left(-\frac{1}{2\pi j}\oint \log|\det \mathbf{W}(z,n)|z^{-1}\,\mathrm{d}z\right) = \mathrm{tr}\left[\mathrm{d}\mathbf{X}_0(n)\right] \tag{3.35}$$

where $\mathrm{tr}\left[\cdot\right]$ is the matrix trace operation. Combining (3.34) and (3.35) gives

$$\mathrm{d}\left(J\mathbf{W}(z,n)\right) = \mathbf{g}^{\mathsf{T}}(\hat{\mathbf{s}}(n))\,\mathrm{d}\mathbf{X}(z,n)\hat{\mathbf{s}}(n) - \mathrm{tr}\left[\mathrm{d}\mathbf{X}_0(n)\right]. \tag{3.36}$$

The differential given here is in terms of $\mathrm{d}\mathbf{X}(z,n)$. However, Amari et al. show that the natural gradient algorithm can be explicitly given as [7]

$$\mathbf{W}_p(n+1) = \mathbf{W}_p(n) - \mu(n)\left[\frac{\mathrm{d}\left(J\mathbf{W}(z,n)\right)}{\mathrm{d}\mathbf{X}_p(n)}\right]\mathbf{W}(z,n). \tag{3.37}$$

The resulting updates are

$$\mathbf{W}_p(n+1) = \mathbf{W}_p(n) + \mu(n)\left[\mathbf{W}_p(n) - \mathbf{g}(\hat{\mathbf{s}}(n-L))\mathbf{u}^{\mathsf{T}}(n-p)\right] \tag{3.38}$$

$$\mathbf{u}(n) = \sum_{i=0}^{L} \mathbf{W}_{L-i}^{\mathsf{T}}(n)\hat{\mathbf{s}}(n-i) \tag{3.39}$$

where $\mathbf{g}(\hat{\mathbf{s}})$ is the Bussgang nonlinearity depending on the statistics of the source signals and $\mu(n)$ is the step size. The function $\mathbf{g}(\hat{\mathbf{s}})$ is optimal for

$$g(\hat{s}_i) = -\frac{\partial \log(p_i(s_i))}{\partial s_i} = \frac{-p_{s_i}(s_i)}{p_{s_i}(s_i)}. \tag{3.40}$$

An important aspect of this natural gradient method is the notion of equivariance. By post-multiplying both sides of Eq. (3.37) by $\mathbf{A}(z)$ we have

$$\mathbf{C}(z, n+1) = \mathbf{C}(z,n) - \mu(n)\left[\frac{\mathrm{d}\left(J\mathbf{W}(z,n)\right)}{\mathrm{d}\mathbf{X}_p(n)}\right]\mathbf{C}(z,n). \tag{3.41}$$

We can see from Eq. (3.36) that

$$\mathrm{d}\left(J\mathbf{W}(z,n)\right) = \mathbf{g}^{\mathsf{T}}\left\{\mathbf{W}(z,n)\mathbf{x}(n)\right\}\,\mathrm{d}\mathbf{X}(z,n)\mathbf{W}(z,n)\mathbf{x}(n) - \mathrm{tr}\left[\,\mathrm{d}\mathbf{X}_0(n)\right] \tag{3.42}$$

$$= \mathbf{g}^{\mathsf{T}}\left\{\mathbf{W}(z,n)\mathbf{A}(z)\mathbf{s}(n)\right\}\,\mathrm{d}\mathbf{X}(z,n)\mathbf{W}(z,n)\mathbf{A}(z)\mathbf{s}(n) - \mathrm{tr}\left[\,\mathrm{d}\mathbf{X}_0(n)\right] \tag{3.43}$$

$$= \mathbf{g}^{\mathsf{T}}\left\{\mathbf{C}(z,n)\mathbf{s}(n)\right\}\,\mathrm{d}\mathbf{X}(z,n)\mathbf{C}(z,n)\mathbf{s}(n) - \mathrm{tr}\left[\,\mathrm{d}\mathbf{X}_0(n)\right]. \tag{3.44}$$

Thus we can see that the evolution of $\mathbf{C}(z,n)$ does not explicitly depend on the mixing channel $\mathbf{A}(z)$.

Equivariance was first applied to blind source separation in [12] for instantaneous mixtures. The equivariant property for instantaneous mixtures states that the convergence of an equivariant algorithm does not depend explicitly on the mixing process, i.e., does not depend on the mixing matrix $\mathbf{A}$. This means that equivariant algorithms provide uniform performance, and thus convergence can be attained no matter what form the mixing matrix takes. This notion of equivariance can be extended to convolutive mixtures by stating that for an algorithm to be equivariant, its behavior only depends on the combined filter $\mathbf{C}(z,n)$. Amari et al. note, however, that this result

only holds for infinite-impulse response (IIR) equalizers, since the ability of an FIR filter to properly equalize a channel depends not only on the length of the filters but also on the initialization of those filters. They also note that the equivariance of an algorithm does not guarantee that $\mathbf{W}(z, n)$ is adequate to equalize the channel, nor does it guarantee that the convergence does not suffer from poor initializations of the demixing filters.

One drawback of the methods mentioned here is that they both assume that the source signals are i.i.d. This is fine if we are dealing with QAM or other communication signals, but speech does not fall into this category. They can however be applied to signals with temporal dependence and still produce some satisfactory results [16]. One artifact of applying the Amari method is the spectral flattening of the source signals. Despite this fact, multichannel blind deconvolution algorithms have been applied to signals with a dependent temporal structure, especially speech, and they have a reasonable performance [16, 29].

CHAPTER IV
ALGORITHM EVALUATION

As previously discussed, convolutive blind source separation algorithms can be grouped into time-domain (TD) or frequency-domain (FD) approaches. These categories will now be further divided into more subcategories. This will give a justification for the choice of the different algorithms chosen to be evaluated. In [28], Hild categorized almost 90 published techniques according to their structure and criterion. By structure, he meant how the estimated signals are calculated. There are two options for this category: TD and FD. Time-domain methods can then be further divided into subcategories depending on whether they use a feedforward or feedbackward model. The feedforward or feedbackward architecture was not considered as a basis for categorization in this dissertation.

The categorization used to select the algorithms to be evaluated is shown in Figure 4.1.

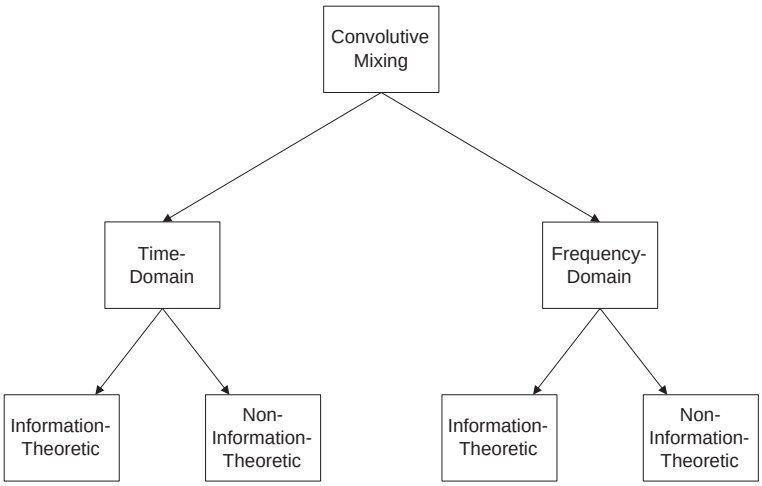


Figure 4.1: Categorization of blind source separation algorithms

4.1 Time-Domain Methods

This section will introduce the TD algorithms that were evaluated for this dissertation. Again an algorithm will be considered a TD one if the updates of the mixing matrix are done in the time-domain.

4.1.1 Torkkola's extension of the Bell/Sejnowski algorithm - INFOMAX Algorithm

One of the seminal papers on information-theoretic, instantaneous based blind source separation is the paper by Bell and Sejnowski [8]. The proposed algorithm by Bell and Sejnowski trained the separating matrix $\mathbf{W}$ based on the information maximization principle. In [44], Torkkola extended the Bell/Sejnowski algorithm to multichannel, convolutive mixing. In his method, he updated the unmixing filter coefficients in the time-domain as well.

Beginning with the original algorithm proposed by Bell and Sejnowski in [8], the restated instantaneous model for discrete-time signals is

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n)$$
$$\hat{\mathbf{s}}(n) = \mathbf{W}\mathbf{x}(n).$$

The separating matrix $\mathbf{W}$ is learned by minimizing the mutual information between the components of $\mathbf{y}(n) = g(\hat{\mathbf{s}}(n)) = \frac{-p_i'}{p_i}$, where $g(\hat{\mathbf{s}})$ is a nonlinear function that is related to the pdf of the source signals. This algorithm can be called a "density matching" technique [45] because, in a sense, $\mathbf{W}$ is learned so that the estimated signals, $\hat{\mathbf{s}}(n)$, have the same pdf as the original sources signals $\mathbf{s}(n)$. The demixing filters are trained by utilizing the stochastic-gradient ascent rule, i.e.,

$$\mathbf{W} = \mathbf{W} + \mu\Delta\mathbf{W}. \tag{4.1}$$

The resulting updates for the separating matrix are

$$\Delta\mathbf{W} = \left[\mathbf{W}^{\mathsf{T}}\right]^{-1} + g(\hat{\mathbf{s}}(n))\mathbf{x}^{\mathsf{T}}(n). \tag{4.2}$$

To avoid the matrix inversion of $\mathbf{W}$ at every step, the natural gradient can be utilized by post multiplying Eq. (4.2) by $\mathbf{W}^\mathsf{T}\mathbf{W}$ to give [4, 5]

$$\Delta\mathbf{W} = \left(\mathbf{I} + g(\hat{\mathbf{s}}(n))\mathbf{x}^\mathsf{T}(n)\right)\mathbf{W}. \tag{4.3}$$

In their original paper, Bell and Sejnowski modeled the cumulative distribution function (cdf) of supergaussian sources with the logistic function $cdf(s_i) = 1/(1+e^{-s_i})$, which results in a nonlinear function of

$$g(\hat{s}_i) = 1 - 2/(1+e^{-\hat{s}_i}). \tag{4.4}$$

Torkkola extends Eq. (4.3) to convolutive mixtures by utilizing a feedbackward architecture for the demixing. The final updates are found to be [45]

$$\Delta w_{i0i} = g(\hat{s}_i)x_i(n) + 1/w_{i0i} \tag{4.5}$$

$$\Delta w_{iki} = g(\hat{s}_i)x_i(n-k) \tag{4.6}$$

$$\Delta w_{ikj} = g(\hat{s}_i)\hat{s}_j(n-k) \tag{4.7}$$

where

$$\hat{s}_i(n) = \sum_{k=0}^{L_{ii}} w_{iki}x_i(n-k) + \sum_{k=1}^{L_{ij}} w_{ikj}\hat{s}_j(n-k). \tag{4.8}$$

Here L_{ij} is the length of the impulse response from the j^{th} observed signal to the i^{th} estimated source. The zero-delay weights in Eq. (4.5) maximize the information passed through the nonlinearity, the weights given in Eq. (4.6) decorrelate each output from the input mixtures, and the weights in Eq. (4.7) decorrelate each output $g(\hat{s}_i)$ from all other sources $\hat{s}_j$ [45]. Since speech is used as the original source signals, temporal decorrelation between the input and the output is unwanted. To overcome this temporal whitening, the non-zero delays in Eq. (4.6) are set to zero after every iteration.

4.2 Modifications of the Multichannel Blind Deconvolution Algorithms

Section 3.1.2 briefly introduced two different multichannel blind deconvolution algorithms. As mentioned earlier these algorithms are not best suited for speech signals because of the assumption of temporal independence. Fortunately, these algorithms have been modified in order to be applied to speech signals.

4.2.1 Nonholonomic Blind Source Separation - NHBSS Algorithm

Section 3.1.3 introduced the natural gradient multichannel blind deconvolution algorithm proposed by Amari et al. in [7]. As mentioned before, this algorithm has the drawback of temporally decorrelating the output signals due to the i.i.d. assumption. Since this dissertation deals specifically with speech, temporal decorrelation is an undesirable effect. In [43], Douglas and Sun modified the algorithm given in Eq. (3.38) and Eq. (3.39) to incorporate the coefficients of a linear predictor in order to mitigate the whitening effects of the algorithm. Yet another solution to this temporal decorrelation effect was proposed by Douglas and Sun in [19], where the natural gradient algorithm was modified to impose a nonholonomic constraint on the separated signals, so that there is no need for any post processing in order to eliminate the whitening effect. The resulting updates for the nonholonomic version of the natural gradient algorithm given in section 3.1.3 are

$$\mathbf{W}_p(n+1) = \mathbf{W}_p(n) + \mu(k)\left[\mathbf{V}_p(n) - \mathbf{g}(\hat{\mathbf{s}}(n-L))\mathbf{u}^{\mathsf{T}}(n-p)\right] \tag{4.9}$$

where the $(i,j)^{th}$ entry of $\mathbf{V}_p(n)$ is

$$v_{ijp}(n) = g(\hat{\mathbf{s}}_i(n-L))u_{ij}(n-p) \tag{4.10}$$

$$u_{ij}(n) = \sum_{q=0}^{L} w_{ij(L-q)}\hat{\mathbf{s}}_i(n-q). \tag{4.11}$$

4.2.2 Generalized Blind Source Separation using Second-Order Statistics - GENSOS Algorithm

The final time-domain algorithm to be discussed uses only second-order statistics in which to estimate the demixing filters. In general, BSS algorithms based on second-

order statistics utilize one of two different signal properties. The first property is the nonwhiteness property in which separation is achieved by simultaneously diagonalizing several correlation matrices over different time lags. The second property is the nonstationarity property, in which the BSS problem is solved by simultaneous diagonalization of short-time output correlation matrices at different time intervals [11]. Buchner et. al. in [11] derive a generic time domain algorithm incorporating both of these properties. They begin by introducing a matrix formulation of the convolutive mixing model. Reformulating the convolutive mixing model, Eq. (3.1) becomes

$$x_p(n) = \sum_{q=1}^{N_s} \sum_{k=0}^{L-1} a_{qp}(k) s_q(n-k) \tag{4.12}$$

for $p = 1, \ldots, N_x$. Here $a_{qp}(k)$ are the coefficients of the impulse response from the q^{th} source to the p^{th} microphone. With this new formulation, the demixing process can be written as

$$y_q(n) = \hat{s}_q(n) = \sum_{p=1}^{N_x} \sum_{k=0}^{L-1} w_{pq}(k) x_p(n-k) \tag{4.13}$$

for $q = 1, \ldots, N_s$. Inspecting Eq. (4.13) it can be seen that the output signals $y_q(n)$ at time n are given by

$$y_q(n) = \sum_{p=1}^{N_x} \mathbf{x}_p^{\mathsf{T}}(n) \mathbf{w}_{pq} \tag{4.14}$$

where $\mathbf{x}_p$ contains the latest L samples of the p^{th} microphone signal x_p, i.e.,

$$\mathbf{x}_p(n) = [x_p(n), x_p(n-1), \ldots, x_p(n-L+1)]^{\mathsf{T}}.$$

The vector $\mathbf{w}_{pq}$ denotes the impulse response from the p^{th} microphone signal to the q^{th} output signal, i.e.,

$$\mathbf{w}_{pq} = [w_{pq,0}, w_{pq,1}, \ldots, w_{pq,L-1}]^{\mathsf{T}}.$$

To incorporate the nonstationarity property, the variable N will be introduced to denote the length of the output signal block used to estimate the short-time correlations

of the output signals. Defining a block output signal of length N, the block output vector can be written as

$$\mathbf{y}_q(m) = \sum_{p=1}^{N_x} \mathbf{U}_p^{\mathsf{T}}(m)\mathbf{w}_{pq} \tag{4.15}$$

where m is the block index and

$$\mathbf{y}_q(m) = [y_q(mL), y_q(mL+1), \ldots, y_q(mL+N-1)]^{\mathsf{T}} \tag{4.16}$$

$$\underset{N\times L}{\mathbf{U}_p^{\mathsf{T}}(m)} = [\mathbf{x}_p(mL), \mathbf{x}_p(mL+1), \ldots, \mathbf{x}_p(mL+N-1)] \tag{4.17}$$

$$= \begin{bmatrix} x_p(mL) & \cdots & x_p(mL-L+1) \\ x_p(mL+1) & \cdots & x_p(mL-L+2) \\ \vdots & \ddots & \vdots \\ x_p(mL+N-1) & \cdots & x_p(mL-L+N) \end{bmatrix}. \tag{4.18}$$

Buchner et al. note that in order to obtain a correlation matrix, $N \geq N_x L$ [10, 11].

To incorporate time-lags for the nonwhiteness property, L of the output vectors in Eq. (4.17) are captured by

$$\underset{N\times L}{\mathbf{Y}_q(m)} = \begin{bmatrix} y_q(mL) & \cdots & y_q(mL-L+1) \\ y_q(mL+1) & \cdots & y_q(mL-L+2) \\ \vdots & \ddots & \vdots \\ y_q(mL+N-1) & \cdots & y_q(mL-L+N) \end{bmatrix}. \tag{4.19}$$

Using Eq. (4.19), Eq. (4.15) can extended to

$$\mathbf{Y}_q(m) = \sum_{p=1}^{N_x} \mathbf{X}_p(m)\mathbf{W}_{pq} \tag{4.20}$$

where

$$\underset{N\times 2L}{\mathbf{X}_p(m)} = [\mathbf{U}_p(m), \mathbf{U}_p(m-1)] \tag{4.21}$$

and

$$\underset{2L\times L}{\mathbf{W}_{pq}} = \begin{bmatrix} w_{pq,0} & 0 & \cdots & 0 \\ w_{pq,1} & w_{p1,0} & \ddots & \vdots \\ \vdots & w_{p1,1} & \ddots & 0 \\ w_{pq,L-1} & \vdots & \ddots & w_{p1,0} \\ 0 & w_{pq,L-1} & \ddots & w_{p1,1} \\ \vdots & 0 & \ddots & w_{p1,L-1} \\ 0 & \cdots & 0 & 0 \end{bmatrix}. \tag{4.22}$$

It is important to note that $\mathbf{U}_p^{\mathsf{T}}(m)$ for $p = 1, \ldots, N_x$ are Toeplitz matrices and $\mathbf{W}_{pq}$ are Sylvester matrices. Finally combing Eqs. (4.20), (4.21) and (4.22) for all channels, the mixing process becomes

$$\mathbf{Y}(m) = \mathbf{X}(m)\mathbf{W} \tag{4.23}$$

with

$$\underset{N\times N_s L}{\mathbf{Y}(m)} = [\mathbf{Y}_1(m), \ldots, \mathbf{Y}_{N_s}(m)] \tag{4.24}$$

$$\underset{N\times 2N_x L}{\mathbf{X}(m)} = [\mathbf{X}_1(m), \ldots, \mathbf{Y}_{N_x}(m)] \tag{4.25}$$

$$\underset{2N_x L\times N_s L}{\mathbf{W}} = \begin{bmatrix} \mathbf{W}_{11} & \cdots & \mathbf{W}_{1,N_x} \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{N_s,1} & \cdots & \mathbf{W}_{N_s,N_x} \end{bmatrix}. \tag{4.26}$$

Short-time correlation matrices can now be defined as[1]

$$\underset{2N_x L\times 2N_x L}{\mathbf{R}_{xx}} = \mathbf{X}^H\mathbf{X} \tag{4.27}$$

$$\underset{N_s L\times N_s L}{\mathbf{R}_{yy}} = \mathbf{Y}^H\mathbf{Y} \tag{4.28}$$

$$\tag{4.29}$$

[1]The block index m will be omitted henceforth.

The cost function is then defined as [10, 11]

$$J(m) = \sum_{i=0}^{m} \beta(i,m) \left\{ \log\det \text{bdiag}\left\{\mathbf{Y}^H\mathbf{Y}\right\} - \log\det \mathbf{Y}^H\mathbf{Y} \right\} \tag{4.30}$$

where $\beta(i,m)$ is a weighting function that can allow for both an offline and online realization. The bdiag operation, following the notation of [11], zeros all off-diagonal matrices in the partitioned matrix $\mathbf{R}_{yy}$. Since the matrix formulation in Eq. (4.23) is used to calculate the short-time correlation matrices $\mathbf{Y}^H\mathbf{Y}$, the cost function $J(m)$ inherently includes all time lags of all auto-correlations and cross-correlations of the output signals. Thus J is minimized, and in fact $J = 0$, when the output cross-correlations over all time lags vanish, thus achieving separation.

The method employed to update the matrix $\mathbf{W}$ is a gradient descent procedure with updates given by

$$\mathbf{W}(m) = \mathbf{W}(m-1) - \mu\Delta\mathbf{W}(m) \tag{4.31}$$

where

$$\Delta\mathbf{W}(m) = \nabla_{\mathbf{w}} J(m) = 2\frac{\partial J(m)}{\partial \mathbf{W}^*}. \tag{4.32}$$

Calculating the gradient $\frac{\partial J(m)}{\partial \mathbf{W}^*}$ gives

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{W}^*} \log\det\left[\mathbf{Y}^H\mathbf{Y}\right] =& \frac{\partial}{\partial \mathbf{W}^*} \log\det\left[(\mathbf{X}\mathbf{W})^H\mathbf{X}\mathbf{W}\right] \\
=& \frac{\partial}{\partial \mathbf{W}^*} \log\det\left[\mathbf{W}^H\mathbf{X}^H\mathbf{X}\mathbf{W}\right] \\
=& \frac{\partial}{\partial \mathbf{W}^*} \log\det\left[\mathbf{W}^H\mathbf{R}_{xx}\mathbf{W}\right] \\
=& 2\mathbf{R}_{xx}\mathbf{W}\left(\mathbf{W}^H\mathbf{R}_{xx}\mathbf{W}\right)^{-1}
\end{aligned}$$

and

$$\frac{\partial}{\partial \mathbf{W}^*} \log\det \text{bdiag}\left\{\mathbf{Y}^H\mathbf{Y}\right\} = 2\mathbf{R}_{xx}\mathbf{W}\left(\text{bdiag}\left\{\mathbf{W}^H\mathbf{R}_{xx}\mathbf{W}\right\}\right)^{-1}.$$

Using these equations it follows from Eq. (4.30)

$$\nabla_{\mathbf{w}} J(m) = 4\sum_{i=0}^{m} \beta(i,m)\mathbf{R}_{xy}\mathbf{R}_{yy}^{-1}\left(\mathbf{R}_{yy} - \text{bdiag}\left\{\mathbf{R}_{yy}\right\}\right) \cdot \left(\text{bdiag}\left\{\mathbf{R}_{yy}\right\}\right)^{-1}. \tag{4.33}$$

As mentioned in earlier sections, the standard gradient descent can be modified to the natural gradient or relative gradient by post multiplying Eq. (4.33) by $\Delta\mathbf{W}(m) = \nabla_{\mathbf{w}} J(m)\mathbf{W}^H\mathbf{W}$. For the formulation described here this needs to be modified to

$$\Delta\mathbf{W}(m) = \mathbf{W}\mathbf{W}^H\nabla_{\mathbf{w}} J(m).$$

This leads to the modified update as

$$\Delta\mathbf{W}(m) = 4\sum_{i=0}^{m}\beta(i,m)\mathbf{W}\left(\mathbf{R}_{yy} - \text{bdiag}\left\{\mathbf{R}_{yy}\right\}\right)\cdot\left(\text{bdiag}\left\{\mathbf{R}_{yy}\right\}\right)^{-1}. \tag{4.34}$$

The weighting function $\beta(i,m)$ allows for different realizations of the update in Eq. (4.34). For online realizations [2], the weighting function is chosen as

$$\beta(i,m) = (1-\lambda)\lambda^{m-i}\epsilon_{0,m}(i) \tag{4.35}$$

where λ is an exponential forgetting factor such that $0 \leq \lambda \leq 1$. In order to perform the updates for $\mathbf{W}$ offline, the weighting function is

$$\beta(i,m) = 1/K_{sig} \tag{4.36}$$

where K_{sig} is the total number of blocks of the observed signals. Only the offline counterpart of this algorithm was implemented for this dissertation. Thus for the offline version, the mixed signals are segmented into K_{sig} blocks, and then the entire signal is processed in order to estimate $\mathbf{W}$. The offline version becomes

$$\mathbf{W}^j = \mathbf{W}^{j-1} - \frac{\mu}{K_{sig}}\sum_{i=0}^{K_{sig}-1}\Delta\mathbf{W}(i) \tag{4.37}$$

where j denotes the current iteration.

There are a few things worth mentioning about the final algorithm given in Eq. (4.34). As shown in Eq. (4.22), $\mathbf{W}$ should be updated with the constraint that it keeps the Sylvester matrix structure. It is sufficient to enforce this constraint by taking a certain column or the L^{th} row as a reference after every iteration to generate the Sylvester form.

The next issue to be addressed is the inversion of the correlation matrices in Eq. (4.34). If the correlation matrix $\mathbf{R}_{yy}$ is ill-conditioned, it must be properly regularized prior to inversion by adding a small constant on the diagonal. In [1], Aichner et al. propose to regularize the correlation matrices by attenuating the off-diagonals of $\mathbf{R}_{y_q y_q}$ by

$$\check{\mathbf{R}}_{\mathbf{y}_q\mathbf{y}_q} = \rho\mathbf{R}_{\mathbf{y}_q\mathbf{y}_q} + (1-\rho)\text{diag}\left\{\mathbf{R}_{\mathbf{y}_q\mathbf{y}_q}\right\} \tag{4.38}$$

where the diag operation sets all off-diagonals to zero, and the weighting factor ρ is chosen so that $0 \leq \rho \leq 1$. The author has found during the course of this research that better results can be sometimes achieved by employing a slightly different regularization scheme. This regularization schemes simply adds a small percentage of the largest value of $|\mathbf{R}_{\mathbf{y}_q\mathbf{y}_q}|$, i.e.

$$\check{\mathbf{R}}_{\mathbf{y}_q\mathbf{y}_q} = \mathbf{R}_{\mathbf{y}_q\mathbf{y}_q} + \rho \max|\mathbf{R}_{\mathbf{y}_q\mathbf{y}_q}|\mathbf{I} \tag{4.39}$$

where $\mathbf{I}$ is the identity matrix.

4.3 Frequency-Domain Methods

In this section, the frequency-domain algorithms chosen to be evaluated for this dissertation will be introduced. The advantage of performing blind source separation in the frequency-domain includes a reduction in the computational complexity. By moving to the frequency-domain it is possible to transform the convolutions in the time-domain to multiplications in the frequency-domain.

There are really two different ways to implement frequency-domain techniques [45]. The first is to do all of the separation in the frequency-domain, including the estimation of the original source signals. This has the advantage of transforming the convolutive mixtures into multiple instantaneous mixtures. It then not only learns the separating matrix, but also computes the separated signals in the frequency-domain. The problem with this is that there may be arbitrary permutations for every frequency bin. Thus complete time-domain reconstruction requires a proper reordering of the signals for every bin.

The second method consists of performing the actual separation in the time-domain and only a few aspects of the updates are done in the frequency-domain. This avoids the permutation problem by placing some constraints on the unmixing matrix. Two algorithms that perform blind source separation in this fashion will now be introduced.

4.3.1 Bussgang Algorithms in the frequency-domain

It was already mentioned in section 3.1.2 how Lambert extended Bussgang deconvolution algorithms to the blind source separation of convolutive mixtures. This section will now describe how in [29, 31] Lambert extended scalar matrix algebra to FIR matrices in order to perform blind source separation. Lambert's derived multichannel blind LMS algorithm was given as

$$\underline{\mathbf{W}}(n+1) = \underline{\mathbf{W}}(n) + \mu(\hat{\mathbf{s}}(n) - g(\hat{\mathbf{s}}(n)))\mathbf{x}^*(n). \tag{4.40}$$

Lambert shows in [29] that the algorithm given in Eq. (4.40) can be updated in the frequency-domain more efficiently. He does so with the use of an algebra for FIR matrices. He notes that any function $g(\cdot)$ acting on a filter h is defined as

$$g(h) = \text{IFFT}\{g(\text{FFT}\{[\mathbf{0}\ h\ \mathbf{0}]\})\} \tag{4.41}$$

where $\mathbf{0}$ represents a vector of zeros so that the approximation holds. Using this approximation, Eq. (4.40) in the frequency-domain becomes

$$\underline{\mathbf{W}} = \underline{\mathbf{W}} + \mu(\hat{\mathbf{S}} - \text{FFT}\{g(\hat{\mathbf{s}})\})\mathbf{X}^H. \tag{4.42}$$

where $\underline{\mathbf{W}}, \hat{\mathbf{S}}, \mathbf{X}^H$ represent the Fourier transforms of the FIR filter matrix, the estimated source signals, and the observed signals, respectively. Again this algorithm is derived assuming that the sources are nongaussian and i.i.d. Obviously, this is not true for speech since there is temporal correlation in speech signals. Speech is also sufficiently nonstationary such that convergence might be impossible. In [29] Lambert notes that a finite difference approximation is more robust to nonstationarity than

the normal MBLMS, thus convergence issues with speech can be somewhat negated. Our experience has also confirmed this finding.

4.3.1.1 Blind Serial Update - BSU Algorithm

The finite difference approximation to a true gradient is found by evaluating the cost function at two points separated by a small distance, i.e., $J(n) - J(n+\delta)$. With this approximation, one can estimate the update using stochastic measures such as correlation functions for $\underline{\mathbf{W}}$ as

$$\Delta\underline{\mathbf{W}} = J(n). \tag{4.43}$$

Lambert then applies this finite difference approximation to the gradient to the serial or relative-gradient updates as introduced in [12] giving an update of the demixing matrix as

$$\Delta\underline{\mathbf{W}} = J(n)\underline{\mathbf{W}}. \tag{4.44}$$

Using these approximations, the cost function for MBLMS in the time-domain becomes

$$J(n) = \delta(0) - E\left\{\hat{\mathbf{s}}(n+k)g(\hat{\mathbf{s}}(n))\right\}. \tag{4.45}$$

Furthermore, in [29] Lambert made use of the Bussgang property in the frequency-domain to formulate Eq. (4.45) in the frequency-domain arriving at the update for $\underline{\mathbf{W}}$ as

$$\Delta\underline{\mathbf{W}}(n) = \left[1 - \mathbf{R}_g(n)\right]\underline{\mathbf{W}}(n) \tag{4.46}$$

where $\mathbf{R}_g(n) = E\left\{\mathrm{FFT}^*(\hat{\mathbf{s}}(n))\mathrm{FFT}\{g(\hat{\mathbf{s}}(n))\}\right\}$ is the Bussgang property in the frequency-domain.

Although the MBLMS algorithm was originally introduced in section 3.1.2 as a time-domain algorithm, this algorithm was implemented using the frequency-domain form of Eq. (4.46). This was chosen for two reasons. First, performing the updates in the frequency-domain forces the updates to be done in a block-wise fashion. This block-wise update can help the stability of the algorithm due to the averaging effect

of the nonstationarity. Second, performing the updates in the frequency-domain, fast convolution techniques can be employed to reduce the computational complexity [20, 41].

4.3.2 Parra's Joint Block Diagonalization - JBD Algorithm

In [34], Parra and Spence use the non-stationarity of speech in order to perform separation. In their method they simultaneously diagonalize multiple cross-correlations at different time lags. This is actually implemented in the frequency-domain by diagonalizing cross-power spectra. The thought behind this is that decorrelating the observed signals at multiple time lags forces the observed signals to be independent, thus achieving separation. In fact by forcing independence between the observed signals for multiple delays corresponding to the delays of the channel filter taps, separation can be achieved. This algorithm does the actual reconstruction of the estimated source signals in the time-domain.

Parra modeled the mixing process as

$$\mathbf{x}(n) = \sum_{k=0}^{P} \mathbf{A}(k)\mathbf{s}(n-k) + \mathbf{v}(n) \tag{4.47}$$

and the unmixing process as

$$\hat{\mathbf{s}}(n) = \sum_{k=0}^{L} \mathbf{W}(k)\mathbf{x}(n-k). \tag{4.48}$$

Here, $\mathbf{v}(n)$ is additive noise. Now consider the cross-correlations of the observations $R_x(\tau)$, assuming that they are stationary

$$R_x(\tau) = E\left\{\mathbf{x}(n), \mathbf{x}(n+\tau)\right\}. \tag{4.49}$$

In the z-domain, this can be written as

$$R_x(z) = \mathbf{A}(z)\mathbf{\Lambda}_s(z)\mathbf{A}^H(z) + \mathbf{\Lambda}_v(z) \tag{4.50}$$

where $\mathbf{A}(z)$ is the matrix of z-transforms of the FIR mixing filter. $\mathbf{\Lambda}_s(z)$ and $\mathbf{\Lambda}_v(z)$ are z-transforms of the autocorrelation matrices of the sources and noise, respectively.

$\mathbf{\Lambda}_s(z)$ and $\mathbf{\Lambda}_v(z)$ are diagonal due to the independence assumption. It is possible, however, to approximate the linear convolution as circular convolutions [29]. By doing so, one can use the DFT instead of the z-transform, provided the frame size T is chosen much larger than the channel filter length, i.e. $T >> P$. Thus

$$\mathbf{x}(\omega, n) \approx \mathbf{A}(\omega)\mathbf{s}(\omega, n) + \mathbf{v}(\omega, n) \tag{4.51}$$

where $\mathbf{x}(\omega, n)$ is the DFT of frame size T starting at time n. In other words $\mathbf{x}(\omega, n) = \sum_{\tau=0}^{T-1} e^{-2\pi j\omega\tau/T}\mathbf{x}(n+\tau)$.

For non-stationary signals, the cross-correlation is time dependent. Estimation of the cross-power spectrum with a resolution of $1/T$ can be difficult if the stationarity of the source signal is on the order of T or less. However, any cross-power spectrum average that diagonalizes the source signals will be sufficient [34]. Parra and Spence chose the sample average

$$\bar{R}_x(\omega, n) = \frac{1}{N}\sum_{k=0}^{N-1} \mathbf{x}(\omega, n+kT)\mathbf{x}^H(\omega, n+kT) \tag{4.52}$$

$$= \mathbf{A}(\omega)\mathbf{\Lambda}_s(\omega, n)\mathbf{A}^H(\omega) + \mathbf{\Lambda}_v(\omega, n). \tag{4.53}$$

If N is large enough, $\mathbf{\Lambda}_s(\omega, n)$ and $\mathbf{\Lambda}_v(\omega, n)$ can be modeled as diagonal due to the independence assumption. It is necessary for the source signals to be non-stationary so that $\mathbf{\Lambda}_s(\omega, n)$ changes over time for a given frequency. Otherwise Eq. (4.53) will not be linearly independent.

This algorithm estimates a multi-path feed-backward FIR model as in Eq. (4.48) so that the estimated sources have cross-power-spectra satisfying

$$\mathbf{\Lambda}_s(\omega, n) = \mathbf{W}(\omega)[\bar{R}_x(\omega, n) - \mathbf{\Lambda}_v(\omega, n)]\mathbf{W}^H(\omega, n). \tag{4.54}$$

If the signals do not vary rapidly, non-overlapping time windows can be chosen for $\bar{R}_x(\omega, t_k)$, i.e. $t_k = kTN$, so that independent conditions are obtained for every

time interval. The unmixing matrix $\mathbf{W}$ is found via an LS estimate for K different times[2]

$$E(\omega, k) = \mathbf{W}(\omega)[\bar{R}_x(\omega, k) - \mathbf{\Lambda}_v(\omega, k)]\mathbf{W}^H(\omega) - \mathbf{\Lambda}_s(\omega, k) \tag{4.55}$$

where $E(\omega, k)$ is the error at time t_k. So the LS estimate is found by minimizing the squared error, i.e.

$$\hat{\mathbf{W}}, \hat{\mathbf{\Lambda}}_s, \hat{\mathbf{\Lambda}}_n = \underset{\mathbf{W}, \mathbf{\Lambda}_s, \mathbf{\Lambda}_v}{\arg\min} \qquad J = \sum_{\omega=1}^{T} \sum_{k=1}^{K} ||E(\omega, k)||^2 \tag{4.56}$$

subject to

$$\mathbf{W}(\tau) = 0, \tau > L << T \tag{4.57}$$

$$W_{ii} = 1. \tag{4.58}$$

Note that Eq. (4.57) puts a time-domain constraint on the filter length L of the demixing filter. This constraint ensures that the frequencies represent independent problems. It also forces the filter response to be zero for $\tau > L << T$. The LS solutions for the parameters in Eq. (4.56) are found by computing the gradients with respect to the filter coefficients in $\mathbf{W}(\omega)$. The gradients of the cost function are

$$\frac{\partial J}{\partial \mathbf{W}^*(\omega)} = 2 \sum_{k=1}^{K} E(\omega, k)\mathbf{W}(\omega)[\bar{R}_x(\omega, k) - \mathbf{\Lambda}_v(\omega, k)] \tag{4.59}$$

$$\frac{\partial J}{\partial \mathbf{\Lambda}_s^*(\omega, k)} = -\text{diag}[E(\omega, k)] \tag{4.60}$$

$$\frac{\partial J}{\partial \mathbf{\Lambda}_v^*(\omega, k)} = -\text{diag}[\mathbf{W}^H(\omega)E(\omega, k)\mathbf{W}(\omega)]. \tag{4.61}$$

The solutions for the estimates are the minimum of Eq. (4.59) and Eq. (4.61). The optimal $\mathbf{\Lambda}_s(\omega, k)$ for a given $\mathbf{W}(\omega)$ and $\mathbf{\Lambda}_v(\omega, k)$ can be explicitly found by setting Eq. (4.60) to zero at every gradient step. This gives

$$\hat{\mathbf{\Lambda}}_s(k) = \text{diag}[\mathbf{W}(\omega)\hat{R}_x(\omega, k)\mathbf{W}^{\mathsf{T}}(\omega) - \mathbf{\Lambda}_v(\omega, k)].$$

[2]Following Parra, $R_x(\omega, t_k)$ will be abbreviated as $R_x(\omega, k)$. This same notation will be used for all other variables as well

Parra notes that the gradient terms scale with the square of R_x. Since the gradient steps for different frequencies have different magnitudes, for optimal convergence it is necessary to use different adaptation steps for each frequency. Or one can simply normalize the powers, resulting in equal steps for each frequency. This can be easily done by defining a weighted cost function

$$J = \sum_{\omega=1}^{T} \sum_{k=1}^{K} m(\omega) ||E(\omega, k)||^2 \tag{4.62}$$

with

$$m(\omega) = \left(\sum_{k=1}^{K} ||R_x(\omega, k)||^2 \right)^{-1}. \tag{4.63}$$

Recall that transforming convolutive blind source separation problems to the frequency-domain, there is an inherent problem of arbitrary permutations in every frequency bin. However, Parra noticed that the constraint on the filter length, $W(\tau) = 0$ for $\tau > L << T$, inherently solves the permutation problem. This constraint requires the coefficients to be zero for $\tau > L$, which effectively forces the solutions to the gradients to be smooth in the frequency-domain.

This spectral smoothness constraint is enforced by projecting the unconstrained gradients in Eqs. (4.59)- (4.61) to the subspace of permissible solutions. So a projector that zeros the appropriate delays for every channel is required. This projector is given by

$$P_J = FZF^{-1} \tag{4.64}$$

where $F_{ik} = \frac{1}{\sqrt{T}} e^{-j2\pi ik}$ and $Z_{ii} = 1$ for $i < L$ and zero otherwise. The constraint of $W_{ii}(\omega) = 1$ is enforced by setting the diagonal terms of the gradient to zero.

4.4 Summary of Algorithms

This chapter introduced the algorithms to be evaluated in Chapter V. Table 1.1 gave a summary of the algorithms presented in this chapter. We will repeat it here for convenience in Table 4.1.

Table 4.1: Set of chosen algorithms for evaluation.

Method	**Section Covered**	**Category**
INFOMAX [44]	4.1.1	TD-IT
NHBSS [19]	4.2.1	TD-IT
GENSOS [11]	4.2.2	TD-NIT
BSU [31]	4.3.1.1	FD-IT
JBD [34]	4.3.2	FD-NIT

TD = Time-Domain

FD = Frequency-Domain

IT = Information-Theoretic

NIT = Non-Information-Theoretic

CHAPTER V
RESULTS

5.1 Introduction

Before simulations of various blind source separation algorithms are presented, a common measure of performance for separation quality will now be presented. The measure of performance will be the Signal-to-Interference Ratio (SIR) [21,40,47]. The SIR measurement is a ratio between the energy of the desired separated signal and the cross-talk between channels, i.e.,

$$SIR = 10 \log_{10} \frac{||\text{target source}||^2}{||\text{interfering sources}||^2}. \tag{5.1}$$

In the following sections of this chapter, the results for various scenarios will be presented. First, the ability of the algorithms to deal with simple instantaneous mixing will be studied. Following the instantaneous mixing, three convolutive mixing situations will be presented, starting with a channel that simply delays the input signals and increasing the severity of the channels to the third example.

Once these situations are discussed, results will be shown regarding signals that have some sort of similarity. The first example will be that of the same person saying two different sentences. This situation hopes to test the ability of the algorithms to deal with speech sources that have similar spectral content, since an individual person has characteristics specific to his/her own voice [51]. Then results will be given for sources that have similar temporal structure. For this simulation, two different speakers say the same sentence.

Next the issue of real-world mixing will be addressed. The recording setup will first be presented, and then the results for the different recording situations will be presented.

Finally results will be shown to evaluate the increase in performance, if any, of the algorithms when presented with additional mixed signals. In this case, an extra microphone signal will be added in order to evaluate this performance increase.

5.2 Parameter Setup

All algorithms evaluated in this chapter are derived using some gradient technique and thus require a learning rate. The parameter μ will refer to this learning rate for all algorithms. Also all of these algorithms were implemented in a batch, offline fashion. Thus the entire signal is presented to the algorithms several times in order to reach convergence.

5.2.1 INFOMAX/NHBSS/BSU Parameters

Other than the learning rate, μ, the INFOMAX, NHBSS, and BSU algorithms do not require any other parameters.

5.2.2 JBD Parameters

The JBD algorithm requires several parameters to be specified. First, define the number of matrices to diagonalize as K, the number of intervals in which to estimate the spectrum as N, and the number of points in the FFT as N_{fft}.

As mentioned in Chapter III, FD approaches have the drawback of requiring large amounts of data to cover long mixing filters. For the JBD method, we must have

$$K \cdot N \cdot N_{\text{fft}} < size(\mathbf{x}) \tag{5.2}$$

where $size(x)$ represents the number of samples of the observed mixtures. We will use Eq. (5.2) to choose N for a given N_{fft} and K. For the remainder of this paper, the N_{fft} parameter is set to $N_{\text{fft}} = 8 \cdot L$, following Parra in Spence, such that the arbitrary permutations for each frequency bin are avoided. Also, it will be assumed that N is chosen as the largest integer value that satisfies Eq. (5.2).

5.2.3 GENSOS Parameters

For the GENSOS algorithm, there are several parameters in which to tune for every simulation. This includes the adaptation step size, μ. Due to the fact that the GENSOS algorithm requires inversion of correlation matrices, this algorithm has a

different adaptation step for each microphone signal. The step size μ_i is the step size that acts on all demixing filters that contribute to the i^{th} separated signal. Furthermore, due to this inversion, upon reaching convergence, the algorithm tends to be a bit unstable. For this reason, an elementary step-size control was used in order to mitigate this instability. The step-size control mechanism utilized for this algorithm was to update the parameters $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_{N_s}]$ proportional to the cost function $J(m)$ given in Eq. (4.30). If $J(m) < J(m-1)$, then increase $\boldsymbol{\mu}$ by a small amount. If $J(m) > J(m-1)$, then decrease $\boldsymbol{\mu}$ dramatically. Finally, the values of $\boldsymbol{\mu}$ are restricted to a minimum and maximum value, i.e., $\boldsymbol{\mu}_{\min} \leq \boldsymbol{\mu} \leq \boldsymbol{\mu}_{\max}$. The step-size control algorithm then can be described as

$$\mu_i(m+1) = \begin{cases} \alpha \cdot \mu_i(m) & \text{if } J(m) > a \cdot J(m-1),\ \alpha < 1 \\ \lambda \cdot \mu_i(m) & \text{if } J(m) < b \cdot J(m-1),\ \lambda > 1 \\ \mu_i(m) & \text{otherwise.} \end{cases} \tag{5.3}$$

In the author's experience, choosing $a = 1.02$, $b = 1$, and $\lambda = 1.01$ with $\boldsymbol{\mu}_{\max} = 1.5\boldsymbol{\mu}$ give good results for a wide variety of scenarios. Unless otherwise stated, these will be the assumed values for these parameters. Although this procedure was developed independently, it is essentially identical to the procedure described in [3].

Finally, we have the matter of the correlation matrix normalization. Unless otherwise stated, the normalization scheme used was that of Eq. (4.39) with $\rho = 0.5$.

5.3 Synthetic Mixing Simulations

The source signals for the following simulations are 2.8 s of a male and female speaker sampled at 11025 Hz and are shown in Figure 5.1. The sentences spoken were sentences found in the TIMIT database, although they were not taken from the database itself. The sentences were:

- (Male)s_1 = `Good service should be rewarded by big tips`.

- (Female)s_2 = `You always come up with pathological examples`.

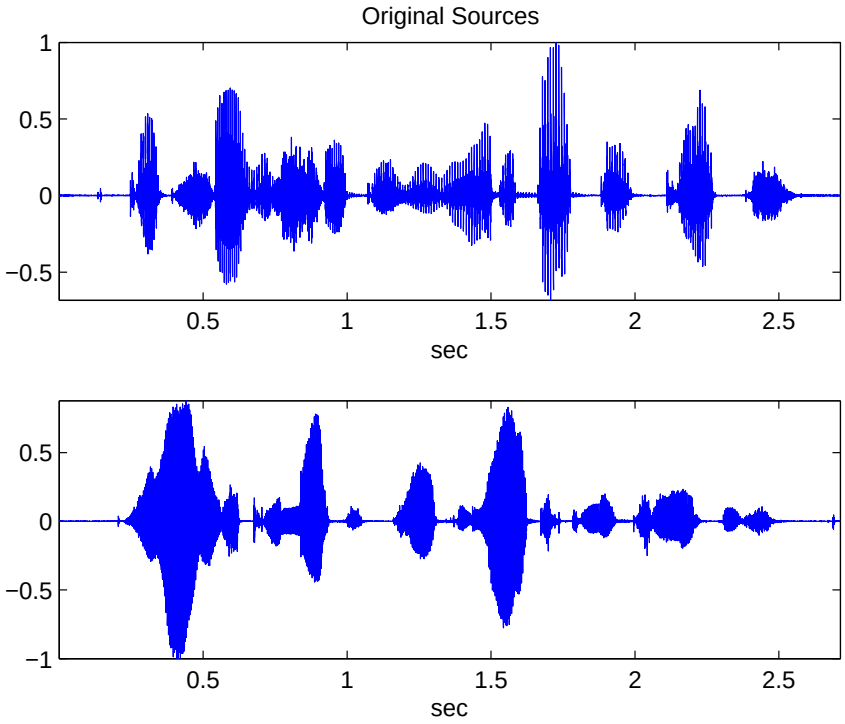


Figure 5.1: Original source signals

5.3.1 Instantaneous Mixtures

The first experiment consisted of instantaneously mixed signals. The mixing matrix used was

$$\mathbf{A} = \begin{bmatrix} 2 & .4 \\ .5 & -.6 \end{bmatrix}. \tag{5.4}$$

It is possible to perfectly invert the channel in the instantaneous case, and therefore the upper limit on the SIR is ∞. Thus, for the results presented here, it will suffice to simply state that all algorithms were capable of achieving an SIR improvement of over 40dB for each channel. For all algorithms, audible separation was perfect.

5.3.2 Delayed Mixing

The next scenario tested was that of delayed and mixed signals. Delayed mixtures are one step closer to true convolutive mixing, however, in this case, there is only one

non-zero coefficient per filter. The coefficients from Eq. (5.4) were used with delays added to each element. The channel used for this situation was

$$\begin{aligned} A_{11}(z) &= 2 \\ A_{12}(z) &= 0.4z^{-24} \\ A_{21}(z) &= 0.5z^{-17} \\ A_{22}(z) &= -0.6z^{-11}, \end{aligned} \tag{5.5}$$

shown in the z-domain for convenience.

5.3.2.1 Parameters and Results

As described in section 5.2, each algorithm has a different set of parameters. The parameters for each algorithm is summarized in Table 5.1.

Table 5.1: Parameter Summary for the Delayed Example

Algorithm	**Parameters**
INFOMAX	$\mu = 0.00005$
NHBSS	$\boldsymbol{\mu} = [0.0001, 0.0001]$
GENSOS	$\boldsymbol{\mu}_{\min} = [0.02, 0.0006]$ $\boldsymbol{\mu} = [0.05, 0.0029]$ $\alpha = [0.8, 0.3]$
BSU	$\mu = 5 \times 10^{-6} \cdot L$
JBD	$\mu = 0.5,\ K = 10$

The improvement in SIR for the individual algorithms are shown in Figure 5.2. By inspecting these plots, all algorithms, with the exception of the BSU algorithm, had similar performance. A point that needs mentioning is the smoothness of the graphs. The INFOMAX algorithm is the only method that had similar performance for all L. For the other algorithms, the SIR does not smoothly increase or decrease

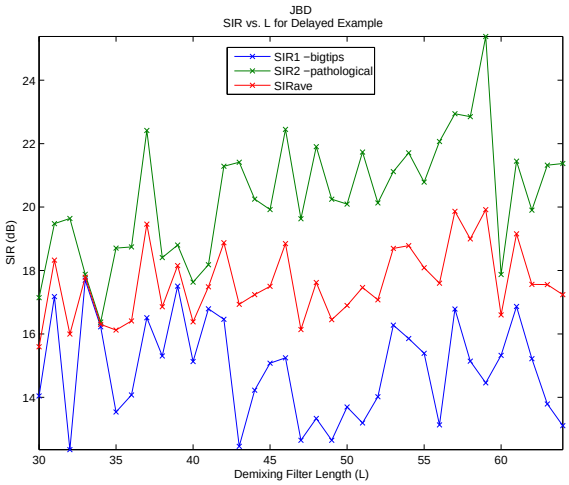
with L. This is a somewhat surprising result, since this example deals with a channel that simply delays the inputs. For this simple channel, the inverse of the channel can be found exactly with a demixing filter of the same length of the channel. Therefore the SIR is actually expected to be the same if not slightly worse as L is increased, because this does not add to the ability of the demixing filers to invert the channel, it only adds more parameters in which to estimate. The results displayed here show that only the INFOMAX algorithm shows this expected result, in which the SIR is not improved by adding more taps to the demixing filter, and in fact a decreasing trend can be seen in Figure 5.2(c). The other algorithms are somewhat sporadic for the different values of L. For the JBD algorithm, this may be explained by the fact that, for each L, we are not holding all of the other parameters constant. For the JBD algorithm, Parra and Spence note that we must have a window length for the FFT as $8L$. So as L increases, so does N_{fft}, which in turn forces N to change in order to satisfy Eq. (5.2). Thus, more than one parameter is changing for each choice of L.

The NHBSS algorithm shows a vast difference in SIR, depending on the choice of L. This was observed throughout this dissertation. The author observed that the NHBSS algorithm tended to not stay at its solution once it achieved convergence. In other words, once the NHBSS algorithm reached its optimal solution, if the data continued to be processed, the demixing filters would continue to update and in fact cause the separation to be worse. This is most likely the explanation for the large deviation in SIR around the range of $40 \leq L \leq 50$.

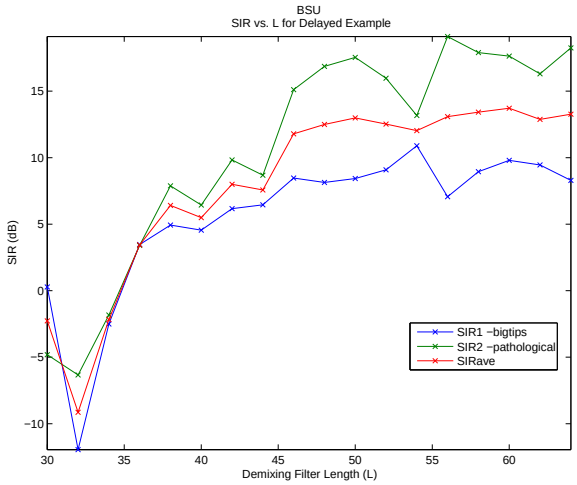
An interesting thing to point out about the GENSOS algorithm is the large decrease in SIR when $L = 60$. The author believes that this sudden and big decrease in performance is due to the signals used in the simulation. The author noted that when L is in the region around 55 to 65, the correlation matrices became very ill-conditioned, even with the regularization techniques discussed before. The author believes that this is the cause for the bad performance when $L = 60$. The author

confirmed this by using different speech signals for the same channel and noticed that the problem did not appear when L reached 60.

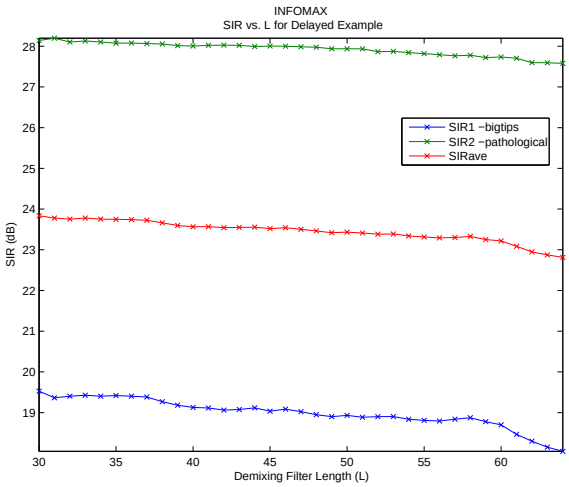
Figure 5.3 compares all the algorithms for the delayed example. This figure shows the average SIR for each algorithm. The average SIR is computed by averaging the SIRs of the estimated signals, i.e. $SIR_{\text{ave}} = \frac{1}{2}(SIR_1 + SIR_2)$. Figure 5.3 shows that the INFOMAX and GENSOS algorithms slightly outperform the other algorithms, although not by much. This is done, however, at a significant increase in computational cost. The JBD algorithm is by far the fastest algorithm, and its performance is not far behind the GENSOS and INFOMAX algorithms. The author does admit that the GENSOS algorithm was not implemented in its real-time form, as in [2].



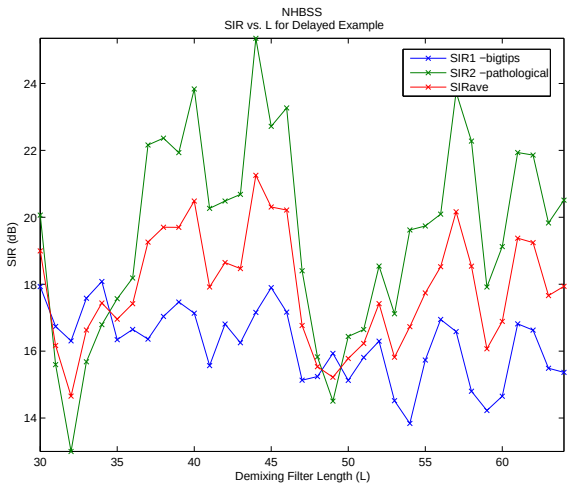
(a): JBD



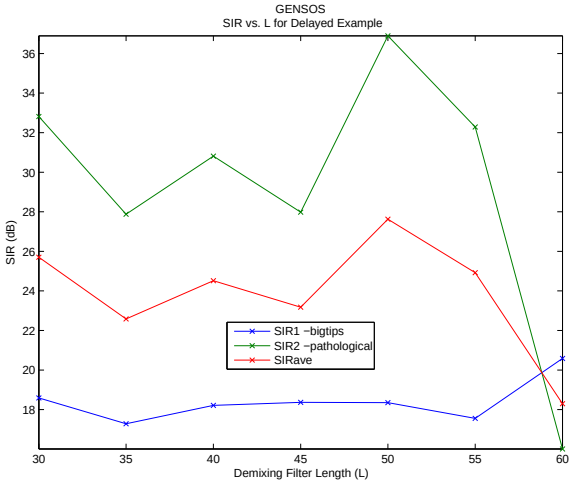
(b): BSU



(c): INFOMAX



(d): NHBSS



(e): GENSOS

Figure 5.2: SIR versus L for the Delayed Example

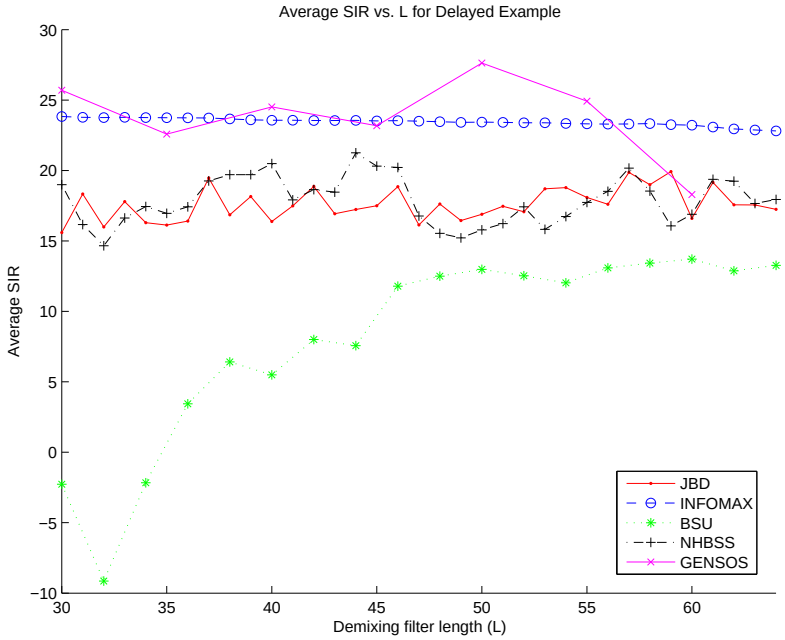


Figure 5.3: Comparison of the Average SIR for the Delayed Example

5.3.3 Convolutive Mixing

The ability of the algorithms to handle convolutive mixing will now be discussed. Three simulations were conducted, starting with a channel with a main path and a single echo, and ending with a more complex mixing channel.

5.3.3.1 Convolutive Mixing: Example 1

The next example will explore the ability of the methods to separate signals that are convolutively mixed, beginning with a direct path for each sensor along with only 1 echo. Again, the source signals used are $2.8s$ of a male and female speaker. The data is synthetically mixed, using a direct path and a single delay of between $1ms$ and $2ms$. The echo strengths will be forced to be less than the direct path. Figure 5.4 shows the impulse responses of the channels for this simulation.

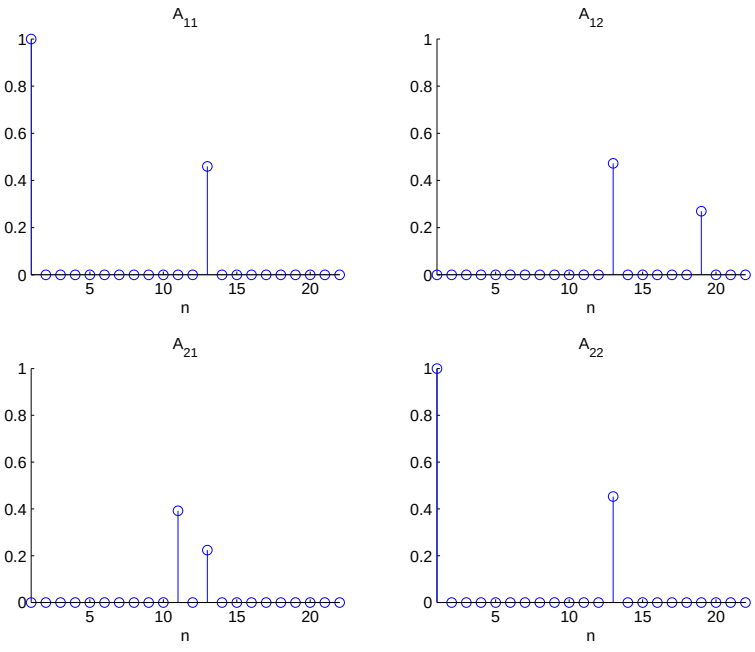


Figure 5.4: The mixing matrix A for the first convolutive mixture example

5.3.3.2 Parameters and Results

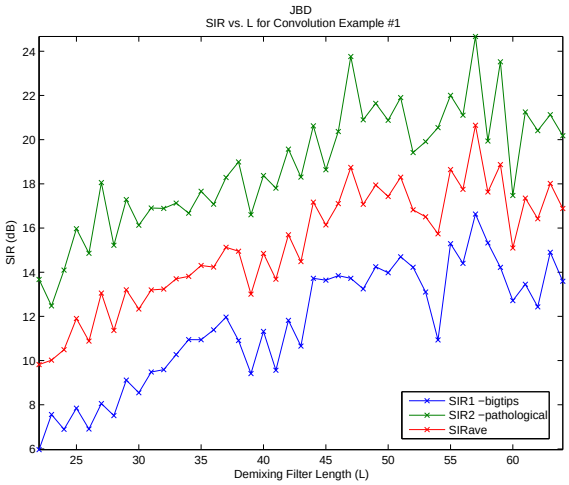
The parameter summary for this experiment is shown in Table 5.2.

Table 5.2: Parameter Summary for the First Convolution Example

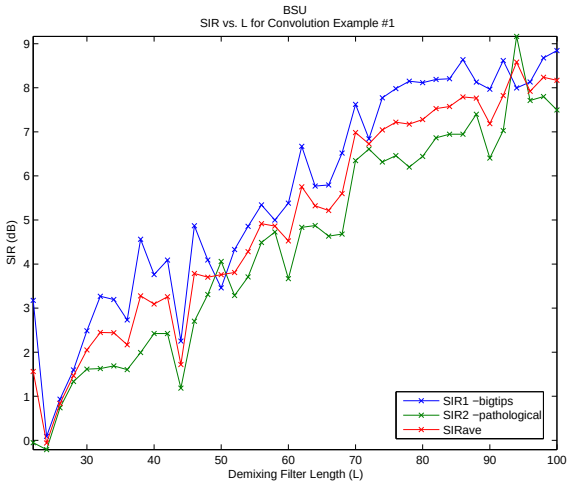
Algorithm	**Parameters**
INFOMAX	$\mu = 0.00005$
NHBSS	$\boldsymbol{\mu} = [0.0001, 0.0001]$
GENSOS	$\boldsymbol{\mu}_{\min} = [0.005, 0.04]$ $\boldsymbol{\mu} = [0.01, 0.05]$ $\alpha = [0.5, 0.75]$
BSU	$\mu = 5 \times 10^{-6} \cdot L$
JBD	$\mu = 0.5,\ K = 10$

The improvement in SIR for the individual algorithms are shown in Figure 5.5. These figures show an improved ability of the algorithms to separate the mixtures as we increase the demixing filter length L. This increase in performance levels off for almost all algorithms when L reaches about 45. The exceptions to this are the BSU and NHBSS algorithms. The BSU algorithm again performs poorly compared to other algorithms. It is also the only algorithm that requires longer filters in order to separate the signals. The NHBSS algorithm had almost uniform performance for all choices of L, suggesting that for this channel, it requires shorter filters in order to achieve the same performance as the other algorithms.

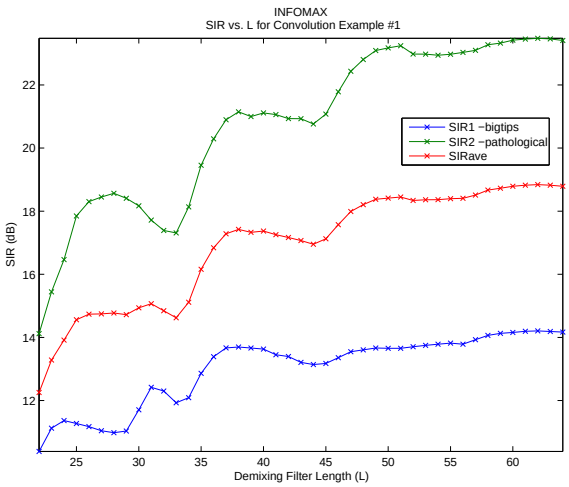
The comparison of all the algorithms is shown in Figure 5.6. This figure shows again that the algorithms performed similarly with the exception of the BSU algorithm.



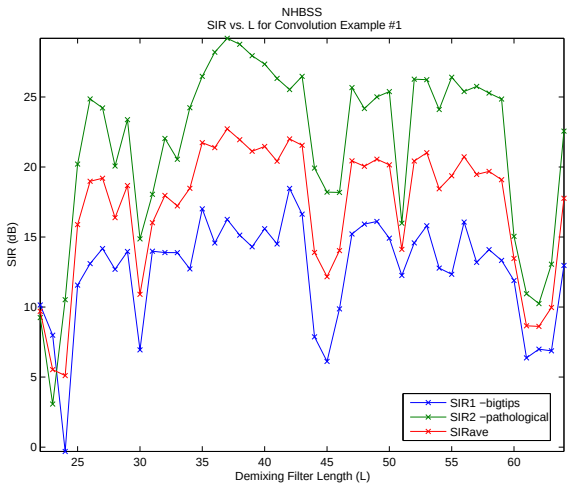
(a): JBD



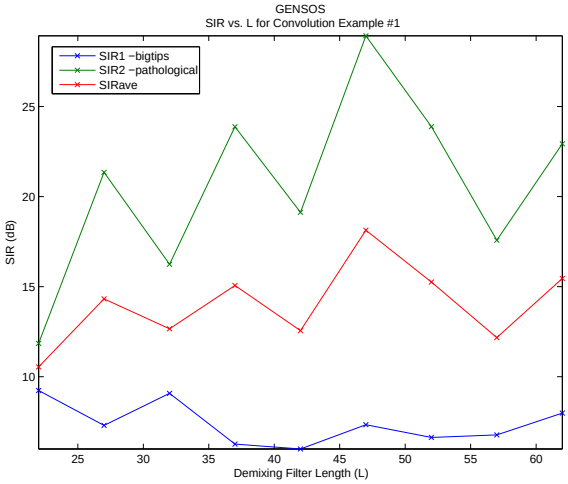
(b): BSU



(c): INFOMAX



(d): NHBSS



(e): GENSOS

Figure 5.5: SIR versus L for the First Convolution Example

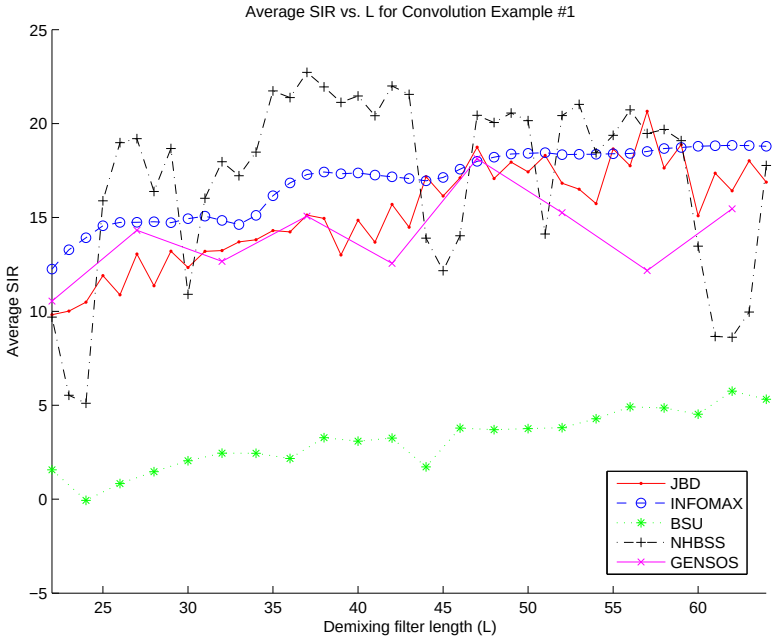


Figure 5.6: Comparison of the Average SIR for the First Convolution Example

5.3.3.3 Convolutive Mixing: Example 2

For the next example, a longer channel was used. The impulse response of a real room can be modeled as exponentially decaying noise that is produced by a exponential or Cauchy distributed process [40]. The impulse responses in this experiment were modeled as an exponentially distributed process with a maximum filter length of 10 taps, as shown in Figure 5.7.

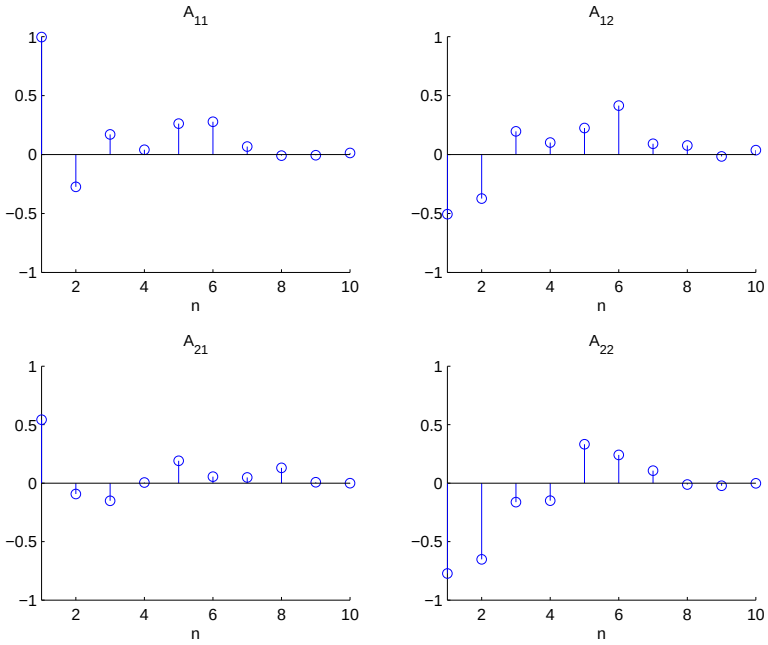


Figure 5.7: The mixing matrix A for the second convolutive mixture example

5.3.3.4 Parameters and Results

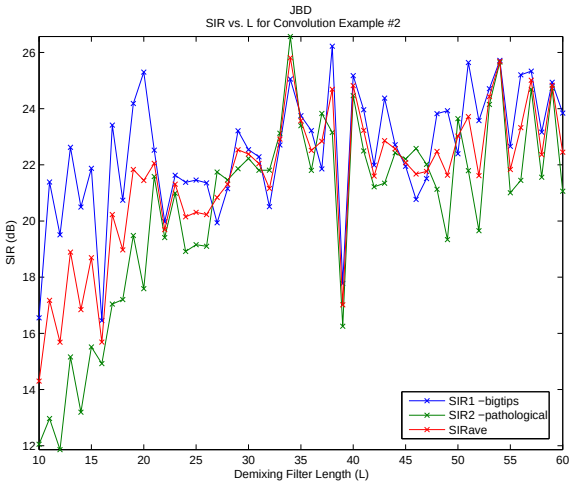
The parameter setup for this experiment is shown in Table 5.3.

The improvement in SIR for the individual algorithms are shown in Figure 5.8. A comparison of all of the algorithms is given in Figure 5.9.

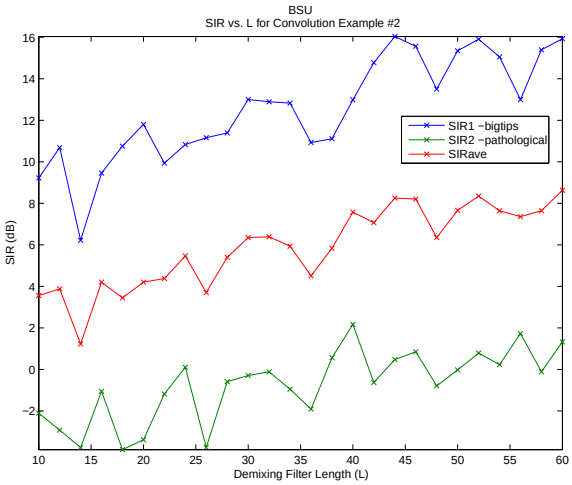
Figure 5.9 again shows similar performance for the algorithms. Although the BSU algorithm does show an improvement in the SIR, it continues to show subpar performance compared to the other four algorithms.

Table 5.3: Parameter Summary for the Second Convolution Example

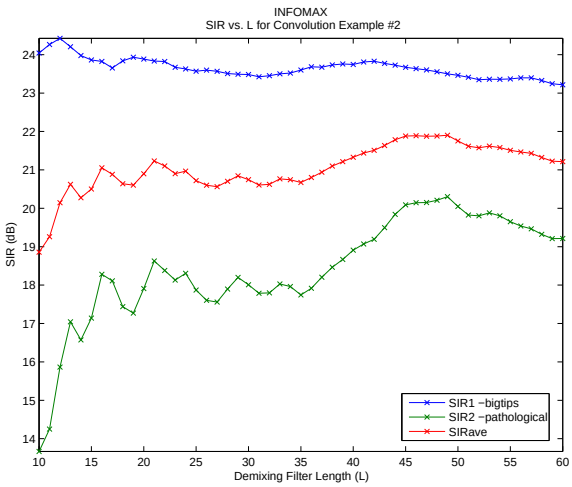
Algorithm	**Parameters**
INFOMAX	$\mu = 0.00005$
NHBSS	$\boldsymbol{\mu} = [0.0001, 0.0001]$
GENSOS	$\boldsymbol{\mu}_{\min} = [0.0045, 0.005]$ $\boldsymbol{\mu} = [0.08, 0.06]$ $\alpha = [0.8, 0.69]$
BSU	$\mu = 5 \times 10^{-6} \cdot L$
JBD	$\mu = 0.5,\ K = 10$



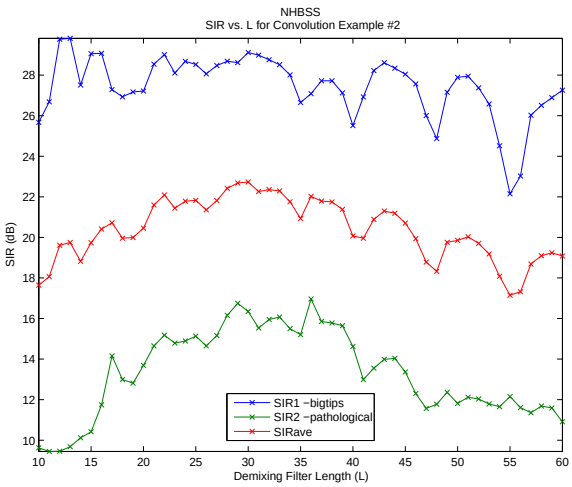
(a): JBD



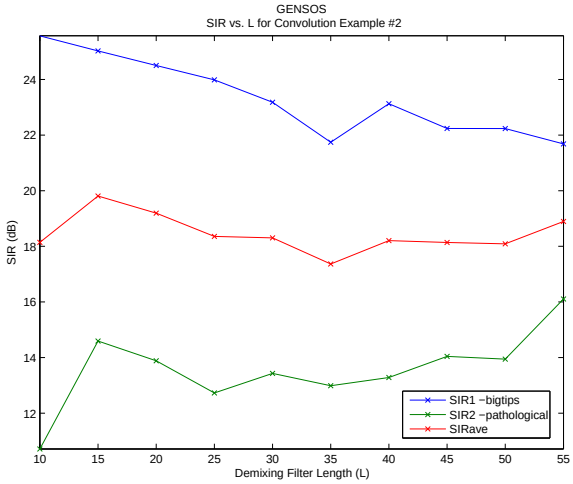
(b): BSU



(c): INFOMAX



(d): NHBSS



(e): GENSOS

Figure 5.8: SIR versus L for the Second Convolution Example

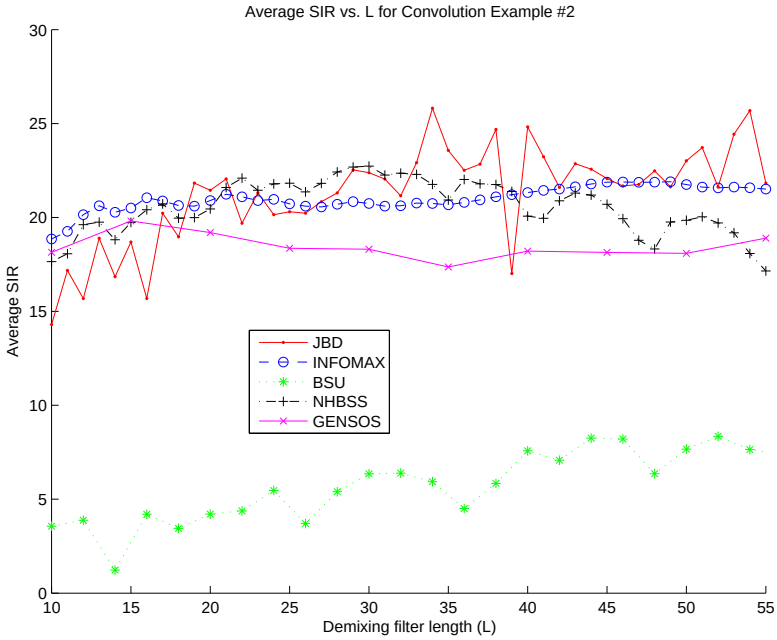


Figure 5.9: Comparison of the Average SIR for the Second Convolution Example

5.3.3.5 Convolutive Mixing: Example 3

We have shown examples where all of the algorithms successfully separate convolutively mixed signals. We will now show an example where the algorithms do not perform as well. For this example, we again used the exponential channel model of the previous example. The resulting channel impulse responses are shown in Figure 5.10.

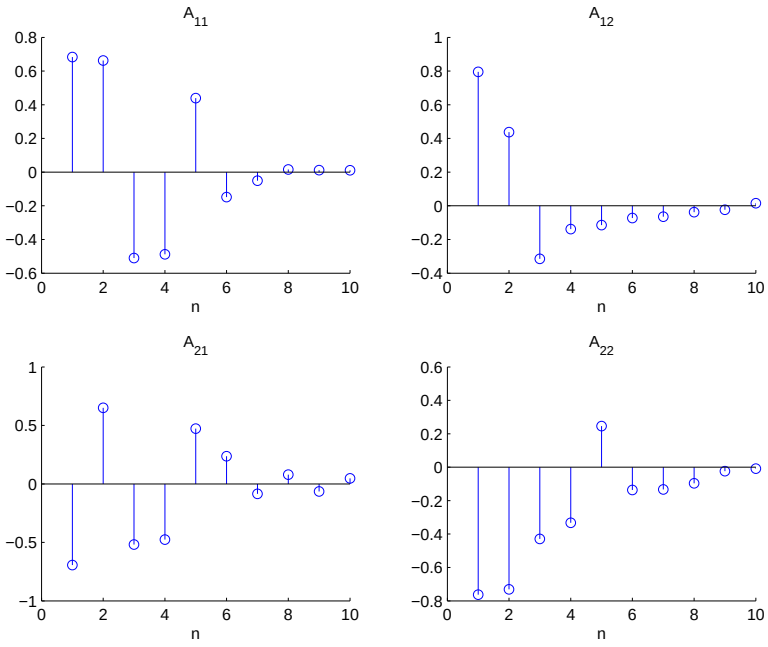


Figure 5.10: The mixing matrix A for the third convolutive mixture example

5.3.3.6 Parameters and Results

The parameters chosen for this experiment are given in Table 5.4.

The improvement in SIR for the individual algorithms are shown in Figure 5.11. The comparison of the algorithms is shown in Figure 5.12.

Each of the algorithms had limited to no ability to separate the mixed signals for this example. Figure 5.11(a) shows that the JBD algorithm in fact made the SIR worse for the second signal since the SIR for the second signal is negative. The same results occurred for the BSU algorithm, although to a greater extent. This suggests

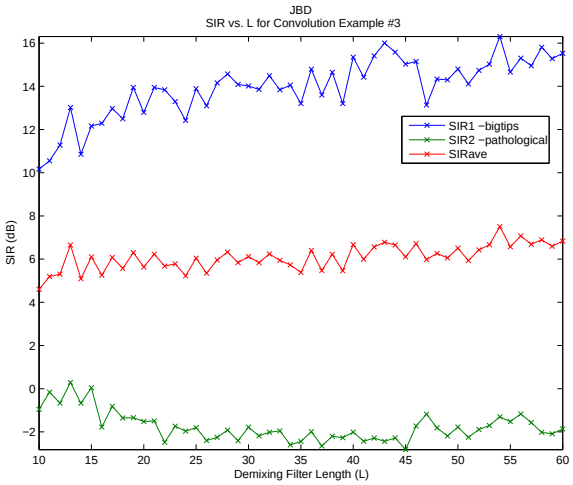
Table 5.4: Parameter Summary for the Third Convolution Example

Algorithm	**Parameters**
INFOMAX	$\mu = 0.00005$
NHBSS	$\boldsymbol{\mu} = [0.0001, 0.0001]$
GENSOS	$\boldsymbol{\mu}_{\min} = [0.03, 0.03]$ $\boldsymbol{\mu} = [0.08, 0.08]$ $\alpha = [0.5, 0.7]$
BSU	$\mu = 5 \times 10^{-6} \cdot L$
JBD	$\mu = 0.5,\ K = 10$

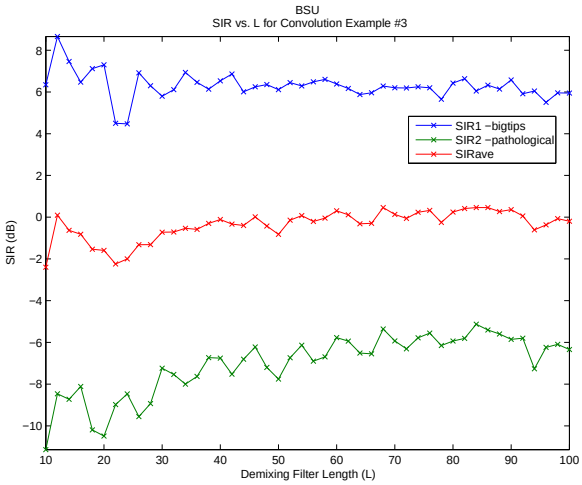
that the JBD algorithm may have trouble dealing with channels that have deep zeros, as does this example.

The NHBSS algorithm shows a strong decrease in performance as the demixing filter length increases. The author noticed during this research that the NHBSS algorithm was the most sensitive to choosing the correct filter length L of the demixing filters.

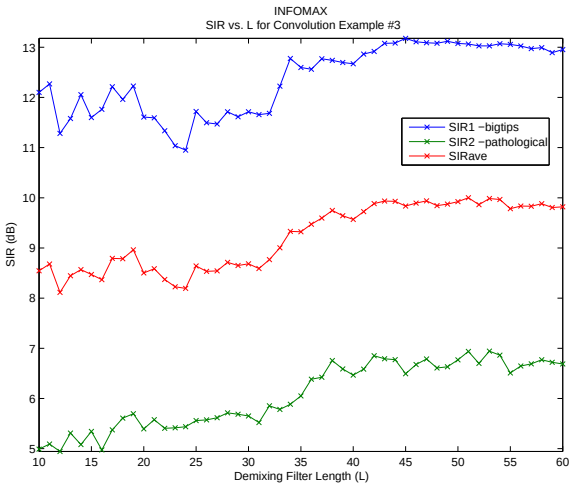
Figure 5.12 shows the compared results for this experiment. It is important to note that although it appears the JBD performed as well as the NHBSS, INFOMAX and GENSOS algorithm, it did in fact degrade the SIR for the second signal, whereas the INFOMAX, NHBSS and GENSOS algorithms all showed an improvement in SIR for both separated signals.



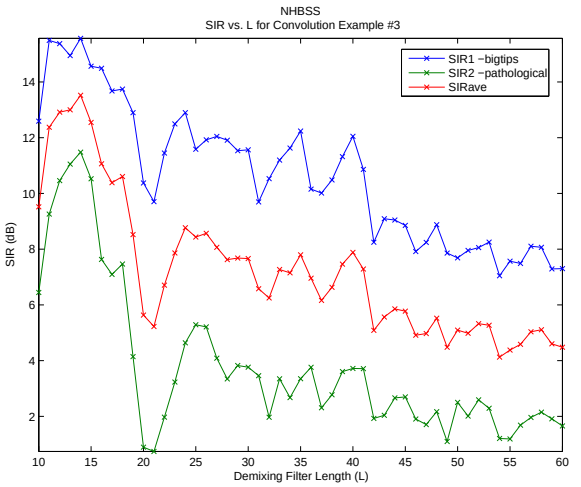
(a): JBD



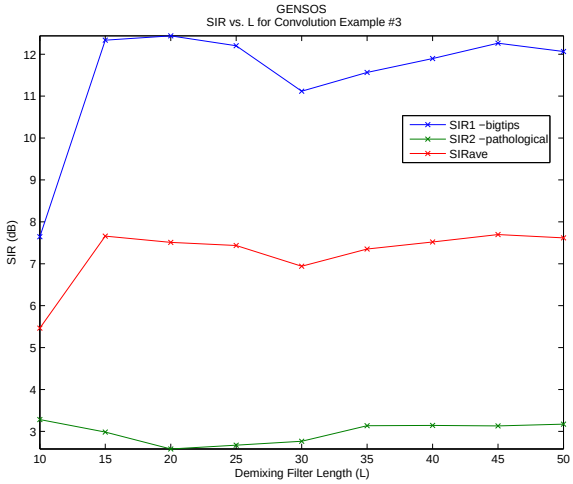
(b): BSU



(c): INFOMAX



(d): NHBSS



(e): GENSOS

Figure 5.11: SIR versus L for the Third Convolution Example

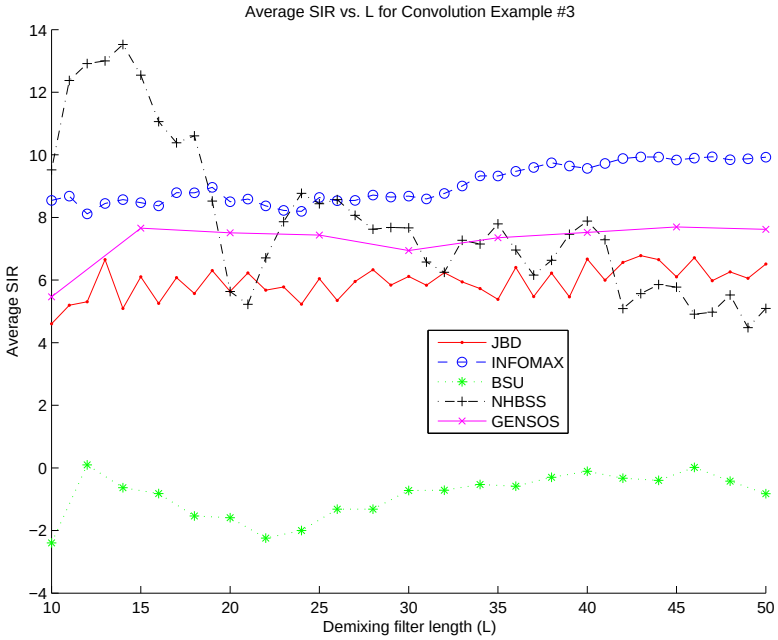


Figure 5.12: Comparison of the Average SIR for the Third Convolution Example

5.3.4 Signals with Similarity

In this section, the ability of the algorithms to separate signals with similarity will be evaluated.

5.3.4.1 Same Person Saying Two Different Sentences

This experiment consists of a single female saying two different sentences. Although this does not represent many true scenarios, unless we have twins speaking at the same time, it will give a performance measure on how well the different algorithms can deal with similar voices. The sentences spoken by the speaker were

- $s_1 =$ `Biblical scholars argue history`.

- $s_2 =$ `You always come up with pathological examples`.

The time-domain and frequency-domain plots of the original sources are shown in Figure 5.13 and Figure 5.14, respectively. As can be seen from Figure 5.14, the spectra of the two source signals are similar due to the fundamental properties of a person's voice [51].

The mixing process chosen for this example is the same of that of the second convolution example as given in Figure 5.7.

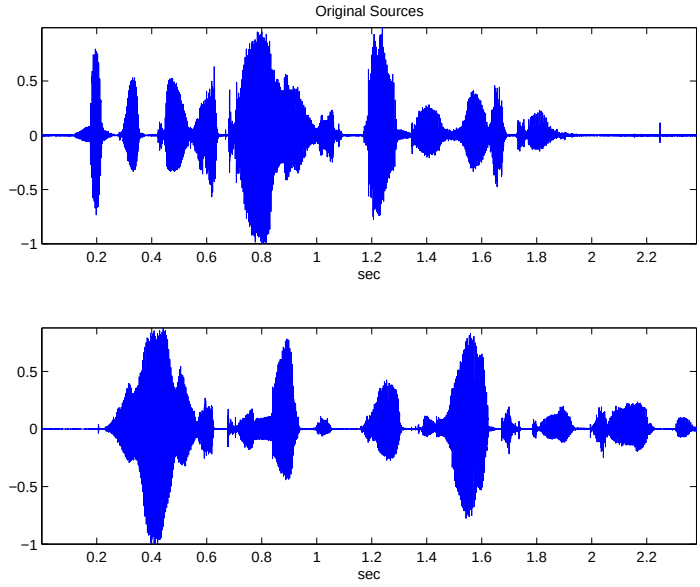


Figure 5.13: Original source signals in the time-domain.

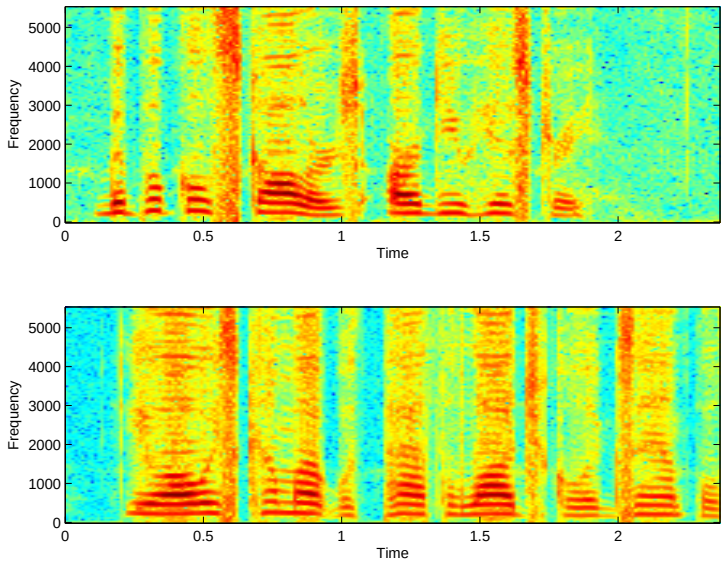


Figure 5.14: Spectrogram of the original sources showing similar spectral content.

5.3.4.2 Parameters and Results

The parameters for this simulation are shown in Table 5.5

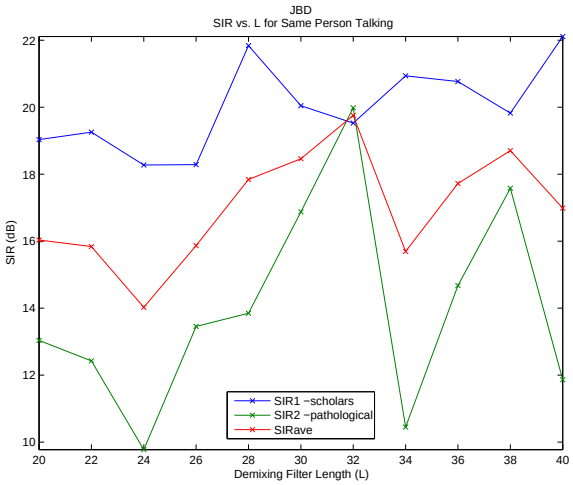
Table 5.5: Parameter Summary for the Same Talker Example

Algorithm	**Parameters**
INFOMAX	$\mu = 1.8 \times 10^{-4}$
NHBSS	$\boldsymbol{\mu} = [0.0001, 0.0001]$
GENSOS	$\boldsymbol{\mu}_{\min} = [0.015, 0.0052]$ $\boldsymbol{\mu} = [0.06, 0.03]$ $\alpha = [0.6, 0.6]$
BSU	$\mu = 6 \times 10^{-5} \cdot L$
JBD	$\mu = 1,\ K = 8$

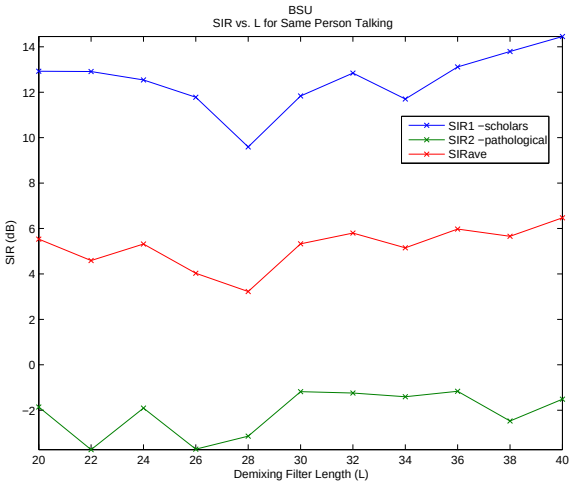
The improvement in SIR for the individual algorithms are shown in Figure 5.15. A comparison of all of the algorithms is given in Figure 5.16.

These results suggest that spectral similarity between speakers does not seem to pose a problem in the different algorithms' abilities to separate the signals. In fact by comparing Figure 5.16 and Figure 5.9, little to no degradation in performance is seen by using the same person speaking. Although this is by no means a comprehensive example, this does imply that signals with similar spectral content do not pose a big problem for these selected blind source separation algorithms.

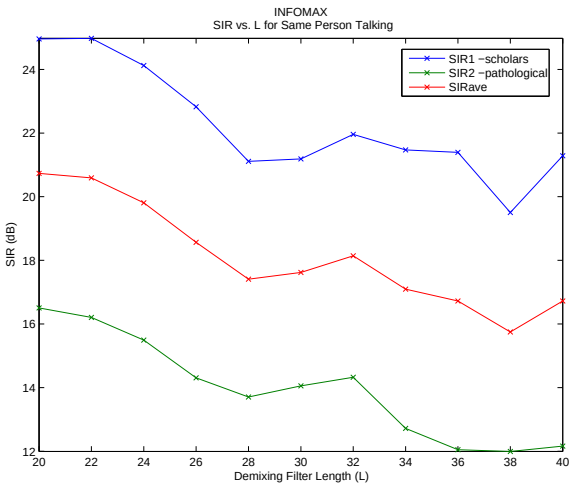
The BSU algorithm again shows very limited performance compared to the other four algorithms.



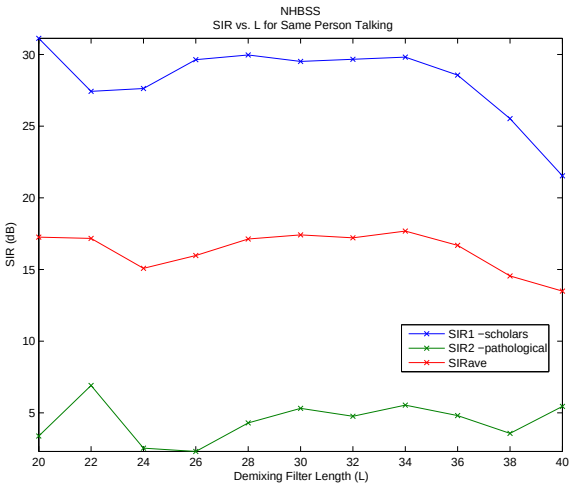
(a): JBD



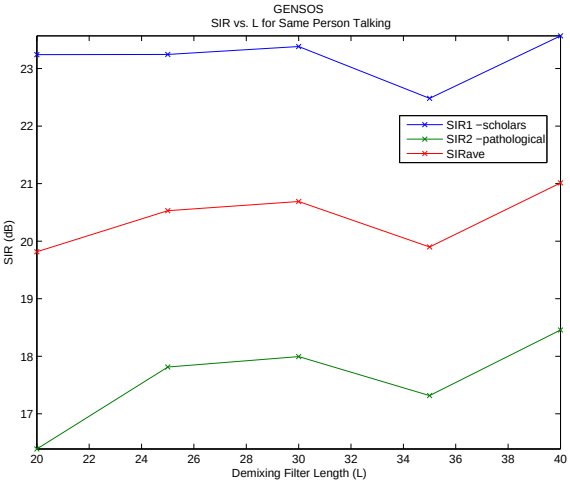
(b): BSU



(c): INFOMAX



(d): NHBSS



(e): GENSOS

Figure 5.15: SIR versus L for the Same Speaker Example

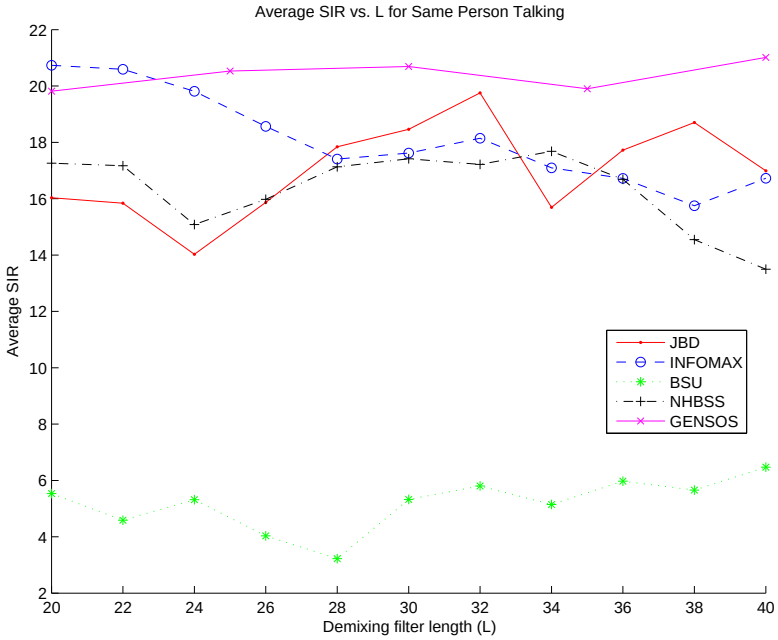


Figure 5.16: Comparison of the Average SIR for the Same Speaker Example

5.3.4.3 Two Different Speakers Saying The Same Sentence

This experiment consists of 5s of a male and female saying the exact same sentence, shown in Figure 5.17. The signals were sampled at 16kHz. These signals were taken from the Orator speech corpus [39]. The sentence spoken was

- `Die Ziele, die wir jetzt verfolgen, sind die gleichen und müssen auch auf die gleiche Weise behandelt werden.`

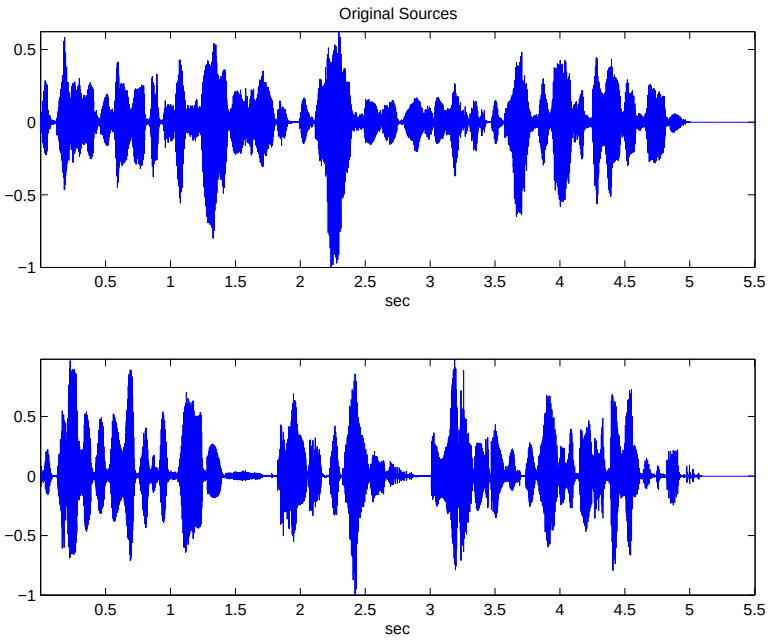


Figure 5.17: Original source signals

The mixing process chosen for this example is again the same of that of the second convolution example, as given in Figure 5.7.

5.3.4.4 Parameters and Results

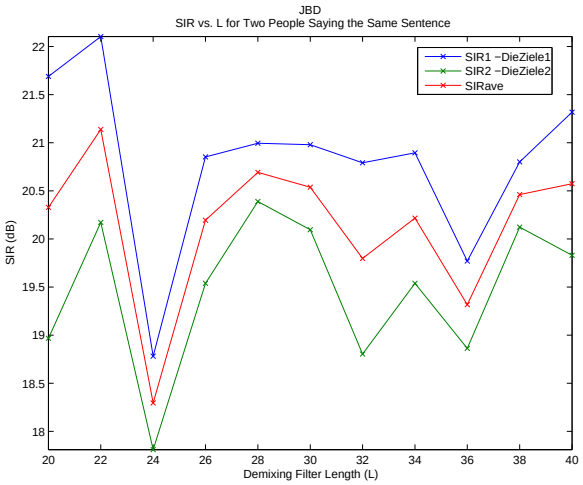
The parameters for this experiment is given in Table 5.6.

The improvement in SIR for the individual algorithms are shown in Figures 5.18. A comparison of all of the algorithms is given in Figure 5.19.

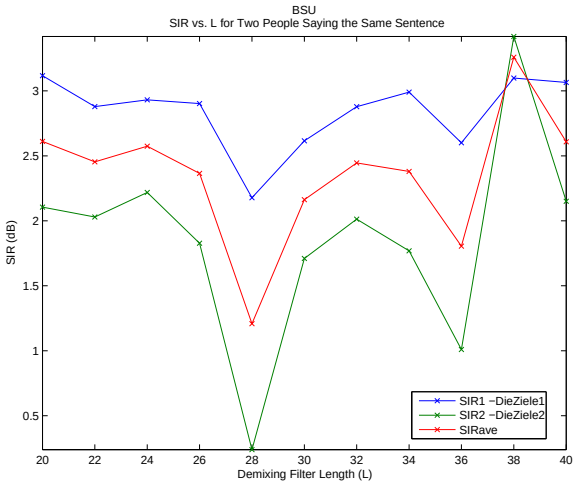
Table 5.6: Parameter Summary for the Same Sentence Example

Algorithm	**Parameters**
INFOMAX	$\mu = 1.8 \times 10^{-4}$
NHBSS	$\boldsymbol{\mu} = [0.0001, 0.0001]$
GENSOS	$\boldsymbol{\mu}_{\min} = [0.035, 0.0065]$ $\boldsymbol{\mu} = [0.06, 0.03]$ $\alpha = [0.8, 0.3]$
BSU	$\mu = 1.5 \times 10^{-5} \cdot L$
JBD	$\mu = 1,\ K = 8$

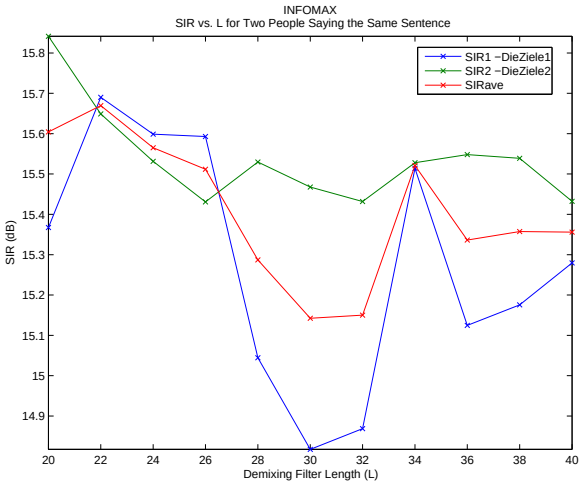
These results show that these algorithms are able to separate the signals even though they have the same temporal characteristics. These are somewhat surprising results, given that the most important assumption of blind source separation is independence of the original source signals. Since the speakers are saying the same things we expect some temporal dependence between the signals. Perhaps the different spectral content of the signals and different "interpretations" of the same wording give the algorithms enough diversity to separate the signals. The BSU continues to have poor performance compared to the other algorithms.



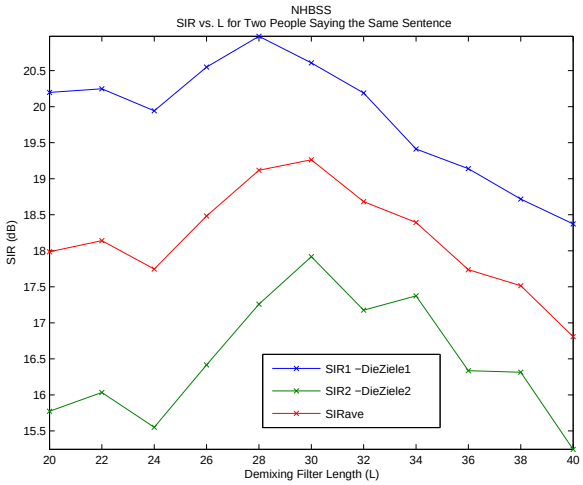
(a): JBD



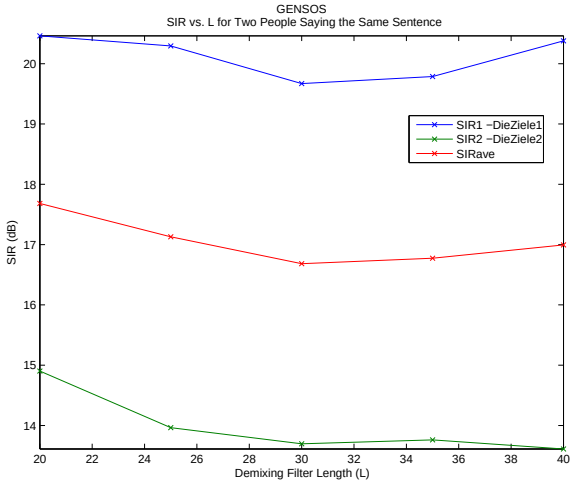
(b): BSU



(c): INFOMAX



(d): NHBSS



(e): GENSOS

Figure 5.18: SIR versus L for the Same Sentence Example

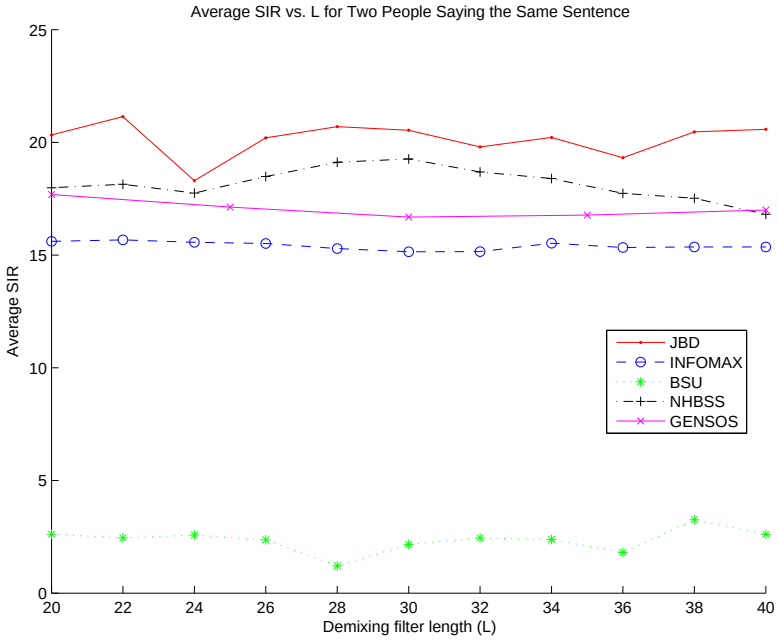


Figure 5.19: Comparison of the Average SIR for the Same Sentence Example

5.3.5 Adding extra channels

Simulations were conducted in order to determine whether adding more microphones improved the performance of the algorithms. The results showed that little to no improvement was made by adding extra microphones, thus they will not be shown here. In [34], Parra and Spence showed significant improvement in SIR by adding more microphones. Reasons for this discrepancy are still unknown. In [50], Westner shows how the addition of extra microphones may add to the ability of blind source separation algorithms to separate convolutively mixed signals. A potential reason for the lack of performance enhancement is the fact that only the NHBSS and JBD algorithms are derived without the assumption of an equal number of microphones as source signals. For this reason, the algorithms as they are presented may not be able to exploit the extra information gained by using more microphones than sources. A more in depth investigation for this case of more microphones than sensors is thus a part of the future work.

The author found that adding more microphones than sources did not degrade performance, it simply did not increase the separability of the mixed signals significantly, in most cases the improvement in separation, if any, was only one dB. The author did notice, however, that having extra sources did not hinder the separability of the algorithm to separate the signals. For example, assume there are two speakers and 4 microphones. Since there are four microphones, it is possible with these algorithms to separate four different original source signals. However, in this case, there are only 2 sources in which to separate. In most literature, this scenario of overdetermined blind source separation is solved by first estimating the number of sources via Principal Component Analysis (PCA). Upon estimation of the number of sources, the blind source separation problem is then reduced to the standard ICA problem in which there are equal number of sources and microphones. The author noticed, however, that this reduction is not necessarily needed. The different algorithms were able to separate the signals in which there were more microphones. For the two sources

and four microphone setup, the algorithms estimated the two original sources along with two other, small amplitude mixtures of the same sources. Thus it is possible to still separate the signals without the need for PCA to reduce the problem. The only thing left to do in the overdetermined case is then to find where the original source signals are, e.g., which separated signals correspond to the original source signals and which signals are unneeded. This is done at a higher computational cost due to the extra microphone signals. At the time this dissertation was written, the author was unaware of a comparison between the tradeoff between the computational complexity of using extra microphones versus the dimension reduction of PCA.

5.3.6 Conclusion for Synthetic Mixtures

The experiments presented in this section showed that the chosen algorithms were able to separate convolutively mixed signals in several different scenarios. The BSU algorithm was the only algorithm that failed to offer much improvement in separation quality for all of the experiments.

5.3.6.1 Comments on Stability

The INFOMAX, BSU and JBD algorithms were very stable for these simulations. Little was needed in the ways of optimizing parameters. Their convergence was very stable, and upon convergence, the demixing filters did not diverge from their final solution.

The GENSOS algorithm tended to become unstable, most likely due to the inversion of the correlation matrices. This was remedied with the adaptive step size control as described by Eq. (5.3). Although this technique is not necessarily needed for all scenarios, it did help increase the convergence speed without leading to instability.

The NHBSS algorithm seemed to be the most unstable of the five algorithms. The NHBSS algorithm would begin to diverge upon reaching the optimized filter coefficients of the demixing system, unless processing was terminated or the step size was dramatically reduced.

5.3.6.2 Comments on Computational Complexity

The JBD, NHBSS, INFOMAX, and the GENSOS algorithms all performed similarly for each of the examples presented here. With that being said, the JBD algorithm definitely has the edge when it comes to computational complexity. Due to the fact that it is a frequency-domain algorithm, JBD is significantly faster than the other algorithms. Admittedly however, there is much room for improvement for both the NHBSS and GENSOS algorithms. The implementations used for these experiments were not optimized using fast convolution techniques. The reason for this was to be able to attain the optimal results for the NHBSS and GENSOS algorithms. In [19,43] Douglas and Sun make use of a block-based update employing fast FFT based convolution. This would decrease the computational complexity of the NHBSS algorithm. In [1,2], Aichner et al. make assumptions that lead to an efficient real-time implementation of the GENSOS algorithm. However, these faster implementations sacrifice a small amount of performance for this decrease in complexity.

The computational complexity of the GENSOS algorithm implemented in this dissertation dramatically increases as L increases. In fact, the computational complexity of the algorithm described here is $O(L^3)$ [11]. Thus, the time to produce results using this algorithm were around 15 hours for demixing filters lengths on the order of 1000. But again the GENSOS algorithm was implemented in its most generic form and can be easily extended to a real-time algorithm.

The INFOMAX algorithm showed the greatest improvement in SIR for almost all experiments. This however is again done at a higher computational cost. To the author's knowledge at the time this dissertation was written, no real-time nor faster implementations are currently available for the INFOMAX algorithm due to its feedback architecture. However Lee and Orglmeister extended the original Infomax algorithm of Bell and Sejnowski to employ fast convolution techniques using a feedforward architecture in [33]. This, however, is not the feedbackward architecture discussed in this dissertation.

5.4 Recorded Signals Simulations

In this section, the performance of the five chosen algorithms will be compared by using recorded signals. The recordings were made with a Fostex MR-8HD multitracker and MXL 990 microphones. The microphones were unidirectional with a cardioid pick-up pattern. The Fostex MR-8HD is capable of sampling four channels simultaneously at 44.1kHz. The recorded signals were then downsampled to 16kHz using Adobe Audition. The signals were recorded using the one-at-a-time method as described by Schobben in [40]. This was done in order to quantitatively evaluate the separation performance.

The one-at-a-time method of recording allows for the calculation of the SIR of the separated signals. In this technique , the recordings are made when only one person is speaking. So for the two-speaker, two-microphone case, the first person talks and his/her voice is recorded at both microphones and then the second person speaks. Thus we will have four recordings, the first person at microphones one and two and the second speaker at microphones one and two. The mixed data is then obtained by summing up the individual voices at each microphone. This is well justified since sound waves are additive. The only downside to this technique is that no change in the acoustical environment can take place between the two people talking. Thus all speakers must be present and in the same position, even when they are silent. Care must also be taken such that the background sounds do not change when the different speakers are recorded.

Using this technique, the SIR for the j^{th} separated signal can be calculated as

$$SIR_j = 10 \log \frac{E\left\{(\hat{s}_{j,s_j})^2\right\}}{E\left\{(\sum_{i \neq j} \hat{s}_{j,s_i}\right\})^2} \tag{5.6}$$

where $\hat{s}_{j,s_j}$ is the j^{th} separated signal with only s_i active.

5.4.1 Recordings Taken in a Large Conference Room

The first recording tests were done in a large conference room in the Electrical and Computer Engineering Department at Texas Tech University. Four microphones were spaced 0.25m (10 inches) apart in a linear array. A male and female speaker were placed directly in front of the end microphones at a distance of 0.76m (30 inches). Figure 5.20, drawn to scale, shows the dimensions of the room as well as the setup of the speakers and microphones. The speech signals were 11 seconds long, again at a sample rate of 16kHz.

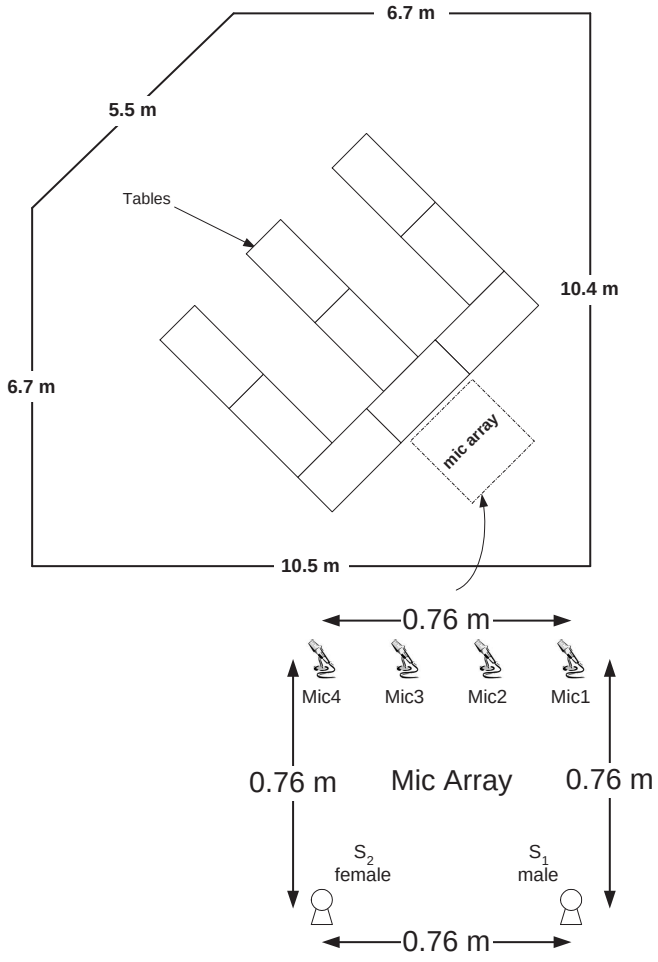


Figure 5.20: Layout of the Large Conference Room. The ceiling height was 3.15m (124 inches).

Using the one-at-a-time method of recording, it is possible to select any subset of the microphone array. First, define Setup 1 by selecting microphones one and four from the microphone array. Setup 2 will consist of selecting microphones two and three from the array.

Choosing Setup 1 would seem to be the best choice, because the speakers are directly in front of the microphones. This would be the easiest setup for the two speaker/microphones case since the SIR of the mixtures would be highest for the microphone directly in front of the speaker. Despite this fact, the author chose to use Setup 2. This was done in order to present a harder scenario for the algorithms, since there is not a dominant presence of either speaker at either microphone.

Experiments were also performed using Setup 1. When compared with experiments using Setup 2, practically equal performance was achieved by all algorithms. However, because of the strong presence of the original sources in the microphone directly in front of them, Setup 1 had a lower improvement in SIR for most algorithms. Thus by choosing Setup 2, better comparison on the performance of the algorithms can be judged.

As mentioned in section 5.3.5, adding more microphones did not dramatically improve the ability of the algorithms to separate the signals. Thus no simulations on extra microphones will be shown here.

The parameters for this experiment are shown in Tables 5.7, 5.8, and 5.9. Again each algorithm is iterated until convergence.

An adaptive step-size method was used for the GENSOS algorithm in all synthetic mixture experiments. Due to the large filter lengths simulated here, calculation of the objective function added a large computational strain. Due to this extra added computational need, the adaptive step size method was not used for these simulations, as the increase in convergence speed was not enough to justify the added computational load.

Table 5.7: Parameter Summary for the Large Conference Room, $L = 256$.

Algorithm	**Parameters**
INFOMAX	$\mu = 10^{-5}$
NHBSS	$\boldsymbol{\mu} = [10^{-5}, 10^{-5}]$
GENSOS	$\boldsymbol{\mu} = [0.2, 0.1]$
BSU	$\mu = 0.01$
JBD	$\mu = 1,\ K = 8$

Table 5.8: Parameter Summary for the Large Conference Room, $L = 512$.

Algorithm	**Parameters**
INFOMAX	$\mu = 2 \cdot 10^{-5}$
NHBSS	$\boldsymbol{\mu} = [8 \cdot 10^{-6}, 8 \cdot 10^{-6}]$
GENSOS	$\boldsymbol{\mu} = [0.2, 0.1]$
BSU	$\mu = 0.01$
JBD	$\mu = 1,\ K = 8$

Table 5.9: Parameter Summary for the Large Conference Room, $L = 1024$.

Algorithm	**Parameters**
INFOMAX	$\mu = 5 \cdot 10^{-6}$
NHBSS	$\boldsymbol{\mu} = [10^{-5}, 10^{-5}]$
GENSOS	$\boldsymbol{\mu} = [0.2, 0.2]$
BSU	$\mu = 0.01$
JBD	$\mu = 1,\ K = 8$

The SIR was calculated for three different lengths of the demixing filter: $L = 256, L = 512,$ and $L = 1024$. The values of L were chosen as a power of two in order to allow for use of the Fast Fourier Transform for the frequency-domain algorithms (BSU, JBD).

Table 5.10: Improvement in signal-to-interference ratios in dB for the conference room experiment. The original mixtures had an SIR of 1.5dB and 2.1dB for microphone two and microphone three, respectively.

Algorithm	$L = 256$		$L = 512$		$L = 1024$	
	$\mathbf{SIR_1}$	$\mathbf{SIR_2}$	$\mathbf{SIR_1}$	$\mathbf{SIR_2}$	$\mathbf{SIR_1}$	$\mathbf{SIR_2}$
JBD	9.3	10.5	10.0	12.7	10.6	12.0
BSU	2.0	-2.1	4.2	-0.3	4.6	1.2
NHBSS	8.2	7.7	8.0	7.3	10.7	7.9
INFOMAX	9.7	8.8	7.6	4.7	5.1	3.7
GENSOS	8.0	11.6	9.4	11.4	8.7	14.3

Table 5.10 shows the signal-to-interference ratios for each choice of L. The BSU algorithm again showed very poor performance. There is a noticeable increase in SIR of the separated signals for the BSU algorithm as the filter length L increases, inferring that a longer demixing filter may improve the ability of the BSU algorithm to separate the mixed signals. In fact, increasing L to 2048 resulted in an improvement in SIR of 6.0 and 2.6 for the estimated signals. This method in general required significantly longer demixing filters in order to achieve separation. All other algorithms showed a decrease in separation quality once the filter length was increased past $L = 1024$ due to over-estimating the filter length of the mixing system.

The JBD, NHBSS, and GENSOS algorithms all showed similar performance, with an edge going to the JBD algorithm. The JBD algorithm proved to be fairly insensitive to overestimating the demixing filter length as shown by the fact that little change in SIR occurs upon increasing the demixing filter from 512 to 1024.

The INFOMAX algorithm showed a definite inability to deal with longer filter lengths. By inspecting Table 5.10, the performance of the INFOMAX algorithm sharply decreases. In fact, as L increased, the INFOMAX algorithm became quite unstable, and its poor performance as L increases reflects this instability.

The NHBSS algorithm showed a decrease in separation quality when increasing L from 256 to 512, and then an increase in performance when L is chosen as 1024. This matches the results of the synthetic mixing case, in which the performance of this algorithm depended strongly on the choice of the demixing filter length. In Figure 5.5(a), a difference of about 10dB in SIR can be seen by choosing an appropriate length of the demixing filter. The results shown here for the real convolutive mixing showed this same behavior, suggesting that this algorithm is sensitive to correctly choosing the length of the demixing filters. The NHBSS algorithm is the only method that showed this kind of behavior.

It is interesting to note that of the three algorithms that performed well in this scenario, the JBD and GENSOS algorithms were able to separate the second signal better than the first, whereas the NHBSS algorithm showed better separation quality in the first signal. This may be explained by the original source signals themselves. Upon listening to the observed signals before they are mixed, it is easy to notice that the first speaker is much more monotone with less change in emphasis throughout the recordings, whereas the second speaker shows tone and volume variation. Since both the JBD and GENSOS algorithm exploit the nonstationarity of the speech signals, it would not be a stretch to say that they were better able to separate the second source, due to its nonstationarity. The NHBSS algorithm does not utilize the nonstationarity of speech, and perhaps this is the cause for its better performance on the more monotone signal.

5.4.2 Conclusion for Large Conference Room Experiment

Out of the five algorithms evaluated using recorded signals, three of them showed the ability to separate the mixed signals. The BSU algorithm continued to show poor performance, just as in the synthetic mixture experiments. Contrary to the synthetic mixtures, the INFOMAX algorithm had a limited ability to separate the recorded signals. It especially showed a performance degradation as the length of the demixing filters was increased. The JBD, NHBSS, and GENSOS algorithm all had success in separating the signals.

5.4.2.1 Comments on Stability

INFOMAX was the only algorithm that suffered from instability. The step size for this method must be chosen to be very small, and then many iterations must be done in order to reach convergence. Also, when the INFOMAX algorithm became close to convergence, many times the step size had to be reduced in order to keep it from becoming unstable. This was done manually by simply hard coding the step size change at some point during the optimization, and then continuing to iterate until the optimal solution was found.

This instability lead to its poorer performance compared to the other algorithms. For the synthetic mixture experiments, the INFOMAX algorithm was one of the best performing algorithms, but for the real recordings presented in this section, it was one of the worst performing. This suggests that the INFOMAX algorithm has trouble dealing with long mixing filters.

The NHBSS algorithm did not appear to be unstable, but it did tend to want to begin diverging once the optimal demixing filters were found. Unlike the JBD, BSU, and GENSOS algorithms, continuing to process the data once convergence has been reached lead to degraded performance. Special care was taken with the NHBSS algorithm in order to stop updating the demixing filters immediately once the optimal solution was found.

The JBD and BSU algorithm both showed to be very stable. All that needed to be done was to choose the step size up to the stability margin. Upon convergence, the demixing filters showed no noticeable change in the coefficients, no matter how much longer the updates were allowed to continue.

Unlike the synthetic mixture experiments, the GENSOS algorithm proved to be very stable as well and thus no adaptive step size control was needed. It also was not sensitive to over processing of the demixing filters, meaning once the algorithm converged, it varied very little upon further iteration over the data.

5.4.2.2 Comments on Computational Complexity

It was mentioned in section 5.3.6 that the JBD and BSU algorithms are significantly faster than other algorithms. This becomes even more apparent in these simulations as the demixing filter length is long. In fact, for the experiment where $L = 1024$, the frequency-domain algorithms (JBD, BSU) convergence was reached after a few minutes, whereas for the time-domain algorithms, convergence was anywhere between 10 to 15 hours.

5.4.3 Recordings Taken in a Living Room

The next recording tests were done in a much smaller room. The room was the living room of the author, where many different items were in the room. This included a piano, both carpeted and tiled floor, a television, and bookcases. This is a harder acoustical environment than the first example, where the recordings where made in a completely carpeted and large conference room.

Four microphones were again placed 0.25m (10 inches) apart in a linear array. A male and female speaker were placed directly in front of the end microphones at a distance of 0.76m (30 inches), and eight seconds of speech data was taken. Figure 5.21 shows the setup of the speakers and microphones.

Again the one-at-a-time method was used to make the recordings. As in the previous case, the microphones used were microphones two and three. Again no substantial improvement in SIR was found by using more microphones for any algorithm.

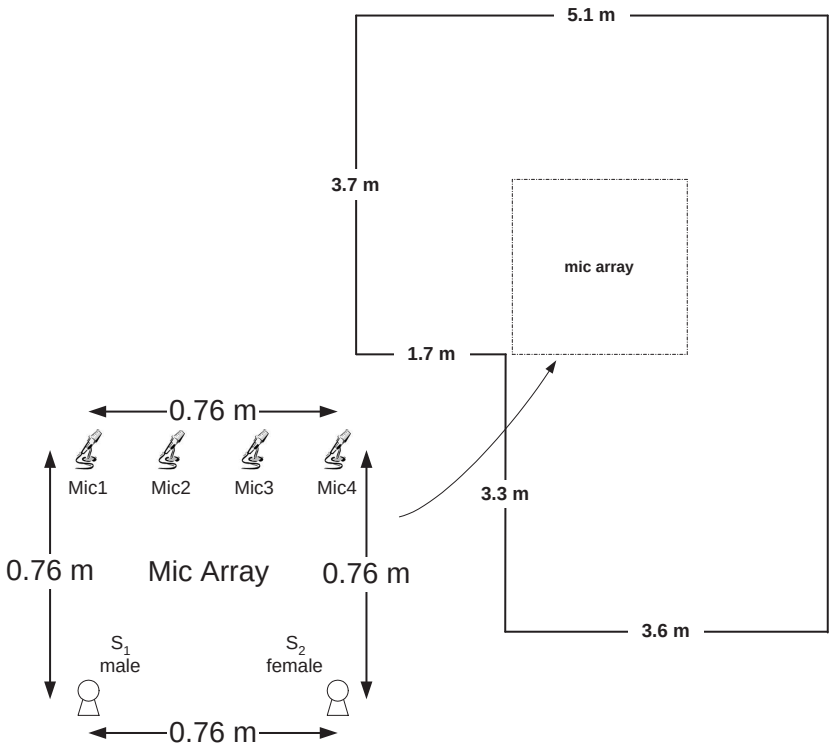


Figure 5.21: Layout of the Living Room. The ceiling height was 2.42m (96 inches).

The SIR was calculated for three different lengths of the demixing filter: $L = 256, L = 512, L = 1024$. The parameters for each simulation are shown in Tables 5.11, 5.12, and 5.13.

The resulting improvements in SIR are shown in Table 5.14. As in the previous simulations, the BSU and INFOMAX algorithms showed poor results compared with the other three. The INFOMAX algorithm again suffered from instability as L is increased, thus suggesting that this algorithm does not deal well with long mixing systems. These recordings, as expected, proved to be a more difficult acoustic environment than the large conference room, due to all of the objects present in the room where these recordings were acquired.

Table 5.11: Parameter Summary for the Living Room, $L = 256$.

Algorithm	**Parameters**
INFOMAX	$\mu = 2 \cdot 10^{-6}$
NHBSS	$\boldsymbol{\mu} = [10^{-5}, 10^{-5}]$
GENSOS	$\boldsymbol{\mu} = [0.5, 0.1]$
BSU	$\mu = 0.01$
JBD	$\mu = 1,\ K = 5$

Table 5.12: Parameter Summary for the Living Room, $L = 512$.

Algorithm	**Parameters**
INFOMAX	$\mu = 5 \cdot 10^{-6}$
NHBSS	$\boldsymbol{\mu} = [8 \cdot 10^{-6}, 8 \cdot 10^{-6}]$
GENSOS	$\boldsymbol{\mu} = [0.5, 0.05]$
BSU	$\mu = 0.01$
JBD	$\mu = 1,\ K = 5$

Table 5.13: Parameter Summary for the Living Room, $L = 1024$.

Algorithm	**Parameters**
INFOMAX	$\mu = 10^{-5}$
NHBSS	$\boldsymbol{\mu} = [10^{-5}, 10^{-5}]$
GENSOS	$\boldsymbol{\mu} = [0.5, 0.05]$
BSU	$\mu = 0.01$
JBD	$\mu = 1,\ K = 5$

Table 5.14: Improvement in signal-to-interference ratios in dB for the living room experiment. The original mixtures had an SIR of 2.4dB and 0.2dB for microphone two and microphone three, respectively.

Algorithm	$L = 256$		$L = 512$		$L = 1024$	
	$\mathbf{SIR_1}$	$\mathbf{SIR_2}$	$\mathbf{SIR_1}$	$\mathbf{SIR_2}$	$\mathbf{SIR_1}$	$\mathbf{SIR_2}$
JBD	7.3	10.0	8.7	11.2	8.4	11.1
BSU	3.5	-1.0	3.5	0.1	4.0	1.2
NHBSS	7.2	10.2	7.3	9.8	9.5	11.1
INFOMAX	6.1	3.1	4.8	4.6	4.5	5.2
GENSOS	8.5	11.9	9.5	13.2	8.2	12.3

The JBD, NHBSS, and GENSOS algorithms again were the only algorithms that showed the ability achieve separation. They all, however performed, about 2dB worse than they did in the large conference room. This again was more than likely due to a more complex channel resulting from the many objects in the small room. The JBD and GENSOS algorithms again showed better interference rejection for the female speaker as compared to the more monotone male speaker.

CHAPTER VI
CONCLUSIONS AND FUTURE WORK

Five different blind source separation algorithms were evaluated in several different scenarios. Two frequency-domain algorithms were chosen for evaluation along with three time-domain algorithms. Several conclusions can be drawn from the experiments discussed in this paper, although they are not necessarily conclusive.

The BSU was one of the frequency-domain algorithms evaluated in this dissertation. This algorithm was the worst performing of the five different algorithm. The BSU generally required a significantly longer impulse response in order to successfully separate a mixture of signals. This held true not only for real world recordings but also for all of the synthetic mixtures as well. Since this algorithm is a frequency-domain method, the added filter length did not add much computational load during processing, but it rarely achieved comparable performance with the other algorithms. This is probably due to the fact that the BSU algorithm was originally derived assuming that the source signals were i.i.d. Lambert noticed that the serial update, as opposed to a gradient search method, was more robust to nonstationarity and thus applied the MBLMS algorithm to speech signals, so to be fair, this algorithm should not be expected to perform as well as those designed for specifically for speech signals. Although it did show poor performance when compared to the other algorithms, it was capable of providing some separation for most cases.

The JBD algorithm was the second frequency-domain algorithm evaluated in this dissertation. This method proved to be one of the best performing algorithms. It not only achieved equal if not greater separation than the other algorithms, but it was by far the fastest algorithm. One of the strongest characteristics of the JBD algorithm was its stability. Upon convergence, the JBD algorithm tended to stay at its optimal solution, and did not deviate from its solution. The only drawback of this algorithm is that it had the most parameters in which to tune. However in the

author's experience, the JBD algorithm was not sensitive to these parameters, and comparable performance was achieved within a reasonable range of these parameters.

The time-domain INFOMAX algorithm proved to be one of the best performers for the synthetic mixtures. It failed to meet expectations, however, with the real world recordings. It not only performed poorly with the recordings, but it also showed to be quite unstable for recorded mixtures. This instability and limited ability to achieve separation was most likely due to the long impulse responses required to invert the acoustical environment.

The second time-domain algorithm evaluated in this dissertation was the NHBSS algorithm. This algorithm showed an ability to separate the real world recordings. The biggest drawback found by the author of this algorithm is that the NHBSS algorithm did not seem to want to stay at its optimal solution. Upon reaching convergence, unless processing of the data was stopped or the step size was dramatically decreased, the NHBSS algorithm would begin to slowly diverge. This gives rise to the need for an adaptive step size control.

The GENSOS algorithm was the third time-domain method evaluated in this dissertation. This algorithm proved to be very successful in separating mixed signals. One of the strong points of this algorithm is that is has an inherent normalization by the autocorrelation matrices. This inherent normalization also leads to one of its drawbacks, being the inversion of the autocorrelation matrices. This inversion requires regularization of the autocorrelation matrices prior to inversion. The inversion itself also increases the computational complexity of this algorithm, especially for long filter lengths. Despite this fact, GENSOS proved to be robust to a number of different mixing scenarios, provided the correlation matrices were properly regularized.

6.1 Directions for Future Work

Several topics that would be interesting directions for future work will be discussed in this section.

1. All algorithms presented here were implemented in their offline forms. The ability of an algorithm to perform in a real-time manner is most appropriate for many applications. Hearing aids and teleconferencing are two applications in which a real-time method is necessary. The GENSOS algorithm has already been successfully applied to a real-time application in [2]. All other algorithms, to the author's knowledge, have not been extended to allow operation in a real-time fashion. The NHBSS algorithm can be implemented in a block-wise fashion, and thus is capable of utilizing fast convolution techniques [19]. This, however, is still not a true real-time algorithm. The INFOMAX algorithm has also been implemented via fast convolution techniques in [33], but that paper utilized a feedforward architecture. Real-time extensions and comparisons of these algorithms need to be explored.

2. The choice of the step size for each algorithm was optimized up to the stability margin. This process was done by trial and error, and the optimal step size was not necessarily easily found. While all of the experiments performed for this research were done in a offline manner, the step size still proved to be the most important parameter in which to choose. In [17] and [49] adaptive step size control was presented for instantaneous blind source separation. There seems to be little literature, however, on step size control for convolutive mixtures and thus would be an interesting topic of study.

3. The findings in this dissertation indicated little to no improvement in separation quality when incorporating more microphones than mixtures. This is contrary to the results shown by Parra and Spence in [34] and by Westner in [50]. A more in depth study of having more microphones than sources is needed, including a trade-off between separation quality and increase in computational complexity.

4. The computational complexity of the algorithms was briefly discussed. This, however, was done only by noting the time taken for each algorithm to reach

convergence. Using computation time in order to discuss computational complexity is not only dependent upon the machine the simulations were run, but also on the ability of the programmer to efficiently implement the algorithms. An in depth study is needed to evaluate the computational complexity of each of the algorithms for a better comparison of their performance.

REFERENCES

[1] Robert Aichner, Herbert Buchner, and Walter Kellermann. A novel normalization and regularization scheme for broadband convolutive blind source separation. In *Sixth Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, Charleston, SC, March 2006. To appear.

[2] Robert Aichner, Herbert Buchner, Fei yan, and Walter Kellermann. Real-time convolutive blind source separation based on a broadband approach. In Carlos García Puntonet and Alberto Prieto, editors, *Fifth Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, volume 3195 of *Lecture Notes in Computer Science*, pages 840–848, Granada, Spain, 2004. Springer.

[3] Robert Aichner, Herbert Buchner, Fei Yan, and Walter Kellermann. A real-time blind source separation scheme and its application to reverberent and noisy environments. *Signal Processing*, 2006. to appear(download at http://dx.doi.org using digital object identifier: 10.1016/j.sigpro.2005.06.022).

[4] Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

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

[6] Shun-ichi Amari, S. Douglas, Andrzej Cichocki, and H. Yang. Novel on-line adaptive learning algorithms for blind deconvolution using the natural gradient approach, 1997.

[7] Shun-ichi Amari, Scott C. Douglas, Andrzej Cichocki, and H. Yang. Multichannel blind deconvolution and equalization using the natural gradient. In *Proceedings of the IEEE Workshop on Signal Processing and Advances in Wireless Communication*, pages 101–104, Paris, France, April 1997.

[8] Anthony J. Bell and Terrence J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.*, 7(6):1129–1159, 1995.

[9] Sandro Bellini. Bussgang techniques for blind deconvolution and equalization. In Simon Haykin, editor, *Blind Deconvolution*, chapter 2, pages 8–59. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1994.

[10] Herbert Buchner, Robert Aichner, and Walter Kellermann. A generalization of a class of blind source separation algorithms for convolutive mixtures. In *Proc. 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA 2003)*, pages 945–950, Nara, Japan, April 2003.

[11] Herbert Buchner, Robert Aichner, and Walter Kellermann. A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics. *IEEE Transactions on Speech and Audio Processing*, 13(1):120–134, 2005.

[12] Jean-François Cardoso. The equivariant approach to source separation. In *Proc. NOLTA*, pages 55–60, 1995.

[13] Jean-François Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, 4(4):112–114, 1997.

[14] Jean-François Cardoso. The three easy routes to independent component analysis; contrasts and geometry. In *Proc. ICA 2001, San Diego*, 2001.

[15] Jean-François Cardoso and Antoine Souloumiac. Blind beamforming for non gaussian signals. *IEE Proceedings-F*, 140(6):362–370, 1993.

[16] Scott C. Douglas. Blind signal separation and blind deconvolution. In Yu Hen Hu and Jenq-Neng Hwang, editors, *Handbook of Neural Network Signal Processing*, chapter 7. CRCPress, 2002.

[17] Scott C. Douglas and Andrzej Cichocki. Adaptive step size techniques for decorrelation and blind source separation. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1191–1195, Pacific Grove, CA, 1998.

[18] Scott C. Douglas and Simon Haykin. Relationships between blind deconvolution and blind source separation. In Simon Haykin, editor, *Unsupervised Adaptive Filtering Volume II: Blind Deconvolution*, volume 2, chapter 3, pages 113–145. John Wiley & Sons, 2000.

[19] Scott C. Douglas and Xiaoan Sun. Blind separation of acoustical mixtures without time-domain deconvolution or decorrelation. In *Neural Networks for Signal Processing XI(NNSP 2001*, pages 323–332, Falmouth, MA, 2001.

[20] Earl R. Ferrara. Fast implementation of lms filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-28(4):474–475, August 1980.

[21] Cédric Févotte, Rémi Gribonval, and Emmanuel Vincent. BSS_EVAL toolbox user guide. Technical Report 1706, IRISA, Rennes, France, 2005. http://www.irisa.fr/metiss/bss_eval/.

[22] Dominique Godard. Self-recovering equalization and carrier tracking in two-dimensional data communication systems. *IEEE Transactions on Communications*, 28(11):1867–1875, November 1980.

[23] Simon Haykin, editor. *Unsupervised Adaptive Filtering Volume I: Blind Source Separation*, volume 1 of *Adaptive and Learning Systems for Signal Processing, Communications, and Control*. John Wiley & Sons, 2000.

[24] Simon Haykin, editor. *Unsupervised Adaptive Filtering Volume II: Blind Deconvolution*, volume 2 of *Adaptive and Learning Systems for Signal Processing, Communications, and Control*. John Wiley & Sons, 2000.

[25] Simon Haykin. *Adaptive Filter Theory*. Prentice Hall, Upper Saddle River, New Jersey 07458, 4th edition, 2002.

[26] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. John Wiley & Sons, 1st. edition, 2001.

[27] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[28] Kenneth E. Hild II. *Blind Separation Of Convolutive Mixtures Using Renyi's Divergence*. PhD thesis, University of Florida, 2003.

[29] Russell H. Lambert. *Multichannel Blind Deconvolution:FIR Matrix Algebra And Separation Of Multipath Mixtures*. PhD thesis, University of Southern California, May 1996.

[30] Russell H. Lambert and Anthony J. Bell. Blind separation of multiple speakers in a multipath environment. In *ICASSP '97: Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, volume 1, page 423, Washington, DC, USA, 1997. IEEE Computer Society.

[31] Russell H. Lambert and Chrysostomos L. Nikias. Blind deconvolution of multipath mixtures. In Simon Haykin, editor, *Unsupervised Adaptive Filtering Volume I: Blind Source Separation*, volume 1, chapter 8, pages 321–375. John Wiley & Sons, 2000.

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

[33] Te-Won Lee, Anthony J. Bell, and Reinhold Orglmeister. Blind source separation of real-world signals. In *Proc. ICNN*, pages 2129–2135, Houston, 1997.

[34] Lucas C. Parra and Clay Spence. Convolutive blind separation of non-stationary sources. *IEEE Transactions on Speech and Audio Processing*, pages 320–7, 2000.

[35] Andrew J. Patterson and Tanja Karp. Application of blind source separation to speech signals. Texas Systems Day, Southern Methodist University, 2003.

[36] Barak A. Pearlmutter and Lucas C. Parra. Maximum likelihood blind source separation: A context-sensitive generalization of ICA. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 613. The MIT Press, 1997.

[37] Dinh-Tuan Pham. Mutual information approach to blind separation of stationary sources. *IEEE Transactions on Information Theory*, 48(7), 2002.

[38] M. D. Plumbley, S. A. Abdallah, J. P. Bello, M. E. Davies, G. Monti, and M. B. Sandler. Automatic music transcription and audio source separation. *Cybernetics and Systems*, 33(6):603–627, 2002.

[39] Holger Quast. Automatic recognition of nonverbal speech: An approach to model the perception of para- and extralinguistic vocal communication with neural networks. Technical report, Machine Perception Lab Tech Report 2002. Institute for Neural Computation, UCSD, 2002.

[40] Daniel Schobben, Kari Torkkola, and Paris Smaragdis. Evaluation of blind signal separation methods, 1999.

[41] J. J. Shynk. Frequency-domain and multirate adaptive filters. *IEEE Signal Processing Magazine*, 9:14–37, January 1992.

[42] Paris Smaragdis. Information theoretic approaches to source separation. Master's thesis, MIT Media Lab, June 1997.

[43] Xiaoan Sun and Scott C. Douglas. A natural gradient convolutive blind source separation algorithm for speech mixtures. In *Proc. EUSPICO*, December 2001.

[44] Kari Torkkola. Blind separation of convolved sources based on information maximization. In *IEEE Workshop on Neural Networks for Signal Processing, Kyoto, Japan*, pages 23–432, September 4-6 1996.

[45] Kari Torkkola. Blind separation of delayed and convolved sources. In Simon Haykin, editor, *Unsupervised Adaptive Filtering Volume I: Blind Source Separation*, volume 1, chapter 8, pages 321–375. John Wiley & Sons, 2000.

[46] André J. W. van der Kouwe, DeLiang Wang, and Guy J. Brown. A comparison of auditory and blind separation techniques for speech segregation. *IEEE Transactions on Signal Processing*, 9(3):189–195, March 2001.

[47] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio separation. *To appear in IEEE Transactions on Speech and Audio Processing*, 2005.

[48] Emmanuelle Vincent. Musical source separation using time-frequency source priors. *To be published in IEEE Transactions on Speech and Audio Processing, special issue on Statistical and Perceptual Audio Processing*, 2005.

[49] Thomas P. von Hoff and Allen G. Lindgren. Adaptive step-size control in blind source separation. *Neurocomputing*, 49(1-4):119–138, 2002.

[50] Alex Westner and Jr. V. Michael Bove. Blind separation of real world audio signals using overdetermined mixtures. In *Proceedings of the First International Workshop on Independent Component Analysis and Blind Signal Separation (ICA '99)*, Aussois, France, January 1999.

[51] E. Zwicker and H. Fastl. *Psychoacoustics: Facts and Models*. Springer Verlag, Berlin, Germany, 1990.