Real Time Blind Source Separation in Reverberant Environments

by

Timothy Sherry

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Master of Engineering
in electronics and signal processing.

Victoria University of Wellington
2014

Abstract

An online convolutive blind source separation solution has been developed for use in reverberant environments with stationary sources. Results are presented for simulation and real world data. The system achieves a separation SINR of 16.8 dB when operating on a two source mixture, with a total acoustic delay was 270 ms. This is on par with, and in many respects outperforms various published algorithms [1],[2]. A number of instantaneous blind source separation algorithms have been developed, including a block wise and recursive ICA algorithm, and a clustering based algorithm, able to obtain up to 110 dB SIR performance. The system has been realised in both Matlab and C, and is modular, allowing for easy update of the ICA algorithm that is the core of the unmixing process.

Acknowledgments

I would like to express my deep gratitude to Professor Bastiaan Kleijn as my research supervisor for his guidance, encouragement and timely critiques of this research work, and for keeping the project on schedule. I would also like to thank Dr. Changxue Ma and GN Resound for their contributions and guidance in producing this research. I would like to thank Dr. Paul Teal for the use of the Electroacoustics laboratory and assistance. And to also thank Dr. Marcus Frean as my secondary supervisor for his assistance.

I would like to thank Arian Miralavi, Wenyu Jin, Fatih Ustok and the rest of the CASP group of Victoria University of Wellington for their assistance and friendship over my time at the University. I also thank Annika Greve and Luke Frogley for their proof reading contributions.

Finally, I wish to thank my family, friends and flatmates for their support and encouragement throughout my study.

Contents

Chapter 1

Introduction

For most people, the ability to converse in a crowded room, whilst challenging, is a skill we use unconsciously. However, for a significant portion of the population hearing loss significantly impairs this skill. This skill is referred to as the cocktail party effect, the auditory ability to focus on a particular source whilst suppressing a wide range of extraneous stimuli [3] (refer to 2.1). For the hearing impaired, the loss of perception across the acoustic spectrum severely diminishes this ability [4]. Modern hearing aids take steps towards alleviating this issue. A beamforming technique is applied to focus on sounds originating in front of the user [5]. By facing the person they wish to converse with there, is an improvement in volume of the speaker. However, due to reduced degrees of freedom when restricted to the two microphones, there is still significant room for improvement.

Blind Source Separation (BSS) offers a possible solution to the cocktail party problem in a reverberant environment. Beamforming restricts the mixing system to a steering vector, and uses this to allow efficient selection of the correct system to use in unmixing the sources [6]. However, in a reverberant environment, the effective direction of arrival is frequency dependent, and the mixing system may not be fully described by a simple direction of arrival. BSS relaxes the restrictions on viable mixing systems [1].

[1]The details of the relaxation of the systems is detailed in the literature review.

Then by using the statistical properties of the target sources, a learning algorithm is able to optimise the coefficents to recover the corrupted source. This allows identification of processes capable of recovering sources which have been mixed by a wide variety of convolutive processes, rather than only process defined by the free-field description as is the case for a classical beamformer [7].

This relaxation of restrictions comes at a significant cost, the search space of unmixing systems has been widened drastically, especially when the number of microphones increases [8] [2]. As a result performing the processing required to estimate these coefficents within a reasonable time is a significant challenge. A variety of BSS algorithms have been proposed [9, 10, 1, 11], which vary significantly in efficacy of separation, and computational complexity.

The goal of this thesis is the development of a BSS system capable of both estimating an unmixing system for use on a real world reverberant process within a reasonable amount of time, and applying it with minimal delay between recording and delivery to the user. Ideally the delay would be so low as to be perceived as instantaneous. Modularity of the unmixing system, so that differing algorithms may be comparatively evaluated for execution time and separation efficacy, is also a prime consideration. Finally evaluation of the practical application of contemporary state of the art BSS approaches has been performed and conclusions are drawn about the challenges solved and still faced by this approach.

GN resound, a world leading audiological instrumentation company have worked closely with us in the development of this system. Their aim is to employ a BSS system based on this work, with the first system targeted at a classroom environment.

[2] Consider that a far field beamformer has two degrees of freedom, an azimuth and elevation angle. A near field beamformer has 3 (x,y,z). A convolutive blind source system has N^2T , where N is the number of microphones and T the maximum considered delay

1.0.1 Technical Requirements

The thesis aims to improve on current BSS algorithms, and developing a system to run in real time. This system needs to fulfill a number of technical requirements. It must have an audio processing latency of less than 100 ms. It was also decided that the estimation of the unmixing operator should take less than 5 seconds to solve. The system should be scaleable in the number of inputs, and robust against loss of inputs. The system should be able to handle varying numbers of users within the room. The system needs to obtain these results on modern hardware whilst maintaining real time operation.

Chapter 2

Literature Review

2.1 Cocktail Party Phenomenon

The cocktail party phenomenon describes and characterizes the ability of humans to focus on a particular acoustic source in a highly complex noise environment. Most humans (and animals) possess some skill in performing this task. However, hearing loss and disorders adversly affecting processing and filtering with in the brain such as autism spectrum disorders, can cause severely diminished ability in performing this task.

We envision a cocktail party, where a number of independent conversations are in progress within an enclosed space. Each listener within this room wishes to focus on a particular conversation, and experiences the other conversations as interference. In addition there are an unknown number of noise sources, from shuffling feet to clinking glasses, many of which are nonlinear. The listener wishes to filter out the relevant conversation while suppressing all interference and noise sources. Most people with adept hearing are able to perform this filtering operation subconsciously.

When looked at from a computational hearing perspective the problem is complex, especially when the listener obtains just two observations of the acoustic scene. It is supposed that some form of binaural directivity

is used [4], and that a time-frequency masking is applied after [12]. However, the field is still an active area of research [13].

Approaches to solving the cocktail party problem can be divided into two classes. Beamforming to selectively target sources within a space can be used to separate speakers, where an acoustic search through the target area is used to identify speakers within the room. Conventional beamformer designs do not consider reverberation, instead assuming the free space propagation of sound from sources within a space. BSS methods instead focus on separation of sources statistically, eschewing prior knowledge of the physical parameters of the environment. As a result they cover a wider range of mixing systems, and in theory are capable of improved performance in reverberant environments. However, they rely on statistical assumptions about the signals they are attempting to separate, and the relaxation on the mixing parameters results in an expanded search space for finding the correct parameters. As we wish to solve the problem for the reverberant case the project is focused on the latter approach, and an overview of current works and techniques in BSS is given below.

2.2 The Mixing Process

To develop a source separation system of any form, an understanding of the mixing process that is corrupting the data we are trying to recover is critical. In the following discussion lower case symbols will denote vectors and upper case will denote matrices and t denotes the time index. In the simplest case the mixing process is instantaneous. We represent this as:

$$x(t) = As(t) \tag{2.1}$$

where $x(t)$ is a vector of observed data, A a time invariant matrix, and $s(t)$ vector of observed source signals. In source separation we estimate part or all of $s(t)$. To do so some constraints on A, or $s(t)$, or both, must be applied, so that something meaningful can be recovered. Choice of these

constraints will be covered in 2.4 and following sections. The matrix A can be considered to describe a set of channels, one for each microphone-speaker pairing.

A common extension of 2.1 is to include an additive noise source, denoted $n(t)$. The noise source is generally a zero mean Gaussian random process, denoted as $\mathcal{N}(0, \Sigma)$:

$$x(t) = As(t) + n(t) \tag{2.2}$$

Time varying linear processes may also be described in a similar manner, by allowing A to evolve over time. This is denoted by giving the A matrix a time index:

$$x(t) = A(t)s(t) + n(t) \tag{2.3}$$

A non-linear process may be described as some vector valued function of the sources $s(t)$:

$$x(t) = F\{s(t)\} + n(t) \tag{2.4}$$

A time variant process can be represented by allowing the function to evolve over time, denoted as F_t.

$$x(t) = F_t\{s(t)\} + n(t) \tag{2.5}$$

Linear processes describe a wide variety of mixing systems, for example functional magnetic resonance imaging applications will often model the mixing process of the head as a time invariant linear process with additive gaussian noise [14]. However, in other cases it is necessary to describe mixing systems with some form of delay. The most basic case is a noiseless delay and sum mixing process:

$$x_m(t) = \sum_{i=1}^{N} A_{mi} s_i(t - \tau_{mi}) \tag{2.6}$$

Here x_m is the observed signal at microphone m[1]. Every source is delayed by a particular time step τ, which is unique to each source to microphone channel. every source also receives a relative gain, described by A_{mi} corresponding to the relevant entry in the matrix A from 2.1. The delay and sum process can be extended to the noisy case in the same manner as the linear noiseless case. The time varying and non-linear cases require more care. This process encompasses the linear case [2], and can be used to describe a significantly wider group of processes. The freespace acoustic propogation function can be considered as a delay and sum mixing process [15].

To extend the representation to a fully convolutive process, the observed signals are described as the sum of each source signal convolved with the filter described by $A_{mi}(t)$. The process now describes a system which introduces some non-trivial filtering to each source-microphone channel.

$$x_m(t) = \sum_{i=1}^{N} \sum_{t=1}^{T} A_{mi}(t) s_i(t) \tag{2.7}$$

The matrix A is now three dimensional, with an added time dimension. While the length of the time dimension is ideally infinite, it is necessary to truncate once the filter gains fall below a cutoff. For audio applications, the maximum T60 time of all the channels is the natural choice.

By considering the convolution theorem of the Fourier transform, it is possible to represent the convolution in the time domain and a set of multiplicative, or instantaneous linear mixtures, in the frequency domain. This is both a convenient and powerful representation of this system, and

[1]As an acoustic system is being developed, the sensing devices will generally by referred to as microphones, and the sources referred to as speakers. However, the systems described can be applied to any relevant mixture where the assumptions made about the sources and mixing processes holds, regardless of the generating sources or the sensors used.

[2]consider the case where $\tau_i m = 0 \forall i, m$. This directly corresponds to 2.1

its use and implications will be discussed in 2.7.

$$\begin{aligned} x_m(\omega) &= \sum_{i=1}^{N} A_m i(\omega) s_i(\omega) \\ &= A(\omega)s(\omega) \end{aligned} \tag{2.8}$$

Again we can extend the frequency domain case to consider additive noise by including a $n_m(\omega)$ term. By the fourier addition theorem this noise source independent of the mixing filter A and can be considered as:

$$\begin{aligned} x_m(\omega) &= \sum_{i=1}^{N} A_{mi}(\omega) s_i(\omega) + \sum_{i=1}^{N} n_m(\omega) \\ &= A(\omega)s(\omega) + n(\omega) \end{aligned} \tag{2.9}$$

A time varying convolutive process can be described by allowing the filter bank, $A(t, \omega)$, to evolve over time. The following section covers the main physical and algorithmic concepts surrounding a convolutive blind source separation system. We focus on an ICA implementation and present innovative work, which has been undertaken in the field of convolutive ICA.

$$\begin{aligned} x_m(\omega) &= \sum_{i=1}^{N} A_{mi}(t, \omega) s_i(\omega) + n_m(\omega) \\ &= A(t, \omega)s(\omega) + n(\omega) \end{aligned} \tag{2.10}$$

While non-linear convolutive models exist and are used in the development of a number of source separation schemes [16], they have not been used in the course of this work and have therefore been omitted.

2.3 The Image Source Method

The aim of this work is to develop an acoustic source separation system. As such, a model which can accurately describe how acoustic signals propagate within a room is necessary for testing purposes. The image method

[**?**] can be used to generate typical impulse responses between an idealized point source and receiver in a simple rectangular room. By forming virtual replications of the sources by reflecting the source location through the walls of the room, the impulse response of the room becomes a sum of delayed and scaled delta functions. Each delta function is the response between a virtual source and the receiver given by free space multiplied by the wall damping coefficients for all the walls it was reflected off. Figure 2.1 depicts the symmetric expansion of a room used to represent the reverberations.

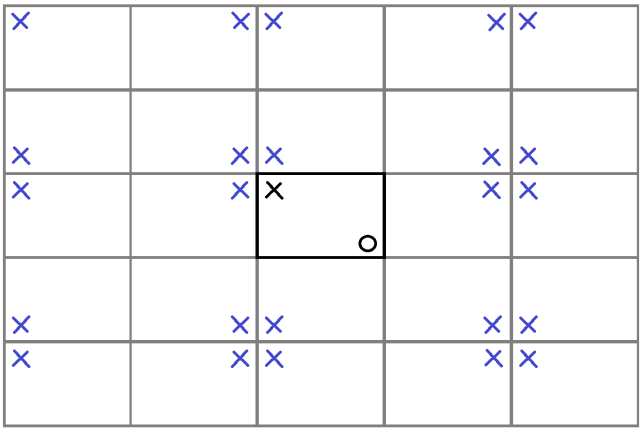

Figure 2.1: The symmetric tessellation of the sources used by the image method to compute the impulse response of a rectangular room. The black circle is the microphone and the black cross is the real source, the blue crosses are the tessellated image locations of the source.

The image method considers each wall as a linear acoustic reflector, and which is a realistic approximation. It is also limited to use on rooms that have the ability to tessellate, and are simple in structure. Practically this limits us to the case of the regular polygonal rooms. Figure 2.2 shows a typical room transfer function generated by the image method for a 5 by 4 metre room.

A single frequency point source generating sound in free space can be

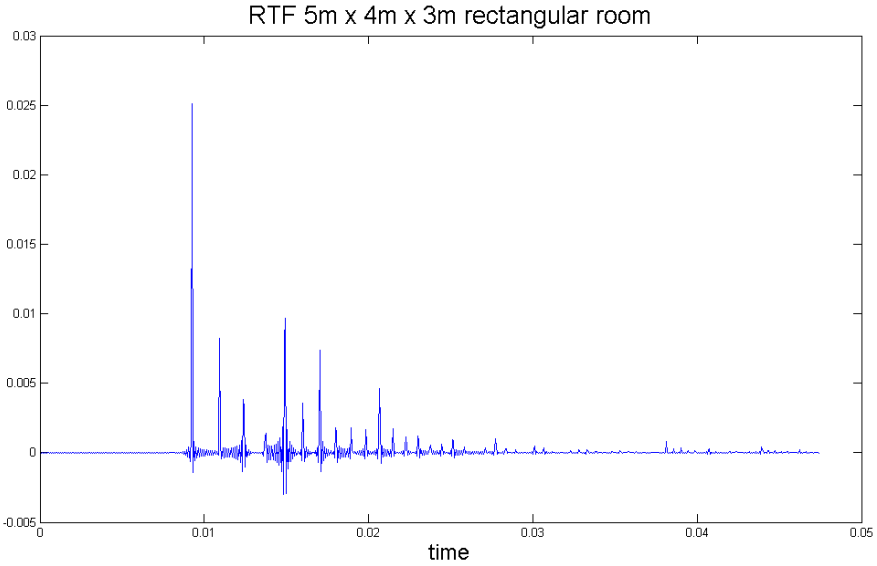

Figure 2.2: Room Transfer Function for a 5 by 4 metre rectangular room

represented by a pressure wave as follows [17]:

$$P(\omega, \text{source}, \text{receiver}) = \frac{\exp\{i\omega(R/c - t)\}}{4\pi R}, \tag{2.11}$$

where P is the pressure wave observed at the receiver produced by the source, ω is $2\pi f$, R is the distance between the source and receiver, c the speed of sound and t time. We can represent the boundary condition of a wall by placing an image symmetrically on the far side of the wall. The image transfer function [17] is given by :

$$P(\omega, \text{image}, \text{receiver}, \epsilon) = \frac{\exp\{i\omega(R/c - t)\}\Pi(\epsilon)}{4\pi R}, \tag{2.12}$$

$$t = \frac{\text{dist}(\text{image}, \text{receiver}}{v_{\text{sound}}}) \tag{2.13}$$

where image is the virtual location of the source on the other side of the walls the image is reflecting from and $\Pi(\epsilon)$ is the product of the reflection coefficients of all the walls the image is reflected by. The overall transfer function for a particular frequency is the summation over all images. To

do this in the frequency domain is trivial, as the Fourier representation of the equation above is a complex delta [18]. This reduces the solution to a sum of complex Fourier coefficients as follows [17]

$$\mathcal{F}\{P(\omega, \text{image}, \text{receiver}, \epsilon)\} = \delta(\frac{\omega}{2\pi} \pm \omega) \exp\{2\pi f - R/c\}\Pi(\epsilon), \tag{2.14}$$

$$\mathcal{F}\{\text{room}, \omega\} = \sum \mathcal{F}\{P(\omega, \text{image}, \text{receiver}, \Pi(\epsilon))\}. \tag{2.15}$$

If we restrict our transfer function to a finite length we can find its Fourier domain representation by considering all the frequencies corresponding to the Fourier transform of that length. This gives an approximation of the Fourier transform of the room transfer function.

2.4 The Delay and Sum Beamformer

The delay and sum beamformer is the simplest beamforming architecture. The aim is to recover the sources mixed by a delay and sum mixing process:

$$x_m(t) = \sum_{i=1}^{N} A_{mi} s_i(t - \tau_{mi}) \tag{2.16}$$

The principle behind the delay and sum beamformer is to delay each microphone in the array by τ_{mi} where m is the microphone being delayed and i is the source we wish to recover. The target source will constructively interfere, while the other j sources will now be delayed by $\tau_{mj} - \tau_{mi}$ for each microphone, resulting in a frequency selective attenuation.

When the microphone array orientation is known, by assuming that the source is distant from the array and therefore the incident sound waves are planar, it is possible to calculate the delay corresponding to a particular direction. When the chosen direction is the same as a source in the sound

field, the resulting signal should approximate the original source. This is the delay and sum beamformer.

2.5 The MVDR Beamformer

The Minimum Variance Distortionless Response (MVDR) beamformer addresses many of the shortcomings of the delay and sum beamformer. For an instantaneous mixing system, with a known spatial correlation, the sensor weights are computed to minimise the total output power of the array, while fixing the gain in a desired direction to unity. we call the vector describing the direction of interest a steering vector, v. The vector of weights corresponding to the MVDR result are:

$$w = \frac{S^{-1}v}{v^H S(-1)v} \tag{2.17}$$

However, the mixing process for acoustic recordings in the real world, as discussed in 2.3, is convolutive in nature. The spatial correlation matrix does not accurately capture the process in the time domain. The system must be linearised to correctly recover the signal as received along the target direction.

As covered in 2.7 a convolutive mixing process can be linearised by the Fourier transform to give:

$$x_m(\omega) = A(\omega)s(\omega) \tag{2.18}$$

By operating in the frequency domain the MVDR weight function can be solved for each frequency, provided the spatial correlation matrix at each frequency is known.

For environments that are not well approximated by the free space model the spatial correlation matrices must be derived via a different method. However, due to sensitivity to modeling errors, and the complexity of models such as the image method required to describe the richer environments, performance is difficult to guarantee. In addition, the system must

be modeled in advance, either by an expert or via a learning algorithm. Finally beamformers suffer from sensitivity to modeling errors. Spatial offsets on the order of a centimeter significantly degrade performance even in a free field environment, and as a result installations require intensive calibration. For these reasons, it was decided that a beamformer approach would not be suited to reverberant source separation.

2.6 Instantaneous Blind Source Separation

Rather than computing weights based on a pre-determined model of the process, weights can be computed based on expectations of the properties of the sources, and weights that result in the maximisation of those properties used to recover the original sources. These methods are referred to as blind source separation (BSS) , as prior knowledge of the spatial arrangement of the generating process is not used, removing reliance upon accurate placement and tracking of the sensor elements. In general the problem is under-determined. The signal properties used vary depending on the approach used. For instantaneous mixing systems, the methods can be divided into three groups: statistical; cyclostationarity and spatial. The statistical methods were chosen as they are the most mature, and tend to be less sensitive to the signal properties, allowing them to be used in a wide range of situations.

2.6.1 Statistical Methods - Independent Component Analysis

Approaches include Bayesian estimation systems, hidden Markov modeling [19], ICA [20], cyclostationarity [21] and time frequency masking techniques [22]. The Independent Component Analysis (ICA) group of approaches to BSS has been chosen due to the popularity of the method within literature and its relatively low computational effort, suggesting

that it can be done within the timescales required for online operation.

Independent Component Analysis (ICA) uses information theoretic properties to separate the source signals mixed by a linear process[23]. It operates on the assumption that the sources we are interested in are non-Gaussian in nature. It can be shown that for a given variance, the probability distribution which has the greatest differential entropy is the Gaussian distribution. By the central limit theorem any linear mixture of non-Gaussian distributions results in a data set with a more Gaussian distribution than the constituent distributions[24].

If we know that the mixed sources in our data are non-Gaussian in nature, finding the least Gaussian representation of a multivariate data set will correspond with the most statistically independent description of the inputs. Speech can be considered non-Gaussian, in both the time and time-frequency domains, as it is sparse in time. The distribution is closer to Laplacian than Gaussian. To find this least Gaussian description the differential entropy of the distribution is compared to the Gaussian distribution of the same variance. Negentropy is another name for this measure. When performing ICA, the negentropy of a distribution must be estimated. The first step to ICA after aggregation of the signals is the removal of redundant dimensions within the data, and whitening of the input signals. Removing all covariances and setting the variance of all the unique signals to unity reduces the negentropy calculation to the comparison between the N dimensional source distributions vs. a unit N dimensional Gaussian. Principal component analysis will obtain the necessary transform efficiently [25]. This reduces the search for the least Gaussian description to rotations in an N dimensional space.

An estimator of the negentropy is applied to rotations of the source distribution. One estimation of the negentropy is based on the principle that the Gaussian distribution is fully defined by its mean and variance, and the higher order statistics are either zero or redundant. Therefore, the higher order statistics such as skewness and kurtosis can be used to esti-

mate the deviation from a Gaussian distribution [8]. A good contrast function will be monotonic as a function of the true entropy for mixtures of the intended source signals, as this reduces the number of local minima in the final search space. Finally a search algorithm, for example a grid search, gradient descent, Markov chain Monte Carlo or expectation maximisation is used to find the representation that maximises the negentropy[26].

The recovered signals have been separated, but their ordering is lost, and will be scaled to have uniform energy. For complex signals the scaling process will introduce a random phase shift as well. For instantaneous mixtures this is usually acceptable however, this poses serious problems when the system is extended to operate on complex mixtures.

2.7 Convolutive Blind Source Separation

To extend the ICA algorithm used in the simple linear case to use in a convolutive mixing environment, the mixing system needs to be linearized. By the Convolution Theorem a convolutive process in the time domain can be described by a multiplicative process in the frequency domain. This implies that finding a solution in the Fourier domain will be a linear problem. Alternatively, the field of research known as auditory scene analysis, provides a range of approaches to convolutive source separation. Auditory scene analysis draws on nature to provide insight into optimum methods of tackling the problem. As the most efficient system for performing BSS is the healthy audiological sensory network of humans, applying similar approaches is likely to offer impressive performance. The three methods most relevant to BSS are: percepual masking [27]; pitch based discrimination [28]; and subband filtering [29]. While this field of research are delivering promising results, the methods are often restricted in the number of sources to be discerned, or in the number of observations they can handle. A more generalist approach, where the system did not rely on a particular arrangement of the microphones was desired.

2.8 Permutation and Scaling Ambiguity

The use of the Fourier linearisation of convolutive mixtures enables the application of linear source separation techniques for convolutive mixtures. However, most linear separation techniques discussed only separate sources up to an unknown scaling and permutation. This must be corrected to enable the recovery of the original speech. Solving this problem is not trivial and is an active field of research [30] [11]. To solve the permutation ambiguity, two complimentary approaches are found in literature: inter-frequency correlation and direction of arrival estimation [31].

2.8.1 Inter-frequency Correlation

The inter-frequency correlation approach assumes that speech occurs across a range of frequencies and has long periods of relative silence. This results in a correlation between frequency bins of a Short Time Fourier Transform (STFT), as coefficients tend to be significant in unison when a speaker is talking and insignificant when they are not. This is the core of the algorithm proposed by NTT in a paper authored by Hiroshi Sawada [9]. This correlation is shown in Figure 2.3 where the periods of activity occur over many frequencies at approximately the same time. Also visible in Figure 2.3 is the independence of these periods of activity and silence between two separate speech signals.

This correlation provides a metric for the likelihood two arbitrary frequency bin signals were generated by the same speech source. A high correlation coefficient corresponds to a high likelihood of a common source, low corresponds to different sources. However, this metric is sensitive to the relative power of each frequency at a given time, and can give incorrect matches. A more robust method involves calculating the power ratio of the recovered sources for each frequency and time slot. The power ratio as described by Hiroshi Sawada[9] is as follows:

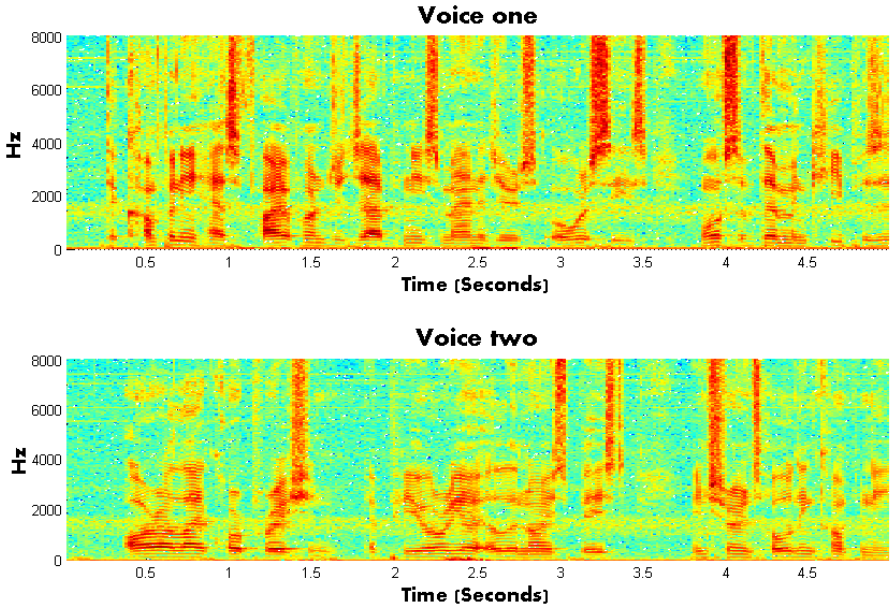

Figure 2.3: Two spectrograms of human speech, red corresponds to greater magnitude coefficients for a frequency-time bin, green to near zero coefficients. Note that for each voice there are periods of activity which occur in unison across many frequencies rather than being randomly dispersed over time. These periods are independent between the voices.

$$\text{pow}\{f,t,s\} = \frac{STFT_s(f,t)}{\sum_{s=1}^{N_{sources}} STFT_s(f,t)} \tag{2.19}$$

Here the strength of a frequency-time point is described relative to the other sources. This has been shown to be more effective for higher numbers of sources [31].

2.8.2 Direction of Arrival

The direction of arrival method applies a beam-forming approach to the permutation issue. It does so by approximating the direction of arrival from source to sensor for each frequency based upon the mixing vector

used to reconstruct the source. It has been exploited by [31] [32] [33].

For each frequency and source in our separation algorithm a complex vector describing how to recover that frequency and source from the mixed observations is given. The magnitude of each coefficient approximates the distance of the source from the sensor. If the sensor location is known then this can be used to estimate the source location through trilateration. Additionally, if the sensor array is compactly arranged, then it could be treated as a far field beam former, and used to generate an estimated angle of arrival of a given source and frequency.

The use of a compact sensor arrangement could be detrimental as it favors sources that are near the array and reduces independence in the recordings. This means the far field beam forming approximation is not relevant. However, trilateration will operate in a near field beamformer, and may provide another means of estimating the correct permutation, assuming the errors introduced by reverberation of the sources is modest. In the case of heavy reverberation this approach to permutation will fail, as the spatial assumptions about direction of arrival are invalidated. As a result we are not focusing so heavily on this approach.

2.8.3 Scaling and Delay Correction

As described in 2.6 there is a random scaling and delay applied to each frequency bin recovered by the ICA algorithm that is fundamental to this process. If not corrected the recovered audio sounds unnatural, and is more difficult to distinguish from the residual noise and interference.

To correct for this problem we aim to recover the signal as it would sound at a particular point within the room. To do so we need to re-scale and correctly phase shift each frequency bin so it has the same relative scale and phase as its generating source has at that particular location. The simplest points to attempt this recovery from are the microphones themselves. If we take the correlation between the source and the micro-

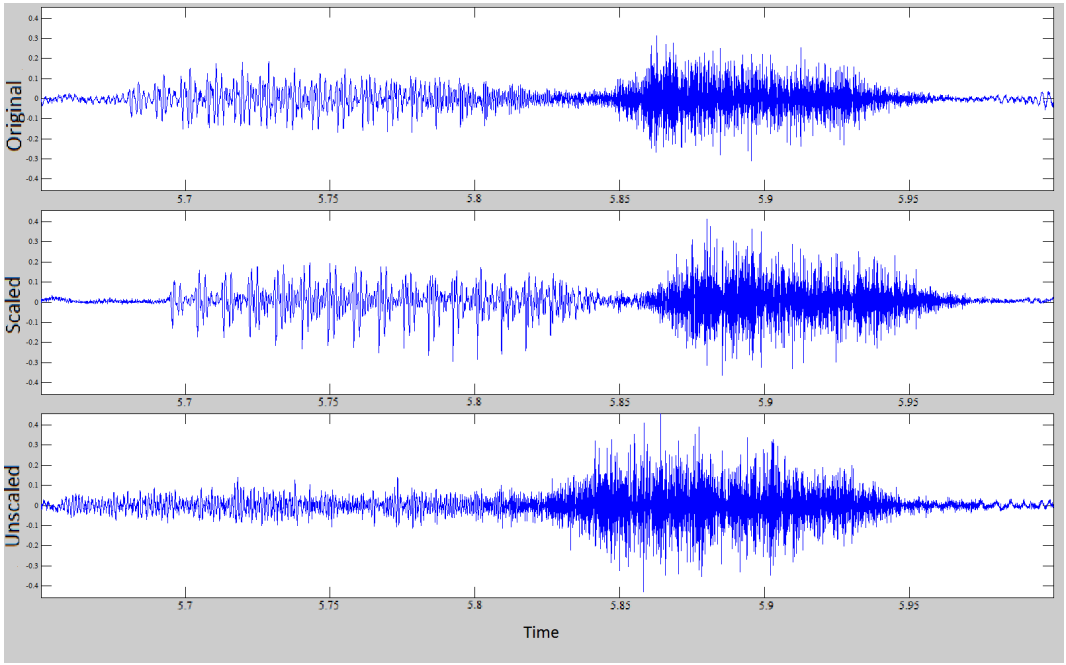

Figure 2.4: 35 ms clip of source audio, a recovered signal with scaling correction applied and a recovered signal without scaling correction. This voice was captured with a PDA, and some noise is present, especially in the 1000-1500 Hz band.

Figure 2.4 shows a 35 ms clip of the original audio, recovered audio with scaling correction and recovered audio without scaling correction. The recovered signal with scaling correction applied exhibits a phase shift from the original, but is relatively unchanged. The uncorrected signal is

neither in phase nor accurately represents the signal; however, its envelope approximates the original signal. In practice the uncorrected signal exhibits a randomised frequency selective phase shift, and while intelligible is not a natural reproduction of a persons voice.

Chapter 3

Method

A state of the art BSS system based on fastICA has been produced. The system runs on a laptop, indefinitely, separating speech signals in real time. The signals need to be generated by static sources, and be relatively constant in output, i.e. any pause in speech would need to be less than the width of one ICA block. These requirements are consistent with the testing environments presented by [31, 1] and others. Additionally it draws parallels with the setup of the CHiME challenge, as here the speaker was defined to be within a relatively small region of space [30]. In addition a modular system was desired, particularly around the source separation, permutation correction, and recovery areas, to allow comparison between contemporary systems.

The design of the system is discussed in the following chapter. The design of the various modules of the system are discussed, starting with the derivation of the complex ICA algorithm for instantaneous unmixing, and the recursive ICA algorithm developed to improve on the convergence time of the system. Then the permutation and scaling solutions allowing operation on convolutive mixtures are described. Finally the realisation of these algorithms in both Matlab and C are discussed.

3.1 Complex ICA

ICA is a successful algorithm in the field of BSS. The SINR results for speech signals are on par with the systems described in section 2.6, more than two microphones are used to collect data. In addition it is computationally efficient when compared to algorithms with comparable accuracy, such as those used in auditory scene analysis and subband filtering. In particular the fastICA algorithm, developed by Aapo Hyvarinen [23] converges particularly fast. As the BSS algorithm will operate in the frequency domain, it will operate on complex inputs. An algorithm extending fastICA for use on complex inputs has been developed, and is derived here.

As described in section 2.6.1, maximising the non-gaussianity of a mixed set of non-gaussian signals will recover the original signals. ICA accomplishes this for linear mixtures by maximizing the negentropy. FastICA uses a Newton method to elicit convergence in a minimum number of iterations for real datasets. In the convolutive case, the negentropy of a number of complex random variables must be estimated, and therefore, the mixing coefficients are complex in nature. We need to estimate the Jacobian, and the Hessian, of the negentropy within a complex domain for our Newton algorithm to work.

Before deriving the algorithm, it is worth discussing the difference between the probability density function (pdf) of the sources, and the data we observe from those sources. As we assume the source is a stochastic process, the values it takes can be modeled by a pdf. However, we can only make observations of the data produced by our source, and must infer the generating pdf from these observations. It is the underlying pdf we are trying to make as non-Gaussian as possible, and we do so by maximising attributes of the pdf, in particular the negentropy, for each recovered signal in the source data.

The negentropy function is always real-valued, and non-negative, even in the complex domain. According to the Cauchy-Riemann criteria, the

complex gradient of such a function can be found by considering it as a function of two sets of variables, one of the real components and one of the real values of the imaginary components. By taking the partial derivatives with respect to each component we can find the complex gradient of our function. This can be extended to the Jacobian of a complex multivariate function with a real output. Let f be a function of the complex domain, g and h be the partial functions of f corresponding to operations upon the real and imaginary components respectively:

$$f(z) = g(z_r) + i\ h(z_i), \tag{3.1}$$

$$f'(z) = g'_r + i\ h'_i, \tag{3.2}$$

$$f'(z) = \frac{dg}{dz_r} g(z_r)dr + i\frac{dh}{dz_i} h(z_i)dz_i. \tag{3.3}$$

We cannot find the negentropy directly as we only have a finite sample of the random variables' distribution. To estimate the negentropy from first principles would involve finding the approximate pdf of the Random Variable (RV), and taking the differential entropy with respect to a Gaussian. This process would be computationally intensive. However, a estimate of the negentropy is unnecessary, our only requirement is that locations of local optima are preserved between the estimate and the true negentropy. Cumulative functions that measure the higher order moments of the RV, for example the kurtosis function, are used, as they describe the difference of a given RV to that of the Gaussian distribution. Correct selection of a contrast function given the distribution of the data it will operate on is critical. Some examples of contrast functions are:

$$g(x) = E\{\tanh(X)\}, \tag{3.4}$$

$$g(x) = E\{(X)^3\}, \tag{3.5}$$

$$g(x) = E\{\frac{1}{\epsilon + X^2}\}. \tag{3.6}$$

where X is an observed random variable, and ϵ is a geometric factor. The contrast functions described above are dependent on the variance

of the RV they are operating on. The variance of each mixture needs to be identical to obtain usable results. Thus we apply the constraint, $E\{|w^H X|^2\} = 1$, where w is the unmixing vector recovering a particular source generating the observed random variables X. $w^H x$ is the current estimate of one source signal within the observed data. Currently (3.6) is used as the contrast function.

We wish to find an optimal unmixing vector, $w_{optimal} = \text{argmax}\{f(w)\}$ where $f(w) = g(w^H x)$ subject to the equality constraint above. The Lagrangian corresponding to this problem is:

$$L(w, \mu) = f(w) + \mu\,(|w^H x|^2 - 1). \tag{3.7}$$

As the function and constraint are defined by a random variable we rewrite the above equation in terms of expectations:

$$L(w, \mu) = E\{g(|w^H X|^2)\} - \mu E\{|w^H X|^2\}. \tag{3.8}$$

Values of w for which the Lagrangian derivative is zero are the constrained optima.

$$\begin{aligned} \bigtriangledown L(w, \mu) &= \bigtriangledown E\{g(|w^H x|^2))\} + \bigtriangledown \mu E\{|w^H x|^2\}, \\ 0 &= \bigtriangledown E\{g(|w^H x|^2))\} + \mu \bigtriangledown (E\{|w^H x|^2\}). \end{aligned} \tag{3.9}$$

The Newton method is used to solve (3.9). The Newton method finds the roots of the derivative of the Lagrangian, and as a result the constrained optima. An approximation of the Jacobian of $\bigtriangledown E\{g(|w^H x|^2))\}$ is:

$$\begin{aligned} \bigtriangledown^2 E\{G|w^H x|^2)\} &= 2E\{(\bigtriangledown^2 |w^H x|^2) g(|w^H x|^2) + 2(\bigtriangledown |w^H x|^2)(\bigtriangledown |w^H x|^2)^T g'(|w^H x|^2)\}, \\ &\approx 2E\{g(|w^H x|^2) + (|w^H x|^2) g'(|w^H x|^2)\} I. \end{aligned} \tag{3.10}$$

This approximation is made by separating out the expectations. The Jacobian of $\mu \bigtriangledown (E\{|w^H x|^2\})$ is $\mu \bigtriangledown^2 E\{|w^H x|^2)\} = 2\mu\, I$, making the total Jacobian of (3.9):

$$J \approx 2E\{g(|w^H x|^2) + (|w^H x|^2)g'(|w^H x|^2) - \mu\}I. \tag{3.11}$$

The Newton solution iterates w_n by taking the Lagrangian and dividing by the Jacobian above:

$$w_{n+1} = w_n - \frac{L(w_n)}{J(w_n)} \tag{3.12}$$

$$= w_n - \frac{\bigtriangledown E\{g(|w_n^H x|^2)\} + \mu \bigtriangledown (E\{|w_n^H x|^2\})}{2E\{g(|w_n^H x|^2) + (|w_n^H x|^2)g'(|w_n^H x|^2) - \mu\}} \tag{3.13}$$

$$= w_n - \frac{E\{x(|w_n^H x|) * g(|w_n^H x|^2\} - \mu w_n}{E\{g(|w_n^H x|^2) + (|w_n^H x|^2)g'(|w_n^H x|^2) - \mu\}}. \tag{3.14}$$

The step between 3.13 and 3.14 is made by calculating the Jacobian of 3.7. We can remove the μ term by multiplying (3.14) by $\mu - E\{g(|w_n^H x|^2) + |w_n^H x|^2 g'(|w_n^H x|^2)\}$. This gives us the simplified update:

$$w_{n+1} = E\{x(w_n^H x)g(|w_n^H x|^2)\} - E\{g(|w_n^H x|^2) + |w_n^H x|^2 g'(|w_n^H x|^2)\}w \tag{3.15}$$

The Newton algorithm is iterated until $w_{n+1} = w_n \pm \epsilon$ where we assume we have reached convergence. The process is repeated for each signal present within the data.

3.2 Recursive ICA

The aim of BSS is to recover stochastic source signals S present within an observed mixture X of the sources. For a linear mixing system one solution is to find an unmixing matrix W^H such that $Y = WX$ and $S = PY$, that is W recovers the source signals S up to some permutation matrix P. Resolving the permutation ambiguity is not considered as there is no natural ordering to the signals we will observe [31]. However, we do consider an on-line method for finding W.

The fastICA algorithm is well suited to finding an unmixing filter for instantaneous mixtures using the ICA method. However, fastICA is a

batch mode algorithm, a recursive implementation of the fastICA algorithm would be more suited to a real time BSS system. If the system were to refine its unmixing filters at each sampling time, the delay in the time to react to changes in the environment could be significantly reduced.

Before deriving the recursive algorithm it is useful to have an understanding of how it should be constructed. We want to derive an adaptive filtering implementation of the fastICA algorithm. we aim to solve the following system:

$$d(t) = \bar{x}(t)w_{\text{optimal}}. \tag{3.16}$$

We wish to find $d(t)$, the desired output, from $\bar{x}(t)$, a vector of noisy input signals, by filtering with the filter w_{optimal}. note that w is the filter to recover a single source $d(t)$. As we assume the mixing system to be linear, this filter is trivial and has a one sample response length. W refers to the matrix containing $w_1, w_2....w_n$ which recovers n sources. In situations where the distortion of the desired signal is non-stationary, w_{optimal} would itself be varying. In general an the filtering weights w can be updated by

$$w_{n+1} = w_n + \phi h_\epsilon, \tag{3.17}$$

where w_n is an FIR filter estimated by the adaptive filter, h_ϵ is the estimated error in the previous filter and ϕ is the learning rate of the algorithm.

The error function h_ϵ is generally given by:

$$w_\epsilon = E\{\bar{x(t)}e^*(t)\}, \tag{3.18}$$

where $\bar{x(t)}$ is the noisy observations and $e^*(t)$ is the error between the observed output and the desired output, that is $d(t) - \hat{d}(t)$ where $\hat{d}(n) = w_n\bar{x(t)}$ and $d(t)$ is the desired output. For consistency X is used in place of $\bar{x}$ to denote the vector of observations, and x^i denotes a single observation.

An ideal algorithm for finding W_n would use an infinite number of samples to calculate (3.17), as this would provide a theoretically perfect reconstruction of W_n. However, as sampling takes a finite amount of time, as does computation, there are practical limits to the number of

samples considered. In addition, the system observed is highly unlikely to remain static over long periods of time, restricting the number of samples available to reconstruct a particular static unmixing matrix.

Taking a window of the data, rather than the whole data set, generally performs better than a full length averaging function when the signal is nonstationary. Longer window lengths result in more accurate reconstruction, but should not be so long that the system would likely have shifted before completion of separation:

$$\begin{aligned} W_{n+1} &= W_n + \frac{1}{L}\sum_{i=n-L+1}^{k} h(W_n, x_i), \\ &= W_n + \Delta W_n \end{aligned} \tag{3.19}$$

where k is an index which shifts the window along the sampling timeline, L defines the window length and h is that of (3.17). It is intuitive to think of the update in terms of the previous W_n and in terms of its discrete derivative ΔW_n which is the updating term. This is equivalent to finding W_n for a fixed length section of the overall recording starting at index $n-L$.

The algorithm described above has significant redundancy in its computation at each step of index n. Assuming that $W_n \approx W_{n+1}$, that is the system has nearly converged on W_{optimal}, then the only new term in the sum is that of x_n and the previously used term of x_{n-L} is now unused. The update can be simplified to:

$$W_{n+1} = W_n + \Delta W_n + \frac{1}{L}(h(W_n, x_k) - h(W_n, x_{k-L}), \tag{3.20}$$

Applying this method directly is infeasable, as the value for W_n is not constant as has been assumed and any shifts in W_n will result in errors when the $h(W_n, x_{k-L})$ is subtracted instead of $h(W_{n-L}, x_{k-L})$. Use of an exponentially decaying window will cause these errors to die out.[1] This

[1] In addition the exponential window will ensure numerical errors are also accounted for

allows the algorithm to converge:

$$W_{n+1} = W_n + (1-\gamma)\Delta W_n + \gamma h(W_n, x(k+L)), \tag{3.21}$$

where γ is a forgetting factor defining the effective memory of the system. For the case of $\gamma = 1$, this results in a memoryless estimate of W_n, where previous estimates are ignored. A γ of near zero would result in diminished updates of W_n. For the memoryless case fast convergence can be expected at the cost of non-robust steady state performance, for low γ the reverse is true, convergence would be slow with good steady state operation. There are a number of proposed adaptive approaches to forgetting [34], the formulation of (3.21) is one of the most straightforward. A more optimal approach may be found.

For the case where the algorithm starts with zero memory, the addition of an adaptive value for γ would improve convergence times without sacrificing steady state performance. Currently the system uses a heuristically determined time-varying γ of the form:

$$\gamma(n) = \frac{\gamma_0}{n^\lambda} + \epsilon. \tag{3.22}$$

The value for γ starts at $\gamma_0 + \epsilon$. It then decays exponentially, as controlled by λ, down to a minimum value of ϵ. ϵ defines the steady state operation, higher ϵ values should be used for signals where W is highly time-variant.

An assumption was made in the derivation of (3.20) that $W_n \approx W_{n+1}$. This is a fair assumption when the system has almost converged. However, if the system is not near convergence, then the earlier terms within the sum are non-optimal and should all be recalculated to ensure optimum results. A compromise can be made by instead calculating $h(W_n, x_i)$ for the most recent values of x_i. This is at the cost of added complexity proportional to the size of the window, as the costly function $h(W_n, x_i)$ must be computed for all values within the window rather than just for the latest observation. The new scheme involves updates as follows:

$$W_{n+1} = W_n + (1-\gamma)\Delta W_n + \gamma \frac{1}{T} \sum_{\tau=0}^{T} h(W_n, x(k+L-\tau)), \tag{3.23}$$

where T is the window length for which the most current estimate of the unmixing filter W_n is used to compute the update ΔW_n.

3.3 Clustering Based Source Separation

An alternative to ICA is to cluster samples based on the direction of arrival to the microphone array. It is similar to the source number and clustering algorithm described by Loesch [35].

Speech is a sparse signal, where there are periods of activity interspersed by long periods of silence. In a normal conversation, which could be considered half duplex, this sparsity extends to speakers, that is only one speaker is active at a time. This knowledge can be exploited when estimating the unmixing system, indeed this sparsity is the property that ICA is maximising in its search for the source signals.

Loesch proposes that each observation at each frequency can be defined in terms of an angular direction of arrival at the microphone array. In addition each observation can be attributed to one source, due to the speakers only speaking one at a time. Clustering of data around these directions of arrival corresponds to a direction of arrival along which a source lies. This can be used to both identify the number of sources present in a recording, and identify the beamforming angle corresponding to each source.

Extending this concept, one does not have to consider the beamforming case, and can define each observation in each frequency bin in terms of a polar coordinate. By clustering based on the angular arguments the vectors corresponding to the unmixing vectors of the underlying sources can be obtained. Knowledge of the arrangement of the microphone array is not necessary, and separation can still occur under reverberant conditions that would significantly affect the angle of arrival assumed by the beamformer. This flexibility comes at the cost of slower clustering due to a reduction in points as each frequency bin is considered separately,

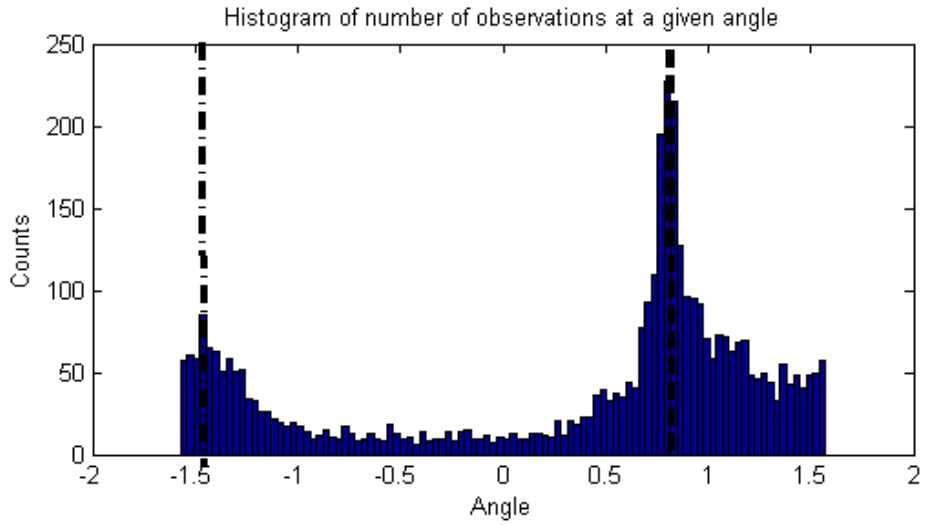

Figure 3.1: A histogram of counts at each angle for the sparse source plotted in 3.2. 5000 observations spread over 100 bins. note the two distinct peaks in the observed data.

and introduction of the permutation problem present in most convolutive techniques. As the angle between any two points can be described by a single angle, even if the data lies in a much higher dimensional space, the clustering algorithm scales by $O(n)$ with increasing microphone array size, compared to $O(n^2)$ seen for ICA, suggesting that for larger arrays the clustering algorithm will be computationally favorable.

The algorithm showed promise, but currently fails to handle crowded mixtures, where speakers are talking over each other constantly. As a result it was decided that the ICA algorithms were more likely to succeed, and development proceeded along that path.

3.4 Short Time Fourier Transform

The most basic approach would involve taking the Discrete Fourier Transform (DFT) of a whole recording of speech; apply the ICA algorithm to the DFT signals and inverse transform the separated signals back into the

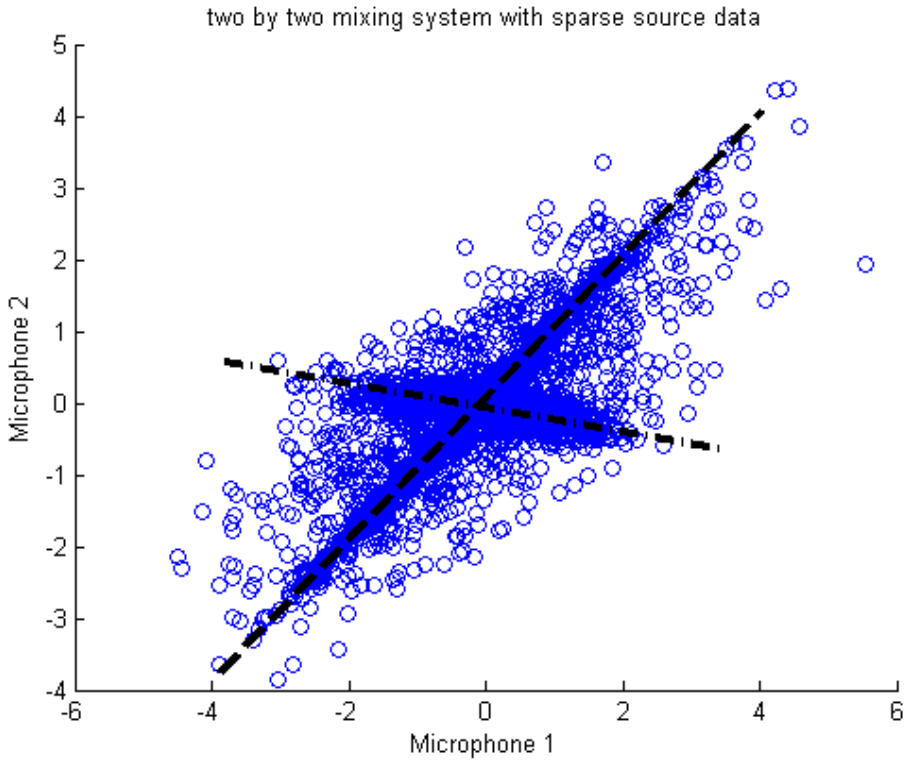

Figure 3.2: A scatter plot of an instantaneous mixture of two sparse sources. we have not considered the angle given by beamforming, only the observed angle described by $\text{atan}(\frac{X_1}{X_2})$.

time domain. However, this approach assumes that the mixing system operates uniformly across all frequencies, in other words a delayed delta function. Outside of anechoic chambers, no environment has such a perfect response, and as such this system will fail to correctly identify the original sources.

To account for more complex environments a lapped, windowed Fourier transform is applied to the data and the ICA algorithm is applied separately to each frequency bin present in the window. An appropriate sample of windows must be considered to accurately capture the statistics. As suggested by Delcroix [1], the window length should be at least twice

the length of the reverberation time.

In our algorithm a Gabor transform with a root-Hann window is used to generate the windowed blocks. A modified Discrete Cosine Transform (DCT) would be preferable over the Gabor transform as it provides critical sampling with smooth windowing. This is the algorithmic sweet spot between complexity, which increases with non-critical sampling, and the minimum process distortion afforded by smooth windowing. However, the DCT makes assumptions about the phase of the signal in the forward transform and in doing so does not allow us to recover the signal with all the frequencies in phase, leaving us with the use of the Gabor approach. The Gabor transforms result is complex, meaning we need to apply a linear separation algorithm capable of operating on complex signals and complex mixing coefficients.

First the data is windowed using the root-Hann window. The use of the root-Hann window on the source data and subsequent windowing on the post processing result, rather than simply windowing with a Hann window the source data and performing an overlap save without windowing, leads to a unitary transform. This has the added advantage of distortion meaning the same in both the time and frequency domains, and reduces processing distortion near window edges. The Gabor transform is taken for the windows, resulting in a time-frequency representation of the data. The fast ICA algorithm is then applied to each frequency individually, and searches for the independent sources that produced a particular frequency gain observed at each successive window time. Assuming that the delay present in the room is less than half the width of each window, the frequency gain at each window should be a linear combination of the frequency components present in each source, and separable via the ICA algorithm. The result is then inverse Gabor transformed, recovering the original sources. This technique reduces computational complexity sufficiently to enable real-time operation.

3.5 Real Time System Implementation

The system was realised in Matlab first to prove the concept, then was ported to C to obtain optimal performance. The design and implementation are described, focusing on the architecture rather than the libraries used to obtain the results.

3.5.1 Multi-threaded Design

The original system was designed to work on a fixed length recording of convolutively mixed speech. It performs an analysis of these recordings, calculates the independent components, the correct permutation, and returns all of the unmixed sources present in the recordings. This process takes around 15 seconds, resulting in a delay before the user can hear any result.

In the practical implementation of this system, the user would prefer to hear the unrecovered recordings rather than hear nothing at all. In addition the user would prefer that the recovered sources were presented instantaneously, even if the presented result is not optimal. To deliver this a multi-threaded solution has been developed, where there is one thread capturing all the recordings and instantaneously returns the desired source to the listener. In addition it collects a sample of the recordings, and once it has a long enough sample it sends it to the second thread. The thread performs the estimation of the unmixing operator in the background, returning a new unmixing solution to the first when it has completed its task. This system has been implemented with the parallel computing toolbox in Matlab, specifically using the single process multiple data (SPMD) structure.

Figure 3.3 depicts the operation of the system. The individual sections are covered below when their operation is not obvious.

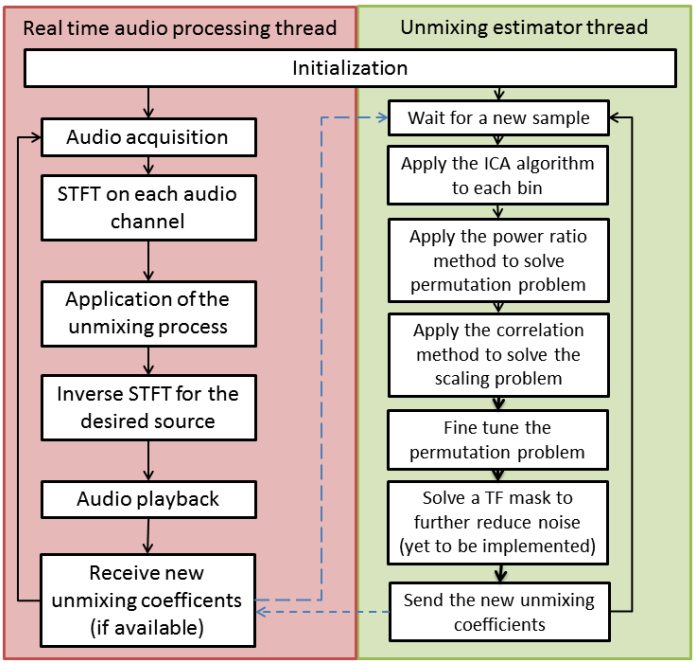

Figure 3.3: The program flow diagram of the online system.

3.5.2 SPMD and the Parallel Computing Toolbox

Recently Matlab has moved towards providing parallel computing options to users of the language through the use of the parallel computing toolbox. The SPMD structure in Matlab allows the creation of independent threads that are able to operate in parallel. The threads have independent memory, but are able to send messages to each other precluding the memory access issues that can make multi-threaded applications significantly more difficult to debug than single threaded applications. To use the SPMD structure the user must first create a pool of workers, or labs as they are referred to in the Matlab documentation. These labs are then assigned to undertake the various function calls that are undertaken within the SMPD structure. The main process treats the SMPD structure effectively as a holistic function call, and waits until the SMPD structure has completed all operations before continuing its tasks.

To implement the real time unmixing system two functions were written intended to operate simultaneously on separate threads. Both functions operate in an indefinite fashion, waiting for a stop signal to be delivered by the user from the GUI.

3.5.3 The DSP Toolbox

The Digital Signal Proceesing (DSP) toolbox provided by Matlab contains functions to gather, process, record and play audio in real time. The AudioRecorder object allows for recording data directly from the speakers in real time. The related AudioPlayer, AudioFileReader and AudioFileWriter objects are also used as their names suggest. A number of important functions are made available, including data type, sampling frequency and buffer size. We currently operate with a sampling frequency of 16 kHz, and use 32 bit floating point for the samples. The buffer length provides a tradeoff between system robustness to interruptions at the cost of delay. Currently the shortest buffer length that can be used before suffering data loss is 0.3 seconds in both the player and recorder. This appears to be a limitation in Matlab, and in theory could be reduced significantly if implemented on a different platform. However, it still demonstrates the ability of the system to separate the sources in near real time using directly observed data.

3.5.4 The Real Time Audio Processing Thread

The real time audio processing thread begins by instantiating the speakers, the microphones or simulation file as directed, and the file writer. All of these processes are implemented using the DSP toolbox. It then initializes the unmixing matrix, which at first is simply the identity matrix resulting in each source being the input from a given microphone.

Following instatiation the system enters the persistent unmixing loop. This loop captures the incoming signal data, takes the Gabor window of

this signal and stores this window for later use by the unmixing thread. The Gabor window is then unmixed using the current unmixing system. This result is reverse Gabor transformed back into the time domain, saved to file and played through the speakers.

Once the thread has collected a sufficiently large set of windows it sends this data set to the unmixing operator estimation thread. It also checks if the unmixing operator estimation thread has returned a new unmixing solution, or if the user is sending a request to halt operation.

The unmixing task takes on average 25 ms per loop iteration for a window size of 1024 samples, equivalent to 64 ms. Therefore, the system should be capable of near instantaneous demixing. However, delays present in the DSP toolbox increase this delay to around 0.8 seconds. Implementation in C++ or similar should be able to reduce this processing delay to a usable time.

3.5.5 The Unmixing Operator Estimation and Permutation Correction Thread

A second thread to find a process to separate the sources and solve the permutation problem runs independently of the first program. Whenever a new set of windows is available it runs the conventional fastICA algorithm on each frequency bin in the data set, returning scaled and permuted sources for each bin.

The system then estimates the most likely permutation of the sources from the ICA algorithm. This is obtained via the inter-frequency correlation method as described in section 2.8.1. Finally the scaling ambiguity is corrected using the method described in 2.8.3. The microphone with highest average gain for a given source is used as the microphone used to correct the scaling factor. This approach is not ideal, as we do not calibrate the microphones. A microphone with an inherent gain will effectively cause the apparent gain in that microphone to be reduced. A measure

to capture the signal to interference ratio of the source in a particular microphone is more suitable.

3.6 Realisation in C

The system has also been written in C, as the overheads in matlab were likely hampering the systems performance. It follows the same architecture as the Matlab code, and uses the FFTW library [**?**] for the complex audio processing routines. Matlabs inbuilt C compiler was considered, but a number of the advanced libraries are not supported by it, specifically the audio IO and threading sections. As a result the only sections that could be automatically converted to C were short enough that writing them directly was not an issue and resulted in clearer, more concise code.

Chapter 4

Results

There are two separate systems to test, the ability for the system to operate in real time, and the efficacy of separation achieved by the algorithm itself. The real time capability can be defined by the lag experienced by the listener between the emmission of sound by the speakers and the delivery of the recovered speech to the user. We will consider a lag of 100ms to be sufficently small, as this corresponds with the approximate crossover between the perception of reverberation and echo [36]. The separation efficacy of the system can be defined by the improvement in SINR between the recovered signal and its generating source, versus the baseline SINR observed in the microphone recordings.

The ability of the source separation algorithms to operate on linear mixtures will be covered first. Then an analysis of the performance on convolutive mixtures on simulated and real world data will be given. Finally the real time performance figures will be published.

4.1 Instantaneous Performance

Two of the three instantaneous algorithms implemented over the course of this work, the block wise ICA algorithm and the recursive ICA algorithm were evaluated on their ability to recover instantaneously mixed speech.

The clustering algorithm did not perform at a comparative level to the ICA algorithms and is omitted.

4.1.1 Simulation Environment

The ICA and recursive ICA algorithms have been tested on instantaneously mixed speech sources, of around two seconds in length, at a sampling rate of 16 kHz. Their performance was measured by taking the signal to interference and noise ratio between the recovered sources and the original signals. The derivation of SINR for a linear system is covered.

SINR

We assume that the source we recover is of the form:

$$r_n = \alpha s_c + \beta n_n + \gamma i_n, \tag{4.1}$$

where r_n is the signal recovered by the n^{th} column in the unmixing matrix W, s_c is the source recovered by r_n scaled such that it is zero mean unit variance. n_n is some additive, zero mean, unit variance noise source and i_n is some additive zero mean, unit variance interference source that is some combination of $s_{!=c}$, the other sources present. Note that the source index differs from the recovered index due to ambiguity in permutation. α is the RMS power of the signal, β the RMS power of the noise and γ the RMS power of the interference in r_n. The SINR would be equal to $\frac{\alpha^2}{\beta^2+\gamma^2}$. However, we do not know α, β, γ, n_n or i_n. The standard method used in the field [37] for computing SINR given only r_n and s_c is derived.

We find that:

$$\begin{aligned}
\text{corr}(r_n, s_c) &= \frac{1}{T} E\{r_n s_c^H\}, \\
&= \frac{1}{T} E(\alpha s_c + \beta n_n + \gamma i_n) s_c^H, \\
&= \frac{1}{T} (\alpha E\{s_c s_c^H\} + \beta E\{n_n s_n^H\} + \gamma E\{i_n s_c^H)\}, \\
&= \frac{1}{T} (\alpha E\{s_c s_c^H\} + 0 + 0), \\
&= \alpha \text{corr}(s_c, s_c).
\end{aligned} \tag{4.2}$$

We can use 4.2 to derive α. However, we cannot directly apply the same to β or γ, as we know neither n_n nor i_n. Instead we can find $\alpha + \beta + \gamma$ by:

$$\begin{aligned}
\text{corr}(r_n, r_n) &= \frac{1}{T} E\{r_n r_n^H\}, \\
&= \frac{1}{T} E\{(\alpha s_c + \beta n_n + \gamma i_n)(\alpha s_c + \beta n_n + \gamma i_n)^H\}, \\
&= \alpha^2 \text{corr}(s_c, s_c) + \beta^2 \text{corr}(n_n, n_n) + \gamma^2 \text{corr}(i_n, i_n).
\end{aligned} \tag{4.3}$$

The correlation of all the signals is equal to one, as they are all unit variance in our example data. Therefore 4.3 is equal to $\alpha^2 + \beta^2 + \gamma^2$ and 4.2 is equal to α. We then find the signal coherence between s_c and r_n.

$$C_{s_c,r_n} = \frac{|\text{corr}(s_c(t), r_n(t))|^2}{\text{corr}(r_n, r_n)\text{corr}(s_c, s_c)} C_{s_c,r_n} = \frac{\alpha^2}{\alpha^2 + \beta^2 + \gamma^2} \tag{4.4}$$

The SINR can be found directly from the coherence as follows

$$\begin{aligned}
1 &= \frac{C_{s_c,r_n}(\alpha^2 + \beta^2 + \gamma^2)}{\alpha^2}, \\
&= \frac{C_{s_c,r_n}\alpha^2}{\alpha^2} + \frac{C_{s_c,r_n}(\beta^2 + \gamma^2)}{\alpha^2}, \\
&= C_{s_c,r_n} + \frac{C_{s_c,r_n}(\beta^2 + \gamma^2)}{\alpha^2}, \\
\frac{1 - C_{s_c,r_n}}{C_{s_c,r_n}} &= \frac{(\beta^2 + \gamma^2)}{\alpha^2}.
\end{aligned} \tag{4.5}$$

The SINR for a recovered signal can be found by taking the coherence with respect to its target source.

4.1.2 Instantaneous Results

A linear mixing test was used to evaluate the recursiveICA algorithm with respect to the fastICA algorithm. The test consisted of separating two uniformly distributed signals, with additive gaussian noise on the observations. The noise levels were minus 10, 20 and 30 dB with respect to the observed power at each microphone, as well as the noiseless case. The performance of the algorithm was plotted for forgetting factors, γ, of 0.1, 0.05, 0.01 and an adaptive rate of:

$$\gamma = 0.8^{(t/600)} + 0.001, \tag{4.6}$$

where t is the time since the initation of the algorithm in samples. While the adaptive algorithms never outperform the fastICA algorithm in terms of SINR values, they have the advantage of obtaining their unmixing solution in real time, whereas the fastICA result is only obtained after full analysis of the signal. The fastICA SINR can be considered as the upper performance limit of the adaptive algorithms for a given timestep.

Figure 4.1 shows the SINR readings for the various solutions in the case where there is no additive noise. The fastICA algorithm obtains an unmixing system with an SINR of 123 dB, which would be imperceptible to humans. The adaptive algorithm with the adaptive forgetting factor performs less predictably than the fastICA algorithm, but is capable of obtaining similar performance. After it has converged the minimum SIR was 90 dB. The fixed learning rate algorithms obtain solutions of around 40 dB, as shown by the red plots in figure 4.1. This is likely due to the increased estimation noise present at higher learning rates. Their convergence time is also closely tied to their learning rate. A higher learning rate results faster convergence and vice versa.

Figures 4.2, and 4.3 show the effect of increasing additive gaussian noise of the convergence of the system. Here the adaptive algorithms perform on par with the block wise fastICA algorithm. The adaptive learning rate still results in the fastest convergence rates as well as the

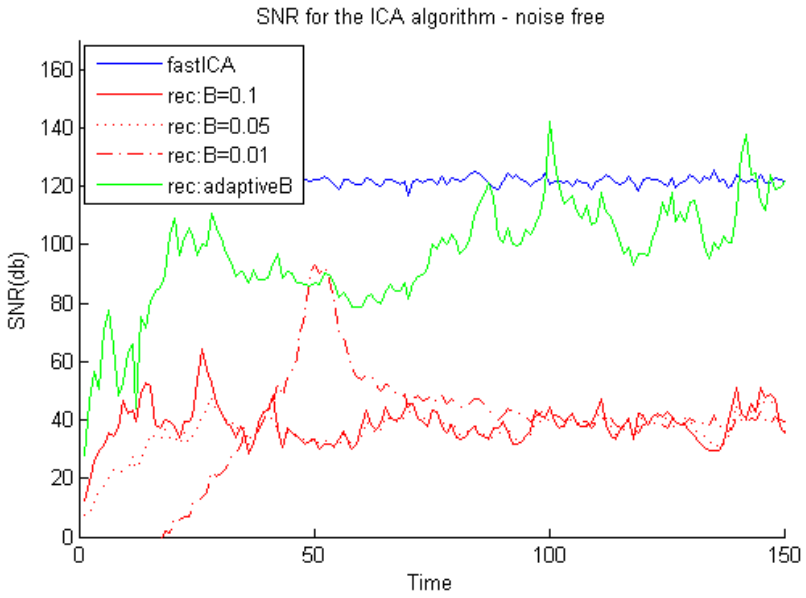

Figure 4.1: SINR values in the noiseless case, we observe the adaptive learning rate algorithm is obtaining higher convergence than the fixed learning rate algorithms. Note that the variation in the SINR over time is partly due to the periodicity assumption in its calculation.

highest convergence of the recursive algorithms at all noise levels. The results show the system entering a noise limited maximum SINR, as all algorithms converge to the additive noise cap in both the -30 dB and -10 dB cases.

4.2 Online Convolutive Separation Performance for Simulated Data

The efficacy of separation when the system is applied to simulated and real world data is presented in the following two sections, and the methods used to obtain these results are outlined. The C algorithm has been tested in parts to ensure it is equivalent to the Matlab code. All testing of the

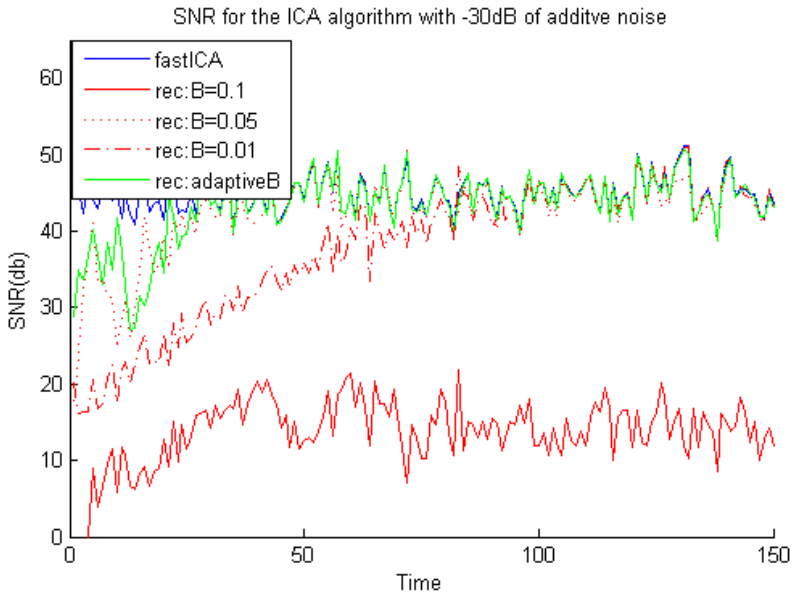

Figure 4.2: SINR values with additive gaussian noise of -30 dB.

complete system has been performed on the Matlab version.

4.2.1 Simulation Environment

The simulation environment uses the image method described in section 2.3 to generate transfer functions of four speaker to four microphone mixtures in a virtual room of 5 x 8 x 3 metres. The image method was used to generate these transfer functions with reflection coefficients of 0.8 for the walls and ceiling, 0.3 for the floor. This reflects a carpeted room with dry wall ceilings and walls. Microphones are placed randomly about the room. Speakers are placed randomly in the XY plane, but restricted to being within 0.75 m and 1.8 m in the Z plane to simulate speaking positions for sitting or standing individuals.

The transfer functions were then convolved with the each source to simulate the observed signal at a particular microphone for that source, and the final observation is the sum of all of the observed signals. The sys-

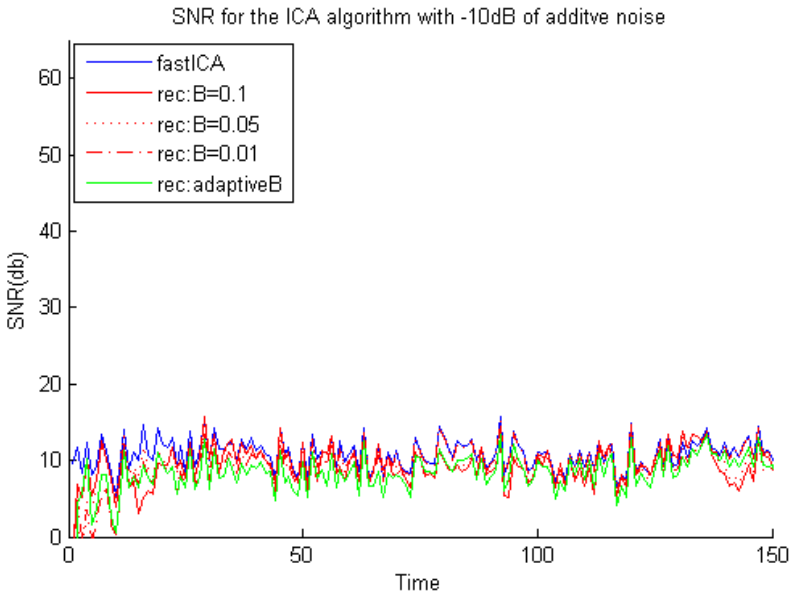

Figure 4.3: SINR values with additive gaussian noise of -10 dB.

tem described by 3 was then used to unmix the sources from the observed signals.

4.2.2 Simulation Results

Separation efficacy was tested for mixtures of two to eight voices through up to eight microphones. We remind the reader that system was not designed to handle the overcomplete case, that is where the number of voices exceeds the number of microphones. The performance of the system for the following number of voices by number of microphones will be discussed: three by three; six by six; and eight by eight.

Figure 4.4 depicts the separation efficacy of the algorithm for a three by three simulation. The envelope of the recovered signals can be seen to correlate strongly with the original signals. The rightmost set demonstrates the scaling system, as the structures of the recovered source are time aligned with the corresponding structures in the mixture, rather than

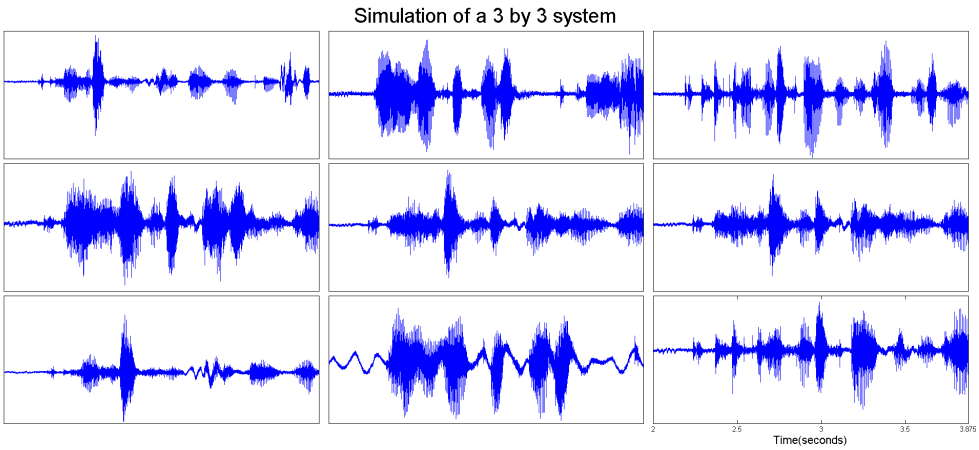

Figure 4.4: Comparison between the generating source signals (top), simulated mixed signals (middle) and recovered signals (bottom) in the time domain for a three by three mixture. Note that the envelope of the recovered signal at the bottom strongly correlates with the original source signal in the top figure.

that of the original source. This results in a well scaled, and phase aligned signal, delayed slightly, as discussed in section 2.8.3. The centre signal demonstrates the weakness of the scaling approach, as a strong low frequency wave has corrupted the recovered signal. This is due to the loss of directionality of low frequency sound, which does not permit separation through a beam-forming vector reconstitution. In this case the corrupting signal is 7.5 Hz below 20 Hz and imperceptible to humans. This effect has been observed in frequencies of up to 60 Hz in the simulated results and 140 Hz in the Real world tests.

Figure 4.5 depicts the separation efficacy of the algorithm for a six by six simulation, showing a selection of three of the sources. While the mixtures clearly have an increased number of underlying sources, the recovered signals show similar levels of corruption to the three by three case. In addition the low frequency corruption observed in the middle signal of figure 4.4 is not exhibited here.

In the eight by eight simulation (depicted in Figure 4.6) we observe a

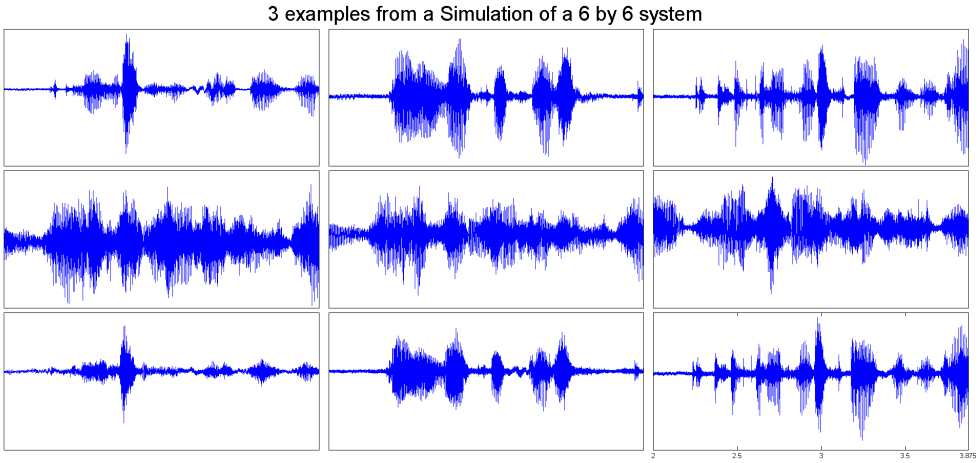

Figure 4.5: Comparison between the generating source (top), simulated mixture (middle) and recovered (bottom) signals in the time domain for a six by six mixture.The mixtures are visibly more crowded, However, the recovered signals have similar, if not better acuity than those of the three by three mixture.

relatively minor increase in additive noise over the six speaker case. The central recovered signal again exhibits a low frequency corruption.

The signal to interference ratio, as experienced at each frequency is displayed in figure 4.7. The two by two simulation shows an improvement of 16.1 dB on average, while the 8 by 8 simulation shows an average improvement of 12 dB.

Figure 4.8 compares the spectrogram of a particular source with a simulated microphone and the recovered estimation of the source. The ability to capture the original sources harmonic structure is demonstrated by the aligned red bands in the source and recovered signals (top and bottom figures respectively). There is a reduction in energy, particularly in the low frequency bands of the recovered signal. This is due to incorrect permutation alignment, and results in that particular bin being suppressed in the scaling stage and causing the observable reduction in energy.

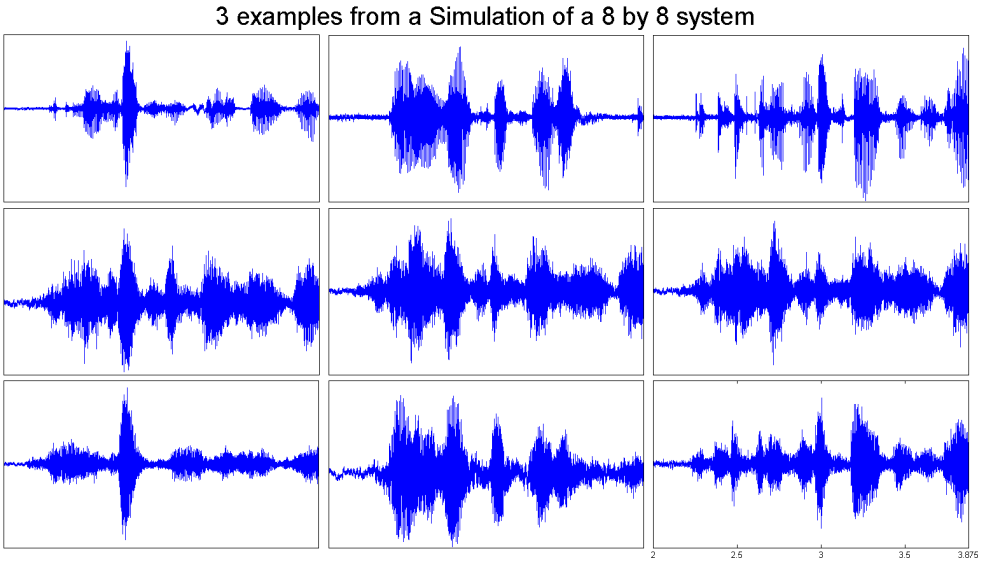

Figure 4.6: Comparison between the generating source(top), simulated mixture(middle) and recovered(bottom) signals in the time domain for an eight by eight mixture.The mixtures are more crowded again, and there is an increase in additive noise for all the signals present.

4.3 Online Convolutive Separation Performance for Real World Data

The simulation environment of 4.2 has two major oversights:

- it assumes perfect point sources with omnidirectional directivity.
- It assumes the room is a perfect box, with no furnishings or interior reflectors.

People speaking are not point sources and are directive. Therefore, the transfer functions between each speaker and microphone differs from that given by the image method. It is likely that the transfer function for a directive pattern would be easier to estimate as the energy of the signal will be spread over fewer reflections, resulting in less reverberation.

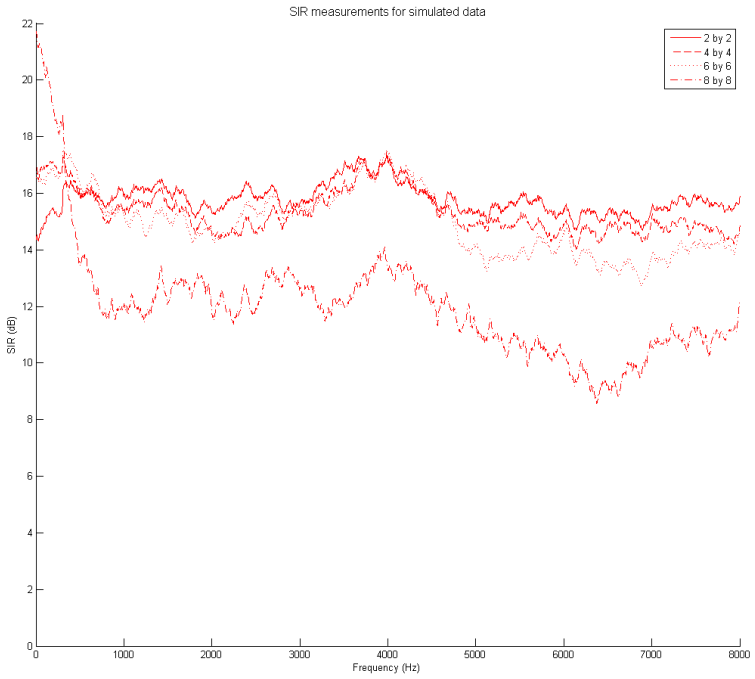

Figure 4.7: The signal to interference ratio, as experienced at each frequency, averaged over all sources and over five runs. The two by two simulation shows an improvement of 16.1 dB on average, while the 8 by 8 simulation shows an average improvement of 12 dB, with the other two simulations falling betwen these results.

Furnishings and non-regular room structure significantly alter the room impulse response from that idealized by the image method, especially in the late reverberation period.

It was decided that a real world test to verify the operation of the system, and justify the use of the simulation in initial testing was required.

4.3.1 Experimental Setup

The electroacoustic lab at VUW was chosen as the test room, as it is sound proofed to reduce the effects of ambient noise. The room is relatively

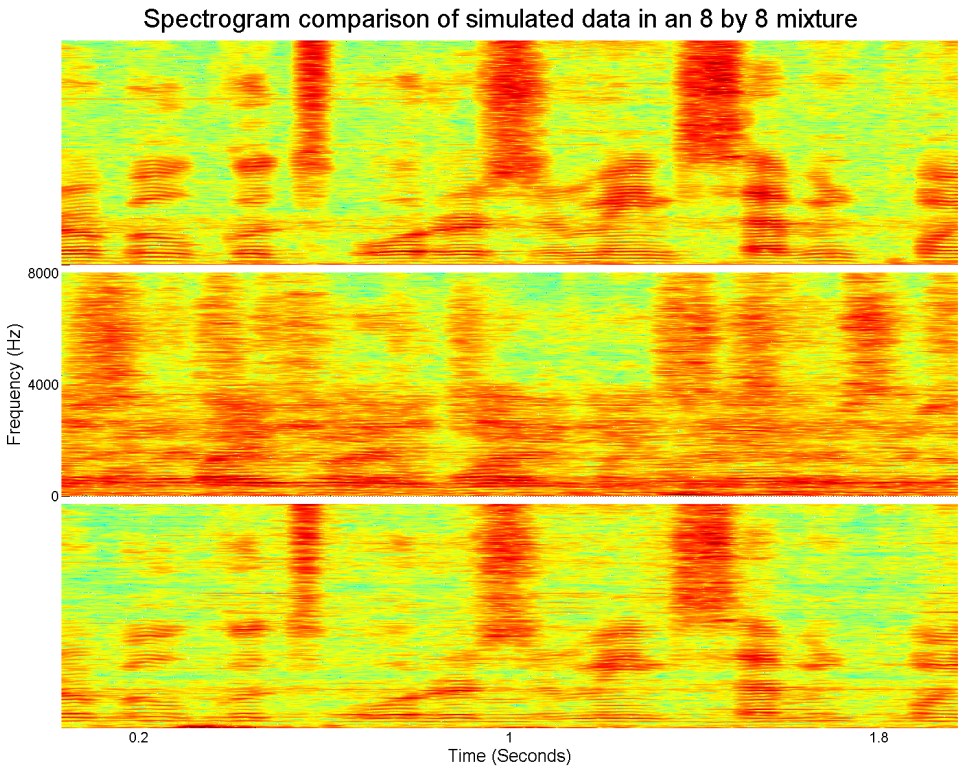

Figure 4.8: Comparison between the generating source (top), recorded mixture(middle) and recovered signal(bottom) in spectrogram form. The harmonic structures present in the source file are mostly recovered from the mixture, however, gaps in certain freqency ranges can be observed at all frequencies.

'dead' acoustically (reflected in a low T60 time) and as such, many real environments would likely have more challenging acoustics. However, the ability to control external noise sources and the access to the necessary components made it a favorable choice. Eight Mackie MR5 speakers are used to play speech recordings and eight Studio Project B3 microphones record the sound from within the room. Using speakers gives control over the speech, and allows us to make accurate comparisons between the recovered and original signals. The Presonus FP10 preamplifier was used as the sound card handling audio IO.

Figure 4.9: The speaker array in the electroacoustics laboratory comprising 24 Mackie MR5 monitors in a circular array. To provide a mixture of closely spaced and spread out emitters, the four leftmost speakers played the first four sources, and then the last four speakers were assigned to the rest of the array at even intervals. The first three microphones within the array can be seen in the foreground

The room is 3 m by 3 m by 2.5 m. It has some baffling on the walls and as a result they have lower than average sonic reflectivity. The T60 time of the room is 0.25 seconds. Figures 4.9 and 4.10 show the test equipment, and 4.11 depicts where the elements are located within the room in an overhead view. The layout was chosen to provide a challenge to the system with a mix of both tightly spaced and spread out speakers. This should make the mixing coefficients highly correlated for the tightly spaced speakers. In addition the microphone array has a mixture of randomized placement and a circular array, as well as a mixture of central and peripheral location. The apparent randomness of the layout was chosen to prove the robustness of the system to spatial arrangements.

Figure 4.10: The 24 microphone array, missing the three microphones from figure 4.9. Five microphones spread evenly around the array were used to complete the 8 microphone system. This simulated a far field array, as it was located in the corner of the room.

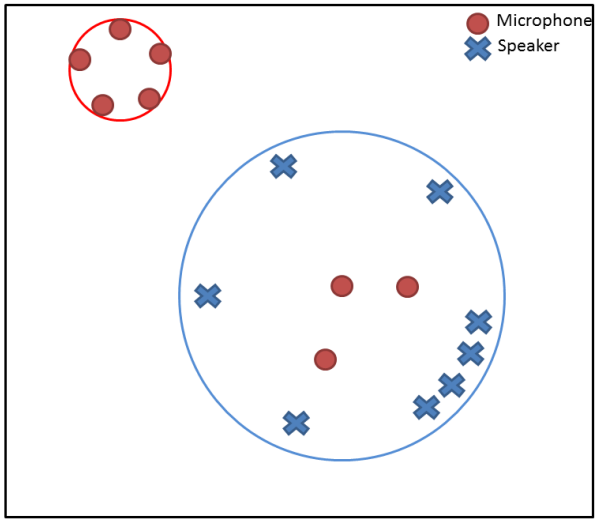

Figure 4.11: The layout of the electroacoustics laboratory, and the active elements of the system.

4.3.2 Real World Results

The same tests applied to the simulation have been used to analyse the performance of the system when applied to data captured in the testing framework described above.

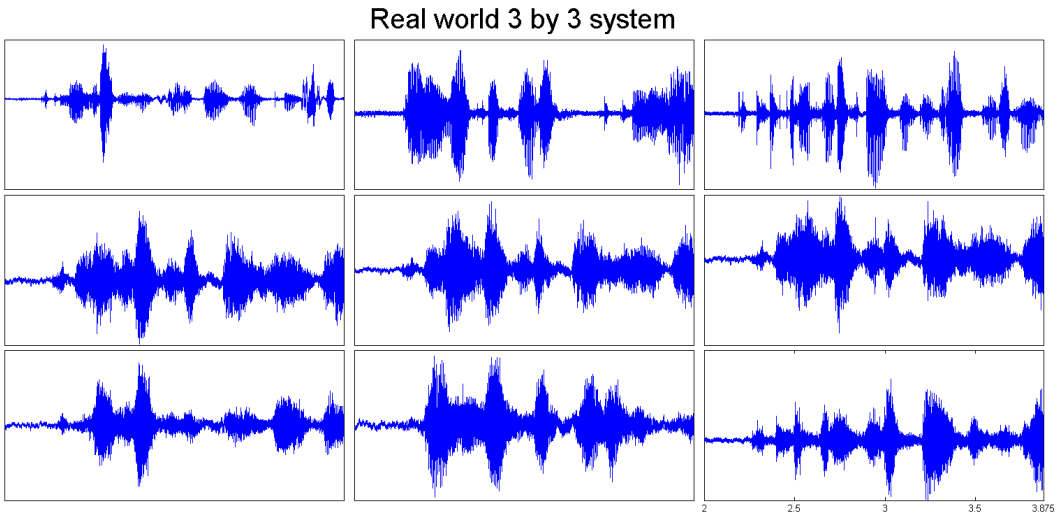

Figure 4.12: Comparison between the generating source(top), simulated mixture(middle) and recovered(bottom) signals in the time domain for a three by three mixture. The relative noise of each signal is significantly increased over the equivalent simulation.

Figure 4.12 depicts the performance obtained in the three by three case. In contrast with the simulation, a significant increase in additive noise can be observed in the three recovered signals. However, the system still improves significantly over the mixed recordings in two out of three cases. The central case demonstrates a failure to recover the original source. While hard to demonstrate visually, the intelligibility of the speech is improved over the mixed recording, as the noise is focused in the low frequency range and is significantly distorted.

The eight by eight case is depicted in Figure 4.13. Again we observe and increase in noise over the equivalent simulation. In the central case there is a significant level of interference, which is the presence of a second

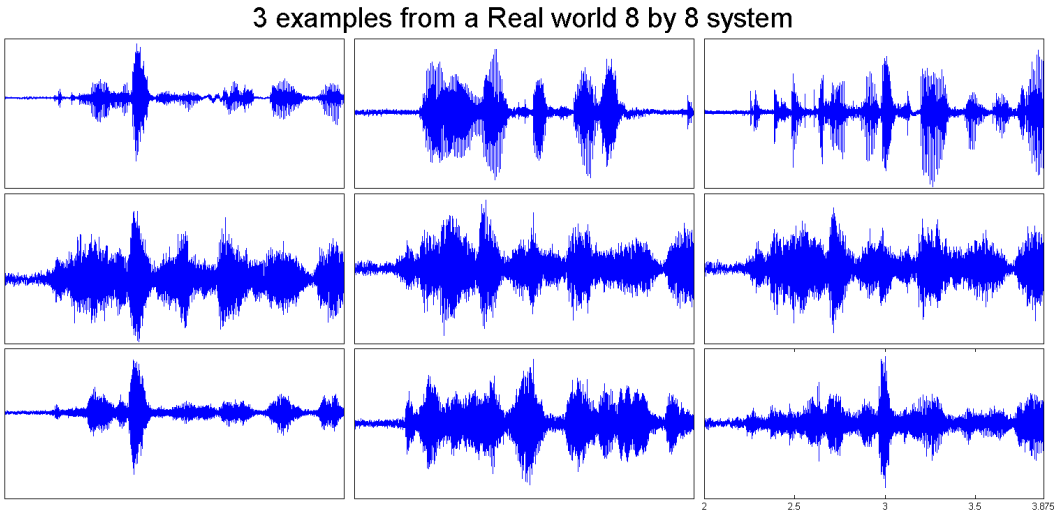

Figure 4.13: Comparison between the generating source(top), simulated mixture(middle) and recovered(bottom) signals in the time domain for an eight by eight mixture.The mixtures are more crowded again, and there is an increase in additive noise for all the signals present.

source. The source has a higher than average correlation, possibly due to the fact that the speech signals were looped and the two had similar lengths, resulting in unintended artificial correlation being introduced.

Figure 4.14 shows the SINR results for the real data set. The two by two case shows an SINR improvement of 10.9 dB and the 8 by 8 case shows an improvement of 8.6dB. Finally Figure 4.15 gives a comparative look at the two results, highlighting the loss of separation performance observed when moving from a simulation to a real world system. This is likely due to the processing noise.

Figure 4.16 compares the spectrogram of a particular source with a simulated microphone and the recovered estimation of the source. The ability to capture the original sources harmonic structure is demonstrated by the aligned red bands in the source and recovered signals (top and bottom figures respectively).

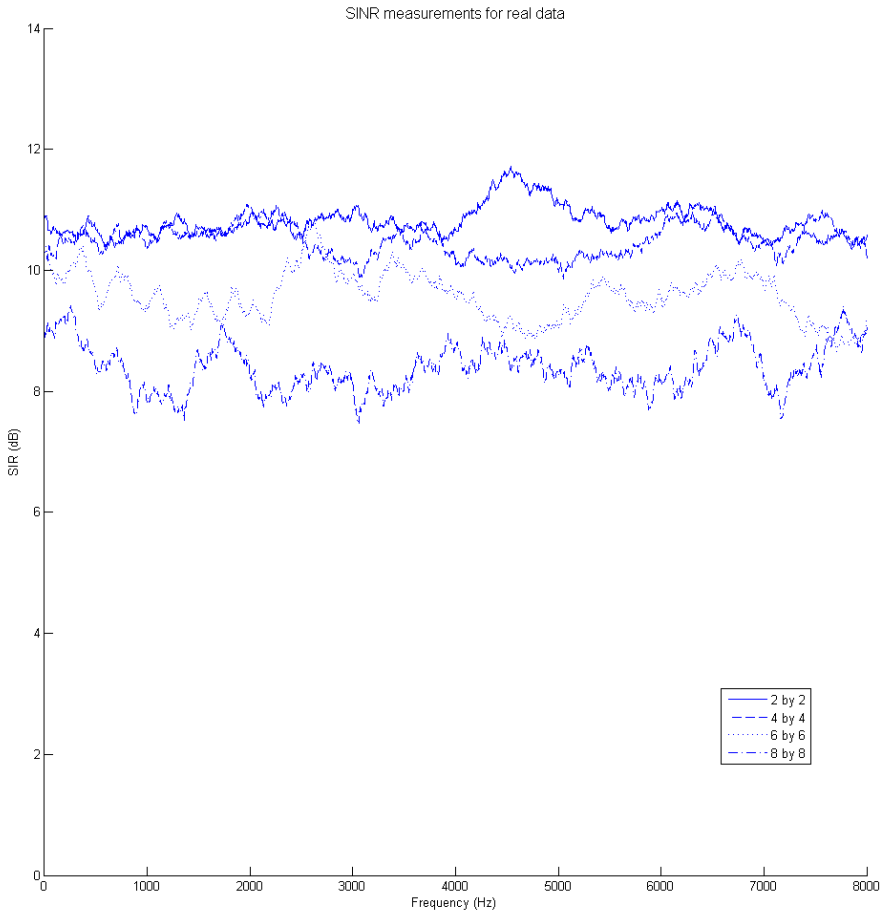

Figure 4.14: The signal to interference ratio, as experienced at each frequency, averaged over all sources and over five runs. The two by two simulation shows an improvement of 10.9 dB on average, while the 8 by 8 simulation shows an average improvement of 8.6 dB.

Figure 4.15: Comparison between SINR figures for the simulated and real data sets.

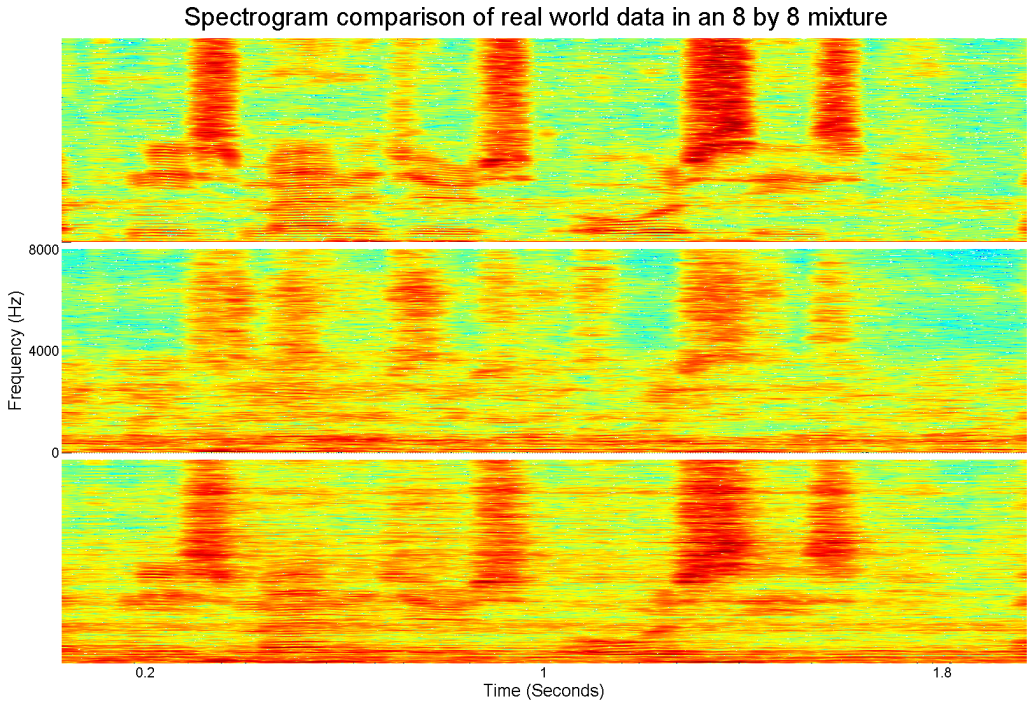

Figure 4.16: Comparison between the generating source (top), recorded mixture (middle) and recovered signal (bottom) in spectrogram form. The harmonic structures present in the source file are mostly recovered from the mixture, but gaps in certain freqency ranges can be observed at all frequencies. In additon noise can be observed filling in periods of total silence in the original source.

4.4 Real Time Performance

To test the real time performance, measurement of the delays experienced in the C implementation was required. In particular the acoustic delay and the estimation delay were characterised.

4.4.1 Acoustic Delay

An important measure of system performance is the total delay from audio capture to playback. To measure this a test using the presonus FP10 audio capture device has been designed to allow for measurement of the total delay between capture of voice and playback through the speakers. The test setup is depicted in Figure 4.17.

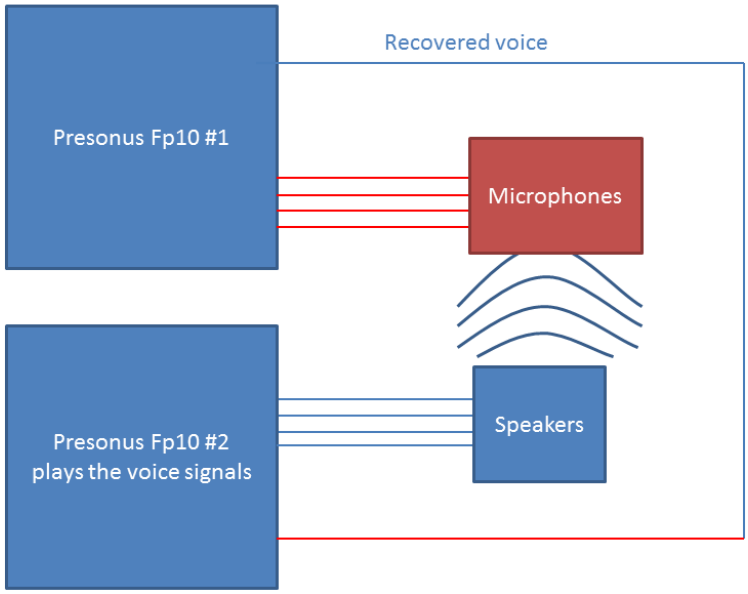

Figure 4.17: wiring diagram of the test setup used to calculate the total delay introduced while processing, note the two FP10 units are asynchronous.

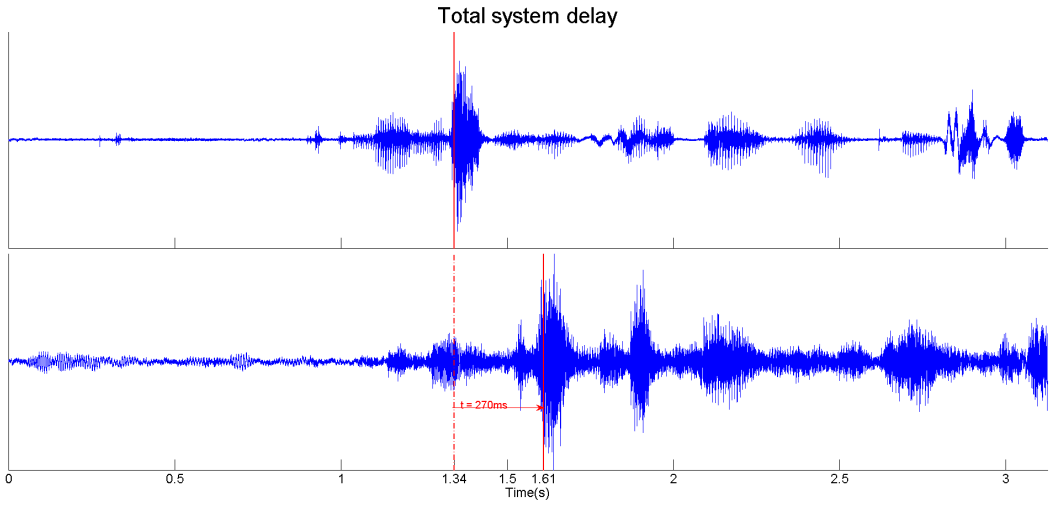

Figure 4.18: The total delay of the real time system is observed to be approximately 270 ms.

Figure 4.18 depicts the delay experienced by the user of the real time system. The vertical red lines show the chosen attack, and the dotted line is the projection of the attack from the speaker output to the measured input. This shows a total delay of 270 milliseconds. This means that the recovered sources fall into the perceivable echo region, and the user would likely perceive the delay that is present.

4.4.2 Delay Between Unmixing Operators

There is also a measurable delay due to the time it takes to estimate the unmixing process. This delay describes the minimum time required to respond to a change in the environment, either the movement of a speaker within the room, or a change in the layout of the room (for example the opening of a door). This time varies with the parameters of the system, in particular the window length of the Gabor transform, the number of frames used in computing the estimate and the number of observations.

Table 1: Execution times for varying system parameters

$N_{microphones}$	N_{Frames}	Window	T_{block}	T_{Matlab}	T_C	$T_{\%}$
2	100	2048	12.8 s	1800ms	120ms	1%
6	100	2048	12.8 s	27000ms	1500ms	12%
8	100	2048	12.8 s	55000ms	3700ms	28%
10	100	2048	12.8 s	105000ms	7500ms	58%
6	200	2048	25.6 s	51000ms	2800ms	10%
6	100	4096	25.6 s	52000ms	2900ms	11%

The table above compares the effect of altering the unmixing parameters on execution time. These parameters are: the number of microphones; the number of frames used by the algorithm to estimate the operator for recovering the sources; the number of samples per Gabor frame; the time required to collect the given number of frames; the execution times of the Matlab and C programs respectively; and the execution time of the C algorithm, as a percentage of the time in column three. The percentage of the time taken to estimate the block is an important reference, as keeping up with the block time allows for all of the data to be used by the unmixing process, due to the way the data is processed as described in Section 3.5.5. The execution time as a fraction of the time to capture the data block is observed to be nearly independent of both the window length and frame number, and scales exponentially with the number of observations.

Chapter 5

Discussion

Previous work in this field of BSS has focused on obtaining high fidelity source separation under a variety of conditions. However, relatively few publications in the field have tested their systems against real world mixtures. Pedersen [38] states that of the 400 papers he reviewed in the field, only 37 of them tested against real world data. Of those that have published this data the work of Sawada [39] and Ukai [40] are among the top of the field. Few have also attempted to do so in a real time fashion, most operate on a full length recording before playing back the sources. The work of Taniguchi [37] and Kim [41] are two well cited researchers who have published results for real time Blind Source Separation Systems.

The maximum published signal to interference ratio for a BSS scheme operating on real world data is 20 dB in the two by two case, by both Sawada and Ukai. In the eight by eight case the maximum achieved SIR was 12 dB [42] by Mukai.

In the case of real time algorithms, Tanguchi reported an SIR improvement of 12 dB for simulated data, for two sources. Kim reported an SIR improvement of 16 dB for simulated data, for three sources. In both cases the simulation is equivalent to that proposed in Section 4.

The proposed system has been shown to obtain a 15.6 dB improvement in SIR in simulation for the two sources case, and an improvement of

12.1 dB in the 8 sources case. This is on par to the figures published in both the offline and online cases. When operating on real world data, the SINR figures drop to 10.5 dB in the two source case and 8.2 dB in the eight source case. While less than the improvements seen in the offline results published by Sawada and Ukai, the fact that the system exhibits this performance on a high number of sources, and scales favourably is promising. In addition, the system could in theory operate in real time on an increased number of sources simultaneously. Extrapolating from the execution time figures, operation on ten or more sources simultaneously is possible, with an SINR figures around 6 to 7 dB. In addition optimized placement of microphones would likely improve the performance.

Currently a number of challenges must be solved before these systems are ready for real world applications. Firstly current systems require the speakers to converse constantly. If there is a pause of over 5 seconds, then the set of unmixing vectors for that voice is lost. In real speech this is common, in normal conversation one participant will pause for periods of time well over 5 seconds while listening to another. No publications have attempted to correct this shortcoming of contemporary BSS systems, assuming that the system will not wander significantly in this period. From experimentation this is not the case.

This is where the clustering based algorithm, described in 3.3 could prove more effective than ICA. Its operation depends on the periods of time where only a few of the people in the room are speaking. In addition the clustering system is computationally efficient, scaling especially well for high numbers of recording elements. This is because the distance measure used for clustering is an angle, and therefore can be considered as a scalar between two points in a space of any dimension. Operating on scalar values, rather than increasingly long vectors due to the extra microphones, is significantly more efficient.

Finally the delay of 270 ms is unacceptable for real world use in the case of someone who is hearing impaired. The sound delivered is a side

channel to the tools such as lip reading, sign language, and body language. Providing this information in a timely manner is critical to having a system which integrates seamlessly. To not do so would only distract them and they would likely not use it.

When timed internally the algorithmic delay between the incoming audio interrupt and the write of the data to the output buffer is 48 ms. This suggests that from a computational standpoint, it is possible to achieve less than 100 ms delay. The most significant delay is the time spent collecting a full Gabor window frame. Audio must be stored for at least that long before it can begin to be processed in the Fourier domain, which is the natural approach.

There are a number of approaches to solving this. One possibility is to operate on shorter Gabor frames in the audio processing step, and processing out the reverberation observed at previous steps. This is a form of Subband processing, and algorithms such as Nordholms [7] would be applicable. Another option is to convert the unmixing operator into a set of ARMA filters, which can trade delay for stability. Aichner [43] used a geometric beamforming technique to obtain a BSS solution in the time domain, avoiding the issue. He published SIR figures of between 6 and 10 dB, for simulated data. His work used ARMA filters of over 2048 taps. His work may provide insight into how to formulate long time domain filters which remain stable.

Chapter 6

Conclusion and Future Work

A cutting edge BSS system operating on real world data, in real time, has been implemented in Matlab and C. The proposed system shows that it is possible to operate on systems with a high microphone and source count, verified to work on up to eight sources. It also shows that this is achievable with an audio processing delay of 48 ms. It is conceivable that the delay caused by the Gabor transform can be reduced to an imperceptible level, without major alteration to the core of the algorithm. The achieved SIR of 15.6 dB for a two source mixture is comparable to contemporary blind source separation algorithms. The 12.1 dB SIR achieved for an eight source mixture exceeds the results obtained by Mukai. The work on the clustering based algorithm is promising, particularly in solving the problem of extended source silence. Applying this clustering algorithm in a recursive fashion over extended timescales could prove fruitful.

The contemporary beamformer is currently the industry standard approach to spatially selective filtering. However, it does not address the problem of reverberation, and its performance degrades quickly in reverberant environments. In addition a beamformer is sensitive to the spatial positioning of its microphones, errors of a few centimetres will result in significant degradation. As a result a rigid array is often used, limiting the options for microphone placement. BSS has the potential to solve these

shortcomings, it can account for reverberation just as well as for the free space system, and as it requires no calibration, will adjust to changes in the system layout.

However, a number of shortcomings mean that while blind source separation has been used successfully in certain fields, particularly functional Magnetic Resonance Imaging (fMRI), it has not been applied in any commercial acoustic systems. There are a number of reasons this is the case. Firstly they are resource intensive algorithms, and it is often assumed that obtaining acceptable results from a system with a high number of speakers, or microphones, is not practical. This thesis suggests that stable, real time operation for the eight by eight case is possible, and that it is reasonable to expect operation past the ten by ten case.

Secondly the ability to handle conversational speech is necessary. Without handling the long pauses present in real world communication, practical applications are limited. As mentioned in Chapter 5, clustering algorithms may prove well suited to this problem. In addition, the problem is similar to the concept of landmarking in Simultaneous Localisation And Mapping (SLAM) algorithms, where a landmark may go unseen for an extended period of time. Quickly and consistently identifying corresponding landmarks is key to a robust SLAM algorithm, and may provide insight. With more research, these problems can likely be solved, and in doing so real world applications may be found.

6.1 Future work

The current system operates well under controlled conditions, but has two shortcomings that, once solved, will provide significant improvements in performance and robustness: Firstly, it is assumed that all speakers in a room will constantly talk. This is not generally the case; speakers will for the most part not be speaking over each other from within a conversation, although people involved in independent conversations will. As a result there may be extended periods, conceivably hours, where the mixing process is relatively static, but certain speakers are quiet for significant periods of time (on the order of 10 minutes). Tracking these silent voices in the room would allow for improved acquisition when those people decide to start speaking again. This problem is complex, there needs to be some method of discarding unmixing vectors where the speaker has left the environment, and for deciding if a new speaker is really new or is just a stored voice becoming active again. However, this is the major issue standing between the current system and real world operation.

The second is that the speakers produce signals that are sparse in time and frequency; however, the permutation and scaling correction must assign a signal to each frequency, even if the source is not producing any signal at that frequency. A solution is a time-frequency mask, which would allow certain frequencies and certain time slots within a frequency band to be suppressed. There are a number of researchers working on this [45][46], and it is one of the techniques employed by Sawada in their state of the art DOLPHIN II system that was the top performer in the CHiME speech challenge [30]. This should reduce noise significantly in frequencies that are not excited by a particular speakers voice.

Bibliography

[1] M. Delcroix, K. Kinoshita, T. Nakatani, S. Araki, A. Ogawa, T. Hori, S. Watanabe, M. Fujimoto, T. Yoshioka, and T. Oba, "Speech recognition in the presence of highly non-stationary noise based on spatial, spectral and temporal speech/noise modeling combined with dynamic variance adaptation," in *Proc. 1st Int. Workshop on Machine Listening in Multisource Environments (CHiME)*, 2011, pp. 12–17.

[2] T. Nakatani, M. Souden, S. Araki, T. Yoshioka, T. Hori, and A. Ogawa, "Coupling beamforming with spatial and spectral feature based spectral enhancement and its application to meeting recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 7249–7253.

[3] N. Wood and N. Cowan, "The cocktail party phenomenon revisited: how frequent are attention shifts to one's name in an irrelevant auditory channel?" *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 21, no. 1, p. 255, 1995.

[4] M. L. Hawley, R. Y. Litovsky, and J. F. Culling, "The benefit of binaural hearing in a cocktail party: Effect of location and type of interferer," *The Journal of the Acoustical Society of America*, vol. 115, p. 833, 2004.

[5] E. Lindemann, "Dynamic intensity beamforming system for noise reduction in a binaural hearing aid," Apr. 23 1996, uS Patent 5,511,128.

[6] P. Chevalier, J.-P. Delmas, and A. Oukaci, "Optimal widely linear mvdr beamforming for noncircular signals," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3573–3576.

[7] N. Grbic and S. Nordholm, "Soft constrained subband beamforming for hands-free speech enhancement," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1. IEEE, 2002, pp. I–885.

[8] C. Pan, J. Chen, and J. Benesty, "On the noisereduction performance of the mvdr beamformer innoisy and reverberant environments," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 815–819.

[9] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 5, pp. 530–538, 2004.

[10] E. Bingham and A. Hyvarinen, "ICA of complex valued signals: a fast and robust deflationary algorithm," in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, vol. 3, 2000, pp. 357–362 vol.3.

[11] N. Ito, S. Araki, and T. Nakatani, "Permutation-free convolutive blind source separation via full-band clustering based on frequency-independent source presence priors," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 3238–3242.

[12] S. Haykin and Z. Chen, "The cocktail party problem," *Neural computation*, vol. 17, no. 9, pp. 1875–1902, 2005.

[13] M. Siegert, H. Römer, and M. Hartbauer, "Maintaining acoustic communication at a cocktail party: heterospecific masking noise improves signal detection through frequency separation," *The Journal of experimental biology*, vol. 216, no. 24, pp. 4655–4665, 2013.

[14] E. J. Jensen, I. Hargreaves, A. Bass, P. Pexman, B. G. Goodyear, and P. Federico, "Cortical reorganization and reduced efficiency of visual word recognition in right temporal lobe epilepsy: A functional mri study," *Epilepsy research*, vol. 93, no. 2, pp. 155–163, 2011.

[15] D. Van Compernolle, "Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE, 1990, pp. 833–836.

[16] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[17] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, p. 943, 1979.

[18] B. Shinn-Cunningham, J. Desloge, and N. Kopco, "Empirical and modeled acoustic transfer functions in a simple room: effects of distance and direction," in *Applications of Signal Processing to Audio and Acoustics, 2001 IEEE Workshop on the*, 2001, pp. 183–186.

[19] L. Xu, "Temporal byy learning for state space approach, hidden markov model, and blind source separation," *Signal Processing, IEEE Transactions on*, vol. 48, no. 7, pp. 2132–2144, 2000.

[20] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *Signal Processing, IEEE Transactions on*, vol. 45, no. 2, pp. 434–444, 1997.

[21] A. Ferreol and P. Chevalier, "On the behavior of current second and higher order blind source separation methods for cyclostationary sources," *Signal Processing, IEEE Transactions on*, vol. 48, no. 6, pp. 1712–1725, 2000.

[22] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *Signal Processing, IEEE transactions on*, vol. 52, no. 7, pp. 1830–1847, 2004.

[23] A. Hyvarinen, "Gaussian moments for noisy independent component analysis," *Signal Processing Letters, IEEE*, vol. 6, no. 6, pp. 145–147, 1999.

[24] J. A. Rice, *Mathematical Statistics and Data Analysis*, 3rd ed. Duxbury Press, Apr. 2001. [Online]. Available: http://www.worldcat.org/isbn/0534399428

[25] S. Moussaoui, D. Brie, A. Mohammad-Djafari, and C. Carteret, "Separation of non-negative mixture of non-negative sources using a Bayesian approach and MCMC sampling," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4133–4145, 2006.

[26] S. Moussaoui, C. Carteret, D. Brie, and A. Mohammad-Djafari, "Bayesian analysis of spectral mixture data using Markov chain Monte Carlo methods," *Chemometrics and Intelligent Laboratory Systems*, vol. 81, no. 2, pp. 137 – 148, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169743905001747

[27] R. Carhart, T. W. Tillman, and E. S. Greetis, "Perceptual masking in multiple sound backgrounds," *The Journal of the Acoustical Society of America*, vol. 45, no. 3, pp. 694–703, 1969.

[28] D. Gerhard, "Pitch-based acoustic feature analysis for the discrimination of speech and monophonic singing," *Canadian Acoustics*, vol. 30, no. 3, pp. 152–153, 2002.

[29] A. Cichocki and P. Georgiev, "Blind source separation algorithms with matrix constraints," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 86, no. 3, pp. 522–531, 2003.

[30] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, M. Matassoni *et al.*, "The second 'CHiME' speech separation and recognition challenge: Datasets, tasks and baselines," in *2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[31] H. Sawada, S. Araki, R. Mukai, and S. Makino, "Grouping separated frequency components by estimating propagation model parameters in frequency-domain blind source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1592–1604, 2007.

[32] J. Even and N. Hagita, "Resolving FD-BSS permutation for arbitrary array in presence of spatial aliasing," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 13–16.

[33] W. Baumann, D. Kolossa, and R. Orglmeister, "Beamforming-based convolutive source separation," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 5, 2003, pp. V–357–60 vol.5.

[34] J. Cooper and K. Worden, "On-line physical parameter estimation with adaptive forgetting factors," *Mechanical Systems and Signal Processing*, vol. 14, no. 5, pp. 705 – 730, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0888327000913220

[35] B. Loesch and B. Yang, "Source number estimation and clustering for underdetermined blind source separation," in *Proc. IWAENC*, 2008.

[36] J. Allen, D. Berkley, and J. Blauert, "Multimicrophone signal-processing technique to remove room reverberation from speech signals," *The Journal of the Acoustical Society of America*, vol. 62, no. 4, pp. 912–915, 1977.

[37] T. Taniguchi, N. Ono, A. Kawamura, and S. Sagayama, "An auxiliary-function approach to online independent vector analysis for real-time blind source separation," in *Hands-free Speech Communication and Microphone Arrays (HSCMA), 2014 4th Joint Workshop on*. IEEE, 2014, pp. 107–111.

[38] M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra, "A survey of convolutive blind source separation methods," *Multichannel Speech Processing Handbook*, pp. 1065–1084, 2007.

[39] H. Sawada, S. Araki, and S. Makino, "Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 3, pp. 516–527, 2011.

[40] U. Satoshi, T. Takatani, H. Saruwatari, K. Shikano, R. Mukai, and H. Sawada, "Multistage simo-model-based blind source separation combining frequency-domain ica and time-domain ica," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 88, no. 3, pp. 642–650, 2005.

[41] T. Kim, "Real-time independent vector analysis for convolutive blind source separation," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, no. 7, pp. 1431–1438, July 2010.

[42] R. Mukai, H. Sawada, S. Araki, and S. Makino, "Frequency domain blind source separation for many speech signals," in *Independent Component Analysis and Blind Signal Separation*. Springer, 2004, pp. 461–469.

[43] R. Aichner, S. Araki, S. Makino, T. Nishikawa, and H. Saruwatari, "Time domain blind source separation of non-stationary convolved signals by utilizing geometric beamforming," in *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, 2002, pp. 445–454.

[44] E. W. Tiemersma, S. L. Bronzwaer, O. Lyytikäinen, J. E. Degener, P. Schrijnemakers, N. Bruinsma, J. Monen, W. Witte, and H. Grundmann, "Methicillin-resistant staphylococcus aureus in europe, 1999–2002," 2004.

[45] R. M. Toroghi, F. Faubel, and D. Klakow, "Multi-channel speech separation with soft time-frequency masking," in *SAPA-SCALE Conference*, 2012.

[46] M. Souden, S. Araki, K. Kinoshita, T. Nakatani, and H. Sawada, "A multichannel MMSE-based framework for speech source separation and noise reduction," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 9, pp. 1913–1928, 2013.