Audio Source Separation using Independent Component Analysis

Nikolaos Mitianoudis
Department of Electronic Engineering,
Queen Mary, University of London

Thesis submitted in partial fulfilment
of the requirements of the University of London
for the degree of Doctor of Philosophy

April, 2004

Abstract

Audio source separation is the problem of automated separation of audio sources present in a room, using a set of differently placed microphones, capturing the auditory scene. The whole problem resembles the task a human can solve in a cocktail party situation, where using two sensors (ears), the brain can focus on a specific source of interest, suppressing all other sources present (*cocktail party problem*).

In this thesis, we examine the audio source separation problem using the general framework of *Independent Component Analysis* (ICA). For the greatest part of the analysis, we will assume that we have equal number of sensors and sound objects. Firstly, we explore the case that the auditory scene is modeled as *instantaneous mixtures* of the auditory objects, to establish the basic tools for the analysis.

The case of real room recordings, modeled as *convolutive mixtures* of the auditory objects, is then introduced. A novel *Fast Frequency Domain ICA* framework is introduced, using two possible implementations. In addition, a robust *Likelihood Ratio Jump solution* to the *permutation problem* of ordering sources along the frequency axis is presented. The idea of exploiting the extra geometrical information, such as the microphone spacing, in order to perform permutation alignment using *beamforming* is then examined. Moreover, the idea of "*intelligent*" source separation of a desired source is introduced. Previous work on instrument recognition is combined with source separation, as an attempt to emulate the human brain's selectivity of sources. The problem of more sources than sensors is also addressed along with other extensions of the original framework.

A great number of audio source separation problems can be addressed successfully using Independent Component Analysis. The thesis concludes by highlighting some of the as yet unsolved problems to tackle the actual audio source separation problem in full.

Στους γονείς μου και την αδελφή μου
για την αγάπη τους.

Acknowledgements

Completing a Philosophy Doctorate in a new and very challenging subject is usually a journey through a long and winding road, where one has to tame more oneself than the actual phenomena in research. Luckily, I was not alone in this trip. The following were my company in this journey, and I would like to say a big thanks, as this work might not have been possible without them.

Primarily, I would like to thank my supervisor Dr. Michael E. Davies for his close supervision and very fruitful collaboration in and outside the aspects of my project.

Thanks to the Electronic departments of Queen Mary College and King's College, University of London for funding my research and covering my living expenses in London for the second/third and first years of my study respectively. Also for the financial support to attend and present my work in several conferences and journals.

To Prof. Mark Sandler and Dr. Mark Plumbley for always keeping up with my research, always being there for me.

To Christopher Duxbury for all the technical discussions and for being a huge musical influence on me over these years, reminding me that I always wanted to be a musician (at least trying to). To Dr. Juan Bello for introducing me to the excellent Venezuelan Rum and just being a great friend and technical support over these years. To Dr. Laurent Daudet for being my technical helpdesk for a year and his hospitality and company in Paris and Santorini. To Giuliano Monti for inspiring me to smile in all situations of life. To Josh, JJ, Dawn, Samer, Paul, Chris Harte, Nico, Jofre, Ade and all other people who studied or demonstrated with me at the DSP group for all the technical discussions, collaboration and fun we had either at King's or Queen Mary.

Finally, I would like to thank all my friends in London and Thessaloniki, and especially, my family, for their constant encouragement and love.

Contents

1 Introduction **1**

1.1 What is Audio Source Separation ? 1

1.1.1 Computational Auditory Scene Analysis (CASA) . . . 2

1.1.2 Beamforming 3

1.1.3 Blind Source Separation 4

1.2 Applications of Audio Source Separation 5

1.3 Thesis Overview 6

1.4 Publications derived from this work 8

2 Blind Source Separation using Independent Component Analysis **10**

2.1 Introduction 10

2.2 Instantaneous mixtures 12

2.2.1 Model formulation 12

2.2.2 Problem Definition 13

2.2.3 Principal Component Analysis 14

2.2.4 Independent Component Analysis 17

2.2.5 ICA by Maximum Likelihood Estimation 19

2.2.6 ICA by Entropy Maximisation 23

2.2.7 ICA by Maximisation of nonGaussianity 24

2.2.8 ICA by Tensorial Methods 32

2.2.9 ICA by Nonlinear Decorrelation 34

2.2.10 Performance Evaluation of ICA methods for instantaneous mixtures . . . 36
2.3 More sources than sensors . . . 38
2.3.1 Problem Definition . . . 38
2.3.2 Is source separation possible? . . . 38
2.3.3 Estimating the sources given the mixing matrix . . . 41
2.3.4 Estimating the mixing matrix given the sources . . . 43
2.4 Convolutive mixtures . . . 49
2.4.1 Problem Definition . . . 49
2.4.2 Time-Domain Methods . . . 51
2.4.3 Frequency-Domain Methods . . . 53
2.5 Conclusion . . . 61

3 Fast ICA solutions for convolutive mixtures 62
3.1 Introduction . . . 62
3.2 Solutions for the scale ambiguity . . . 62
3.2.1 Previous approaches . . . 63
3.2.2 Mapping to the observation space . . . 63
3.3 Solutions for the permutation ambiguity . . . 65
3.3.1 Source modelling approaches . . . 66
3.3.2 Channel modelling approaches . . . 67
3.3.3 A novel source modelling approach . . . 68
3.4 Fast frequency domain ICA algorithms . . . 75
3.4.1 A fast frequency domain algorithm . . . 76
3.4.2 An alternative approach . . . 77
3.4.3 Similarities between the two Fast-ICA solutions . . . 80
3.5 A unifying frequency domain framework . . . 81
3.6 Evaluation . . . 82
3.6.1 Performance metrics for convolutive mixtures . . . 83
3.6.2 Experiment 1 . . . 84
3.6.3 Experiment 2 . . . 84
3.6.4 Performance Measurements . . . 87

3.6.5 Computational Cost 87
3.7 Other Extensions 91
3.7.1 Aliasing in the Frequency domain framework 91
3.7.2 Effect of frame size 95
3.8 Conclusion 98

4 Using Beamforming for permutation alignment 100
4.1 Introduction 100
4.2 Array Signal Processing 101
4.2.1 Definition 101
4.2.2 Number of sources and Directions Of Arrival (DOA) estimation 104
4.2.3 Beamforming - Separation 107
4.2.4 Frequency Domain Beamforming 108
4.3 ICA as a Beamformer 110
4.4 Beamforming as a solution to the permutation ambiguity . . 113
4.4.1 DOA estimation ambiguity 113
4.4.2 Permutation alignment ambiguity 115
4.5 A novel method for permutation alignment using beamforming 117
4.6 Experiments 118
4.6.1 Experiment 1 - Single Delay 118
4.6.2 Experiment 2 - Real room recording 122
4.7 Sensitivity Analysis 131
4.7.1 Beamformer's sensitivity to movement 135
4.7.2 Distortion introduced due to movement 137
4.8 Conclusion 139

5 Intelligent Audio Source Separation 142
5.1 Introduction 142
5.2 Instrument Recognition 143
5.2.1 Preprocessing 145
5.2.2 Feature Extraction 145

5.2.3 Instrument Modelling 148
5.2.4 Instrument Recognition 149
5.3 Intelligent ICA 150
5.3.1 Intelligent FastICA 151
5.3.2 Bayesian Approach 153
5.4 Experiments 156
5.4.1 Intelligent FastICA 157
5.4.2 Bayesian Approach 158
5.5 Conclusion 166

6 Conclusions-Future Work **168**
6.1 Summary and Conclusions 168
6.2 Open problems 173
6.2.1 Additive noise 173
6.2.2 Dereverberation 174
6.2.3 More sources than sensors in a convolutive environment 175
6.2.4 Real-time implementation 176
6.2.5 Non-stationary mixing 177

List of Figures

2.1 The general noiseless audio source separation problem. 11
2.2 The instantaneous mixtures source separation problem. . . . 14
2.3 Scatter plot of 2 linearly mixed superGaussian data sets (left), PCA applied to the data sets (right). 17
2.4 Scatter plot of 2 linearly mixed superGaussian data sets (left), ICA applied to the data sets (right). 26
2.5 Scatter plot of 2 linearly mixed subGaussian (uniform) data sets (left), ICA applied to the data sets (right). 27
2.6 Scatter plot of 2 linearly mixed data sets with different distribution (left), ICA applied to the data sets (right). 27
2.7 3 audio sources 2 sensors scenario in the time domain (left) and the sparse MDCT domain (right). 40
2.8 Hyvärinen's clustering algorithm results for the 2 sensors-3 sources scenario. 44
2.9 Zibulevski's clustering approach can be confused when two sources are very closely located. 46
2.10 The real room source separation scenario. 50
2.11 Lee's frequency domain framework: Unmixing in the frequency domain, source modelling in the time domain. 55
2.12 Smaragdis' frequency domain framework: Unmixing and source modelling in the frequency domain. 56

2.13 An illustration of the permutation problem in frequency domain ICA. The arbitrary permutation of the successfully separated components along frequency results in the reconstructed sources remain mixed. 60

3.1 Exploring the statistical properties of short audio segments. Histograms of three different 62.5$msec$ segments in the time domain (a),(b),(c) and the corresponding histograms in the frequency domain (d), (e), (f). 69
3.2 Permutation problem illustrated. Separated sources using the Smaragdis algorithm (left) and the algorithm proposed in section 3.4.1 (right). Permutation inconsistencies are highlighted with arrows. 85
3.3 The four filters modelling the room acoustics created by Westner's *roommix* function. 86
3.4 Comparison of the fast FD-ICA algorithm with the natural gradient approach in the Westner case. We can see the improvement in convergence speed and in separation quality . . 88
3.5 Measuring distortion along frequency for the NG FD-ICA and the fast FD-ICA case. 89
3.6 Filter bank characteristic of a 16-point DFT. 93
3.7 Difference in distortion between the case of 50% overlap and 90% overlap for source 1 at microphone 1 (left plot), and microphone 2 (right plot) 95
3.8 Frame size effect on signal's statistical properties (i.e. estimated kurtosis/sample). 96
3.9 A possible framework to solve the frame size problem. 98

4.1 An array signal processing setup: 3 sensors and 2 sources with Directions of Arrival θ_1 and θ_2 respectively. 102
4.2 Directivity pattern of a three microphone array. 104

4.3 Example of the MuSIC algorithm in a 2 sources-3 sensors scenario. The two sources emitting at 45^o and 60^o. Two directions of arrival estimated by the MuSIC algorithm at $\theta_1 = 45^o$, $\theta_2 = 60^o$. 107
4.4 Directivity pattern along frequency for a single delay case. . . 110
4.5 Directivity pattern along frequency for a real room transfer function. We can spot a main DOA along frequency, however, it seems to be slightly shifted due to the multipath of the real room transfer function. 113
4.6 Average Beampatterns along certain frequency bands for both sources. 116
4.7 A plot of $P(\theta)$ as described in eq. 4.31 gives two distinct DOAs θ_1 and θ_2. 116
4.8 Directivity patterns for the two sources. Permutation problem exists even in the single delay case without any further steps. 120
4.9 The Likelihood Ratio jump solution seems to align the permutations in the single delay case. 121
4.10 Plotting P(θ) (eq. 4.31) using the first 2KHz for the single delay case. Two distinct DOAs are visible. 122
4.11 Permutations aligned using the Directivity Patterns in the single delay case. We can see some problems in the mid-higher frequencies. 123
4.12 Using the MuSIC Directivity Patterns methodology for permutation alignment. The permutation problem is demonstrated here. 124
4.13 Plotting the MuSIC Directivity Patterns for the Likelihood Ratio solution for the single delay case. 125
4.14 Accurate DOA estimates using the MuSIC algorithm. 126
4.15 Accurate permutation alignment using the MuSIC directivity patterns. 127

4.16 Experimental 2 sensor 2 sources setup in a real lecture room. 128
4.17 Directivity patterns for the two sources. Permutation problem exists in the real room case. No steps were taken for the permutation problem, resulting into nothing comprehensible. 129
4.18 The Likelihood Ratio jump solution seems to align most of the permutations. Certain mistakes are visible, especially in the higher frequencies. 130
4.19 Plotting P(θ) (eq. 4.31) using the first 2KHz for the single delay case. Two distinct DOAs are visible for the real room case. 131
4.20 Permutations aligned using the Directivity Patterns in the real room case. We can see good performance in the lower frequencies but some inconsistencies in the mid-higher frequencies. 132
4.21 Using the MuSIC Directivity Patterns methodology for permutation alignment. No steps for the permutation problem are taken. The permutation problem is visible. 133
4.22 Plotting the MuSIC Directivity Patterns for the Likelihood Ratio solution for the real room case 134
4.23 Accurate DOA estimates using the MuSIC algorithm in the real room case. 135
4.24 Permutation alignment using the MuSIC directivity patterns in the real room case. 136
4.25 Comparing beamforming patterns at (a) 160Hz and (b) 750Hz. 137
4.26 Distortion increases as a function of frequency in the case of a misaligned beamformer. 138

4.27 Distortion introduced due to movement.(a) Correct beamforming for right source and correct mapping for left source, (b) left source moves, correct beamforming for right source and incorrect mapping for left source, (c) correct beamforming for left source and correct mapping for right source, (d) left source moves, incorrect beamforming for left source and correct mapping for right source. 140

5.1 A general flow diagram for instrument recognition model training. 144
5.2 MFCC triangular filterbank in the Mel-frequency domain (left) and in the frequency domain (right) 147
5.3 A general flow diagram for performing instrument recognition. 150
5.4 A scatter plot of the two sources, two sensors case. Getting an estimate of the most nonGaussian component can give an estimate of the other component in the prewhitened 2D space. 152
5.5 Fast convergence of the intelligent FastICA scheme. The stars on the unit circle denote the algorithm's iterations. 158
5.6 Plot of $G(w)$ (eq. 5.11), as a function of θ for the two instruments (accordeon (up) and acoustic guitar (bottom)). 160
5.7 Plot of cohort normalised $G(w)$ (eq. 5.16), as a function of θ for the two instruments (accordeon and acoustic guitar). . . . 161
5.8 Plot of $G(w)$ (eq. 5.11), as a function of θ for the two instruments (violin (up) and piano (below)). 162
5.9 Plot of cohort normalised $G(w)$ (eq. 5.16), as a function of θ for the two instruments (violin and piano). 163
5.10 Slow convergence of the numerical optimisation of the cohort normalised likelihood. The stars on the unit circle denote the algorithm's iterations. 163
5.11 Plot of the cohort normalised $G(w)$ (eq. 5.11), as a function of θ_1, θ_2 in the 3×3 case for the acoustic guitar case. 164

5.12 Plot of the cohort normalised $G(w)$ (eq. 5.11), as a function of θ_1, θ_2 in the 3×3 case for the accordeon case. 165
5.13 Plot of the cohort normalised $G(w)$ (eq. 5.11), as a function of θ_1, θ_2 in the 3×3 case for the violin case. 165

List of Tables

3.1 ISNR (dB) measurements for the two versions of the fast FD-ICA framework (after 50 iterations) and the natural gradient algorithm (after 500 iterations). We observe that the two algorithms perform similarly. 90

3.2 Average along frequency Distortion (dB) performance for differing amounts of oversampling. 94

5.1 Inaccuracy of Maximum Likelihood estimation in instrument recognition. We also demonstrate the models' performance in instrument recognition and in presence of additive Gaussian noise and linear mixtures of the three instruments individually with accordeon. All results scaled by 10^3 and v_{acc} represents feature vectors from accordeon samples. 155

Nomenclature

$\gg$	much greater.
$\neq$	not equal to.
$\propto$	proportional to.
$\approx$	approximate to.
$\leftarrow$	substitute.
$\underline{x}$	vector.
$\tilde{x}$	estimate of x.
$\forall$	for all
$\in$	belongs to
$p(x)$	pdf of x.
$p(x\|y)$	conditional pdf of x given y.
$\mathcal{N}(\mu, \sigma^2)$	Gaussian random variable with μ mean and σ^2 variance.
$\mathbb{Z}$	set of integer numbers.
$\mathbb{R}$	set of real numbers.
j	$\sqrt{-1}$.
x^*	complex conjugate of x.
A^{-1}	inverse of a square matrix.
A^T	transpose of a matrix.
A^H	Hermitian (complex-conjugate transpose) of a matrix.
A^+	pseudoinverse of a matrix.
$\underline{x}^T$	transpose of a vector.
$\underline{x}^H$	Hermitian (complex-conjugate transpose) of a vector.
$\det(A)$	determinant of a square matrix.
$\mathcal{E}\{\cdot\}$	expectation.
$\|\|\underline{x}\|\|_1$	L1 norm of a vector.
$\|\|\underline{x}\|\|$	L2 norm of a vector.
$\|\|A\|\|_F$	Frobenius norm of a matrix.

$\text{rank}(A)$	rank of a matrix.
$\text{diag}(d_1, \ldots, d_N)$	diagonal matrix with diagonal elements $d_1, \ldots, d_N$.
$\text{sgn}(\cdot)$	sign operator.
$\text{kurt}(\cdot)$	kurtosis.
$\text{STFT}\{\cdot\}$	Short-time Fourier Transform operator.
$\text{DCT}\{\cdot\}$	Discrete Cosine Transform operator.
$\Re\{\cdot\}$	Real part of a complex number.
$\Im\{\cdot\}$	Imaginary part of a complex number.
$*$	linear convolution.
$\otimes$	circular convolution.
$\mathsf{span}\{A\}$	subspace spanned by the columns of A.
$\lvert \cdot \rvert$	absolute value.
$\max_u J(u)$	maximise $J(u)$ in terms of u.
$\arg\max_u J(u)$	the argument u that maximises $J(u)$.

Chapter 1

Introduction

1.1 What is Audio Source Separation ?

Humans exhibit a remarkable ability to extract a sound object of interest from an auditory scene. The human brain can perform this everyday task in real time using only the information acquired from a pair of sensors, i.e. our ears. Imagine the situation of walking down a busy street with a friend. Our ears capture a huge variety of sound sources: car noise, other people speaking, a friend speaking, mobile phones ringing. However, we can focus and isolate a specific source that is of interest at this point. For example, we may listen to what our friend is saying. Getting bored, we can overhear somebody else's conversation, pay attention to an annoying mobile ringtone or even listen to a passing car's engine, only to understand it is a Porsche. The human brain can automatically focus on and separate a specific source of interest.

Audio source separation can be defined as the problem of decomposing a real world sound mixture (auditory scene) into individual audio objects. The automated analysis using a computer that captures an auditory scene through a number of sensors is the main objective of this thesis. Although this is a relatively simple task for the human auditory system, the automated audio source separation can be considered one of the most challenging topics

in current research.

A number of different methods were proposed to solve the problem.

1.1.1 Computational Auditory Scene Analysis (CASA)

A possible approach to address the problem will be to analyse and finally emulate the way humans perform audio source separation using a computer. *Psychoacoustics* is a special area of research studying how people perceive, process and deduce information from sounds. Such studies construct experimental stimuli consisting of a few simple sounds such as sine tones or noise bursts, and then record human subjects interpretation/perception of these test sounds [Bre99, Ell96]. Audio source separation may be regarded as one aspect of a more general process of auditory organization of these simple structures, which is able to untangle an acoustic mixture in order to retrieve a perceptual description of each constituent sound source [vdKWB01].

Computational Auditory Scene Analysis (CASA) was one of the first methods that tried to "decrypt" the human auditory system in order to perform an automatic audio source separation system [Ell96, Sma01, vdKWB01, BC94]. Conceptually, CASA may be divided into two stages.

In the first stage, the acoustic mixture is decomposed into sensory elements ("segments"). CASA employs either computer vision techniques or complete ear models (outer and middle ear, cochlear filtering etc) in order to segment the auditory scene into several audio elements.

The second stage ("grouping") then combines segments that are likely to have originated from the same sound source [vdKWB01]. Psychological and psychoacoustic research of this kind has uncovered a number of cues or grouping rules which may describe how to group different parts of an audio signal into a single source, such as *i) common spatial origin, ii) common onset characteristics, i.e., energy appearing at different frequencies at the same time, iii) amplitude or frequency modulations in the harmonics of a musical tone, iv) harmonicity or periodicity, v) proximity in time and frequency, vi) continuity (i.e. temporal coherence).* Usually, CASA employs

one or two sensor signals, as the main goal is to emulate humans way of performing auditory scene analysis [Ell96, Sma01].

1.1.2 Beamforming

Array signal processing is a research topic that developed during the late 70s and 80s mainly for telecommunications, radar, sonar and seismic applications. The general array processing problem consists of obtaining and processing the information about a signal environment from the waveforms received at the sensor array (a known constellation of sensors). Commonly, the signal environment consists of a number of emitting sources plus noise.

Exploiting time difference information from the observed signals, one can estimate the number of sources present in the environment, plus the angles of their arrival towards the array sensor [Sch86]. The use of an array allows for a directional beam pattern. The beam pattern can be adapted to null out signals arriving from directions other than the specified look direction. This technique is known as *spatial filtering* or *adaptive beamforming* [FMF92, VK96].

The reception of sound in large rooms, such as conference rooms and auditoria, is typically contaminated by interfering noise sources and reverberation. One can set up an array of microphones and apply the techniques of *adaptive beamforming* in the same way as in telecommunications to perform several audio processing tasks. We can enhance the received amplitude of a desired sound source, while reducing the effects of the interfering signals and reverberation. Moreover, we can estimate the direction or even the position of the sound sources in the near field [HBE01] present in the room (*source localisation*). Most importantly, if the auditory scene contains more than one source, we can isolate one source of interest, whilst suppressing the others, i.e. perform source separation.

Beamforming assumes some prior knowledge on the geometry of the array, i.e. the distance between the sensors and the way they are distributed in the auditory scene. Usually, linear arrays are used to simplify the compu-

tational complexity. In addition, optimally the array should contain more sensors than the sources in the auditory scene. Exploiting the information of the extra sensors using *subspace* methods, we can localise and separate the audio sources.

1.1.3 Blind Source Separation

In contrast to CASA and beamforming, *blind source separation* is a statistical technique that draws inspiration neither from the mechanisms of auditory function nor from the geometry of the auditory scene. Blind source separation systems can identify sound objects, simply by observing the general statistical profile of the audio sources.

By definition, in blind separation there is no available a priori knowledge concerning the exact statistical distributions of the source signals; no available information about the nature of the process by which the source signals were combined (mixing process). In reality, some assumptions must be made regarding the source signal distributions and a model of the mixing process must be adopted. However, these assumptions remain fairly general without undermining the strength of the method.

A special case of blind source separation is *Independent Component Analysis* (ICA), a blind estimation framework that assumes that the sound objects in the scene are *statistically independent*. This assumption together with a relatively general source statistical profile (source prior) can perform audio source separation. To model the mixing procedure, usually FIR filters are employed to describe the room's transfer function between the sources and the sensors.

In the thesis, we are mainly going to focus on this analysis method and more specifically on Independent Component Analysis (ICA). However, as all the aforementioned approaches try to solve essentially the same problem, it might be beneficial to find some links between these methods in order to produce a more complete audio source separation system. In the thesis, we also explore whether blind source separation can incorporate elements from

beamforming.

1.2 Applications of Audio Source Separation

There are many applications where an audio source separation system can be useful:

- *Noise Suppression for mobile phones/hearing aids.* Having unmixed the sources that exist in an auditory scene, one can remove the unwanted noise sources in a multiple source environment. This can serve as a denoising utility for mobile phones, hearing aids or any other recording facility.

- *Music transcription.* Unmixing a recording to the actual instruments that are playing in the recording is an extremely useful tool for all music transcribers. Listening to an instrument playing solo rather the actual recording facilitates the transcription process. This applies to all automated polyphonic transcription algorithms that have appeared in research. Combining a source separation algorithm with a polyphonic transcriber will lead to a very powerful musical analysis tool.

- *Efficient coding of music.* Each instrument has different pitch, attack, timbre characteristics, requiring different bandwidth for transmission. Decomposing a musical signal into sound objects (instruments) will enable different encoding and compression levels for each instrument, depending on its characteristics. The result will be a more efficient, high quality audio codec. This will be more in line with the general framework of *MPEG-4* for video and audio [Aud].

- *Medical applications.* There are medical applications where an audio source separation algorithm might be useful, such as the separation of foetus's heartbeat from the mother's in the womb.

- *Surveillance applications.* The ability of discriminating between the audio objects of an auditory scene will enhance the performance of surveillance applications.

- *Remixing of studio recordings.* In tomorrow's audio applications, with all the powerful tools that can search for songs similar to the ones we like or that sound like the artist we want, a *personal remixing* of a studio recording according to our liking will be possible with audio source separation. In addition, current stereo recordings can be remixed in 5.1 speaker configuration (five satellite speakers and one subwoofer) without using the original masters.

- *Post-processing of film recordings.* Source separation tools will be very useful for editing and effects in the film industry. A source separation algorithm will help post-adjust actors' voice levels in a film take. Dubbing in different languages and any kind of post-processing will also be facilitated.

1.3 Thesis Overview

This thesis is focused on the audio source separation of real world recordings. In this attempt, we address a couple of specific open problems in the field, as it is explained further on. Our solutions are based on a statistical method called *Independent Component Analysis* aiming to decompose linear mixtures of statistical independent signals.

In ***Chapter 2***, we establish the basic background needed for our analysis. We decompose the audio source separation problem in three basic subproblems. First of all, we examine the case of *instantaneous mixtures* of equal number of sources and sensors, where we introduce *Independent Component Analysis* (ICA) as a tool to perform source separation. Subsequently, the problem of *more sources than sensors* is introduced. Current approaches on this underdetermined problem based on ICA are presented. Finally, we define *convolutive mixtures* as a way to model real world recordings and we

examine the way solutions for the two previous subproblems have evolved to address the convolutive mixtures problem.

In ***Chapter 3***, we focus on the *frequency-domain ICA* (FD-ICA) approaches to tackle the convolutive mixtures problem. We analyse the *scale* and *permutation* ambiguity in frequency-domain ICA in full, providing a novel approach to solve these ambiguities, based on *superGaussian source modelling* and a *Likelihood Ratio* correcting mechanism. In addition, a novel fast Frequency Domain framework is introduced, consisting of two fast "fixed-point" algorithms, adapted to work in the frequency domain. The new framework is benchmarked with various real-world recordings. The aliasing between the frequency bands, introduced by the Fourier transform is under investigation, as well as the effect of the frame size used on the estimator's performance.

In ***Chapter 4***, we investigate the idea of using *frequency-domain beamforming* to eliminate the permutation ambiguity. The idea of interpreting FD-ICA as a FD-beamformer is examined. Various limitations on the currently proposed methods are discussed in this chapter, along with novel *Directions of Arrival* mechanisms to help align the permutations for FD-ICA. A preliminary study on the behaviour of source separation algorithm to sources' movement is conducted. A novel method to use subspace methods in the case of equal number of sources and sensors is presented.

In ***Chapter 5***, we explore the novel idea of performing *"intelligent" source separation*, i.e. a selective extraction of a specific audio source from the auditory scene. Previous instrument/speaker modelling techniques are combined with traditional source separation algorithms to tackle this problem. A fundamental limitation of traditional instrument recognisers in the case of source separation is highlighted.

In ***Chapter 6***, we conclude by outlining several of the issues that were posed, analyzed or introduced in this thesis. Emphasis is given to the novel ideas presented throughout the text. In addition, some of the current open problems in the Audio Source Separation framework are presented. Some

possible routes and solutions for future work in the field are also discussed.

1.4 Publications derived from this work

The following publications have arisen from this work.

Journal papers

- Mitianoudis N. and Davies M., "*Audio Source Separation: Problems and Solutions*", International Journal of Adaptive Control and Signal Processing, Volume: 18, Issue: 3, pages: 299-314, April 2004.
- Davies M., Mitianoudis N., "*A simple mixture model for sparse overcomplete ICA*", IEE proceedings in Vision, Image and Signal Processing, Volume: 151, Issue: 1, pages: 35-43, February 2004.
- Mitianoudis N. and Davies M., "*Audio source separation of convolutive mixtures*", IEEE Transactions on Speech and Audio processing, Volume: 11, issue: 5, pages 489-497, September 2003.

Conference papers

- Mitianoudis N. and Davies M., "*Using Beamforming in the Audio Source Separation Problem*", Seventh International Symposium on Signal Processing and its Applications, Paris, France, July 2003.
- Mitianoudis N. and Davies M., "*Intelligent audio source separation using Independent Component Analysis*", Audio Engineering Society Conference, Munich, May 2002.
- Mitianoudis N. and Davies M., "*New fixed-point solutions for convolved mixtures*", 3rd International Conference on Independent Component Analysis and Source Separation, San Diego, California, December 2001.
- Mitianoudis N. and Davies M., "*A fixed point solution for convolved audio source separation*", IEEE workshop on Applications of Signal

Processing on Audio and Acoustics, New Paltz, New York, October 2001.

- Reiss J., Mitianoudis N. and Sandler M., "*A generalised method for the calculation of mutual information in time-series*", Audio Engineering Society Conference, Amsterdam, May 2001

Chapter 2

Blind Source Separation using Independent Component Analysis

2.1 Introduction

Assume there are N sources transmitting the signals $s_1(n), s_2(n), \ldots, s_N(n)$ via a medium (air, cable, network etc), where n defines the discrete time index. At different points of this medium, there are M sensors that capture the signals $x_1(n), x_2(n), \ldots, x_M(n)$, conveyed by the medium.

Source Separation is the process aiming to separate a number of source signals from a set of observations signals.

We introduce the vectors $\underline{s}(n) = [s_1(n)\ s_2(n)\ \ldots\ s_N(n)]^T$ and $\underline{x}(n) = [x_1(n)\ x_2(n)\ \ldots\ x_M(n)]^T$, representing the source signals and the observed signals respectively. We can generally model the mixing procedure, introduced by the medium, with an operator $A[\cdot]$. Assuming there is some additive noise $\underline{\epsilon}(n)$, we can express the signals captured by the microphones as follows:

$$\underline{x}(n) = A[\underline{s}(n)] + \underline{\epsilon}(n) \tag{2.1}$$

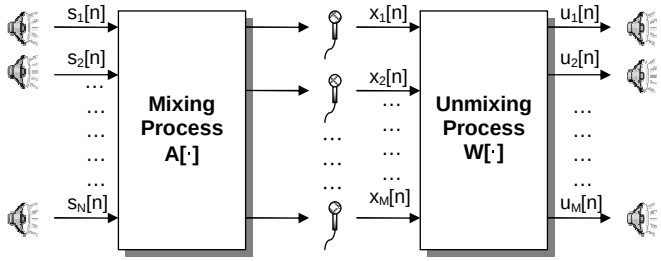

Figure 2.1: The general noiseless audio source separation problem.

Assuming that the system is invertible, we can perform separation by estimating an operator $W[\cdot]$ that can invert the mixing operator $A[\cdot]$.

$$\underline{u}(n) = W[\underline{x}(n)] = W[A[\underline{s}(n)] + \underline{\epsilon}(n)] \approx \underline{s}(n) \tag{2.2}$$

More often, the separation procedure is called *Blind Source Separation (BSS)*. The term *blind* refers to the fact that the method employs only the observed signals to perform separation. No other prior knowledge on the source signals is used. Although this may be considered a drawback, it is in fact the strength of BSS methods, making them a versatile tool for exploiting the spatial diversity provided by an array of sensors [Car98a]. In practice, all BSS methods are *semi-blind*, as some knowledge about the source models is often used. However, these models tend to be quite general, thus preserving the versatility of the method.

BSS methods can be applied to other interesting cases as well. In finance, we can use BSS to find independent factors in financial data [CC01]. In biomedical applications, BSS is sometimes used to remove artifacts from biomedical signals, like EEG [MBJS96], or for analysis. In image processing, BSS can be used to estimate the best independent basis for compression or denoising [HCO99]. However, one very interesting application is the source separation of audio signals.

The cocktail party problem is a real-life illustration of the audio source separation problem, we address in the thesis. The situation of being in a

cocktail party and using our ears to focus on and separate a specific sound source out of all the sound sources present in the room (people talking, background music etc) is defined as the *cocktail party problem*. Research has looked into the way our brain tackles this problem in order to emulate human behaviour on a computer to achieve automatic separation of the sound objects present in the auditory scene.

In order to tackle the audio source separation problem, researchers have divided it to many subproblems. Each subproblem can provide with tools to address the full source separation task. A lot of research has been carried out over the past years in this field. In this thesis, we will look into three of the basic subproblems. First of all, we consider the case of having equal number of sources and sensors in the auditory scene and that the sensors capture weighted versions of each sound source (*instantaneous mixtures*). Then, we look in the case of instantaneous mixtures with fewer sensors than sources (*overcomplete case*). Finally, we explore the case of equal number of sources and sensors but we consider that the sensors capture room reflections as well (*convolutive mixtures*). Other subproblems that can be addressed are dealing with noise and possible dereverb of the sources.

In the following sections, we analyse the basic approaches that were proposed to address the three subproblems mentioned earlier on. This analysis focuses on the particular methods that influenced our approach on source separation later on.

2.2 Instantaneous mixtures

2.2.1 Model formulation

One crude approximation is to assume that the mixing system $A[\cdot]$ in (2.1) is instantaneous, i.e. the microphones capture *instantaneous mixtures* of the audio sources present in the auditory scene (see figure 2.2). Assuming that A is a *mixing* matrix and $\underline{\epsilon}(n)$ models some additive noise, the observed signals $\underline{x}(n)$ can be modeled as follows:

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

$$\begin{bmatrix} x_1(n) \\ x_2(n) \\ \dots \\ x_M(n) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{bmatrix} \begin{bmatrix} s_1(n) \\ s_2(n) \\ \dots \\ s_N(n) \end{bmatrix} + \underline{\epsilon}(n) \tag{2.4}$$

In the cocktail party concept, this assumption implies that each microphone captures a portion of each source. Consequently, each observation is modeled by adding portions of each source. This seems to be a rather simplified model. However, if we are referring to studio recordings, where audio signals are mixed using a mixing desk, the mixed signals can be modelled as summed portions of the original sources. In addition, it is a good starting point for Blind Source Separation algorithms and provides sufficient background for developing more sophisticated models.

For the rest of the analysis in this section, we will assume that we have equal number of microphones and audio sources, i.e. $N = M$. Equally, we have to assume that the mixing matrix A is a *full rank matrix*, as in the opposite case, our problem drops to the *more sources than sensors case* ($N > M$). In addition, we assume that there is *no additive noise in the mixtures*. BSS in the presence of noise is usually addressed as a special separate case of the problem.

2.2.2 Problem Definition

The *Blind Source Separation* problem is concentrated on retrieving the original sources given the observations. In the instantaneous mixtures case, we only have to estimate the unmixing matrix W. We can easily see that if $W = A^{-1}$, we can retrieve the original signals $s(n)$ almost directly.

Given a set of observations $\underline{x}$, estimate the unmixing matrix $W \approx A^{-1}$, that can separate the individual sources present via the linear transform:

$$\underline{u}(n) = W\underline{x}(n) \approx \underline{s}(n) \tag{2.5}$$

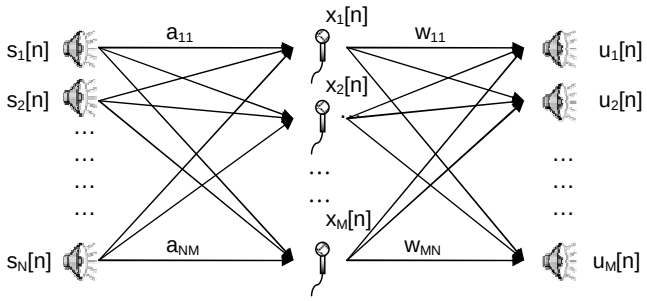

Figure 2.2: The instantaneous mixtures source separation problem.

Usually, our estimate of W should approximate A^{-1}, denoting the quality of our separation. In order to measure the performance of the separation algorithm, we introduce the *performance matrix* P.

$$P = WA \tag{2.6}$$

Ideally, we would expect the matrix P to be close to an identity matrix for an efficient separation algorithm. However, as the separated sources may not come with the same *order* and *scale* as the original sources, the matrix P should ideally be an identity up to a permutation and scale. We will discuss the use of the matrix P for measuring source separation performance later on.

We will now discuss the essentials of two techniques used to perform source separation of instantaneous mixtures : *Principal Component Analysis* (PCA) and *Independent Component Analysis* (ICA). PCA is essentially a decorrelation tool, however, not sufficient to perform source separation. On the other hand, ICA can perform source separation assuming statistical independence of the sound objects.

2.2.3 Principal Component Analysis

Principal Components Analysis (PCA) is a statistical tool used in many applications, such as statistical data analysis, feature extraction and data compression. Its objective is to find a smaller set of variables with less

redundancy that would represent the original signal as accurately as possible [HKO01]. In PCA, the redundancy is measured in terms of *c*orrelation between the observed data series. This will be much more emphasised in the next chapter, where ICA is introduced.

Suppose we have a random vector $\underline{x}$ with N elements and there are T observations of this vector. We are going to apply PCA to transform the signal into uncorrelated components.

For the rest of the thesis, the first analysis step will be to remove possible bias (DC offset from microphones in the audio case) from the observed data. This will simplify the calculation of statistical measures further on.

$$\underline{x} \leftarrow \underline{x} - \mathcal{E}\{\underline{x}\} \tag{2.7}$$

The operator $\mathcal{E}\{\}$ denotes *expectation*. In the thesis, we will use *expectation* for the theoretical analysis. For the practical implementation of the described algorithms, we will substitute the *expectation* with the *sample mean* or the actual expression inside the expectation, depending on the type of learning (*batch* or *stochastic* respectively), as it will be explained later on.

Assume a random variable u_1, which can always be expressed as the linear product of a "weight" vector $\underline{w}_1$ and $\underline{x}$:

$$u_1 = \underline{w}_1^T \underline{x} \tag{2.8}$$

The variable u_1 can be the *first principal component* of $\underline{x}$, only if the variance of u_1 is maximally large. Therefore, we have to estimate the vector $\underline{w}_1$ that maximises the variance of u_1. However, we have to impose the constraint that the norm of $\underline{w}_1$ is always equal to 1, as we are only interested in the orientation of the vector. This will also ensure the stability of the algorithm. The optimisation problem is stated as follows:

$$\max_{\underline{w}_1} J_1(\underline{w}_1), \qquad \text{subject to } ||\underline{w}_1||^2 = \underline{w}_1^T \underline{w}_1 = 1 \tag{2.9}$$

$$\text{where} \qquad J_1(\underline{w}_1) = \mathcal{E}\{u_1^2\} = \underline{w}_1^T \mathcal{E}\{\underline{x}\underline{x}^T\} \underline{w}_1 = \underline{w}_1^T C_x \underline{w}_1 \tag{2.10}$$

The solution to this optimisation problem is given by the *eigenvectors* of the covariance matrix $C_x = \mathcal{E}\{\underline{x}\underline{x}^T\}$. Assume that the eigenvalues of C_x are $d_1, d_2, \ldots, d_N$, where $d_1, d_2, \ldots, d_N > 0$ and $\underline{e}_1, \underline{e}_2, \ldots, \underline{e}_N$ are the eigenvectors of C_x, each corresponding to the same index eigenvalue. The solution maximising (2.9) is :

$$\underline{w}_1 = \underline{e}_1 \tag{2.11}$$

We can generalise the problem of (2.7) to m principal components. However, the constraint that should be added in this case is that each principal component u_m should be uncorrelated with all the previously found principal components. The solutions are the eigenvectors of C_x.

$$\underline{w}_i = \underline{e}_i \qquad \text{for all } i = 1, \ldots, m \tag{2.12}$$

As a result, we have found a linear transform to map our observed signals to m uncorrelated signals (bases) of ascending importance. This decomposition can have many applications. It can be used for compression, as we can keep the most important principal components (bases) of the decomposition and reconstruct the signal using only these.

To calculate the eigenvalues and eigenvectors of the covariance matrix, we use the *Single Value Decomposition* method [MS00]. Multiplying the observations $\underline{x}$ with a matrix containing the eigenvectors of C_x, we transform the observations to a set of *orthogonal* (decorrelated) signals. Multiplying also with a diagonal matrix containing the inverse square root of the corresponding eigenvalues, we transform the observations to a set of *orthonormal* signals (unit variance). This procedure is also known as *prewhitening* or decorrelation of the input data.

PCA is also known as the *Karhunen-Loéve* or *Hotelling* transform. PCA can also be applied in feature extraction, in order to reduce the correlation between the elements of the feature vector. It is also proposed as a pre-processing tool to enhance the performance of *Gaussian Mixture Models* (GMM) [LH99] (see also section 5.2.3). If we use all the principal components, then this procedure is also known as *prewhitening* or decorrelation of

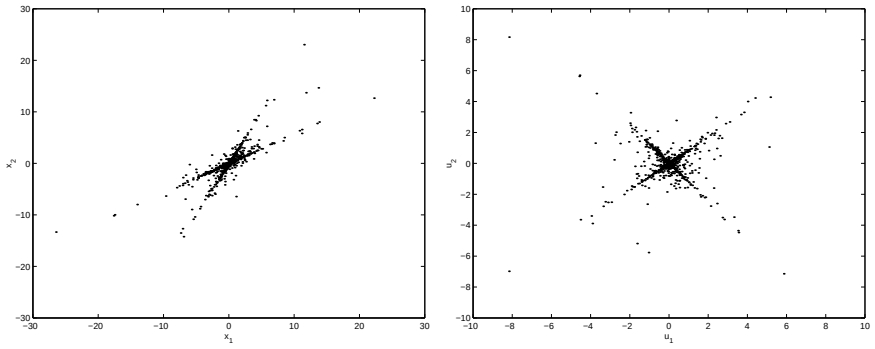

Figure 2.3: Scatter plot of 2 linearly mixed superGaussian data sets (left), PCA applied to the data sets (right).

the input data.

The whole procedure can be summarised, as follows:

1. Calculate the eigenvalues $d_1, d_2, \ldots, d_N$ and the eigenvectors $\underline{e}_1, \underline{e}_2, \ldots, \underline{e}_N$ of the covariance matrix C_x. Ensure that $d_1, d_2, \ldots, d_N > 0$.

2. Form the matrices $V_x = [\underline{e}_1, \underline{e}_2, \ldots, \underline{e}_N]^T$ and $D = diag(d_1, d_2, \ldots, d_N)^{-0.5}$.

3. Apply PCA by

$$\underline{u}_{PCA} = DV_x\underline{x} = V\underline{x} \tag{2.13}$$

In figure 2.3, we can see the *scatter plot* of two observed audio mixtures and the effect of PCA on the mixtures. *Scatter plot* in the 2×2 case is a plot of one observation signal against the other. As we can see after PCA, the principal components are uncorrelated (i.e. they are orthogonal).

However, we can see that PCA did not separate the sources present in the mixtures. The sources would have been separated, if their orientations matched the axis u_1, u_2. This implies that *uncorrelatedness* is not a sufficient criterion for performing source separation.

2.2.4 Independent Component Analysis

Independent Component Analysis (ICA) was firstly introduced as a concept in the early 1980s by J. Herault and C. Jutten without the same

name [AHJ85]. Many researchers around the world worked on BSS and contributed to this field. However, it was not until 1994 that P. Common [Com94] released a paper describing the essentials of this technique and giving its final name. Hitherto, ICA has been applied in many diverse fields, as a tool that can separate linearly mixed independent components.

The general ICA framework

ICA assumes the same instantaneous mixtures model, as described in (2.3). The general *ICA framework* makes the following assumptions:

1. *The source signals $\underline{s}$ are assumed to be statistically independent.* This implies that:

$$p(\underline{s}) = p(s_1, s_2, \ldots, s_N) = p(s_1)p(s_2)\ldots p(s_N) \tag{2.14}$$

2. *At most one of the independent components can have Gaussian statistics.* This is mainly because the mixing matrix A is not identifiable for more than one Gaussian independent components [HKO01, EK03].

For the rest of the analysis in the section, we will assume that A is square and there is no additive noise. The noisy problem and the more sources than sensors case are examined separately as special ICA cases.

Ambiguities in the ICA framework

In addition, there are certain *ambiguities* that characterise all ICA methods.

1. *We cannot determine the order of the independent components.* This is also known as the *permutation ambiguity.* In the instantaneous mixtures case, this is not a great problem, it becomes rather serious in other cases (see section 2.4).

2. *We cannot determine the variances (energies) of the independent components.* This is also known as the *scale ambiguity.* As both A and $\underline{s}$ are unknown, any scalar multiplication on $\underline{s}$ will be lost in the mixing.

The ambiguities of the ICA model can be expressed mathematically as follows:

$$\underline{x} = A\underline{s} = (A\Lambda\Pi)(\Pi^{-1}\Lambda^{-1}\underline{s}) = A_{eq}\underline{s}_{eq} \tag{2.15}$$

where Λ is a diagonal matrix with nonzero diagonal elements, illustrating the *scale ambiguity* and Π is an identity matrix with permuted rows, illustrating the *permutation ambiguity*. As we are only observing $\underline{x}$, our estimates $\underline{u}$ can be unique up to a permutation and scale. Observation signals $\underline{x}$ can always be decomposed into many different A_{eq} and $\underline{s}_{eq}$. However, the possible estimates $\underline{s}_{eq}$ will only be different in scale and permutation. In instantaneous ICA, the ambiguities are not so important, however, we will see that there are some applications, where these ambiguities need to be addressed.

In section 2.2.3, we saw that *prewhitening is actually half ICA*. Prewhitening manages to orthogonalise the sources present in the mixtures, using second-order statistics. However, PCA is not capable of separating the sources, as nonGaussian signals are not identifiable using second-order statistics only. The rotation needed to separate the mixtures is achieved using ICA.

In the next sections, we are going to analyse some of the basic approaches for performing ICA of instantaneous mixtures.

2.2.5 ICA by Maximum Likelihood Estimation

In this part, we will employ *Maximum Likelihood* (ML) estimation to separate the sources present in the instantaneous mixtures [Car97, PP97]. Assuming that $W \approx A^{-1}$ is the unmixing matrix then, we can write:

$$\underline{x} = A\underline{s} \qquad \text{and} \qquad \underline{u} = W\underline{x} \tag{2.16}$$

Following a basic property of linear transformed random vectors

$$p_x(\underline{x}) = |\det(A^{-1})| p_s(\underline{s}) \tag{2.17}$$

Assuming that $p_u(\underline{u}) \approx p_s(\underline{s})$ and statistical independence between the

estimated sources $\underline{u}$, we can write:

$$p_x(\underline{x}) = |\det(W)| p_u(\underline{u}) = |\det(W)| \prod_{i=1}^{N} p_i(u_i) \tag{2.18}$$

Let $W = [\underline{w}_1, \underline{w}_2, \ldots, \underline{w}_N]^T$. Therefore, we can write:

$$p_x(\underline{x}) = |\det(W)| \prod_{i=1}^{N} p_i(\underline{w}_i^T \underline{x}) \tag{2.19}$$

We can present the likelihood of W, as a product of the densities at each observation and optimise the expectation of the log-likelihood. More specifically,

$$L(W) = \prod_{i=1}^{N} p_i(\underline{w}_i^T \underline{x}) |\det(W)| \tag{2.20}$$

$$\mathcal{E}\{\log L(W)\} = \mathcal{E}\{\sum_{i=1}^{N} \log p_i(\underline{w}_i^T \underline{x})\} + \log |\det(W)| \tag{2.21}$$

$$G(W) = \mathcal{E}\{\sum_{i=1}^{N} \log p_i(\underline{w}_i^T \underline{x})\} + \log |\det(W)| \tag{2.22}$$

We will now try to maximise this likelihood expression with respect to W. Using a gradient ascent approach, one can show that:

$$\frac{\partial G(W)}{\partial W} = (W^T)^{-1} + \mathcal{E}\{\phi(W\underline{x})\underline{x}^T\} \tag{2.23}$$

where $\phi(\underline{u}) = [\phi_1(u_1), \ldots, \phi_i(u_i), \ldots, \phi_n(u_n)]^T$ and

$$\phi_i(u_i) = \frac{\partial}{\partial u_i} \log p(u_i) = \frac{1}{p(u_i)} \frac{\partial p(u_i)}{\partial u_i} \tag{2.24}$$

The update rule for ML estimation can then be:

$$W \leftarrow W + \eta \Delta W \tag{2.25}$$

$$\Delta W \propto (W^T)^{-1} + \mathcal{E}\{\phi(W\underline{x})\underline{x}^T\} \tag{2.26}$$

where η is the *learning rate* and ΔW is the update of W.

Amari [ACY96] came to the same result minimising the *Kullback-Leibler* (KL) divergence between the joint and the product of the marginal distributions of the estimates. More importantly, he realised that the parameter space in this optimisation scheme is not Euclidean but has a *Riemannian* metric structure. In such a case, the steepest direction is given by the *natural gradient* instead. The rule tracing the natural gradient is given by multiplying the right-hand side of (2.26) by $W^T W$. This can also be considered an attempt to perform a Newton-type descent by approximating the *Hessian* inverse $(\nabla^2 G)^{-1} \approx W^T W$. The proposed *natural gradient update* algorithm is:

$$\Delta W \propto (I + \mathcal{E}\{\phi(\underline{u})\underline{u}^T\})W \tag{2.27}$$

Activation function choice - Source modelling

The next issue is the choice of the nonlinear function $\phi(\cdot)$. Looking at (2.24), we can see that the activation function is defined by the source signal model of our source signals. There are many possible choices for the *activation function* $\phi(\cdot)$. Hyvärinen [Hyv99d] proposes the following:

1. For *superGaussian sources* (signals with positive kurtosis, e.g. a Laplacian signal):
$$\phi^+(u) = -2\tanh(u) \tag{2.28}$$

2. For *subGaussian sources* (signals with negative kurtosis, e.g. a uniform signal):
$$\phi^-(u) = \tanh(u) - u \tag{2.29}$$

Learning the update

The learning procedure in these update rules can be divided into two categories:

Batch Learning In the learning rules described above, the update requires the calculation of several expectations. In addition, we have a number

of observations of $\underline{x}$ that will be used to train the algorithm. In practice, the expectation is approximated by the sample mean of this function over the observations. This kind of algorithm, where the entire training set is used at every step of the iteration to form the expectation is called *batch learning*. The common batch learning ML-ICA can be summed up, as follows:

1. Assume a training set of N_s vectors $\underline{x}$. For each one, calculate $\underline{u}(n) = W\underline{x}(n)$. Moreover, choose a suitable learning rate η for the data.
2. Calculate $\Delta W = (I + \frac{1}{N_s}\sum_{n=1}^{N_s} \phi(\underline{u}(n))\underline{u}^T(n))W$
3. Update W, i.e. $W \leftarrow W + \eta\Delta W$
4. Repeat steps 2,3 until W convergence.

Online Learning For these update rules, it is necessary to compute the mean values or sample averages of the appropriate functions at each iteration step. This becomes more difficult, as new observation samples keep on coming during the iterations. The statistics of the observation vectors may also be changing and the algorithm should be able to track this. The kind of algorithms, where the whole data set is not used in batch in each iteration, but only the latest observation vector, are called *on-line algorithms*. This implies that the expectation in the learning rule is dropped. The learning rule in (2.27) takes a new form, called *stochastic gradient*, as introduced by Bell and Sejnowski [BS95].

$$\Delta W \propto (I + \phi(\underline{u})\underline{u}^T)W \tag{2.30}$$

This algorithm does not converge deterministically, as it generally tries to follow the gradient.

These training strategies can be used in every learning rule according to the application.

2.2.6 ICA by Entropy Maximisation

Suppose we have a random vector $\underline{s}$ with density $p(\underline{s})$. We can define *differential entropy* as follows [CT91]:

$$H(\underline{s}) = -\int p(\underline{s}) \log p(\underline{s}) d\underline{s} \tag{2.31}$$

A normalised version of entropy is given by *negentropy* J. Negentropy measures the distance of a random variable from the Gaussian distribution of the same covariance. It is defined as follows:

$$J(\underline{s}) = H(\underline{s}_{Gauss}) - H(\underline{s}) \tag{2.32}$$

Another metric from information theory is *mutual information*:

$$I(s_1, s_2, \ldots, s_N) = \sum_{i=1}^{N} H(s_i) - H(\underline{s}) \tag{2.33}$$

Mutual Information can be a good metric of statistical dependence [Com94]. If the random variables $s_1, s_2, \ldots, s_N$ are statistically independent, the Mutual Information is equal to zero.

Bell-Sejnowski method

Bell and Sejnowski [BS95] proved that we can perform Independent Component Analysis by minimising the Mutual Information. Assume that the unmixing matrix is W and $\underline{u} = W\underline{x}$. Using certain properties of differential entropy, we can say that:

$$I(u_1, u_2, \ldots, u_N) = \sum_{i=1}^{N} H(u_i) - H(\underline{x}) - \log|\det(W)| \tag{2.34}$$

The optimisation problem is as follows: We have to estimate the unmixing matrix W that minimises the mutual information in (2.34). In other words, estimate the W that makes separated components more statistically independent. Looking at the definition of differential entropy, we can rewrite it as follows:

$$H(u_i) = -\mathcal{E}\{\log p(u_i)\} \tag{2.35}$$

Now we can rewrite (2.34) as follows:

$$I(u_1, u_2, \ldots, u_N) = -\sum_{i=1}^{N} \mathcal{E}\{\log p(u_i)\} - H(\underline{x}) - \log|\det(W)| \tag{2.36}$$

Assuming that our separated sources are statistically independent, thus they are uncorrelated. Assuming unit variance (can be any constant), we can write that:

$$\mathcal{E}\{\underline{u}\underline{u}^T\} = I \Rightarrow$$

$$W\mathcal{E}\{\underline{x}\underline{x}^T\}W^T = I \Rightarrow$$

$$\det(W)\det(\mathcal{E}\{\underline{x}\underline{x}^T\})\det(W^T) = 1 \tag{2.37}$$

which means that $\det(W)$ must be constant, since $\det(\mathcal{E}\{\underline{x}\underline{x}^T\})$ is not a function of W.

If we compare equation (2.36) and (2.22), we can say that they look really similar, apart from the minus sign and the constant term $H(\underline{x})$. Hence, if we try to minimise (2.36), we will end up with the well-known ML estimation learning rule. Of course, $\partial H(\underline{x})/\partial W = 0$, as $H(\underline{x})$ is not dependent on W.

Starting from a different criterion of independence, we ended up with the same learning rule:

$$\Delta W \propto (W^T)^{-1} + \mathcal{E}\{\phi(W\underline{x})\underline{x}^T\} \tag{2.38}$$

This demonstrates that even though we started from different metrics of statistical independence (mutual information, Kullback-Leibler (KL) divergence, Maximum Likelihood estimation), we conclude to the same update algorithm for the estimation of Independent Components.

2.2.7 ICA by Maximisation of nonGaussianity

Another way to perform ICA is by using another criterion for independence: *nonGaussianity*. It is strange how we can combine nonGaussianity with independence, but we will use the *Central Limit Theorem* to support this [Hyv99d, HKO01, HO97]. Assume that $\underline{x} = A\underline{s}$ and a "weight" vector

$\underline{w}$. The following linear product of $\underline{x}$ and $\underline{w}$ can be one of the independent components, if $\underline{w}^T$ was one of the rows of A^{-1}.

$$u = \underline{w}^T \underline{x} = \underline{q}^T \underline{s} \tag{2.39}$$

As we can see u is a linear combination of the source vectors. The central limit theorem states that the sum of two (or more) independent random variables tends to be more Gaussian than any of the independent component s_i and becomes least Gaussian when it equals one of the s_i. Therefore, if we try to maximise the nonGaussianity of u in terms of $\underline{w}$, we will estimate one of the independent components present in $\underline{x}$.

These algorithms are often *deflationary*. This means that we calculate the first independent component or unmixing vector $\underline{w}$. For the rest, we initiate the learning rule and after every iteration we try to keep the vector orthogonal to the previously estimated vectors $\underline{w}_i$. This is achieved using an orthogonalisation scheme, like *Gram-Schmidt orthogonalisation* [MS00]. Moreover, this implies that data are prewhitened before applying ICA.

There are many ways for measuring nonGaussianity.

Measuring kurtosis

Kurtosis is a fourth order cumulant of a random variable. For a random variable with zero mean, the *normalised kurtosis* is calculated through the formula:

$$kurt(u) = \frac{\mathcal{E}\{u^4\}}{(\mathcal{E}\{u^2\})^2} - 3 \tag{2.40}$$

The basic property of the normalised kurtosis is that for Gaussian random variables, kurtosis is zero. For most nonGaussian random variables, kurtosis is nonzero. As the signals become more *superGaussian*, kurtosis becomes *positive* and increasing in value. In contrast, if the signals become more *subGaussian*, kurtosis becomes *negative* and decreasing in value. For the rest of the analysis, we will refer to the normalised kurtosis as kurtosis.

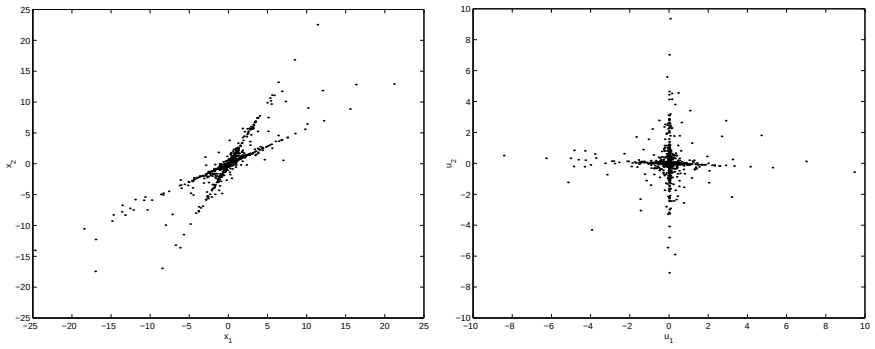

Figure 2.4: Scatter plot of 2 linearly mixed superGaussian data sets (left), ICA applied to the data sets (right).

Hyvärinen introduced a simplified expression for kurtosis. Multiplying (2.40) with the data's squared variance $(\mathcal{E}\{u^2\})^2$ (always positive), we get the following definition (2.41). This expression is easier to optimise, lacking the denominator.

$$kurt(u) = \mathcal{E}\{u^4\} - 3(\mathcal{E}\{u^2\})^2 \tag{2.41}$$

In this approach, we are going to prewhiten the data. This ensures that the sources are uncorrelated and with unit variance, i.e. that the source signals are orthonormal. Then we will have to find the angle of $\underline{w}^T\underline{x}$, where the kurtosis is maximised, i.e. the angle of the most nonGaussian component. Then the orthogonal projection $\underline{w}^T\underline{x}$ will give us the separated component.

Gradient algorithm using kurtosis First of all, the observation signals are prewhitened according to (2.13).

$$\underline{z} = V\underline{x} \tag{2.42}$$

In practice, to maximise the absolute value of kurtosis, we start from a random vector $\underline{w}$ and compute the direction at which the absolute value of the kurtosis of $\underline{w}^T\underline{z}$ is increasing. Maximising the *absolute value of kurtosis* caters for *both superGaussian and subGaussian* signals. Performing gradient

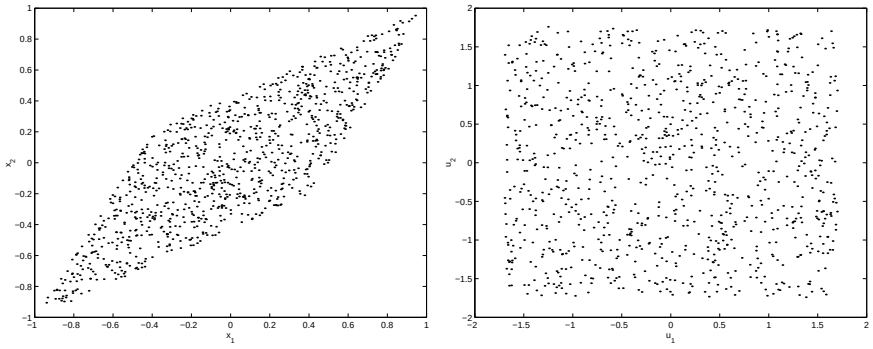

Figure 2.5: Scatter plot of 2 linearly mixed subGaussian (uniform) data sets (left), ICA applied to the data sets (right).

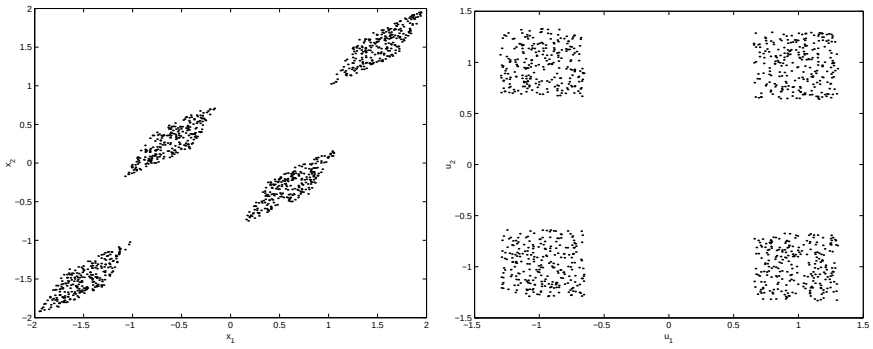

Figure 2.6: Scatter plot of 2 linearly mixed data sets with different distribution (left), ICA applied to the data sets (right).

ascent, under the constraint that $||\underline{w}||^2 = 1$ produces the following:

$$\frac{\partial |kurt(\underline{w}^T \underline{z})|}{\partial \underline{w}} = 4sgn(kurt(\underline{w}^T \underline{z}))[\mathcal{E}\{\underline{z}(\underline{w}^T \underline{z})^3\} - 3\underline{w}||\underline{w}||^2] \tag{2.43}$$

Since we are interested actually only in the direction of the gradient vector, we can obtain the following update:

$$\underline{w} \leftarrow \underline{w} + \eta \Delta \underline{w} \tag{2.44}$$

$$\Delta \underline{w} \propto sgn(kurt(\underline{w}^T \underline{z}))\mathcal{E}\{\underline{z}(\underline{w}^T \underline{z})^3\} \tag{2.45}$$

$$\underline{w} \leftarrow \underline{w}/||\underline{w}|| \tag{2.46}$$

Newton-type algorithm using kurtosis ("Fixed-point" algorithm) To increase speed and robustness, we can develop a Newton-type algorithm for maximising the kurtosis. The derivation of this algorithm is discussed in depth in [HO97]. Using the technique of Lagrange multipliers, one can derive the following fixed-point algorithm

$$\underline{w}^+ \leftarrow \mathcal{E}\{\underline{z}(\underline{w}^T \underline{z})^3\} - 3\underline{w} \tag{2.47}$$

This is registered by Hyvärinen as the "fixed-point algorithm" for ICA [HO97]. The algorithm can be summed up as follows:

Estimate one component

1. Prewhiten data, i.e. $\underline{z} = V\underline{x}$.
2. Begin with a random initial vector $\underline{w}$ that has $||\underline{w}|| = 1$
3. Update $\underline{w}^+ \leftarrow \mathcal{E}\{\underline{z}(\underline{w}^T \underline{z})^3\} - 3\underline{w}$.
4. Normalise $\underline{w}^+ \leftarrow \underline{w}^+/||\underline{w}^+||$
5. Go to step 3, until convergence.

Estimate many components

We can apply the previous algorithm N-times to get all the components that exist in the mixtures. However, we have to ensure that we are looking

for different components each time. Even though we randomly initiate the update rule each time, we may as well fall into the same component. A solution would be to keep the new estimated component, always orthogonal to the previously estimated in the N-dimensional space.

1. Prewhiten data, i.e. $\underline{z} = V\underline{x}$.
2. Begin with a random initial vector $\underline{w}$ that has $||\underline{w}|| = 1$
3. Update $\underline{w}^+ \leftarrow \mathcal{E}\{\underline{z}(\underline{w}^T\underline{z})^3\} - 3\underline{w}$
4. Set $\underline{w}^+ \leftarrow \underline{w}^+ - BB^T\underline{w}^+$
5. Normalise $\underline{w}^+ \leftarrow \underline{w}^+/||\underline{w}^+||$
6. Go to step 3, until $\underline{w}^+$ converges to a value with desired accuracy.

More specifically, B is a projection matrix containing all the vectors $\underline{w}$ calculated for previous components. The transformation in step 4 forces the algorithm to converge to a different component from the ones discovered.

This algorithm is basically much faster than the natural gradient or the Bell-Sejnowski approaches to ICA.

Measuring Negentropy

In section 2.2.6, we defined *negentropy* as the distance of a random variable from the Gaussian distribution. It is evident that it can be used as a measure of nonGaussianity. For a Gaussian random variable, negentropy is zero and nonnegative for all other types of random variables. Negentropy is more justified as a measure of nonGaussianity by statistical theory [HKO01]. One problem is that we can not calculate negentropy directly, but instead we estimate negentropy through approximations. A very good approximation is:

$$J(u) \approx \frac{1}{12}\mathcal{E}\{u^3\}^2 + \frac{1}{48}kurt(u)^2 \tag{2.48}$$

In order to generalise this definition using general higher-order cumulants and not solely kurtosis, we can use a nonquadratic function G and obtain

an approximation of negentropy as follows:

$$J(u) \propto [\mathcal{E}\{G(u)\} - \mathcal{E}\{G(v)\}]^2 \tag{2.49}$$

where v is a Gaussian variable of zero mean and unit variance. Hyvärinen established a fixed-point algorithm of maximising negentropy, called *FastICA* and is analysed in depth in [Hyv99a].

In order to produce a Newton-type ("fixed-point") algorithm, one has to estimate the gradient algorithm. Of course, prewhitening is essential before any processing. The gradient law maximising (2.49) is

$$\Delta\underline{w} \propto \gamma\mathcal{E}\{\underline{z}g(\underline{w}^T\underline{z})\} \tag{2.50}$$

$$\underline{w} \leftarrow \underline{w}/||\underline{w}|| \tag{2.51}$$

where $\gamma = \mathcal{E}\{G(\underline{w}^T\underline{z})\} - \mathcal{E}\{G(v)\}$. In addition, $g(u) = dG(u)/du$. A common choice for this function, amongst others, can be the following:

$$g(u) = \tanh(au), \text{where } 1 \leq a \leq 2 \tag{2.52}$$

To derive the fixed-point algorithm, note that the maxima of the approximation of the negentropy of $\underline{w}^T\underline{z}$ are typically obtained at certain optima of $\mathcal{E}\{G(\underline{w}^T\underline{z})\}$. The optima of $\mathcal{E}\{G(\underline{w}^T\underline{z})\}$, under the constraint that $||\underline{w}||^2 = 1$, are obtained at the point where the gradient of the *Lagrangian* is zero (*Kuhn-Tucker* conditions).

$$F(\underline{z}, \underline{w}) = \mathcal{E}\{\underline{z}g(\underline{w}^T\underline{z})\} + \beta\underline{w} = 0 \tag{2.53}$$

Applying Newton's method to solve the equation, we have:

$$\frac{\partial F}{\partial \underline{w}} = \mathcal{E}\{\underline{z}\underline{z}^T g'(\underline{w}^T\underline{z})\} + \beta I \approx \mathcal{E}\{\underline{z}\underline{z}^T\}\mathcal{E}\{g'(\underline{w}^T\underline{z})\} + \beta I = (\mathcal{E}\{g'(\underline{w}^T\underline{z})\} + \beta)I \tag{2.54}$$

Since the data is prewhitened, $\mathcal{E}\{\underline{z}\underline{z}^T\} = I$. According to Newton's method, the update rule is given by the equation:

$$\underline{w}^+ \leftarrow \underline{w} - [\frac{\partial F}{\partial \underline{w}}]^{-1}F \tag{2.55}$$

After some work on (2.55), we can finally get the following learning rule, which is known as *FastICA*.

$$\underline{w}^+ \leftarrow \mathcal{E}\{\underline{z}g(\underline{w}^T\underline{z})\} - \mathcal{E}\{g'(\underline{w}^T\underline{z})\}\underline{w} \tag{2.56}$$

The whole FastICA algorithm can be summed up as follows:

Estimate one component

1. Begin with a random initial vector $\underline{w}$ that has $||\underline{w}|| = 1$
2. Calculate the covariance matrix C of the observed vectors $\underline{x}$.
3. Update $\underline{w}^+ \leftarrow C^{-1}\mathcal{E}\{\underline{x}g(\underline{w}^T\underline{x})\} - \mathcal{E}\{g'(\underline{w}^T\underline{x})\}\underline{w}$.
4. Normalise $\underline{w}^+ \leftarrow \underline{w}^+/\sqrt{(\underline{w}^+)^T C\underline{w}^+}$
5. Go to step 3, until convergence.

Estimate many components

To estimate all the components, we run the one-unit algorithm N times, keeping the new estimates orthogonal to the previously estimated components.

1. Begin with a random initial vector $\underline{w}$ that has $||\underline{w}|| = 1$
2. Calculate the covariance matrix C of the observed vectors $\underline{x}$.
3. Update $\underline{w}^+ \leftarrow C^{-1}\mathcal{E}\{\underline{x}g(\underline{w}^T\underline{x})\} - \mathcal{E}\{g'(\underline{w}^T\underline{x})\}\underline{w}$.
4. Correct $\underline{w}^+ \leftarrow \underline{w}^+ - \sum_{j=1}^{p} \underline{w}^{+T} C\underline{w}_i\underline{w}_j$
5. Normalise $\underline{w}^+ \leftarrow \underline{w}^+/\sqrt{(\underline{w}^+)^T C\underline{w}^+}$
6. Go to step 3, until $\underline{w}^+$ converges to a value with desired accuracy.

Instead of calculating every independent component separately, sometimes it is more efficient to calculate all components simultaneously. We can use different learning rules (2.56) for all independent components and apply a symmetric decorrelation to prevent the algorithms from converging

to the same components. This can be accomplished by using a symmetric decorrelation:

$$W \leftarrow W(W^T W)^{-1/2} \tag{2.57}$$

where $W = [\underline{w}_1, \underline{w}_2, \ldots, \underline{w}_N]$ is the matrix of the vectors $\underline{w}_i$.

2.2.8 ICA by Tensorial Methods

Assume a zero mean random variable x and the *characteristic function* $\hat{f}(\omega) = \mathcal{E}\{\exp(j\omega x)\}$. We expand the function $\log \hat{f}(\omega)$ to a Taylor series, as follows:

$$\log \hat{f}(\omega) = \kappa_1(j\omega) + \kappa_2(j\omega)^2/2! + \cdots + \kappa_r(j\omega)^r/r! + \ldots \tag{2.58}$$

The coefficients κ_i are called i^{th}*-order cumulants*. In multivariate situations, cumulants are called *cross-cumulants*, similar to cross-covariances. Assume we have the BSS scenario, as introduced in section 2.2.1. Kurtosis of the separated signals can be expressed as a *fourth-order cross-cumulant*. Following some properties, we get:

$$kurt(\sum_i w_i x_i) = cum(\sum_i w_i x_i, \sum_j w_j x_j, \sum_k w_k x_k, \sum_l w_l x_l) \tag{2.59}$$

$$= \sum_{ijkl} w_i^4 w_j^4 w_k^4 w_l^4 cum(x_i, x_j, x_k, x_l) \tag{2.60}$$

The *tensor* is a multi-linear operator defined by the 4^{th} order cumulants, and it is analogous to the covariance matrix for second order moments. The tensor $F = [F_{ij}]$ of a matrix $\Xi = [\xi_{kl}]$.

$$F_{ij} = \sum_{kl} \xi_{kl} cum(x_i, x_j, x_k, x_l) \tag{2.61}$$

As the tensor is a multi-linear operator and due to the symmetry of the cumulant structure, *eigenvalue decomposition* is always possible [Car90]. Assume that Υ is an eigenmatrix and λ the corresponding eigenvalue, we have that the tensor F can be decomposed as follows.

$$F = \lambda \Upsilon \tag{2.62}$$

Assume V is the prewhitening matrix and $\underline{z} = VA\underline{s} = W^T\underline{s}$ are the prewhitened data. After prewhitening, the matrices W and $W^T = VA$ will be orthogonal, where W is the estimated unmixing matrix. Assume that $\underline{w}_m$ is the m^{th} row of W.

One can show that *any matrix in the form* $\Upsilon = \underline{w}_m \underline{w}_m^T$ *can be an eigenmatrix of the following tensor* $F = [F_{ij}]$*, while the corresponding eigenvalues being the kurtoses of the independent components* [Car90, CS93, HKO01].

$$F_{ij} = \sum_{kl} \Upsilon_{kl} cum(z_i, z_j, z_k, z_l) = \sum_{kl} w_{mk} w_{ml} cum(z_i, z_j, z_k, z_l) = \quad (2.63)$$

$$= \cdots = w_{mi} w_{mj} kurt(s_m) \quad (2.64)$$

As a result, if we knew the eigenmatrices of the tensor, we could estimate the rows of the unmixing matrix W, i.e. the independent components. However, if we do not have distinct eigenvalues, then the eigenmatrices are not uniquely defined and consequently the problem is difficult to solve.

Joint Approximate Diagonalisation of Eigenmatrices (JADE)

To overcome this problem, one can view eigenvalue decomposition as diagonalisation. Assuming that the ICA model holds, then the matrix W can diagonalise the tensor F of any matrix Ξ, i.e. the matrix $Q = WFW^T$ is diagonal. This is because F is a linear combination of terms $\underline{w_i w_i}^T$ [CS93]. To approximately diagonalise the matrix Q, we can either minimise the energy of the off-diagonal terms, or maximise the energy of the diagonal terms. Therefore, Cardoso [CS93] proposed to optimise the following cost function:

$$\max_W J_{JADE}(W) = \max_W \sum_i ||diag(WF_iW^T)||^2 \quad (2.65)$$

where F_i denotes the tensor of different matrices Ξ_i. For the choice of Ξ_i, one optimal choice can be the eigenmatrices of the tensor, as they span the same subspace as the tensor, thus retaining all the information about the cumulants. It can be shown that this method is equivalent to minimising

nonlinear correlations (see 2.2.9) [HKO01]. This is the basic principle behind the *JADE* algorithm. JADE can be very slow and computationally expensive with high dimensional data. However, for low dimensional data, it offers a very accurate alternative to the "fixed-point" and natural gradient algorithms.

2.2.9 ICA by Nonlinear Decorrelation

Assume two random variables u_1 and u_2 and two functions $f(u_1)$ and $g(u_2)$, where at least one is nonlinear. We can say that u_1 and u_2 are *nonlinearly decorrelated* [HKO01], if

$$\mathcal{E}\{f(u_1)g(u_2)\} = 0 \tag{2.66}$$

Nonlinear decorrelation can be a criterion for statistical independence. The variables u_1 and u_2 are statistically independent if

$$\mathcal{E}\{f(u_1)g(u_2)\} = \mathcal{E}\{f(u_1)\}\mathcal{E}\{g(u_2)\} = 0 \tag{2.67}$$

for every continuous function f, g that are zero outside a finite interval. We can also show that, in order to satisfy the independence criterion, the functions f, g should be *odd* and u_1, u_2 must have symmetrical probability density functions. In this general framework, we need to address the following: a) how can we choose f, g to satisfy (2.67) and b) how can we nonlinearly decorrelate the variables u_1, u_2. In the next paragraph, we examine two attempts to address these questions.

Hérault-Jutten Algorithm

Assume the 2×2 BSS case of (2.3). Hérault and Jutten [AHJ85] devised the following *feedback* network to unmix the sources.

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} x_1 - m_{12}u_2 \\ x_2 - m_{21}u_1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 0 & m_{12} \\ m_{21} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.68}$$

$$\underline{u} = \underline{x} - M\underline{u} \Rightarrow \underline{u} = (I + M)^{-1}\underline{x} \tag{2.69}$$

Hérault and Jutten adapt m_{12}, m_{21} to reduce nonlinear correlation.

$$\Delta m_{12} = \eta f(u_1)g(u_2) \tag{2.70}$$

$$\Delta m_{21} = \eta f(u_2)g(u_1) \tag{2.71}$$

A common choice for $f(u) = u^3$ and $g(u) = tan^{-1}(u)$. This is a very elegant pioneering solution, however, the inversion is computationally expensive (although $(I + M)^{-1} \approx (I - M)$. In addition, the number of sources has to be small and the global behaviour is not guaranteed.

Cichocki-Unbehauen Algorithm

Based on the previous approach, Cichocki et al [CUMR94] proposed a *feed-forward* network to estimate the unmixing matrix W. The update is given by

$$\Delta W = \eta[\Lambda - f(\underline{u})g(\underline{u}^T)]W \tag{2.72}$$

The matrix Λ is diagonal, whose elements determine the amplitude scaling for the unmixed signals. For the two nonlinear functions, the authors propose the hyperbolic tangent and a polynomial. Moreover, they proved that:

$$\Lambda_{ii} = \mathcal{E}\{f(u_i)g(u_i)\} \tag{2.73}$$

If the algorithm converges to a nonzero unmixing matrix, then the separated sources are nonlinearly decorrelated and hopefully independent. Again, using $f(u) = \tanh(u)$ and $g(u) = u$, we get the natural gradient algorithm, starting from a different perspective.

2.2.10 Performance Evaluation of ICA methods for instantaneous mixtures

Performance metrics

Performance Matrix P One can use the *Performance Matrix P*, described in (2.6), to evaluate ICA algorithms. Observing the performance matrix, one can get the new permutation of the separated sources and also get an estimate of the separation quality. An example follows: Assume we have 3 speakers, linearly mixed, using the following random mixing matrix A. Running the fixed-point algorithm (see 2.2.7), we get the following unmixing matrix W.

$$A = \begin{bmatrix} -0.72 & 0.20 & -0.96 \\ -0.59 & -0.45 & 0.49 \\ -0.60 & -0.60 & -0.10 \end{bmatrix}, W = \begin{bmatrix} 0.62 & -0.76 & 0.13 \\ 0.39 & 0.16 & -0.90 \\ -0.67 & -0.61 & -0.40 \end{bmatrix}$$

$$P = WA = \begin{bmatrix} -0.07 & 0.40 & \underline{-1.00} \\ 0.15 & \underline{0.54} & -0.20 \\ \underline{1.09} & 0.38 & 0.39 \end{bmatrix}$$

Looking at P more closely, we can see that there is a dominant value in every row or column. That corresponds to the separated component. Ideally, the other values in the matrix should be zero. Usually they are not, which implies that the separation is not perfect. The relation between the dominant terms in every row/column with the other can be actually a metric for performance evaluation. Moreover, P shows us the relation between the permutation of the original and separated sources. The position of the greatest term of every row in the matrix denotes the mapping. For example, in the first row (separated source u_1), the greatest term is in column 3. This implies that the third original signal s_3 came out as the first separated signal. Equally, u_2 corresponds to s_2 and u_3 to s_1.

SNR measurement In addition, one can use *Signal-to-Noise Ratio* (SNR) as a separation quality measurement. In other words, we compare the energy

of the original signal with the energy of the difference, using the formula:

$$SNR_{ICA} = 10 \log \frac{\sum_n s^2(n)}{\sum_n (s(n) - u(n))^2} \tag{2.74}$$

Due to the ICA scale ambiguity (amplitude and sign), we must ensure that we compare signals with the same variance and polarity.

Performance Index Moreover, another statistical performance metric was established, exploiting the performance matrix P [HKO01]. As previously mentioned, an ideal matrix P is defined so that on each of its rows and columns, only one of the elements is equal to unity, while all the other elements are zero. Clearly, the following index is minimum for an ideal permutation matrix. The larger the value E is, the poorer the statistical performance for the algorithm.

$$E = \sum_{i=1}^{m} (\sum_{j=1}^{m} \frac{|P_{ij}|}{\max_k |P_{ik}|} - 1) + \sum_{j=1}^{m} (\sum_{i=1}^{m} \frac{|P_{ij}|}{\max_k |P_{kj}|} - 1) \tag{2.75}$$

Separation quality Schobben et al [STS99] discussed the various problems involved with the measurement of BSS methods' performance and proposed a series of performance indexes. However, their indexes require extra calculations, as they actually intervene in the model. The *separation quality* of the j^{th} separated output can be defined as:

$$S_j = 10 \log \frac{\mathcal{E}\{u_{j,s_j}^2\}}{\mathcal{E}\{(\sum_{i \neq j} u_{j,s_i})^2\}} \tag{2.76}$$

with u_{j,s_i} the j^{th} output of the whole mixing-unmixing system when only s_i is active.

Mutual Information measurement Moreover, one could use the *mutual information*, as a measurement of statistical independence and therefore as a performance index, as proposed by Reiss et al [RMS01]. A fast method to estimate the mutual information of time-series was developed to facilitate the calculation.

However, all these metrics can be used only in the case that both the original sources and the mixing matrix are known. Just in the case of SNR, only the original and the separated sources are required.

2.3 More sources than sensors

2.3.1 Problem Definition

In the following analysis, we will assume the *instantaneous mixtures* problem, as introduced in section 2.2.1. However, we will assume that the number of microphones M is less than the number of sources N (*overcomplete case*). Our model can be represented by:

$$\underline{x}(n) = A\underline{s}(n) \tag{2.77}$$

where $A = [\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_N]$ is a $M \times N$ mixing matrix.

In *overcomplete* source separation, however, you can not estimate the unmixed sources using the inverse of the mixing matrix, as in this case A is not square. One can use the *pseudoinverse* of A to get an approximate estimate by $\underline{u}(n) \approx A^{+}\underline{x}(n) = A^{T}(AA^{T})^{-1}\underline{x}(n)$. In literature, however, the pseudoinverse is mainly used to initialise the actual estimation algorithm. As a result, in overcomplete ICA, there are two simultaneous problems, one has to solve:

1. Estimate the mixing matrix A, given an estimate of $\underline{u}(n)$.

2. Estimate the source signals $\underline{u}(n)$, given an estimate of A.

2.3.2 Is source separation possible?

The linear blind source separation problem, in general, has two theoretical issues: the *identifiability* and the *separability* of the problem. *Identifiability* describes the capability of estimating the structure of the linear model up to a scale and permutation and *separability* the capability of retrieving the sources using the estimate of the mixing model. According to Eriksson and

Koivunen [EK03], the "square" linear ICA model ($N = M$) is identifiable if a) all source signals are nonGaussian or b) A is full rank and at most one source is Gaussian.

In the case of overcomplete ICA, it is still possible to identify the mixing matrix from the knowledge of $\underline{x}$ alone, although it is not possible to uniquely recover the sources $\underline{s}$. Although, assuming a probability distribution for $\underline{s}$, one could obtain estimates of the sources, by maximising the likelihood of $p(\underline{x}|A, \underline{s})$. Eriksson and Koivunen [EK03] proved that the general linear ICA model is *unique* up to the following assertions: a) The model is separable, b) all source variables are nonGaussian and $rank(A) = M$ and c) none of the source variables have characteristic function featuring a component in the form $\exp(Q(u))$, where $Q(u)$ is a polynomial of degree at least 2 .

As it is evident from the above analysis, Gaussianity is something that can inhibit the identifiability and separability of the linear ICA model. In the overcomplete case, nonGaussianity (especially superGaussianity) is much more essential to facilitate the source separation task. In the case of audio signals, that will be our main interest, we have certain time-domain statistical profile. Speech signals tend to have a Laplacian distribution, due to the many pauses that exist in the nature of speech. Musical signals tend to have a more Gaussian-like structure that might not affect the ICA algorithm in the square case, however, in the overcomplete case the extra Gaussianity may affect the identifiability of the problem (see figure 2.7(left)). The solution for signals with such statistics for overcomplete ICA is to use a linear, sparse, superGaussian, orthogonal transform $T_{\text{sparse}}\{\cdot\}$. A sparse transform linearly maps the signal to a domain where most of the values are very small, i.e. concentrates the energy of the signals to certain areas. This sparse transform should also be linear. As a result, the mixing matrix A remains unchanged by the signal transformation.

$$\underline{x} = A\underline{s} \longleftrightarrow T_{\text{sparse}}\{\underline{x}\} = AT_{\text{sparse}}\{\underline{s}\} \tag{2.78}$$

where $T_{\text{sparse}}\{\underline{x}\} = [T_{\text{sparse}}\{x_1\} \dots T_{\text{sparse}}\{x_M\}]^T$.

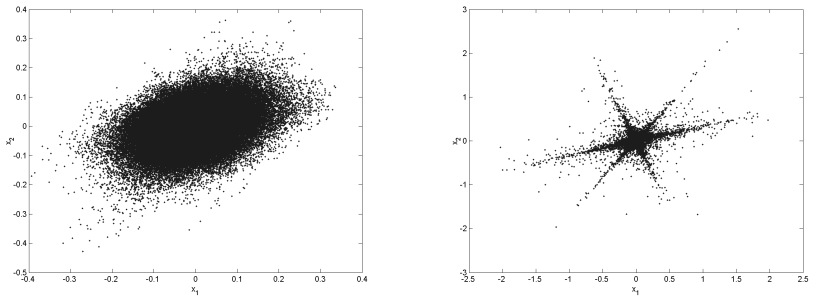

Figure 2.7: 3 audio sources 2 sensors scenario in the time domain (left) and the sparse MDCT domain (right).

It is clear that the estimation of A in the transform domain is equivalent to the estimation in the time-domain, however, with sparser statistics. If the transform is invertible, one can perform the estimation of $\underline{u}$ in the transform domain, otherwise the estimation has to be performed in the time-domain, given the estimate of A.

There are many candidate transforms for this task. The Fourier transform is a sparse, linear, orthogonal transform, however, it is not preferred due to the complex outputs. The *Discrete Cosine Transform* (DCT) is an even sparser, linear, orthogonal transform and can be much more preferable to the Fourier transform as it is real. Using the *Modified DCT* (MDCT) [DS03], a transform that is applied on shorter frames to account for stationarity, can enhance sparsity. In figure 2.7(right), we can see a mixture in the MDCT domain. Sparsity facilitates the estimation of A, as now the orientation of the components is visible. Another candidate can be the *Wavelet* transform, as proposed by Zibulevsky et al [ZKZP02]. Using the sparsest subset of the wavelet decomposition, one can estimate the mixing matrix in a sparse environment, assuming that the sources are all active in that subset. We should note that the choice of sparse transform is clearly signal class dependent.

In our analysis, we will use the MDCT transform as a sparse transform, unless otherwise stated. Next, we will look at methods that try to tackle

the two subproblems of overcomplete ICA.

2.3.3 Estimating the sources given the mixing matrix

This is a problem that does not exist when $M = N$, as you can invert the matrix and get accurate estimates of your sources. In the $M \geq N$ case, the *pseudoinverse* can give accurate estimates of the sources. However, in the overcomplete case, the estimates one can get from the *pseudoinverse* are not accurate. Therefore, we have to resort to other methods to solve the problem.

ML estimation

One solution is to use *Maximum Likelihood* (ML) or *Maximum A Posteriori* (MAP) estimation to retrieve our sources, given the mixing matrix A. Imposing a source model, our sources can be retrieved by:

$$\underline{u} = \arg\max_{\underline{u}} P(\underline{u}|\underline{x}, A) = \arg\max_{\underline{u}} p_u(\underline{u}) P(\underline{x}|A, \underline{u}) P(\underline{u}) \tag{2.79}$$

Therefore, in the noiseless case the sources can be retrieved by

$$\Delta\underline{u} \propto -\partial \log P(\underline{u})/\partial\underline{u} \tag{2.80}$$

However, this gradient based algorithm is not very fast.

Linear Programming

As explained earlier on, usually we employ sparse linear transforms to enhance the quality of separation. Therefore, a Laplacian model for the sources $p(u) \propto \exp^{-|u|}$ can be applied. A good starting point for the algorithm can always be the pseudoinverse solution. However, as we are dealing with very sparse sources, we can initialise the algorithm with zero signals (very sparse source) that might be closer to the model than the pseudoinverse, or any other random initialisation. Lewicki [LS98] proved that source estimation,

assuming Laplacian priors, can be reduced to minimising the $L1$-norm of the estimated sources.

$$\min_{\underline{u}} ||\underline{u}||_1 = \min_{u_i} \sum_i |u_i| = \min_{\underline{u}} [1\ 1 \dots 1]|\underline{u}| \tag{2.81}$$

$$\text{subject to } \underline{x} = A\underline{u}$$

This can be transformed and solved as a linear programming problem. However, solving a linear programming problem for every time sample can be quite computationally expensive and very slow. This can be quite important when you are updating the mixing matrix as well, and you want to find an estimate for the sources, for each estimate of A. In that case, we aim for a solution that can be fast and accurate.

Simplified $L1$-norm minimisation

In order to reduce the computation load of $L1$-norm minimisation (linear programming), it is equivalent to solve the problem as follows: *Assume that only M sources at maximum can be active at each time sample.* Now, we only have to find which of the N sources are more likely to be active in each time slot. As a measure of likelihood for sparse sources, we will use the $L1$-norm $||\underline{u}||_1 = \sum_i |u_i(n)|$. For example, for the case of 2 microphones and 3 sources, assuming that $A = [\underline{a}_1\ \underline{a}_2\ \underline{a}_3]$, we will have:

$$\begin{aligned}
\tilde{\underline{u}}_1(n) &= [\underline{a}_1\ \underline{a}_2]^{-1}\underline{x}(n) \\
\tilde{\underline{u}}_2(n) &= [\underline{a}_2\ \underline{a}_3]^{-1}\underline{x}(n) \\
\tilde{\underline{u}}_3(n) &= [\underline{a}_1\ \underline{a}_3]^{-1}\underline{x}(n)
\end{aligned} \tag{2.82}$$

Then form

$$\text{Ł}_i(n) = ||\tilde{\underline{u}}_i(n)||_1 \tag{2.83}$$

Then, the ML solution would be the one that $\min_i L_i(n)$. Then, you reconstruct the separated sources for each time slot, by using the corresponding $\underline{u}_i(n)$ and pad the other $M - N$ sources with zeros.

This scheme is less computationally expensive than linear programming for small number of sources and sensors. As the number of sources and sensors increases, finding all possible combinations of active sources becomes rather complicated and a proper linear programming solution might be more appropriate in this case.

2.3.4 Estimating the mixing matrix given the sources

Clustering Approaches

Hyvärinen's Approach Hyvärinen [Hyv98] in his analysis shows that maximising the $\log p(A, \underline{s})$ is not an approximation but equivalent to the log-likelihood that Lewicki tries to maximise in [LS98].

Moreover, Hyvärinen forms a very efficient *clustering algorithm* for superGaussian components. In order to perform separation, he assumes that the sources are very sparse. Therefore, for sparse data you can claim that at most only one component is active at each sample. In other words, we attribute each point of the scatter plot to *one source only.* This is a *competitive winner-take-all mechanism.*

1. Initialise $A = [\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_N]$.
2. Collect the points that are close to the directions represented by $\underline{a}_i$. For all $\underline{a}_i$ find the set of points S_i of $\underline{x}$ that
$$|\underline{a}_i^T \underline{x}(n)| \geq |\underline{a}_j^T \underline{x}(n)|, \ \forall \ j \neq i \tag{2.84}$$
3. Update
$$\underline{a}_i \leftarrow \sum_{n \varepsilon S_i} \underline{x}(n)(\underline{a}_i^T \underline{x}(n)) \tag{2.85}$$
$$\underline{a}_i \leftarrow \underline{a}_i / ||\underline{a}_i||, \ \forall \ i = 1, \ldots, N \tag{2.86}$$
4. Repeat 2,3 until convergence.

As we can see, this is a clustering approach, as we force the direction of the mixing matrix to align along the concentration of the points in the scatter plot.

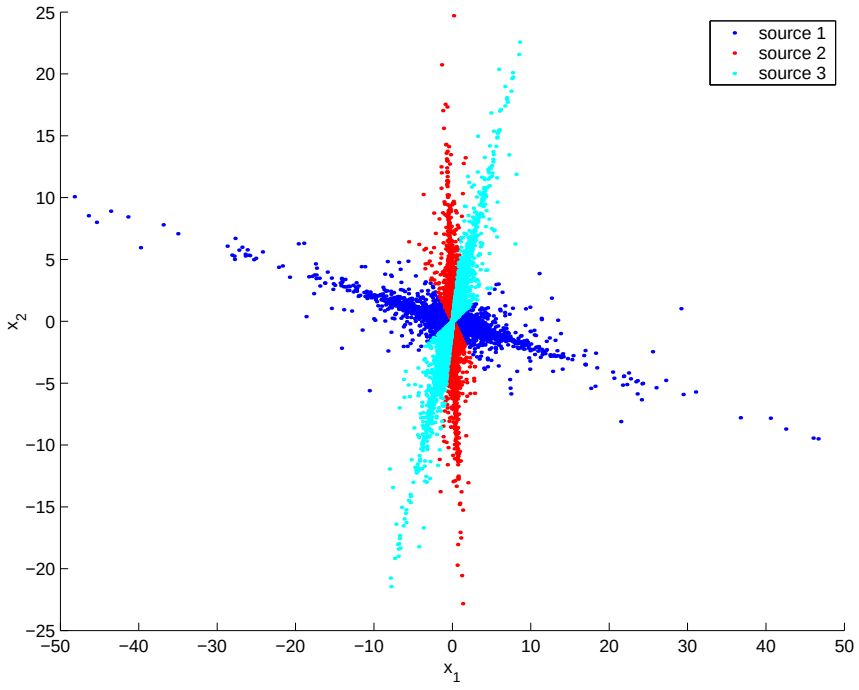

Figure 2.8: Hyvärinen's clustering algorithm results for the 2 sensors-3 sources scenario.

To estimate the sources in this case, all we have to do is construct the vectors $\underline{x}_{Si}(t)$ that contain all the vectors from $\underline{x}(t)$ corresponding to each S_i. Then, the estimates are given by:

$$u_i = \underline{a}_i^T \underline{x}_{Si} \tag{2.87}$$

Zibulevsky's Approach Zibulevsky et al [ZKZP02] proposed another clustering solution for overcomplete source separation. As discussed earlier, the use of a linear sparse transform is required to enhance the performance of overcomplete ICA. One could use the very sparse *Modified Discrete Cosine Transform* (MDCT). Zibulevsky proposes the use of the sparsest subset of the *wavelet decomposition*. His approach

1. Assume a sparse transform $T_{\text{sparse}}\{x\}$ and

$$\underline{z} = T_{\text{sparse}}\{\underline{x}\} \tag{2.88}$$

2. Normalise vectors to unit sphere (M-dimensional sphere)

$$\underline{z} \leftarrow \underline{z}/||\underline{z}|| \tag{2.89}$$

   A useful hint is to remove data points with $||\underline{z}|| \approx 0$.

3. Map all the points to the half unit sphere, by taking the absolute value of the first element of the vector $\underline{z}$:

$$z(1) \leftarrow |z(1)| \tag{2.90}$$

4. Use a clustering algorithm (K-means, Fuzzy C-means) to find the center of clusters formed on the unit half-sphere. The centers of the clusters will give you approximately the columns of A.

5. Estimate sources using linear programming or the simplified linear programming, as explained earlier on. We can even use other clustering algorithms.

The drawback of this method is that it is not accurate enough, as by projecting the data point to the unit-sphere, we are losing information. For example, if two sources are located very closely, even if they are very sparse, the projection to the unit sphere might create a single cluster instead of two separate clusters. An example of the 2D case can be seen in figure 2.9, where although we can visually separate the two sparse signals from their scatter plot, the projection on the unit circle forms one cluster.

Bayesian Approaches

Maximising joint likelihood In [LS98], Lewicki formed a Bayesian approach to overcomplete ICA. He also explored the general case with additive noise ϵ.

$$\underline{x} = A\underline{s} + \underline{\epsilon} \tag{2.91}$$

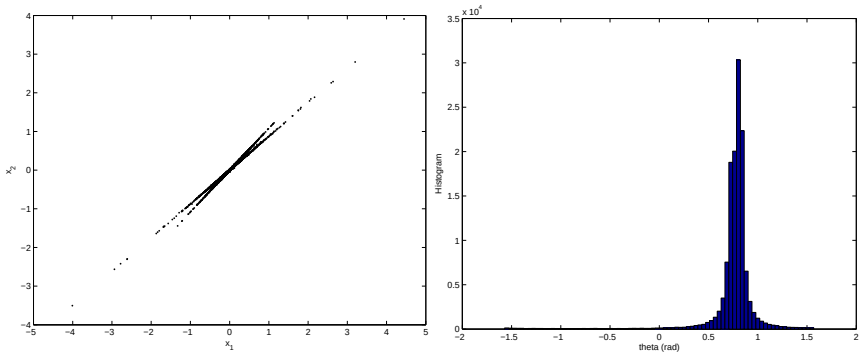

Figure 2.9: Zibulevski's clustering approach can be confused when two sources are very closely located.

Assuming that the noise is Gaussian and isotropic with covariance matrix $C_\epsilon = \sigma_\epsilon^2 I$, one can write down that:

$$\log p(\underline{x}|A, \underline{s}) \propto -\frac{1}{2\sigma_\epsilon^2}(\underline{x} - A\underline{s})^2 \tag{2.92}$$

Now, we have to deal with two problems, as stated before: a) estimate A, b) estimate $\underline{u}$. We have discussed so far various methods for getting an estimate of the sources, given an estimate of A. Now, Lewicki explored a way to get an estimate of A, given an estimate of the sources. Thus, Lewicki thought of maximising the following:

$$\max_A p(\underline{x}|A) = \max_A \int p(\underline{u})p(\underline{x}|A, \underline{u})d\underline{u} \tag{2.93}$$

After approximating $p(\underline{x}|A)$ with a Gaussian around $\underline{u}$ and a mathematical analysis, Lewicki derives a gradient algorithm that resembles the natural gradient.

$$\Delta A \propto -A(\phi(\underline{u})\underline{u}^T + I) \tag{2.94}$$

where $\phi(u)$ represents the activation function. Assuming sparse priors, Lewicki proposed $\phi(u) = \tanh(u)$. Lewicki claims that this approach can work for sources captured in the time-domain, however, it is bound to have better performance in a sparser domain, as analysed earlier. The algorithm can be summarised as follows:

1. Randomly initialise A.
2. Initialise source estimates $\underline{u}$ either with the pseudoinverse or with zero signals.
3. Given the estimated $\underline{u}$, get a new estimate for A.
$$A \leftarrow A - \eta A(\phi(\underline{u})\underline{u}^T + I) \tag{2.95}$$
where η is the learning rate.
4. Given the new estimate for A, find a new estimate for $\underline{u}$ either by solving the linear programming problem for every sample n, or the simplified linear programming, as explained earlier on.
5. Repeat steps 3,4 until convergence.

As this is a gradient algorithm, its convergence depends highly on the choice of learning rate and on signal scaling. This two-step method demonstrated slow convergence in our simulations.

Mixtures of Gaussians - Attias' approach Attias [Att99] proposed to model the sources as a *Mixture of Gaussian* (MoG) and used an *Expectation-Maximisation* (EM) algorithm to estimate the parameters of the model. A MoG is defined as:

$$p(s_i) = \sum_{k=1}^{K} \pi_{ik} \mathcal{N}_{s_i}(\mu_{ik}, \sigma_{ik}^2) \tag{2.96}$$

where K defines the number of Gaussians used, μ_{ik} and σ_{ik} denote the mean and standard deviation of the k^{th} Gaussian and $\pi_{ik} \in [0, 1]$ the weight of each Gaussian. Always, $\sum_{k=1}^{K} \pi_{ik} = 1$. To model the joint density function $p(\underline{s})$, we issue a vector $\underline{q}(t) = [q_1(t), q_2(t), \ldots, q_N(t)]$. Each $q_k(t)$ can take a discrete value from 1 to K and represents the state of the mixture of the k^{th} source at time t. The joint density function $p(\underline{s})$ is itself a MoG in the following form:

$$p(\underline{s}) = \prod_{i=1}^{N} p(s_i) = \sum_{q_1} \cdots \sum_{q_N} \pi_{1,q_1} \ldots \pi_{L,q_N} \prod_{i=1}^{N} \mathcal{N}_{s_i}(\mu_{i,q_i}, \sigma_{i,q_i}^2) \tag{2.97}$$

Assuming additive Gaussian noise of zero mean and covariance J, one can exploit the Gaussian structure to express $p(\underline{x}|A)$. Attias shows that

$$p(\underline{x}|A,J) = \sum_{q_1=1}^{K} \cdots \sum_{q_N=1}^{K} \pi_{1,q_1} \ldots \pi_{N,q_N} \times \ldots \tag{2.98}$$

$$\times \mathcal{N}_x(\underline{a}_1\mu_{1,q_1} + \cdots + \underline{a}_N\mu_{N,q_N}, J + \underline{a}_1\underline{a}_1^T\sigma_{1,q_1}^2 + \cdots + \underline{a}_N\underline{a}_N^T\sigma_{N,q_N}^2)$$

where $A = [\underline{a}_1 \ldots \underline{a}_N]$. In order to estimate the parameters of this model $\mu_{i,q_i}, \sigma_{i,q_i}, \pi_{i,q_i}, A, J$, Attias chose to minimise the Kullback-Leibler distance between the model sensor density $p(\underline{x}|A,J)$ and the observed one $p^o(\underline{x})$. He developed an *Expectation - Maximisation* (EM) algorithm to train the parameters of the model. Again, the whole training procedure is divided into two steps that are repeated for each iteration: a) Adapt the parameters of the model, b) estimate the sources.

Adapt the model

$$A = \mathcal{E}\{\underline{x}\underline{u}^T\}(\mathcal{E}\{\underline{x}\underline{x}^T\})^{-1} \tag{2.99}$$

$$J = \mathcal{E}\{\underline{x}\underline{x}^T\} - \mathcal{E}\{\underline{x}\underline{u}^T\}A^T \tag{2.100}$$

$$\mu_{i,q_i} = \frac{\mathcal{E}\{p(q_i|u_i)u_i\}}{\mathcal{E}\{p(q_i|u_i)\}} \tag{2.101}$$

$$\sigma_{i,q_i}^2 = \frac{\mathcal{E}\{p(q_i|u_i)u_i^2\}}{\mathcal{E}\{p(q_i|u_i)\}} - \mu_{i,q_i}^2 \tag{2.102}$$

$$\pi_{i,q_i} = \mathcal{E}\{p(q_i|u_i)\} \tag{2.103}$$

$$p(q_i|u_i) = \frac{\pi_{i,q_i}p(u_i)}{\sum_{j=1}^{N}\pi_{j,q_j}p(u_j)} \tag{2.104}$$

Estimate the sources

Attias proposed a MAP-estimator, maximising the source posterior $p(\underline{u}|\underline{x})$. More specifically,

$$\underline{u} = \arg\max_{\underline{u}} \log p(\underline{x}|\underline{u}) + \sum_{i=1}^{N} \log p(u_i) \Rightarrow \tag{2.105}$$

$$\Delta\underline{u} = \eta A^T J^{-1}(\underline{x} + A\underline{u}) - \eta\phi(\underline{u}) \tag{2.106}$$

where η is the learning rate and $\phi(u) = \partial \log p(u)/\partial u$, incorporating the source model.

All the Bayesian approaches tend to give complete and more general solutions. However, they tend to be very slow in convergence, compared to the clustering approaches.

2.4 Convolutive mixtures

2.4.1 Problem Definition

In the previous sections, we have mentioned a lot of methods based on the ICA framework that can perform high-quality separation of linearly mixed sources. However, if we try to apply these techniques on observation signals acquired from microphones in a real room environment, we will see that all actually fail to separate the audio sources. The main reason is that the instantaneous mixtures model does not hold in the real room scenario.

Looking at figure 2.10, we can see that in a real recording environment sensors (microphones) record delayed attenuated versions of the source signals, apart from direct path signals. This is mainly due to reflections on the surfaces inside the room (multipath signals). In this sense, the observation signals can be more accurately modelled as:

$$\begin{aligned} x_1(n) &= a_{11}(1)s_{11}(n - T_{11}^1) + \cdots + a_{11}(K_1)s_{11}(n - T_{11}^{K1}) + \\ &\quad + a_{12}(1)s_{12}(n - T_{12}^1) + \cdots + a_{12}(K_2)s_{12}(n - T_{12}^{K2}) \\ x_2(n) &= a_{21}(1)s_{21}(n - T_{21}^1) + \cdots + a_{21}(K_3)s_{21}(n - T_{21}^{K3}) + \\ &\quad + a_{22}(1)s_{22}(n - T_{22}) + \cdots + a_{22}(K_4)s_{22}(n - T_{22}^{K4}) \end{aligned} \tag{2.107}$$

where T_{ij}^k model the k^{th} time delay from the j^{th} source, as observed by the i^{th} microphone. In addition, the coefficients $a_{ij}(k)$ model the room transfer function between the j^{th} source and the i^{th} microphone. Subsequently, we can generalise for the M microphones - N sources case. Assuming the maximum delay of all transfer functions is K, we can write that

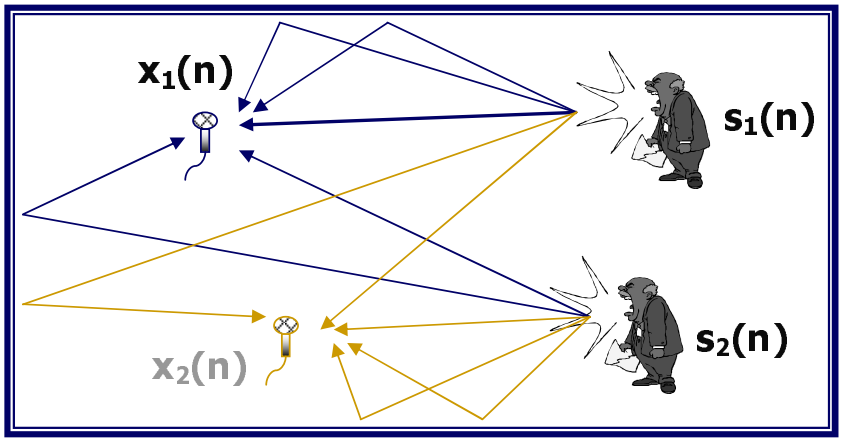

Figure 2.10: The real room source separation scenario.

$$\begin{aligned}
x_1(n) &= \sum_{k=1}^{K} a_{11}(k)s_{11}(n-k) + \cdots + \sum_{k=1}^{K} a_{1N}(k)s_{1N}(n-k) \\
x_2(n) &= \sum_{k=1}^{K} a_{21}(k)s_{21}(n-k) + \cdots + \sum_{k=1}^{K} a_{2N}(k)s_{2N}(n-k) \\
&\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\
x_M(n) &= \sum_{k=1}^{K} a_{M1}(k)s_{M1}(n-k) + \cdots + \sum_{k=1}^{K} a_{MN}(k)s_{MN}(n-k)
\end{aligned} \tag{2.108}$$

Equivalently, we can write

$$\begin{aligned}
x_1(n) &= \underline{a}_{11} * s_1(n) + \quad \cdots \quad + \underline{a}_{1N} * s_N(n) \\
&\cdots \qquad\qquad \cdots \qquad\qquad \cdots \\
x_M(n) &= \underline{a}_{M1} * s_1(n) + \quad \cdots \quad + \underline{a}_{MN} * s_N(n)
\end{aligned} \tag{2.109}$$

$$\begin{bmatrix} x_1(n) \\ x_2(n) \\ \cdots \\ x_M(n) \end{bmatrix} = \begin{bmatrix} \underline{a}_{11} & \cdots & \underline{a}_{1N} \\ \underline{a}_{21} & \cdots & \underline{a}_{2N} \\ \cdots & \cdots & \cdots \\ \underline{a}_{M1} & \cdots & \underline{a}_{MN} \end{bmatrix} * \begin{bmatrix} s_1(n) \\ s_2(n) \\ \cdots \\ s_N(n) \end{bmatrix} \tag{2.110}$$

$$\underline{x}(n) = \begin{bmatrix} \underline{a}_{11} & \cdots & \underline{a}_{1N} \\ \underline{a}_{21} & \cdots & \underline{a}_{2N} \\ \cdots & \cdots & \cdots \\ \underline{a}_{M1} & \cdots & \underline{a}_{MN} \end{bmatrix} * \underline{s}(n) \tag{2.111}$$

The above equation describes the observation signals in the real room case. These mixtures are often referred to as *convolutive mixtures*. In our case and in most ICA applications, the room transfer function $\underline{a}_{ij}$ is usually modelled by a *high-order FIR* filter. To increase accuracy, we could use *lower-order IIR* filters to model room acoustics. However, as IIR filters are less stable and require minimum-phase mixing, we will model the channel using FIR filters [Chr92]. The length of an average room transfer function is usually $> 250msec$, depending on the actual room size and positions of the sources/sensors in the room [Sma97].

The problem we are called to solve in the real room case is how we can unmix the convolutive mixtures using the general ICA framework, as described in 2.2.4. Assuming FIR mixing procedures, we will look for FIR unmixing solutions as well. As a result, we want to estimate FIR filters $\underline{w}_{ij}$ that can unmix the sources.

$$\underline{u}(n) = \begin{bmatrix} \underline{w}_{11} & \cdots & \underline{w}_{1N} \\ \underline{w}_{21} & \cdots & \underline{w}_{2N} \\ \cdots & \cdots & \cdots \\ \underline{w}_{M1} & \cdots & \underline{w}_{MN} \end{bmatrix} * \underline{x}(n) \tag{2.112}$$

In our analysis, we will always assume equal number of microphones and sensors for the convolutive case, i.e. $N = M$. Again, we will assume no additive noise in our model.

2.4.2 Time-Domain Methods

A typical time domain method tries to estimate the unmixing coefficients using the signals in the time domain. An equivalent form of the convolutive mixtures model in (2.112) is :

$$x_i(n) = \sum_{j=1}^{N} \sum_{k=1}^{K} a_{ijk} s_j(n-k) \qquad \forall i = 1, \ldots, N \tag{2.113}$$

We can separate the mixtures, by estimating unmixing filter $\underline{w}_{ij}$, following a *feedforward* or equally an *FIR filter* architecture, as expressed by the

following equation.

$$u_i(n) = \sum_{j=1}^{N} \sum_{k=1}^{K} w_{ijk} x_j(n-k) \qquad \forall i = 1, \ldots, N \tag{2.114}$$

Torkkola [Tor96] proposed a *feedback* architecture to solve the delay-compensation problem. He also generalised the feedback architecture to remove temporal dependencies, stabilising the cross-weights. Lee [LBL97] proposed the following IIR separation structure, assuming that this structure can only invert *minimum-phase acoustic environments* (all zeros of the mixing system and consequently all poles of the unmixing system are inside the unit circle).

$$u_i(n) = x_i(n) - \sum_{j=1}^{N} \sum_{k=0}^{L} w_{jk} u_j(n-k) \qquad \forall i = 1, \ldots, N \tag{2.115}$$

or equivalently

$$\underline{u}(n) = \underline{x}(n) - W_0 \underline{u}(n) - \sum_{k=1}^{L} W_k \underline{u}(n-k) \tag{2.116}$$

The learning procedure, i.e. the estimation of W, is performed by maximising the joint entropy $H(g(u))$, where $g(\cdot)$ is a sigmoid function. In a similar sense to Bell-Sejnowski's rule, taking into account Amari's natural gradient approach, Lee proposes the following learning rule:

$$\Delta W_0 \propto -(I + W_0)(I + \mathcal{E}\{\phi(\underline{u})\underline{u}^T\}) \tag{2.117}$$

$$\Delta W_k \propto -(I + W_k)\mathcal{E}\{\phi(\underline{u})\underline{u}^T(n-k)\}, \ \forall \ k = 1, \ldots, L \tag{2.118}$$

where $\phi(u) = -\partial \log p(u)/\partial u$. All these updates are performed in the time-domain.

There are certain *drawbacks* in using time-domain methods in the source separation context. From adaptive filter theory [Hay96], we know that time domain algorithms are very efficient for small mixing filters (communication channels etc), however they can be computationally expensive for long transfer functions, such as a room transfer function. The solution of using

smaller IIR filter, instead of long FIR filters, will always be prone to numerical instability and the inability to invert non-minimum phase filters [Sma97]. In addition, the problem of *spectral whitening* introduced by a feedforward architecture, was observed and solved by Torkkola [Tor96] using a feedback architecture, however, it showed there are interdeterminacies in the time-domain methods. All these led researchers to search for a new domain to work on the convolutive mixtures problem.

2.4.3 Frequency-Domain Methods

One of the recent methods for performing ICA of convolutive mixtures is the Frequency Domain ICA. Smaragdis [Sma98], Lee et al [LBL97], Parra and Spence [PS00b] proposed moving to the frequency domain, in order to solve the convolution problem.

Looking at the FIR feedforward convolutive mixtures model, one can use the convolutive model in (2.110). The notation used in (2.110) is also known as *FIR matrix algebra* [Lam96]. From adaptive filter theory, we know that such problems can be addressed with a general *multichannel, subband filterbank*. However, there are certain benefits by choosing a Fourier basis filter bank, i.e. the *Fourier transform*. One motivation is that the signals become more superGaussian in the frequency domain, which will be beneficial for any ICA learning algorithm. Another motivation is that applying the Fourier Transform on the previous equation, we can approximate the linear convolution with multiplication. More specifically:

$$STFT\left\{\begin{bmatrix} x_1(n) \\ \dots \\ x_N(n) \end{bmatrix}\right\} = STFT\left\{\begin{bmatrix} \underline{\alpha}_{11} * s_1(n) & \dots & \underline{\alpha}_{1N} * s_N(n) \\ \dots & \dots & \dots \\ \underline{\alpha}_{N1} * s_1(n) & \dots & \underline{\alpha}_{NN} * s_N(n) \end{bmatrix}\right\} \Rightarrow \tag{2.119}$$

$$\Rightarrow \begin{bmatrix} x_1(f,t) \\ \dots \\ x_N(f,t) \end{bmatrix} = \begin{bmatrix} A_{11}(f) & \dots & A_{1N}(f) \\ \dots & \dots & \dots \\ A_{N1}(f) & \dots & A_{NN}(f) \end{bmatrix} \begin{bmatrix} s_1(f,t) \\ \dots \\ s_N(f,t) \end{bmatrix} \tag{2.120}$$

$$\underline{x}(f,t) = A_f \underline{s}(f,t), \qquad \forall\ f = 1, \dots L \tag{2.121}$$

where $x(f,t) = STFT\{x(n)\}$ and L is the number of FFT points. The *Short Time Fourier Transform* (STFT) is used instead of the Fourier Transform, in order to divide the signal into shorter overlapping frames and preserve signal's stationarity. Using the Fourier transform and assuming *statistical independence* between frequency bins, we have transformed a convolutional problem into L instantaneous mixtures problems, i.e. an instantaneous mixtures problem for each frequency bin. In order to transform the convolution into multiplication, one has to use windows larger than the maximum length of the transfer functions, i.e. $L \gg K$. Hence, we can use the very well established theory on separation of instantaneous mixtures and solve this problem. However, this case is not as simple as ICA of instantaneous mixtures. This is due to the following reasons:

1. The dataset in this case are instantaneous mixtures of complex numbers, which implies that we have to ensure the stability and convergence of the original algorithms with complex data.

2. The *scale* and *permutation* ambiguity, which had negligible effect in the instantaneous mixtures case, now play a very important role in this approach, as it will be explained later on.

In the next subsections, we will have a closer look at three basic approaches on Frequency Domain ICA and explain the permutation and scale ambiguity in detail.

Lee's approach

Continuing from the time-domain approach, Lee at al [LBL97] claimed that a FIR unmixing structure would be more beneficial in the audio case, mainly because real room acoustics usually involve non-minimum phase mixing (zeros outside the unit circle). In addition, they proposed moving to the fre-

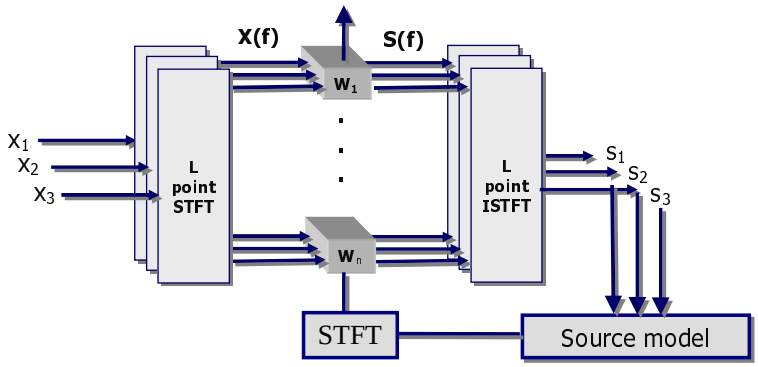

Figure 2.11: Lee's frequency domain framework: Unmixing in the frequency domain, source modelling in the time domain.

quency domain and unmix the sources there, in order to avoid the convolution in the time-domain. Hence, an update rule, similar to Amari's natural gradient (see eq. 2.27), was developed. The unmixing matrix W_f for every frequency bin is estimated in the frequency domain using the following rule:

$$\Delta W_f \propto (I + \mathcal{E}\{STFT\{\phi(\underline{u}(n))\}_f \underline{u}^H(f,t)\})W_f \tag{2.122}$$

The proposed framework is illustrated in figure 2.11. The key point in Lee's approach, apart from unmixing in the frequency domain, is that he prefers to apply the nonlinearity $\phi(u)$ in the time domain. As the nonlinearity contains information about the source models, Lee et al prefer to model their sources in the time-domain. This can have some advantages and disadvantages as it will be explained further on. The main obvious disadvantage of this method is the extra computational complexity introduced by moving the estimated signals from and to the frequency domain for every update, in order to apply the nonlinearity. One advantage is that Lee et al did not encounter the permutation problem, however, this might not always be the case, as it will be discussed later on.

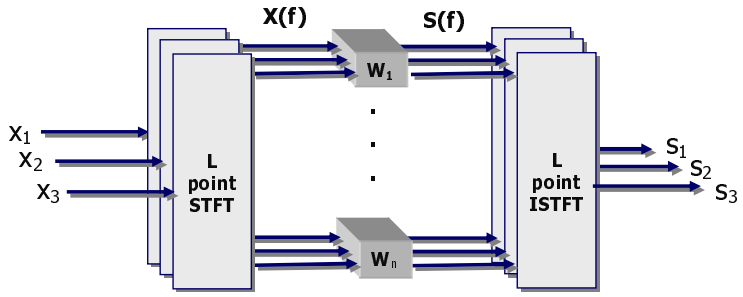

Figure 2.12: Smaragdis' frequency domain framework: Unmixing and source modelling in the frequency domain.

Smaragdis' approach

Smaradgis [Sma98] proposed to work solely in the frequency domain for the convolutive problem, i.e. perform the unmixing and the source modelling in the frequency domain, in order to avoid the extra complexity of moving from the frequency to the time domain and vice versa. Therefore, the system is adapting solely in the frequency domain, independently for each frequency bin. The proposed framework can be seen in figure 2.12.

For each frequency bin, one can assume superGaussian priors for our signals. Signals tend to be more superGaussian in the frequency domain in nature (see paragraph 3.3.3). Starting with a complex prior $p_s(s)$, one can minimize the Kullback-Leibler divergence between the prior and the probability distribution of the actual data $p_u(u)$ and following Amari's paper [ACY96] derive the natural gradient for complex data.

$$\Delta W_f = \eta(I + \mathcal{E}\{\phi(\underline{u}(f,t))\underline{u}(f,t)^H\})W_f \tag{2.123}$$

where η is the learning rate and $\phi(u) = \partial \log p_u(u)/\partial u$. Smaragdis observed that we can not apply the sigmoid $\tanh(u) = (e^u + e^{-u})/(e^u - e^{-u})$ function for complex data, as it has singularities for $u = j\pi(k + 1/2)$, where $k \in \mathbb{Z}$. These singularities can cause instability to the natural gradient rule. As a result, Smaragdis proposed the following split-complex sigmoid

function that is *smooth*, *bounded* and *differentiable* in the complex domain.

$$\phi(u) = \tanh(\Re\{u\}) + j\tanh(\Im\{u\}) \tag{2.124}$$

The natural gradient algorithm is robust and converges in relatively easy acoustic environments. Smaragdis observed the problems arising from scale and permutation ambiguity and proposed some solutions. In addition, he proposed the use of zero-padding before the FFT, as a tool to smooth the spectra, to facilitate the separation algorithm. On the whole, the proposed framework seems to be a robust, general solution to the convolutive mixtures problem.

Parra's approach

Parra and Spence [PS00b] exploited *non-stationarity* and second order statistics of audio signals with additional constraints in the time and frequency domain to propose a new ICA method for separation of convolutive mixtures.

A signal $s(n)$ is considered non-stationary*, if $C_s(n) \neq C_s(n+\tau)$, where $C_s(n) = \mathcal{E}\{\underline{s}(n)\underline{s}(n)^T\}$ is the covariance matrix of $\underline{s}$ and τ a constant.* That is to say that a signal is considered non-stationary, if its statistics change along time. Assume a noisy convolutive mixtures model, as follows:

$$\underline{x}(n) = A * \underline{s}(n) + \underline{\epsilon}(n) \Rightarrow \tag{2.125}$$

$$\underline{x}(f,t) = A(f)\underline{s}(f,t) + \underline{\epsilon}(f,t), \qquad \forall\ f = 1,\ldots,L \tag{2.126}$$

We form the covariance matrix and obtain:

$$C_x(f,k) = E\{\underline{x}\underline{x}^H\} = A_f C_s(f,k) A_f^H + C_\epsilon(f,k) \tag{2.127}$$

Assuming the estimated sources $\underline{u}(f,t)$, $\tilde{C}_\epsilon(f,k)$ the estimated noise covariance, $\tilde{C}_u(f,k)$ the estimated sources covariance and $C_x(f,k)$ the covariance of the observed data. An appropriate error measurement is :

$$E(k) = C_x(f,k) - A_f \tilde{C}_u(f,k) A_f^H - \tilde{C}_\epsilon(f,k) \tag{2.128}$$

As a result, a good cost function to minimise is

$$J(A_f, \tilde{C}_\epsilon, \tilde{C}_u) = \sum_k ||E(k)||_F^2 \tag{2.129}$$

Using the derivatives $\partial J/\partial A, \partial J/\partial \tilde{C}_\epsilon, \partial J/\partial \tilde{C}_u$, one can find estimates for each of the parameters $A_F, \tilde{C}_\epsilon, \tilde{C}_u$.

Assuming a stable FIR unmixing filter W_f, we can rewrite the above equations, in terms of W_f, as follows:

$$\hat{C}_u(f,k) = W_f[C_x(f,k) - C_\epsilon(f,k)]W_f^H \tag{2.130}$$

The cost function that can be employed in this case:

$$J(W_f, \tilde{C}_\epsilon, \tilde{C}_u) = \sum_k ||\hat{C}_u(f,k) - \tilde{C}_u(f,k)||_F^2 \tag{2.131}$$

One can obtain estimates for W_f formulating the gradients of the above contrast function in terms of W_f, C_u and C_ϵ, according to the analysis in [PS00b].

In order to estimate $\underline{u}(f,t)$, one can use W_f in the square case. As Parra tries to cater for the non-square case as well, he proposes a *Least Squares (inverse filtering)*, a *Maximum likelihood* or a *MAP estimate* to retrieve the sources. Methods to retrieve sources, given the mixing matrix A_f, were discussed earlier on. Parra also observed the scale and permutation ambiguity and proposed solutions that are analysed in the next paragraph.

In addition to exploiting nonstationarity, for periodic signals with known statistical profile, one can exploit other second-order information to solve the separation problem, such as *cyclostationarity*. Wang et al [WJSC03] proposed a solution combining fourth order and second order information to perform separation of cyclostationary convolutive mixtures.

Scale ambiguity in frequency domain methods

The *scale ambiguity* in ICA of instantaneous mixtures was analyzed in paragraph 2.2.4. The ICA algorithms are not able to determine the variances (energies) of the independent components. As a result, the algorithms unmix the original signals up to a scaling factor. For instantaneous ICA in the time domain, this is not a problem, as the unmixed signals may be amplified or attenuated after separation, however a normalisation can always rectify the problem.

In frequency domain ICA, we adapt L independent algorithms, one for each frequency bin. Thus, any arbitrary scaling change to each individual update rule will cause spectral deformation to our unmixed signals. In addition, it is not guaranteed that the scaling will be uniformly distorted along frequency, changing the signal envelope after separation.

Smaragdis [Sma98] proposed to keep the unmixing matrix normalised at unit norm, i.e. $||W_f|| = 1$. This implies that the unmixing matrix does not scale the data. This step can be beneficial for the convergence of the algorithm, as it prevents the gradient descent (natural gradient) from diverging much from the optimum.

$$W_f \leftarrow W_f ||W_f||^{-1/N} \tag{2.132}$$

Parra thought of constraining the diagonal elements of the unmixing matrix to unity, i.e. $W_f^{ii} = 1$, in order to avoid any scale deformation of the array. Other methods are proposed to tackle the scale ambiguity in the next chapter.

Permutation ambiguity in frequency domain methods

The *permutation ambiguity* in ICA of instantaneous mixtures was analyzed in paragraph 2.2.4. The ICA algorithms are not able to determine the permutation of the independent components. As a result, the order of the unmixed components is totally random. For instantaneous ICA in the time domain, the order of the unmixed signals is not a problem. Usually, we

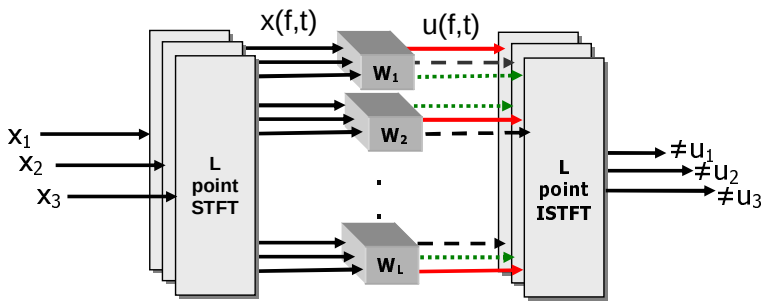

Figure 2.13: An illustration of the permutation problem in frequency domain ICA. The arbitrary permutation of the successfully separated components along frequency results in the reconstructed sources remain mixed.

are interested in retrieving all sources, therefore the permutation is not important at all.

In frequency domain ICA, we adapt L independent algorithms, one for each frequency bin. Therefore, any arbitrary permutation of the sources along the frequency axis, will result in the sources remaining mixed, when reconstructed in the time domain (see figure 2.13). As a result, we must impose some coupling between frequency bins to align the permutations along frequency. Many solutions have been proposed for the permutation ambiguity and will be analyzed in detail in the following chapter, along with a new proposed solution.

Lee et al never experienced the permutation ambiguity. The main reason being that they apply the source model, i.e. the nonlinearity in the time-domain, and as a result they do not have to assume statistical independence between the frequency bins in the frequency-domain source model. This assumption is mainly the cause of the permutation ambiguity in the frequency domain, although there is evidence that even using time domain models, the permutation problem can still exist [PA02].

Smaragdis tried to couple neighbouring bins, assuming that the unmixing matrices of two neighbouring bins should be similar, and proposed the following coupling rule:

$$\Delta W_{f+1} \leftarrow \Delta W_{f+1} + \alpha \Delta W_f \tag{2.133}$$

where $0 \leq \alpha \leq 1$ is constant that weights the influence of the neighbouring bin.

In order to solve the permutation problem, Parra et al put a constraint on the length K of the unmixing FIR filter W. Basically, assuming the FIR structure for the unmixing filter, we expect the frequency response of that filter to be a smooth function as the FIR frequency response is basically polynomial. Therefore, this projection operator tries to keep the unmixing filter as smooth as possible, lining up the correct permutations accordingly.

2.5 Conclusion

In this chapter, we have analysed some of the techniques that have been developed to solve the ICA problem in the case of *instantaneous*, *overcomplete* and *convolutive mixtures*. The aim of this chapter was not to perform a thorough review of the methods developed on the subject but on the other hand, give an overview of the area, emphasizing the approaches that influenced our work. For a more thorough review on ICA problems, applications and methods, one can always refer to Hyvärinen, Oja and Karhunen's book, titled *Independent Component Analysis* [HKO01], or to T.W. Lee's book, titled *Independent Component Analysis - Theory and Applications* [Lee98].

In the next chapters, we will look into a fast frequency domain ICA framework that was introduced to solve the convolutive mixtures problem. A method to solve the permutation problem was introduced. Further on, we will look into a channel modelling solution for the permutation ambiguity, such as beamforming. The idea of performing "intelligent" ICA, i.e. automatically extracting a single source of interest from the mixtures will be explored further on. Finally, some more extensions and considerations on the general frequency domain framework will be presented.

Chapter 3

Fast ICA solutions for convolutive mixtures

3.1 Introduction

In this chapter, we are going to examine fast unmixing solutions for the square convolutive mixtures under the frequency domain framework. The permutation and scale ambiguity are going to be analysed in depth and a novel source modelling solution for the permutation is presented. In addition, in the search for a fast unmixing algorithm in the frequency domain framework, two novel unmixing approaches are presented and evaluated. Finally, we will examine the effect of frame size and the aliasing introduced in the frequency domain framework.

3.2 Solutions for the scale ambiguity

In the previous chapter, we defined the *scale ambiguity* of instantaneous ICA and explained how this interdeterminancy can cause spectral deformation of the separated signals. However, there are methods to remove this ambiguity from the ICA framework.

3.2.1 Previous approaches

The element that can cause the scale ambiguity is the unmixing matrix W_f. Following gradient based laws to update the unmixing matrix without any constraint, the estimate can change in scale (sign and magnitude). As a result, the unmixed sources will be altered in scale.

An obvious approach is to apply a *constraint on the unmixing matrix*. Smaragdis [Sma98] proposed to constrain the matrix by normalising it to unit determinant.

$$W_f \leftarrow W_f / ||W_f||^{1/N} \tag{3.1}$$

where N is the number of the sensors and sources. This constrains the matrix to perform rotations but not scaling. This action is also beneficial for the convergence of the algorithm, as this normalisation prevents the algorithm from overshooting. In a similar effort, Parra and Spence [PS00b] constrained the diagonal elements of the unmixing matrix to unity, i.e. $W_f^{ii} = 1$. This can constrain the scaling of the unmixing matrix W_f.

Another approach would be to constrain the variance of the data. In the frequency domain framework, the signal will have different signal levels at each frequency bin. The updates to W_f are calculated passing through the data at each frequency bin. Therefore, different energy levels may lead the unmixing matrix to different scaling. Normalising the data to unit variance can enforce uniform scaling of the unmixing matrix along frequency.

3.2.2 Mapping to the observation space

A valid solution to address the scale ambiguity is *mapping the sources back to the observation space*, i.e. the microphones' space. The idea is mentioned by Cardoso [Car98b]. In this study, Cardoso mentions that "instead of focusing on the columns of the mixing matrix A, we can focus on the spaces containing each component and then we can get the same separation result, without the ambiguity of *scale* (sign and magnitude)". In other words, by mapping the separated sources back to the observation space of the microphones, we can undo any scale deformation, performed by the unmixing

matrix W, preserving separation though. We can support this argument mathematically with the following analysis. At first, we will assume that the permutation ambiguity is sorted. The 2×2 case will be used for simplicity, but it is straightforward to generalise the analysis to the $N \times N$ case. Assume the following mixing model.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \tag{3.2}$$

Define the signals $\underline{x}_{s_1}$ and $\underline{x}_{s_2}$, as the signals s_1, s_2 observed by the microphones each alone in the auditory scene, i.e.

$$\underline{x}_{s_1} = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} s_1, \qquad \underline{x}_{s_2} = \begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix} s_2 \tag{3.3}$$

As a result,

$$\underline{x} = \underline{x}_{s1} + \underline{x}_{s2} \tag{3.4}$$

We want to estimate the unmixing matrix $W = A^{-1}$ that can separate the sources. Having sorted out the permutation problem, the ICA finally estimates the matrix $\hat{W} = (A\Lambda)^{-1}$, where $\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$ is a diagonal matrix containing the arbitrary scaling introduced by the algorithm. As a result, our separated outputs are scaled.

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \hat{W}\underline{x} = (A\Lambda)^{-1}A\underline{s} = \Lambda^{-1}\underline{s} = \begin{bmatrix} s_1/\lambda_1 \\ s_2/\lambda_2 \end{bmatrix} \tag{3.5}$$

Having estimated $\hat{W}$, we can move the separated signals to the microphones' domain and undo the incorrect scaling. In other words we have to calculate $\underline{x}_{s1}, \underline{x}_{s2}$.

$$\underline{x}_{s_1} = \begin{bmatrix} (\hat{W}^{-1})_{11} \\ (\hat{W}^{-1})_{21} \end{bmatrix} u_1 = \begin{bmatrix} a_{11}\lambda_1 \\ a_{21}\lambda_1 \end{bmatrix} s_1/\lambda_1 = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} s_1 \tag{3.6}$$

$$\underline{x}_{s_2} = \begin{bmatrix} (\hat{W}^{-1})_{12} \\ (\hat{W}^{-1})_{22} \end{bmatrix} u_2 = \begin{bmatrix} a_{12}\lambda_2 \\ a_{22}\lambda_2 \end{bmatrix} s_2/\lambda_2 = \begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix} s_2 \tag{3.7}$$

As we can see, we have removed the arbitrary scaling by projecting the signals back to the microphone's domain and still have the signals unmixed.

Similarly, we can prove that this scheme can remove the scale ambiguity, even when the permutation ambiguity is not sorted. Assume the 2×2 scenario that was proposed previously and a permutation matrix $\Pi = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, denoting that the sources are flipped. As a result, the ICA algorithm has estimated the following matrix $\hat{W}$

$$\hat{W} = (A\Lambda\Pi)^{-1} \tag{3.8}$$

$$\hat{W} = \left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right)^{-1} = \begin{bmatrix} \lambda_2 a_{12} & \lambda_1 a_{11} \\ \lambda_2 a_{22} & \lambda_2 a_{21} \end{bmatrix}^{-1} \tag{3.9}$$

The separated outputs will be:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \hat{W}\underline{x} = (A\Lambda\Pi)^{-1} A\underline{s} = \Pi^{-1}\Lambda^{-1}\underline{s} = \begin{bmatrix} s_2/\lambda_2 \\ s_1/\lambda_1 \end{bmatrix} \tag{3.10}$$

Moving the separated signals to the microphones' domain, we can still undo the incorrect scaling.

$$\underline{x}_{s_1} = \begin{bmatrix} (\hat{W}^{-1})_{11} \\ (\hat{W}^{-1})_{21} \end{bmatrix} u_1 = \begin{bmatrix} a_{12}\lambda_2 \\ a_{22}\lambda_2 \end{bmatrix} s_2/\lambda_2 = \begin{bmatrix} a_{12} \\ a_{22} \end{bmatrix} s_2 \tag{3.11}$$

$$\underline{x}_{s_2} = \begin{bmatrix} (\hat{W}^{-1})_{12} \\ (\hat{W}^{-1})_{22} \end{bmatrix} u_2 = \begin{bmatrix} a_{11}\lambda_1 \\ a_{21}\lambda_1 \end{bmatrix} s_1/\lambda_1 = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} s_1 \tag{3.12}$$

As we can see, the scale ambiguity is removed, despite the existing permutation ambiguity.

3.3 Solutions for the permutation ambiguity

Solving the convolutive problem in the frequency domain, independently for each frequency bin generates the *permutation problem*, since there is the

inherent permutation ambiguity in the rows of W_f [PS00b, Sma98]. This is more complicated than the *ordering ambiguity* in the instantaneous mixtures ICA, since the ordering of the sources must remain the same along the frequency axis. As a result, the ICA algorithm produces different permutations of separated sources along the frequency axis, and therefore the sources remain mixed. In order to solve this problem, we need to impose some sort of coupling between the "independent" unmixing algorithms, so that they converge to the same order of sources.

Many solutions have been proposed to tackle the problem. In general, these solutions fall into two categories: the *source modelling* and the *channel modelling* approaches.

3.3.1 Source modelling approaches

In *source modelling* solutions, the aim is to exploit the coherence and the information between frequency bands, in order to identify the correct alignment between the subbands. In fact, audio signals can rarely be considered independent between frequency bands due to the actual audio structure (harmonic stacks and transients) in both music and speech. As a result, any rule that can group similar objects will align the permutations.

In Lee's approach [LBL97], the signals are modelled in the time-domain (the $\tanh(\cdot)$ nonlinearity is applied in the time-domain). There is a benefit from imposing time-domain source models: *the permutation problem does not seem to exist*. When we apply the source model in the time-domain, we do not assume that the signals are statistically independent along each frequency bin. As a result, the permutations are coupled due to the source model applied to the whole signal and not to its "independent decompositions". However, there is evidence reported that problems similar to the permutation problem do exist [PA02]. This method is computationally expensive, due to the mapping back and forth between the frequency and time domains and do not take advantage of the strong nonGaussianity in the frequency domain.

Ikeda [IM99] tried to match the time envelopes of the signal along the frequencies. This approach models the fact that at the same time index we usually get similar energy stimulation along all frequencies. However, even appropriate matching of energy envelopes along frequency might not be accurate enough, as the energy profile is different for each frequency band for the same signal. In a following subsection, we will see a novel, more accurate way of modelling the idea of localising energy bursts along time.

3.3.2 Channel modelling approaches

In *channel modelling* solutions, the aim is to exploit additional information about the room transfer functions, in order to select the correct permutations. These room transfer functions have certain properties. In source separation, we usually employ long FIR (Moving Average, all-zero) models to estimate the room transfer functions, as their stability is guaranteed. In addition, most room transfer function have a dominant first delay (direct path) term that can be used to identify the angular position of each source signal to the sensor array.

Smaragdis proposed an adaptive scheme to apply some frequency coupling between neighbouring frequency bins. Assuming that the unmixing matrices between neighbouring bins W_f and W_{f-1} should not be too dissimilar, he proposed the following coupling scheme.

$$\Delta W_f \leftarrow \Delta W_f + a\Delta W_{f-1} \tag{3.13}$$

where $0 < a < 1$. This heuristic adaptive solution can be interpreted as placing weakly coupled priors on W_f of the form:

$$p(W_f|W_{f-1}) \propto \exp(-\frac{1}{2\sigma^2}||W_f - W_{f-1}||_F) \tag{3.14}$$

This imposes some weak smoothness constraint across frequency. However, it had limited effect, as it has been reported to fail in several separation cases [Dav00].

Parra et al [PS00b] also worked in the frequency domain using non-stationarity to perform separation. Their solution to the problem was to impose a constraint on the unmixing filter length K. This is achieved by applying a projection operator P to the filter estimates at each iteration, where $P = FZF^{-1}$, F is the Fourier transform and Z is a diagonal operator that projects on the first K terms. In other words, it imposes a *smooth* constraint on the unmixing filters, as they are modelled as FIR filters (polynomials). Again mixed success has been reported for this method, as it seems to get trapped in local minima [IM00]. Both approaches can be characterized as *gradient solutions*, and problems similar to those noted in [Dav00] tend to occur.

Another solution is to use *beamforming* to align the permutations along the frequency axis. All BSS methods make no assumptions about the position of the sources in the 3D space. However, *beamforming* estimates the directions of signal's arrival (DOA) in order to steer the beam of an array of sensors to focus on a specific source, as investigated by Saruwatari et al [SKS01], Ikram and Morgan [IM02], Parra and Alvino [PA02]. The extra geometrical information employed by beamforming is the sensors' arrangement, which is assumed to be fixed. We will analyse the application of beamforming in the BSS concept in detail in the next chapter.

3.3.3 A novel source modelling approach

The next section describes a novel approach that imposes *frequency coupling in the source model*. The method consists of two steps : a) a time-frequency source model to force coupling between frequency bins and b) a likelihood ratio jump to align the permutations.

A time-frequency source model

If we examine the statistical properties of an audio signal over shorter quasi-stationary periods in the time-domain (frames of the STFT), the signal is not always well modelled as superGaussian. Looking at the statistical

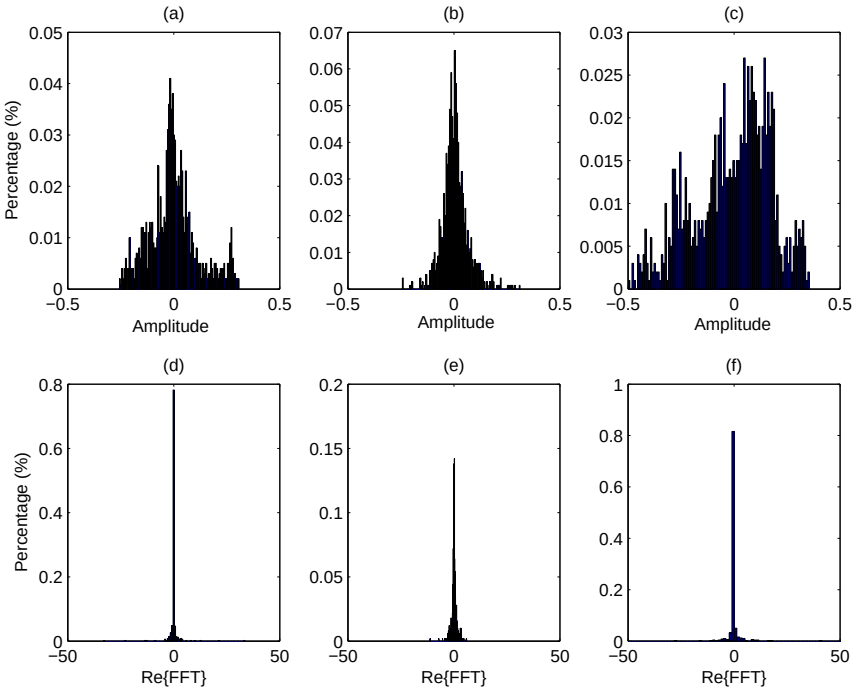

Figure 3.1: Exploring the statistical properties of short audio segments. Histograms of three different 62.5$msec$ segments in the time domain (a),(b),(c) and the corresponding histograms in the frequency domain (d), (e), (f).

properties of these segments in the frequency domain, they can be better modelled as superGaussian, as these sections have very heavy tailed distributions [Dav00]. Figure 3.1 exhibits the histograms of some audio signal segments in the time-domain and the histograms of the real part of Fourier transform of these segments.

This implies that the frequency domain is a better candidate for source modelling. This will provide a better achievable performance, since as noted by various authors (e.g. [Car98a]), the *Cramer-Rao bound* (the performance bound for an estimator) for the estimation of the unmixing matrix in ICA algorithms is related to how close the source distributions are to Gaussian. That is that the more nonGaussian the distributions are, the better the achievable performance of the ICA algorithm.

In addition, most of the superGaussianity measured in the time domain comes from the fluctuating amplitude of the audio signal. The *slowly varying amplitude* profile also gives us valuable information that can be exploited for source separation and is not affected by the permutation problem. Therefore, we can exploit this property to introduce frequency coupling within the STFT structure.

Motivated by this, we introduce the following *time-frequency model*. We will generally assume that the STFT coefficients of the separated sources follow an exponential nonGaussian distribution. In addition, the model needs to incorporate some information about the scaling of the signal with time (i.e. the signal envelope), assuming that it is approximately constant over the analysis window. This can be modelled by a *nonstationary time varying* scale parameter β_k.

$$p(u_k(f,t)) \propto \beta_k(t)^{-1} e^{-h(u_k(f,t)/\beta_k(t))} \tag{3.15}$$

where $h(u)$ defines the general statistical structure (i.e. superGaussianity), the index t represents the time-frame index, f the frequency bin and k is the source index. The key feature is that the β_k term *is not a function of frequency*, but only a function of time. This restriction provides us with sufficient coupling between frequency bins to break the *permutation ambiguity*. The β_k term can be interpreted as a *volume measurement*. Literally, it measures the overall signal amplitude along the frequency axis (all frequencies), emphasising the fact that one source is "louder" at a certain time slot. This "energy burst" indication can force alignment of the permutations along the frequency axis.

To incorporate this model to the Frequency Domain ICA framework, we need to see how the proposed time-frequency model alters the *natural gradient* algorithm in (2.123). Effectively, the source model is represented by the *activation function* $\phi(u)$. Recall we have:

$$\phi(u) = \frac{\partial}{\partial u} \log p(u) = \frac{1}{p(u)} \frac{\partial p(u)}{\partial u} \tag{3.16}$$

The proposed model gives the following activation function:

$$\phi(u_k(f,t)) \propto \beta_k(t)^{-1} h'(u_k(f,t)/\beta_k(t)) \tag{3.17}$$

The natural gradient algorithm is then altered as follows:

$$\Delta W_f = \eta(I + \beta(t)^{-1}\mathcal{E}\{g(\underline{u}(f,t))\underline{u}^H(f,t)\})W_f \tag{3.18}$$

where $\beta(t) = diag(\beta_1(t), \beta_2(t), \ldots, \beta_N(t))$, $g(u) = h'(u)$ and η is the learning rate. The value for $\beta_k(t)$ is estimated adaptively from the separated signals $\underline{u}(f,t)$.

We note that care needs to be taken in defining activation functions for complex data. Below, we will consider activation functions of the form $(u/|u|)f(|u|)$. Although a variety of other activation functions are valid, such as $g(u) = \tanh(\Re\{u\}) + j\tanh(\Im\{u\})$ (split non-linearity), proposed by Smaragdis [Sma98], it seems more intuitive to impose no preference on the phase angles. That is to introduce circularly symmetric priors on complex variables without phase preference. This is essentially the same as the priors on subspaces as proposed by Hyvärinen et al in *Independent Subspace Analysis* (ISA) [HH00]. Assuming complex Laplacian priors in the form of $p(u) \propto exp(-|u|) \Rightarrow h(u) = |u|$, we set $f(|u|) = 1$. The activation function in (3.18) is then the following:

$$g(u) = u/|u|, \qquad \forall |u| \neq 0 \tag{3.19}$$

Although the discontinuity due to $|u|$ implies the cost function will not be smooth at certain points, in practice, the performance of the algorithm appears to be unaffected. MacKay [Mac02] also supported that the above "Laplacian" function can have the same robustness property as the tanh function. Alternatively, we could use a "smoothed" Laplacian prior $p(u) \propto \exp(-|u| + \log|u|)$, as proposed by Zibulevsky [ZKZP02].

Assuming complex Laplacian priors, we can use the following estimate for $\beta_k(t)$:

$$\beta_k(t) = \frac{1}{L}\sum_f |u_k(f,t)| \tag{3.20}$$

Permutation Problem Revisited - The likelihood Ratio Jump

Let us now investigate the effect of this time-frequency model upon the permutation symmetries. Without the $\beta(t)$ term the log likelihood function has an identical maximum for every permutation of the sources at each frequency. Incorporating β, we weight the likelihood of an unmixing matrix at a given frequency with the time envelope induced by the components at other frequencies. Thus, β allows the matching of time envelopes, providing us with a discriminator for the different permutations.

Nonetheless, a direct application of (3.18) does not guarantee that the correct permutation will be found. The β term will break the symmetry, however, it will not necessarily change the cost function enough to completely remove spurious minima. Thus, a gradient optimisation scheme is likely to get trapped in a local minimum. This may explain the poor performance of Parra's solution [PS00b] in certain examples, as observed by Ikram et al [IM00].

As a result, we introduce a post processing mechanism in the algorithm by which the correct permutations are sorted. Fortunately, due to the symmetry of the problem, if we know where one minimum is, we know where they all are. As the sources are separated and therefore statistical independent, if we know where one is, then all the others will be orthogonal to the first one in the N^{th} dimensional space. It is therefore possible to introduce a jump step into the update that chooses the permutation that is most likely.

Here we describe a solution for $N = 2$, using the Laplacian prior. Suppose that for a given set of known W_f, $\beta_k(t)$ and $\underline{u}(f,t) = W_f \underline{x}(f,t)$, we wish to compare two possible choices for source estimates of u:

$$1. \begin{bmatrix} \gamma_{11} & 0 \\ 0 & \gamma_{22} \end{bmatrix} \tilde{u}(f,t) = u(f,t) \tag{3.21}$$

$$2. \begin{bmatrix} 0 & \gamma_{12} \\ \gamma_{21} & 0 \end{bmatrix} \tilde{u}(f,t) = u(f,t) \tag{3.22}$$

where γ_{ij} are rescaling parameters that account for incorrect scaling. To

compare these two possibilities, we will evaluate their likelihood over T time frames.

$$1.\ \log p(u|\gamma_{11},\gamma_{22}) = -T\log(\gamma_{11},\gamma_{22}) + \log p(\tilde{u}) \tag{3.23}$$

$$2.\ \log p(u|\gamma_{12},\gamma_{21}) = -T\log(\gamma_{12},\gamma_{21}) + \log p(\tilde{u}) \tag{3.24}$$

with the values of γ_{ij} chosen to maximise the likelihood. For the Laplacian model these are:

$$\gamma_{ij} = \frac{1}{T}\sum_t \frac{|u_i(f,t)|}{\beta_j(t)} \tag{3.25}$$

We can now evaluate the likelihood of the estimated $u(f,t)$ in terms of the known quantities $u(f,t)$ and γ. For case 1, we have:

$$\log p(\tilde{u}) \propto -\gamma_{11}^{-1}\sum_t \frac{|u_1(f,t)|}{\beta_1(t)} - \gamma_{22}^{-1}\sum_t \frac{|u_2(f,t)|}{\beta_2(t)} \tag{3.26}$$

which reduces to $\log p(\tilde{u}) \propto -2T$. The analysis for case 2 is identical. Therefore, we get:

$$\log \frac{p(\text{“case1”})}{p(\text{“case2”})} = -T\log(\gamma_{11}\gamma_{22}) + T\log(\gamma_{12}\gamma_{21}) \tag{3.27}$$

and we can form the following *likelihood ratio test* (LR):

$$LR = \frac{p(\text{“case1”})}{p(\text{“case2”})} = \frac{\gamma_{12}\gamma_{21}}{\gamma_{11}\gamma_{22}} \tag{3.28}$$

If $LR < 1$, we permute the rows of W_f before proceeding. This likelihood ratio test is performed after calculating the update ΔW_f, lining up permutations that were not sorted by the gradient step.

There are basically *two drawbacks* in this approach. Firstly, this becomes *more complicated for more than 2 sources*, although one possible solution would be to consider the sources in a pairwise fashion. Secondly, the algorithm has to work *only in batch mode*, as usage of a one-sample likelihood is not possible. On the other hand, the algorithm seems to perform well in the majority of cases.

Generalising the Likelihood Ratio Jump

In this section, we will try to generalise the *Likelihood Ratio Jump* solution for the general $N \times N$ case. In fact, the coefficient γ_{ij} can model the probability that the i^{th} source has moved to the j^{th} position (of the original source alignment). For example, the product $\gamma_{31}\gamma_{22}\gamma_{13}$ can model the probability of the following perturbation: sources $3 \rightarrow 1, 2 \rightarrow 2, 1 \rightarrow 3$, for the 3×3 case.

For the $N \times N$ case, we have to examine all different *ordered combinations* of N sources. This gives us $N!$ cases in total that need to be compared. The probability of each case is formed in a similar manner as described for the 2×2 case.

We will briefly demonstrate the 3×3 situation, where we have $3! = 6$ ordered combinations. Consequently, you have to form the following probabilities:

$$L_1 = \log p(\text{“case 1”}) = -\log(\gamma_{11}\gamma_{22}\gamma_{33}) \tag{3.29}$$

$$L_2 = \log p(\text{“case 2”}) = -\log(\gamma_{11}\gamma_{23}\gamma_{32}) \tag{3.30}$$

$$L_3 = \log p(\text{“case 3”}) = -\log(\gamma_{21}\gamma_{12}\gamma_{33}) \tag{3.31}$$

$$L_4 = \log p(\text{“case 4”}) = -\log(\gamma_{21}\gamma_{32}\gamma_{13}) \tag{3.32}$$

$$L_5 = \log p(\text{“case 5”}) = -\log(\gamma_{31}\gamma_{22}\gamma_{13}) \tag{3.33}$$

$$L_6 = \log p(\text{“case 6”}) = -\log(\gamma_{31}\gamma_{12}\gamma_{23}) \tag{3.34}$$

The correct permutation should be given by the $\max(L_1, L_2, L_3, L_4, L_5, L_6)$. As a result, we permute the rows of W_f according to the indices of γ in the maximum L. For example, if L_6 was the maximum, then we have to swap the rows of W_f, as follows: row $3 \rightarrow 1$, row $1 \rightarrow 2$, row $2 \rightarrow 3$.

One could possibly reduce the computational complexity of this scheme by performing a pairwise Likelihood Ratio, i.e. sort out the permutations in pairs.

3.4 Fast frequency domain ICA algorithms

So far, we have only considered a gradient-based optimisation scheme to produce maximum likelihood (or MAP) estimates of the original audio sources. However, all gradient-based optimisation methods have two major drawbacks.

1. Gradient algorithms *converge relatively slowly*. For a common frequency domain ICA scenario, we found that the natural gradient would require around 500 updates to each W_f (iterations) on average for some decent separation quality.

2. Gradient-based algorithms' *stability* depends on the *choice of the learning rate*. Natural signals have greater low frequency values; therefore the time-frequency values tend to have different signal levels for every frequency bin. Inevitably, keeping a constant learning rate for all learning procedures may inhibit the separation quality at certain frequency bands. This may also give a reason why the natural gradient approach does not perform well at high frequencies, as observed by Smaragdis [Sma98]. Other reasons for this behaviour come from the beamforming point of view (see Chapter 4).

For these reasons, we want to replace the natural gradient scheme in the FD-ICA framework with a Newton-type optimisation scheme. Their basic feature is that they converge much faster than gradient algorithms with the same separation quality and while they are more computationally expensive, the number of iterations for convergence is decreased. In addition, they tend to be much more stable, as their learning rate is defined by the inverse of the *Hessian* matrix [MS00]. Hyvärinen et al [BH00, Hyv99d, Hyv99c] introduced several types of Newton-type "fixed-point" algorithms in ICA of instantaneous mixtures, using kurtosis or negentropy.

3.4.1 A fast frequency domain algorithm

In [Hyv99c], Hyvärinen explored the relation between a generalised "fixed-point" (approximate Newton method) ICA algorithm with the maximum likelihood ICA approach on instantaneous mixtures. In the following analysis, we show that it is elementary to extend the algorithm proposed in [Hyv99c] to be applicable to the proposed time-frequency framework.

In the ML-ICA approach for instantaneous mixtures, we form and try to maximise the following likelihood with respect to the unmixing matrix W:

$$F = \log L(\underline{x}|W) = \mathcal{E}\{\log p(\underline{u})\} + \log|det(W)| \tag{3.35}$$

Performing *gradient ascent*, we can derive the Bell-Sejnowski [BS95] algorithm.

In [Hyv99c], Hyvärinen tries to solve the following optimisation problem:

$$\begin{aligned} &\max_{W} \mathcal{E}\{G(W\underline{x})\} \\ &\text{subject to } \mathcal{E}\{\underline{u}\underline{u}^T\} = I \end{aligned} \tag{3.36}$$

where $G(u)$ is a non-quadratic function. The solution for this problem can be estimated by finding the maximum of the following function:

$$K(W) = \mathcal{E}\{G(W\underline{x})\} - \alpha(\mathcal{E}\{\underline{u}\underline{u}^T\} - I) \tag{3.37}$$

where α is the *Lagrange multiplier*. Performing a gradient ascent on $K(W)$, we get:

$$\nabla K = \mathcal{E}\{G'(W\underline{x})\underline{x}^T\} - \alpha CW \tag{3.38}$$

where $C = \mathcal{E}\{\underline{x}\underline{x}^T\}$. If we choose $G(\underline{u}) = \log p(\underline{u})$, then this update law is almost identical to the Bell-Sejnowski law and the natural gradient, with a different term controlling the scaling of the unmixing matrix W. In fact, the algorithm in (3.38) can be viewed as solving a *constrained Maximum Likelihood problem*. After a series of steps (see [Hyv99c]) and using $G(\underline{u}) = \log p(\underline{u})$, we end up to the following learning rule:

$$\Delta W = D[diag(-\alpha_i) + \mathcal{E}\{\phi(\underline{u})\underline{u}^T\}]W \tag{3.39}$$

where $\alpha_i = \mathcal{E}\{u_i\phi(u_i)\}$, $D = diag(1/(\alpha_i - \mathcal{E}\{\phi'(u_i)\}))$. In practice, we observed that this algorithm converges at a faster rate than the gradient based update rules, as it will be demonstrated further on.

Comparing the update rule in (3.39) with the original natural gradient law, we can see that they are similar. Instead of a constant learning rate, there is a learning rate (the D matrix) that adapts to the signal. Hence, the algorithm is less dependent on signal levels and therefore more stable. Hyvärinen states that replacing I with the adaptive term $diag(-\alpha_i)$ is also beneficial for convergence speed. If we use pre-whitened data $\underline{x}$, then the formula in (3.39) is equivalent to the original fixed-point algorithm [Hyv99d], while it is still expressed in terms of the natural gradient algorithm. The most important consequence for us, however, is that the nonlinear activation function $\phi(u)$ in (3.39) has exactly the same interpretation as in the ML-approach.

3.4.2 An alternative approach

Bingham and Hyvärinen proposed a "fast" fixed-point algorithm for independent component analysis of complex valued signals [BH00]. It is an extension of Hyvärinen's FastICA algorithm [HO97, Hyv99a] for complex signals.

First of all, we assume that the observed signals are prewhitened. Now, the sources are orthogonal to each other in the N-dimensional space. Our objective is to maximise a suitable contrast function $J_G(\underline{w})$ in terms of $\underline{w}$. The maxima of this function should give us the independent components. In this optimisation problem, we constrain the contrast function to be in the following form, in order to reduce the complexity of the optimisation:

$$J_G(\underline{w}) = \mathcal{E}\{G(|\underline{w}^H \underline{x}|^2)\} \tag{3.40}$$

where $G(\cdot)$ is a smooth even function that can help us identify the independent components. Choosing $G(u) = u^2$ is equivalent of optimising the kurtosis of the absolute value of the complex data. Bingham and Hyvärinen

proposed a number of other possible candidates for $G(\cdot)$. In the same paper, they prove that complex independent components can be estimated by optimising the nonlinear function, as described by (3.40). Keeping $J_G(\cdot)$ a real-valued function facilitates this optimisation. Working instead directly on complex values, we need to pay more attention to the choice of the contrast function, as reported by Smaragdis [Sma98] and Davies [Dav00].

In addition, we impose the constraint that $\mathcal{E}\{|\underline{w}^H\underline{x}|^2\} = 1$ (i.e. orthonormal components, due to prewhitening). The following optimisation problem is set:

$$\max_{w_j} \sum_{j=1}^{N} J_G(\underline{w}_j) \qquad j = 1, \ldots, N \tag{3.41}$$
$$\text{subject to } \mathcal{E}\{(\underline{w}_k^H\underline{x})(\underline{w}_j^H\underline{x})^*\} = \delta_{kj}$$

where δ_{kj} is the Kronecker delta. The proposed fixed-point algorithm by Bingham and Hyvärinen is summarised by the following formula:

$$\underline{w}^+ \leftarrow \mathcal{E}\{\underline{x}(\underline{w}^H\underline{x})^*\phi(|\underline{w}^H\underline{x}|^2)\} - \mathcal{E}\{\phi(|\underline{w}^H\underline{x}|^2) + |\underline{w}^H\underline{x}|^2\phi'(|\underline{w}^H\underline{x}|^2)\}\underline{w} \tag{3.42}$$

$$\underline{w}^+ \leftarrow \underline{w}^+/||\underline{w}^+|| \tag{3.43}$$

where $\phi(u)$ is an activation function. Instead of calculating every independent component separately, it is preferable for many applications to calculate all components simultaneously. We can use different one-unit algorithms (3.42) for all independent components and apply a symmetric decorrelation to prevent the algorithms from converging to the same component. This can be accomplished by using a symmetric decorrelation:

$$W \leftarrow W(W^HW)^{-1/2} \tag{3.44}$$

where $W = [\underline{w}_1, \underline{w}_2, \ldots, \underline{w}_N]$ is the matrix of the vectors $\underline{w}_i$.

Bingham and Hyvärinen proposed a set of activation functions that can be applied to this fixed-point algorithm. As we can see the problem involves real data, therefore it is easier to choose an activation function. From the

set of the proposed activation functions, we are going to use the following:

$$\phi(u) = 1/(0.1 + u) \tag{3.45}$$

The derivative of the above is:

$$\phi'(u) = -1/(0.1 + u)^2 \tag{3.46}$$

This method achieves fast and accurate separation of complex signals. The small number 0.1 in the denominator prevents singularities of the activation functions for $u \to 0$. We are going to adapt this method to a frequency-domain separation framework. The main advantage of this algorithm is that it was initially designed to perform separation of complex-valued mixtures, therefore being easier to adapt directly in a frequency domain framework.

In other words, the observation signals are transformed into a time-frequency representation using a Short-Time Fourier Transform. As before, we prewhiten the $\underline{x}(f,t)$. Then, we have to calculate the unmixing matrix W_f for every frequency bin. We randomly initialise N learning rules, as described in (3.42) and (3.43) for every frequency bin and iterate until convergence. However, there are no steps to tackle the permutation problem.

We can address the permutation problem firstly, by incorporating the time dependent prior $\beta(t)$ in the learning rule, in order to impose frequency coupling. As we have seen in [BH00], the $\beta(t)$ term can be actually integrated in the activation function $\phi(u)$. In section 3.4.1, we saw that Hyvärinen transformed the basic fixed-point algorithm to a form that was similar to the natural gradient algorithm and we gathered that we could incorporate $\beta(t)$ in the activation function $\phi(u)$ of the fixed-point algorithm, so as to impose frequency coupling. This is the main motivation behind incorporating the $\beta(t)$ term in the activation function of the second fixed-point algorithm, although not with the same probabilistic interpretation. Therefore, equations (3.45) and (3.46) are now transformed in the following form.

$$\phi(u) = 1/(\beta_k(t)(0.1 + u)) \tag{3.47}$$

$$\phi'(u) = \partial\phi(u)/\partial u = -1/(\beta_k(t)(0.1+u)^2) \tag{3.48}$$

where $\beta_k(t)$ refers to the corresponding separated component u_k, as introduced in (3.20). The second step is to apply the likelihood ratio jump solution, described in (3.25), (3.28), so as to keep the same source permutation along the frequency axis. The likelihood ratio jump solution can be directly applied to the second fixed-point algorithm, without any adaptation.

3.4.3 Similarities between the two Fast-ICA solutions

The difference between the two fixed-point algorithms lies in the different contrast function employed in the optimisation problem. In the first fixed-point algorithm, the contrast function is $G_1(\underline{w}^H\underline{x})$, where as in the second fixed-point algorithm the contrast function is $G_2(|\underline{w}^H\underline{x}|^2)$, where $\phi(u) = \partial G(u)/\partial u$ and preferably a definition of kurtosis.

In the first Fast-ICA approach, we try to solve the problem:

$$\max G_1(\underline{w}^H\underline{x}) \qquad \text{subject to } ||\underline{u}||^2 = 1 \tag{3.49}$$

In the second Fast-ICA approach, we try to solve the problem:

$$\max G_2(|\underline{w}^H\underline{x}|^2) \qquad \text{subject to } ||\underline{u}||^2 = 1 \tag{3.50}$$

where G_1, G_2 are non-quadratic functions. In the thesis, we have shown that the method derived from the first problem by Hyvärinen can be seen as Maximum likelihood estimation, if we choose $G_1(u) = \log p(u)$. For $p(u)$, we use a Laplacian prior for the separated sources, i.e. $p(u) \propto e^{-|u|}$.

We can show very easily that the second problem can be regarded as ML estimation. Suppose we choose a non-quadratic function $G_2(u) = \log q(u)$, where $q(u) \propto e^{-\sqrt{|u|}}$. Then, we can show that:

$$G_2(|\underline{w}^H\underline{x}|^2) = \log e^{-\sqrt{|\underline{w}^H\underline{x}|^2}} = \log e^{-|u|} \tag{3.51}$$

In other words, the methods are similar in principle, however, they address the problem using different mathematical formulations. This might explain the similar performance in the source separation problem, as we will see in the next section. They can both be interpreted as ML estimation.

Moreover, we can easily justify some of the activation functions proposed by Hyvärinen for the second approach, i.e.

$$\phi(u) = \frac{\partial}{\partial u} \log q(u) = \frac{-1}{2\sqrt{u+\alpha}} \tag{3.52}$$

The α term is a small number added to stabilise the denominator of the activation function.

3.5 A unifying frequency domain framework

We can now use all the previous analysis to form a unifying framework for the convolutive mixtures problem.

First of all, we *prewhiten* the time-frequency STFT coefficients of the mixtures $\underline{x}(f,t)$ and store the prewhitening matrices V_f for each frequency bin.

The next step is to estimate the unmixing matrix for each frequency bin. We will use either of the two "fixed-point" approaches, using random initialisation for W_f. Moreover, he have to keep the rows of W_f orthogonal with unit norm.

First fixed-point algorithm

$$\Delta W_f = D[diag(-\alpha_i) + \mathcal{E}\{\phi(\underline{u}(f,t))\underline{u}^H(f,t)\}]W_f \tag{3.53}$$

$$W_f \leftarrow W_f(W_f^H W_f)^{-0.5} \tag{3.54}$$

The parameters in this update rule are calculated as previously. In addition, we will use the proposed time-frequency source model, as described earlier, to impose frequency coupling. Therefore, the activation function $\phi(u_k)$ in (3.53) for all $k = 1, \ldots, N$ is:

$$\phi(u_k) = \beta_k^{-1}(t)u_k/|u_k| \qquad \forall u_k \neq 0 \tag{3.55}$$

The derivative $\phi'(u_k)$ used in the calculation of D can be approximated by:

$$\phi'(u_k) = \beta_k^{-1}(t)(|u_k|^{-1} - u_k^2|u_k|^{-3}) \qquad \forall u_k \neq 0 \tag{3.56}$$

Alternate fixed-point algorithm

For every $i = 1, \ldots, N$,

$$\underline{w}_{if}^{+} \leftarrow \mathcal{E}\{\underline{x}(\underline{w}_{if}^H \underline{x})^* \phi(|\underline{w}_{if}^H \underline{x}|^2)\} - \mathcal{E}\{\phi(|\underline{w}_{if}^H \underline{x}|^2) + |\underline{w}_{if}^H \underline{x}|^2 \phi'(|\underline{w}_{if}^H \underline{x}|^2)\}\underline{w}_{if} \tag{3.57}$$

$$W_f \leftarrow W_f (W_f^H W_f)^{-0.5} \tag{3.58}$$

where $W_f = [\underline{w}_{1f}, \underline{w}_{2f}, \ldots, \underline{w}_{Nf}]$. The time-frequency model is introduced by:

$$\phi(u_k) = 1/(\beta_k(t)(0.1 + u_k)) \qquad \phi'(u_k) = -1/(\beta_k(t)(0.1 + u_k)^2) \tag{3.59}$$

The next step is to remove the *permutation ambiguity* by applying the *likelihood ratio jump* solution.

An important issue is the *spectral shape ambiguity*. In [Car98b], Cardoso shows that we can remove this ambiguity by focusing on the observation spaces containing each source rather than on the columns of the mixing matrix. We will use the analysis introduced earlier on to return the separated sources to the observation space, and remove the introduced scaling ambiguity. Thus, we will have N estimates of each source. Denoting the estimated unmixing matrix as W_f, the prewhitening matrix as V_f for each frequency bin f, then the separated sources, *observed at each microphone*, are given by:

$$\tilde{s}_{i,x_j}(f,t) = [V_f^{-1} W_f^{-1}]_{ji} u_i(f,t), \ \forall i, j = 1, \ldots, N \tag{3.60}$$

where $\tilde{s}_{i,x_j}$ is the *i-th* estimated source observed at the *j-th* microphone.

3.6 Evaluation

It is not our intention to provide an exhaustive comparison of the many different approaches to BSS with convolutive mixtures. Instead, we present

several experiments to demonstrate that the proposed fast FD-ICA framework can produce fast and good quality separation, providing a robust solution for the permutation problem.

3.6.1 Performance metrics for convolutive mixtures

We have to introduce a new set of metrics for the evaluation of convolutive mixtures separation systems, mainly inspired by the ones introduced in section 2.2.10. As a result, during the framework's evaluation and in other parts of our work, we used the following metrics.

- *Improvement in Signal-to-Noise Ratio* (ISNR) achieved at each microphone. This metric is also referred to as *Noise Reduction Rate* (NRR) in [SKS01]. Note that ISNR can be used as a performance metric, as the sources are observed at the microphones.

$$ISNR_{i,j} = 10 \log \frac{\mathcal{E}\{(s_{i,x_j}(n) - x_j(n))^2\}}{\mathcal{E}\{(s_{i,x_j}(n) - \tilde{s}_{i,x_j}(n))^2\}} \tag{3.61}$$

  where x_j is the mixed signal at the *j-th* microphone, $\tilde{s}_{i,x_j}$ is the *i-th estimated* source observed at the *j-th* microphone and s_{i,x_j} is the *i-th original* source observed at the *j-th* microphone. This actually compares the signal before and after the unmixing stage with the original signal simulated or recorded alone in the room. As ICA algorithms do not perform dereverberation, it is fairer for the algorithm's performance to compare the estimated signals with the original signals simulated/recorded alone in the room, rather than the original signals.

- *Distortion* along the frequency axis. This is based on the distortion metric proposed by Schobben et al [STS99].

$$D_{i,j}(f) = 10 \log \frac{\mathcal{E}\{|\mathrm{STFT}\{s_{i,x_j}(n)\} - \mathrm{STFT}\{\lambda_{ij}\tilde{s}_{i,x_j}(n)\}|^2\}}{\mathcal{E}\{|\mathrm{STFT}\{s_{i,x_j}(n)\}|^2\}} \tag{3.62}$$

  where $\lambda_{ij} = \mathcal{E}\{s_{i,x_j}(n)^2\}/\mathcal{E}\{\tilde{s}_{i,x_j}(n)^2\}$ is just a scaling parameter, ensuring that the two signals are normalised to the same scaling. This

metric can visualise the performance of the metric along the frequency axis. It can also be interpreted as $1/SNR_{i,j}$.

3.6.2 Experiment 1

In our initial experiment, we created a synthetic convolutive mixture of two speech sources (3 secs at 16 kHz) that illustrates the permutation problem in the Smaragdis algorithm. The synthesised acoustic paths consisted of an initial delay followed by single echo. The echo times were between 1 and 5 milliseconds and echo strengths between 0.1 and 0.5 of the direct path signal.

Spectrograms of the separated sources are given in figure 3.2 along with equivalent separations for the Smaragdis algorithm. It is clear that the permutation inconsistencies that occurred in the Smaragdis case are no longer present. Omitting the LR step in our algorithm seems to produce the same permutation errors as in Smaragdis's case. In both cases, the frame size was 2048 samples (approximately 150ms) with a frame overlap of 50%. However, while the Smaragdis algorithm required about 500 iterations (cycles through all the data) to reach convergence, the fast FD-ICA framework required only 50. This is very typical of the dramatic improvement in efficiency that can be achieved using Fast ICA techniques.

3.6.3 Experiment 2

The second experiment was chosen to test the algorithm's ability in highly reverberant conditions. To do this, we used Westner's room acoustic data. Westner [Wes] placed a number of microphones and loudspeakers in a conference room and measured the transfer function between each speaker and microphone position. Using his *roommix* function, one can simulate any of the measured speaker-microphones configurations in that conference room, generating a very challenging data set. For our experiment, we placed our sources to speaker positions 1 and 2 and we used microphones 2 and 1 to capture the auditory scene, according to Westner's configuration [Wes].

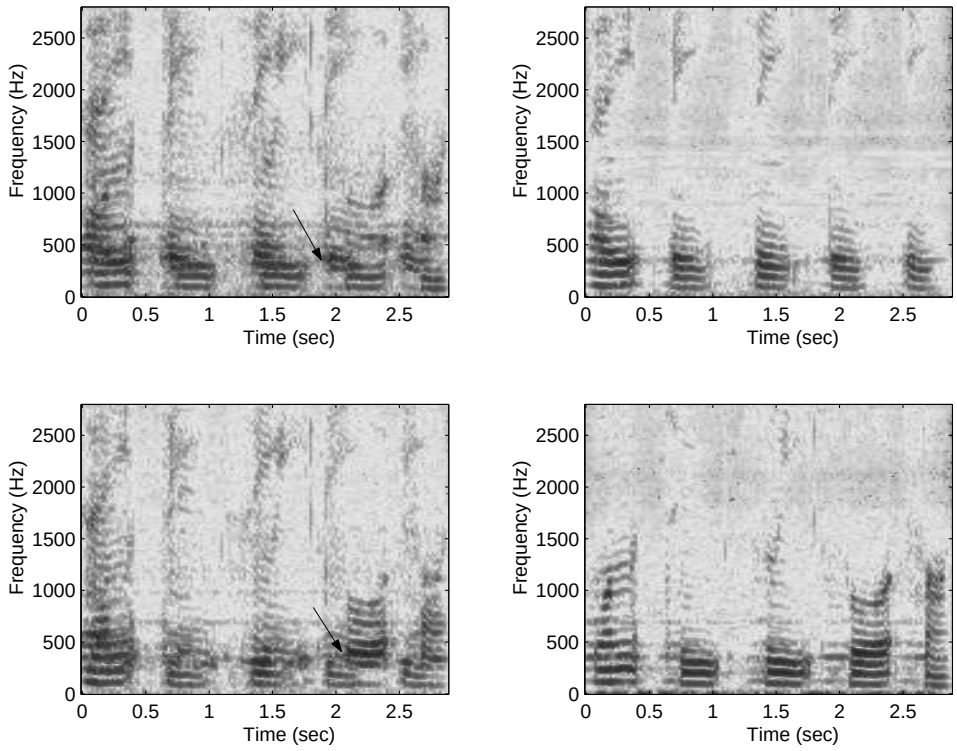

Figure 3.2: Permutation problem illustrated. Separated sources using the Smaragdis algorithm (left) and the algorithm proposed in section 3.4.1 (right). Permutation inconsistencies are highlighted with arrows.

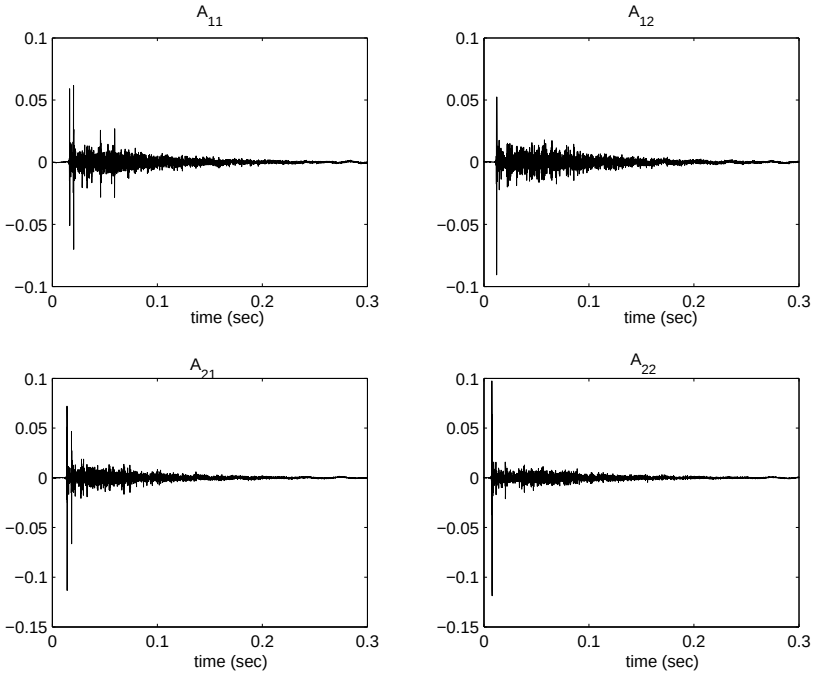

Figure 3.3: The four filters modelling the room acoustics created by Westner's *roommix* function.

An example of the simulated room impulse responses used in this experiment is depicted in figure 3.3. The room acoustics have substantial reverberation for several hundred milliseconds and therefore this experiment is expected to be very challenging.

We applied the algorithm to speech data (around 7secs at 16KHz), using a STFT frame size of around 500 msecs with 75% overlapping and a Hamming window. The fast FD-ICA algorithm managed to reduce the crosstalk by a considerable amount. Choosing a long frame length is inevitable, as it needs to be much greater than the length of the mixing filters, so that the convolution is actually transformed into multiplication in the frequency domain. The fact that reverberation continued beyond the frame length means that the transfer function can not be perfectly modelled.

It should be noted that one drawback of our current approach is that we are attempting to reconstruct the signals at the microphones. Thus, the

reverberation is still present on the separated sources. One possible solution to this problem has recently been proposed in [SD01].

3.6.4 Performance Measurements

To quantify the performance of our two fast implementations and compare it against a natural gradient update scheme, we measured the *Improvement in Signal-to-Noise Ratio* (ISNR) achieved at each microphone. The ISNR results for the experiments described above are presented in table 3.1. These clearly demonstrate the superiority of the fast learning algorithm when faced with a challenging acoustical environment. In addition, we notice that the two approaches have similar performance. Thus, for the rest of the analysis, we will refer generally to the fast FD-ICA framework without specifying which of the two versions we are using.

In figure 3.4, we compare the performance of the fast FD-ICA framework with the natural gradient (NG) algorithm in the Westner case. We can see the improvement in convergence speed and separation quality. In this plot, we can also see that the actual speed of the proposed framework, as it converges in around 20 iterations.

We can also measure the *distortion* along the frequency axis, as proposed by Schobben et al [STS99]. In figure 3.5, we plot $D_{1,1}$ and $D_{1,2}$ for Exp. 2 using fast FD-ICA along frequency. We can see that the distortion remains negative along the greatest part of the spectrum, (significantly lower compared to the NG approach) except for some high-frequency areas and some specific frequency bands. It may be that the source separation problem is ill-determined at these frequency bands or the signal levels are low.

3.6.5 Computational Cost

The computational cost of the Fast FD-ICA framework is slightly increased, compared to the natural gradient framework. We have to consider the extra cost introduced by the fast algorithm and the Likelihood-Ratio jump. In terms of *floating point operations*, the "fixed-point" algorithm requires 1.45

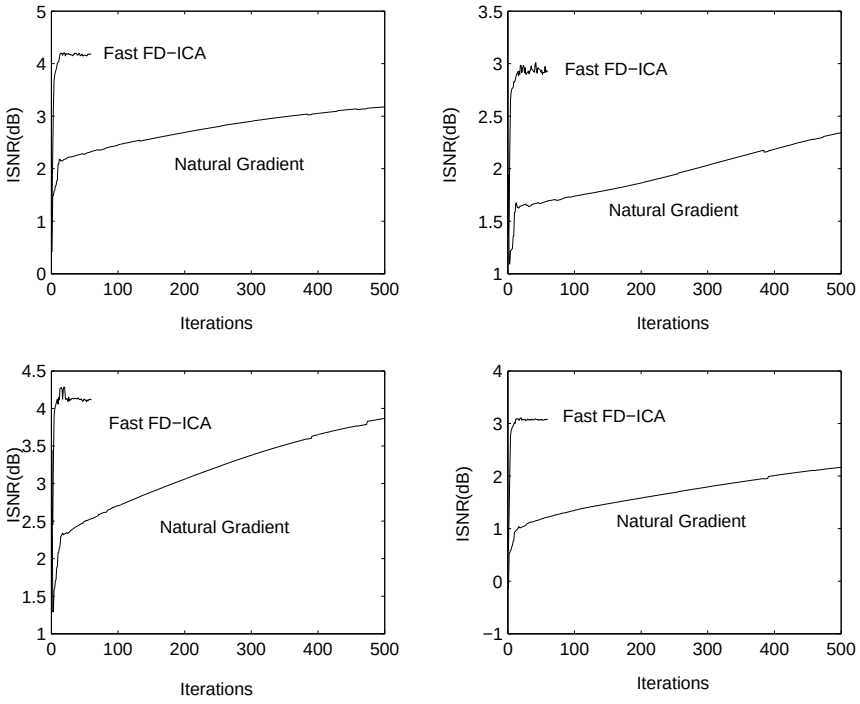

Figure 3.4: Comparison of the fast FD-ICA algorithm with the natural gradient approach in the Westner case. We can see the improvement in convergence speed and in separation quality

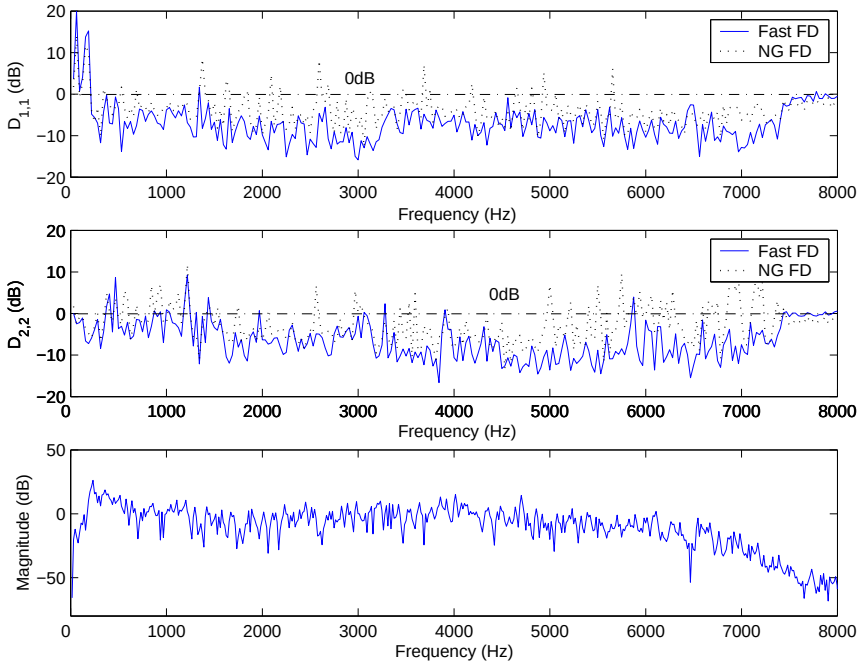

Figure 3.5: Measuring distortion along frequency for the NG FD-ICA and the fast FD-ICA case.

Table 3.1: ISNR (dB) measurements for the two versions of the fast FD-ICA framework (after 50 iterations) and the natural gradient algorithm (after 500 iterations). We observe that the two algorithms perform similarly.

	$\texttt{ISNR}_{1,1}$	$\texttt{ISNR}_{2,1}$	$\texttt{ISNR}_{1,2}$	$\texttt{ISNR}_{2,2}$
Exp.1 Fast FD-ICA 1	8.02	3.92	6.79	4.85
Exp.1 Fast FD-ICA 2	9.1	3.85	6.22	4.56
Exp.1 Nat.Grad.	5.33	1.21	4.92	2.40
Exp.2 Fast FD-ICA 1	4.19	3.09	4.18	3.40
Exp.2 Fast FD-ICA 2	3.94	2.98	3.92	3.26
Exp.2 Nat.Grad.	3.18	2.34	3.87	2.17

times more *flops* per iteration than the natural gradient algorithm. Including the LR jump, it requires 2.02 times more flops per iteration. The above preliminary evaluation was performed using MATLAB's command *flops* that counts the number of floating point operations performed. Considering that the new framework converges in 10-30 times fewer iterations, we can all see the overall gain in computational cost and convergence speed. However, the computational cost of the LR jump increases significantly with more than 2 sources. Working on a pairwise basis with N sources, the cost of the LR jump will scale quadratically with N.

3.7 Other Extensions

In this section, we consider several problems encountered in the frequency-domain ICA framework, concerning the aliasing introduced by the Fourier transform and the effect of the frame length in the Short-Time Fourier Transform on the estimator's performance. Possible solutions to rectify these problems and enhance the performance of the source separation algorithms will be presented.

3.7.1 Aliasing in the Frequency domain framework

One of the first considerations of the convolutive source separation problem is the choice of the unmixing domain, i.e. the domain of adaptive filtering. A great part of the proposed solutions prefer not to work in the *time-domain*, the main reason being the computational cost of the convolution. Although using *fast convolution* schemes [LLYG03], it is possible to reduce the computational cost. However, performing the unmixing in a subband architecture, we can use different adaptation rates in each subband, which is not possible in a time-domain implementation. As mentioned earlier in this chapter, the time-domain is not the ideal framework for source modelling either.

Following this analysis, filtering adaptation and implementation are usually performed in the *frequency domain*, mainly due to the following property of the Fourier Transform:

$$x(n) = \alpha(n) * s(n) \rightleftharpoons X(f) = A(f)S(f) \tag{3.63}$$

where n represents the time index, f frequency index and $*$ represents the *linear convolution*.

Multiplication in the Short-Time Fourier Transform domain is equivalent to *circular convolution* in the time domain. One can approximate the *linear convolution*, as a *circular convolution* and therefore approximate this property on (3.63) by applying the Short-Time Fourier Transform (STFT)

giving:

$$X_i(f,t) \approx \sum_{j=1}^{N} A_{ij}(f) S_j(f,t) \qquad i = 1, \ldots, N \tag{3.64}$$

However, the two types of convolution are equivalent only, if the DFT length L is twice the length of the room transfer functions $L = 2K$ [OS89]. In a real room situation, we can not always ensure that the DFT length is twice the length of the room transfer function. As a result, this approximation usually introduces errors.

Aliasing introduced by the Fourier Transform

To understand the nature of this approximation, it is instructive to consider the DFT (and more specifically the Short Time Fourier Transform) as a bank of *critically-sampled, narrow-band* filters. Critical sampling implies that only 1 in P samples is used in each band (assuming a P-band uniform filterbank). The approximation error then manifests itself in the form of aliasing between neighbouring frequency bins [WP02, GV92]. In figure 3.6, one can see the frequency response of a 16-point DFT filterbank. We can see that, in fact, the Fourier Transform can be interpreted as a very poor filterbank. Aliasing between neighbouring bins starts at relatively high signal levels ($\sim 4dBs$), which can introduce distortion in the analysis and reconstruction part of the source separation algorithm.

Possible solutions to suppress aliasing

There are several methods that can possibly help to overcome the aliasing of the FFT.

If the length of the room transfer functions K is substantially shorter than the DFT length $K \ll L$ then the filter will not change much between frequency bins and the aliasing effect will be suppressed. However, room acoustics tend to have long transfer functions such that we might expect $K > L$. Thus, this argument does not apply in our case.

Aliasing can be reduced by simply oversampling the DFT filterbank

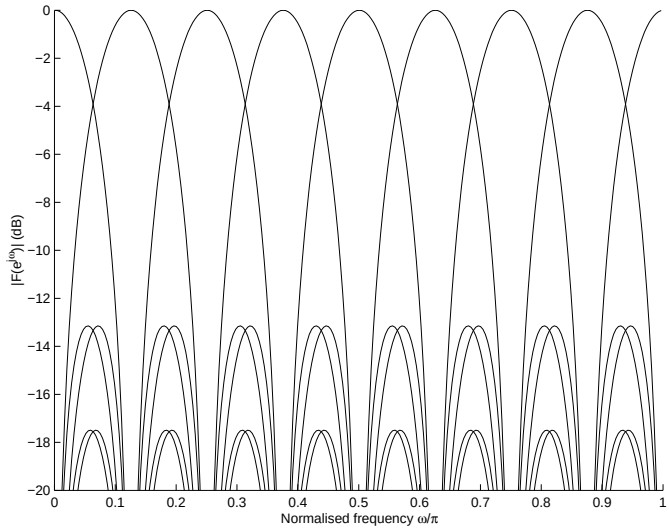

Figure 3.6: Filter bank characteristic of a 16-point DFT.

[WP02, GV92]. Given the requirement for wide gain adjustment in source separation, critical sampling is insufficient. Although oversampling increases the data rate, it is the price that must be paid for gain adjustability without aliasing [BS98]. Lambert also showed the effectiveness of oversampling in alias cancelation, in terms of the filter's Laurent series expansion [Lam96]. Further improvements can also be obtained through the careful selection of subband filter parameters. This implies that we can replace the FFT framework with a properly designed filterbank, featuring ideally "perfect" bandpass filters. Although this is impossible, developments have led to filterbanks with slightly overlapping subbands, designed in such a way that they closely approximate ideal bandpass filters and only small amounts of oversampling are necessary [WP02].

Another possible approach to aliasing reduction is to include adaptive cross terms between neighbouring frequency bins [GV92]. This technique seems to introduce additional complexity and is not very appropriate for the already computationally expensive framework of audio source separation.

Experiments

To see the effect of aliasing, we applied the first Frequency Domain algorithm (see section 3.4.1) using a 4096 point windowed DFT filterbank with differing amounts of oversampling to a mixture of speech signals recorded (separately) in a real room environment. Performance is measured in terms of *Distortion* as introduced by Schobben et al [STS99] and explained in section 3.6.1.

$$D_{i,j}(f) = 10 \log \frac{\mathcal{E}\{|\mathrm{STFT}\{s_{i,x_j}(n)\} - \mathrm{STFT}\{\lambda_{ij}\tilde{s}_{i,x_j}(n)\}|^2\}}{\mathcal{E}\{|\mathrm{STFT}\{s_{i,x_j}(n)\}|^2\}} \tag{3.65}$$

where $\lambda_{ij} = \mathcal{E}\{s_{i,x_j}(n)^2\}/\mathcal{E}\{\tilde{s}_{i,x_j}(n)^2\}$.

Our observations are broadly similar to those in [NGTC01]. The distortion was significantly improved with oversampling. Figure 3.7 shows the *increase* in distortion introduced with 50% frame overlap in comparison to that with 90% overlap as a function of frequency. It is clear that oversampling predominantly benefits the high frequency distortion. The overall distortion values are summarised in Table 3.2.

Table 3.2: Average along frequency Distortion (dB) performance for differing amounts of oversampling.

	$D_{1,1}$	$D_{2,1}$	$D_{1,2}$	$D_{2,2}$
Mixing 50% overlap	-3.78	-4.45	-6.59	-2.69
Mixing 75% overlap	-4.56	-4.90	-7.21	-3.43
Mixing 90% overlap	-5.86	-6.26	-8.80	-4.99

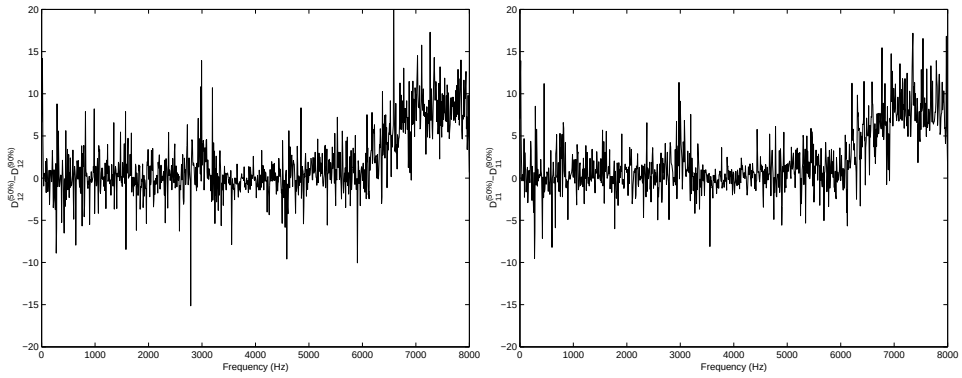

Figure 3.7: Difference in distortion between the case of 50% overlap and 90% overlap for source 1 at microphone 1 (left plot), and microphone 2 (right plot)

3.7.2 Effect of frame size

In audio source separation of signals recorded in a real room, the room transfer functions are usually quite long (i.e. $>$ 100ms). Therefore, to adequately model these, we need to make the frame size large ($>$2048 at $16KHz$ sampling). However, as the frame size increases, the signal captured by the frame tends to be less stationary and we end up averaging over different quasi-stationary segments of audio. The result is the signal tends to be more Gaussian (roughly via the *central limit theorem*). Even without large frame sizes the presence of the reverberation itself will tend to make the signal more Gaussian for the same reasons. As one of the arguments for working in the frequency domain was to increase the nonGaussianity of the signals, we see that there will be a trade-off between large frame sizes that can fully describe the room acoustics and small frame sizes where nonGaussianity is greatest.

To explore this trade-off, we examined the statistics of a single frequency bin of a windowed DFT as a function of frame size. Specifically, we filtered a speech signal with the following filter:

$$h(n) = w(n)e^{-j\omega_0 n} \qquad n = 1, \ldots, K \tag{3.66}$$

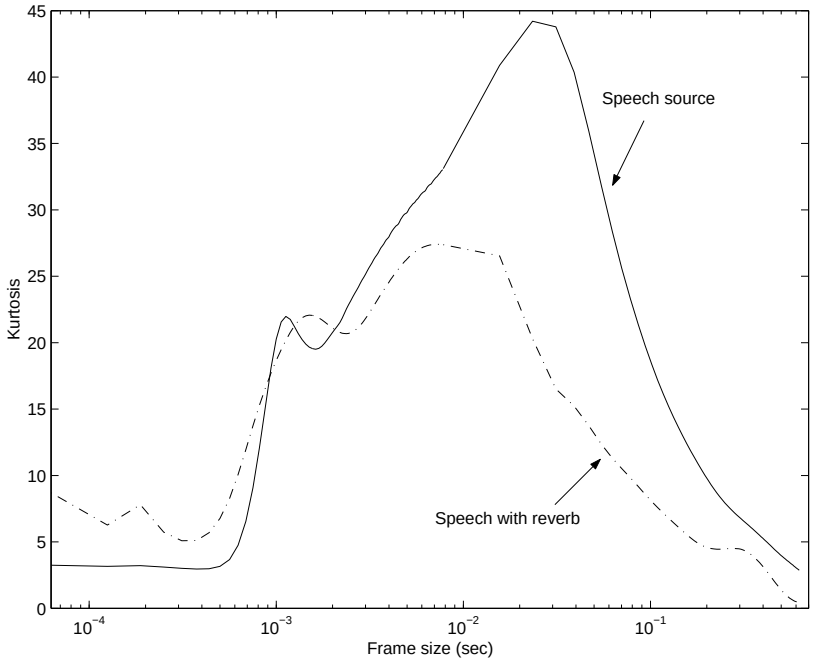

Figure 3.8: Frame size effect on signal's statistical properties (i.e. estimated kurtosis/sample).

where $w(n)$ represents the window, $\omega_0 \in [-\pi, \pi]$ represents the frequency at which we want to study our signal's statistics and finally K is the analysis frame length. We used a Hamming window and observed the signal at 1kHz. We measured the signal's nonGaussianity at frame lengths varying from $6ms$ to $625ms$. *Kurtosis* is used as a significant measure of nonGaussianity. As our data are complex, the "normalised" kurtosis is given by the following expression:

$$kurt(x) = \frac{\mathcal{E}\{\mid x \mid^4\}}{\mathcal{E}\{\mid x \mid^2\}^2} - 2 \tag{3.67}$$

We then repeated the measurement for a reverberant version of the same speech signal (using an estimated room transfer function [Wes]). Figure 3.8 shows the level of estimated kurtosis of the signals as a function of frame size.

The results follow our intuition. For very small frame sizes the estimated kurtosis tends to that for the time domain source model. Although the signals still have positive kurtosis they are only weakly nonGaussian. As the frame size increases, we are able to exploit the sparsity of the sources in the STFT domain.

For the speech signal with no reverberation the estimated kurtosis is maximum for a frame size of about $20 - 30ms$ (this is the frame size that is commonly used for efficient speech coding). Note at this value the estimated kurtosis is 10 times that for the time domain model. Finally, as the frame size grows very large, we begin to average over a sufficient number of stationary segments and the estimated signal kurtosis tends towards zero. The effect of reverberation (dashed line) is to reduce the peak value of the estimated kurtosis. However, the other general trends persist.

One conclusion is that when source modelling in the frequency domain we cannot afford to choose long frame sizes with $K \gg L$ since this will merely render our signal statistics Gaussian. A similar conclusion was drawn by Araki et al [AMNS01].

Instead, we can choose $L \approx K$ and use oversampling as explained in section 3.7.1. However, when in a highly reverberant environment, even this

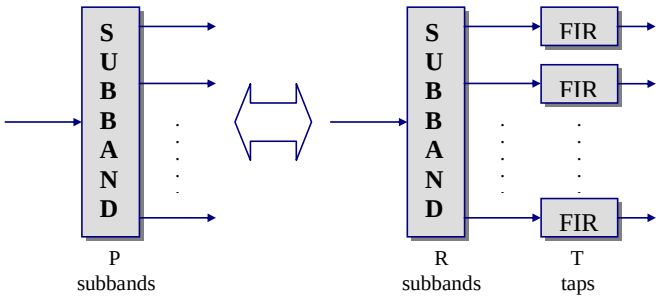

Figure 3.9: A possible framework to solve the frame size problem.

condition may lead to poor performance.

In this situation a possible solution, often adopted in subband adaptive filtering algorithms is to use a mixture of subband and convolutional filtering, as depicted in Figure 3.9, where a P subband filterbank is replaced by a smaller R subband structure and a *short* unmixing filter of size T (such that $P = RT$) [Mit98, HCB95]. This would enable us to work with highly sparse signals, while also reducing to some extent the permutation problem. A typical example would be to decompose a $P = 4096$ FFT filterbank into a $R = 1024$ FFT filterbank and a $T = 4$ tap filter for each subband. Although this might keep the frame size small, we will have to use a convolutive unmixing algorithm for small-size filters, instead of the instantaneous unmixing algorithm. This might increase the computational complexity, however, the effectiveness of this process is under current investigation.

3.8 Conclusion

In this chapter, we have addressed the problem of *convolutive mixtures source separation* using *Frequency-Domain Independent Component Analysis*. A method to solve the *scale ambiguity* was proposed. A novel method to solve the permutation ambiguity using a time-frequency source model along with a Likelihood Ratio jump solution was proposed. The methods seemed to rectify the permutation problem in the majority of the cases. Two

fast "fixed-point" algorithms were adapted to work with complex numbers and incorporate the solution for the permutation problem. All these modules were put together to form a unifying frequency domain ICA framework that manages to perform fast and robust source separation in the majority of the cases. Several tests were performed to test the efficiency of the proposed framework with encouraging results.

Finally, we saw that the STFT can be considered a critically sampled filterbank that introduces aliasing between the subbands. The use of oversampling to reduce the aliasing was investigated, plus other possible options were discussed. The increase in Gaussianity due to the long FFT frames (due to long room transfer functions), used in audio convolutive mixtures, was also discussed. Possible solutions were also presented

Chapter 4

Using Beamforming for permutation alignment

4.1 Introduction

The theory of *Array Signal Processing* was established in the late 70s and early 80s with application to sonar, radar and telecommunication devices. The use of a *structured array of sensors* rather than a simple sensor can enhance the receiver's capabilities in terms of identifiability of sources, directional tracking and enhanced reception [vVB88]. The idea of array signal processing was introduced also in the case of audio signal analysis [GJ82]. The main areas of application for an audio beamformer are *blind deverberation*, *source localisation*, *hearing aids*, *blind enhancement* and *speaker arrays*. A common application of many array signal processing systems was to *steer the overall gain pattern of the array sensors to focus on a desired source coming from a specific direction, while suppressing possible sources coming from other directions* (*beamforming*).

In fact, the source separation systems, as described analytically in Chapter 2, can be regarded as array signal processing systems. A set of sensors arranged randomly in a room to separate the sources present is effectively a beamformer. However, in source separation systems the sensors can be

arbitrarily placed in the room and usually an equal number of sensors and sources is sufficient to perform separation. In array signal processing systems, we assume a known arrangement of the sensors and usually the number of sensors is greater than the number of sources. Using more sensors than sources enables us to use "subspace techniques" to estimate the directions of arrival for each source.

In this chapter, we investigate the relation between beamforming and blind source separation in more detail. We will also look at the application of beamforming for permutation alignment on Frequency-Domain Independent Component Analysis for Blind Source Separation of convolutive mixtures. In addition, we demonstrate that one can apply common "subspace techniques" (such as MuSIC) even in the case of an equal number of sources and sensors.

4.2 Array Signal Processing

In this section, we give a brief background on Array Signal Processing theory that will be used in our analysis later on.

4.2.1 Definition

Assume that you have an array of M sensors $\underline{x}(n) = [x_1(n)\ x_2(n)\ \ldots\ x_M(n)]^T$ capturing an auditory scene, where n represents discrete time index. Assume there are N sources in the auditory scene $\underline{s}(n) = [s_1(n)\ s_2(n)\ \ldots\ s_N(n)]^T$. Suppose that the distance of the sensors from the origin of the array is d_k. The source signals arrive at the origin of the array's coordinates system at an angle θ_i. These angles θ_i are called *Directions of Arrival* (DOA) of the sources in the *far-field* approximation.

Assuming that we deal with *narrowband signals*, we can say that $s_i(n) \approx \alpha e^{j2\pi f_c n}$, where f_c is the carrier frequency. Considering only one source $s_1(n)$ and no reverb, one can say that each sensor captures the incoming signal with a time lag (phase difference) of T_i (see figure 4.1). The delays T_i are functions of the signals' DOA θ_i.

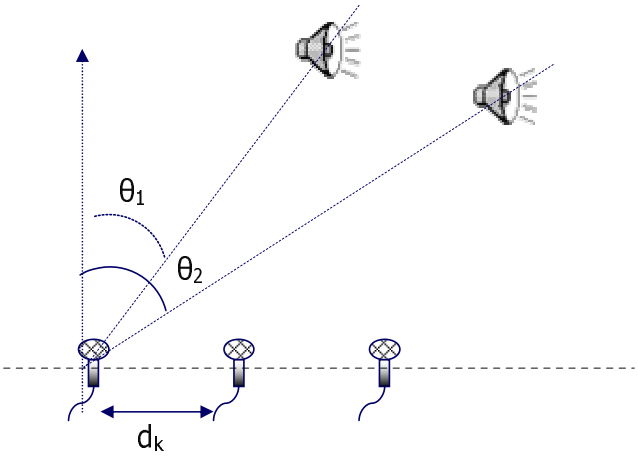

Figure 4.1: An array signal processing setup: 3 sensors and 2 sources with Directions of Arrival θ_1 and θ_2 respectively.

$$x_1(n) = s_1(n) \tag{4.1}$$

$$x_2(n) = s_1(n - T_1) \approx \alpha e^{-j2\pi f_c T_1} s_1(n) \tag{4.2}$$

$$\dots \quad \dots \quad \dots \quad \dots \quad \dots \tag{4.3}$$

$$x_M(n) = s_1(n - T_M) \approx \alpha e^{-j2\pi f_c T_M} s_1(n) \tag{4.4}$$

Assuming equal distance d between the sensors, a *far-field* approximation and discrete-time signals with time index n, we have

$$\underline{x}(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \\ \dots \\ x_M(n) \end{bmatrix} \approx \begin{bmatrix} 1 \\ \alpha e^{-j2\pi f_c T} \\ \dots \\ \alpha e^{-j2\pi f_c (M-1)T)} \end{bmatrix} s_1(n) = \underline{a}(\theta_1) s_1(n) \tag{4.5}$$

where $T = dsin\theta_1/c$, where $c = 340m/sec$ is the velocity of sound in air. For multiple sources, we have

$$\underline{x}(n) = \sum_{k=1}^{N} \underline{a}(\theta_k) s_k(n) = [\underline{a}(\theta_1)\ \underline{a}(\theta_2)\ \dots\ \underline{a}(\theta_N)]\underline{s}(n) \tag{4.6}$$

where $A = [\underline{a}(\theta_1)\ \underline{a}(\theta_2)\ \ldots\ \underline{a}(\theta_N)]$ can be considered as an equivalent to the mixing matrix A (see Chapter 2). As we can see the model of an array is similar to the Blind Source Separation model. However, in array signal processing, the model incorporates more geometrical information (i.e. Directions of Arrival, position of sensors), whereas source separation the mixing model is more general, employing statistical information about the sources only.

The *calibration of the array* is an important parameter in array signal processing. The performance of an array can be limited due to calibration errors relating to the electrical and/or geometrical characteristics of the array. Calibration errors may make it impossible to track the DOA of the incoming sources. Therefore, *array calibration* is significant for any array system.

The ultimate goal of the above mentioned array signal processing scenario is to find a set of weights that can "steer" the gain pattern of the array, so that we can isolate one source of interest and suppress the others. This procedure is usually known as *beamforming* and the resulting filter a *beamformer*.

The problem of beamforming is usually divided into many subproblems. The first aspect is to identify the number of sources that are present. The second subproblem is to identify the sources' DOA $\theta_1, \theta_2, \ldots, \theta_N$. Finally, we unmix the sources by steering the beampattern of the array to provide maximum gain to the direction of the corresponding source or more accurately to place *nulls* (zero gain) to all other sources present in the auditory scene. More specifically,

$$u_i(n) = \sum_{k=1}^{M} w_k^*(\theta_i) x_k(n) = \underline{w}^H(\theta_i)\underline{x}(n) \tag{4.7}$$

where $\underline{w}(\theta_i)$ models the filter coefficients (beamformer) that maximize the overall gain of the array towards the angle θ_i. To retrieve all sources

$$\underline{u}(n) = [\underline{w}(\theta_1)\ \underline{w}(\theta_2)\ \ldots\ \underline{w}(\theta_N)]^H \underline{x}(n) \tag{4.8}$$

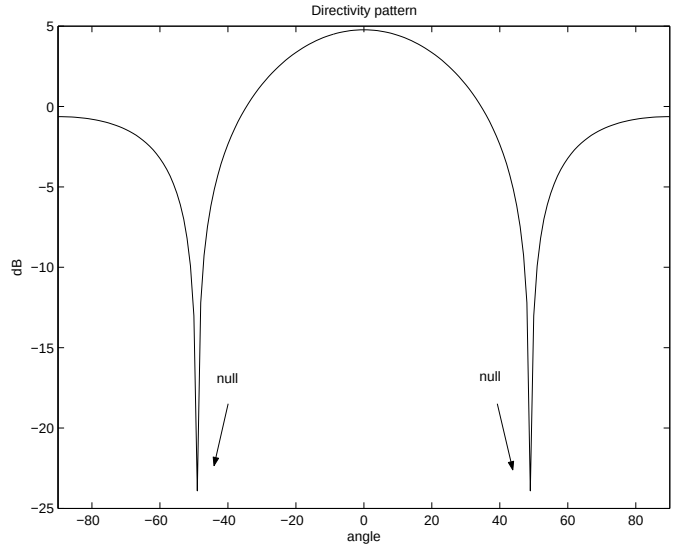

Figure 4.2: Directivity pattern of a three microphone array.

Having estimated the unmixing vector (beamformer), one can plot the gain pattern of the array along $\theta \in [-\pi/2, \pi/2]$. This constitutes the *directivity pattern* of the array. In figure 4.2, we can see the directivity pattern of a three microphone array. We can see that the array's gain pattern is steered at 0°, while two nulls are placed at $\pm 49^\circ$. A beamformer has $M-1$ *Degrees of Freedom* (DOF), i.e. can suppress $M-1$ unwanted sources.

In the following section, we are going to examine each of the subproblems that constitute the full beamforming problem. As stated before, we have to estimate the number of sources N, the Directions of Arrival θ_i and finally the unmixing - beamforming vectors $\underline{w}(\theta_i)$.

4.2.2 Number of sources and Directions Of Arrival (DOA) estimation

If the number of sensors M is greater than the number of sources, we can estimate the number of sources and the DOA using *subspace methods* [MS00].

In the following analysis, we will also assume some additive, isotropic noise $\underline{\epsilon}(n)$ to our model, i.e. $C_\epsilon = \sigma_\epsilon^2 I$, where σ_ϵ is the standard deviation of

the noise. Consider $A = [\underline{a}(\theta_1)\ \underline{a}(\theta_2)\ \ldots\ \underline{a}(\theta_N)]$, then the model is described as follows:

$$\underline{x}(n) = A\underline{s}(n) + \underline{\epsilon}(n) \tag{4.9}$$

Calculating the covariance matrix for $\underline{x}$. We have:

$$C_x = \mathcal{E}\{\underline{x}\ \underline{x}^H\} = A\mathcal{E}\{\underline{s}\ \underline{s}^H\}A^H + \mathcal{E}\{\underline{\epsilon}\ \underline{\epsilon}^H\} \tag{4.10}$$

$$C_x = AC_sA^H + \sigma_\epsilon^2 I \tag{4.11}$$

where C_x is the covariance matrix of $\underline{x}$ and C_s is the covariance matrix of $\underline{s}$.

As $M > N$ the rank of C_x will be equal to N and its eigenvalues will be $\lambda_1, \lambda_2, \ldots, \lambda_N, 0, \ldots, 0$ in the noiseless case. Assuming that the additive noise is isotropic, i.e. $C_\epsilon = \sigma_\epsilon^2 I$ and $\sigma_{s_i}^2 \gg \sigma_\epsilon^2$ then the eigenvalues will be shifted by σ_ϵ^2. More specifically, the eigenvalues of C_x will be $\lambda_1 + \sigma_\epsilon^2, \lambda_2 + \sigma_\epsilon^2, \ldots, \lambda_N + \sigma_\epsilon^2, \sigma_\epsilon^2, \ldots, \sigma_\epsilon^2$ and the corresponding eigenvectors of C_x will be $\underline{e}_1, \underline{e}_2, \ldots, \underline{e}_N, \underline{e}_{N+1}, \ldots, \underline{e}_M$.

As we can see, a criterion in order to determine the number of sources N present in the auditory scene is counting the number of eigenvalues being above a "noise floor". In addition, the small eigenvalues can give an estimate of the level of noise. That is the method usually followed to determine the number of sources and noise level estimation in array processing problems.

Subspace Methods - MuSIC

DOA estimation can be performed in a number of different ways. However, if $M > N$, then a number of *subspace methods*, such as *MuSIC* and *ESPRIT* can be used to provide very accurate estimates for the DOA. In this section, we will briefly describe the *Multiple Signal Classification* (MuSIC) method [Sch86].

Some useful theory from linear algebra can help us establish a practical method to estimate the DOA. This is provided the model fits and the array is calibrated. We can show [MS00] that the space spanned by the columns

of matrix $A = [\underline{a}(\theta_1)\ \underline{a}(\theta_2)\ \ldots\ \underline{a}(\theta_N)]$ is equal to the space spanned by the eigenvectors $\underline{e}_1, \underline{e}_2, \ldots, \underline{e}_N$.

$$\mathsf{span}\{A\} = \mathsf{span}\{[\underline{e}_1, \underline{e}_2, \ldots, \underline{e}_N]\} = \mathsf{span}\{E_s\} \tag{4.12}$$

This subspace can uniquely determine the DOA $\theta_1, \ldots, \theta_N$ of the source signals. To find the DOA, we have to estimate the angles θ that $\underline{a}(\theta) \in \mathsf{span}\{E_s\}$. Assuming that $E_s = [\underline{e}_1, \underline{e}_2, \ldots, \underline{e}_N]$ contains the eigenvectors corresponding to the desired source and $E_n = [\underline{e}_{N+1}, \ldots, \underline{e}_M]$ contains the eigenvectors corresponding to noise, we can form $P = E_s E_s^H$ and $P^{\perp} = (I - E_s E_s^H) = E_n E_n^H$. The DOAs should satisfy the following conditions

$$P\underline{a}(\theta) = \underline{a}(\theta) \text{ or } P^{\perp}\underline{a}(\theta) = 0 \tag{4.13}$$

In practice, we plot the following function

$$M(\theta) = \frac{1}{|P^{\perp}\underline{a}(\theta)|^2} \qquad \forall\ \theta \in [-90, 90] \tag{4.14}$$

The N peaks of the function $M(\theta)$ will denote the DOA of the N sources. An example of the MuSIC algorithm can be seen in fig. 4.3. The MuSIC algorithm is applied in the case of two sources being observed by a symmetric array of three sensors. Plotting $M(\theta)$, we can clearly observe two main peaks, denoting the two directions of arrival.

On the other hand, a well-known problem with some of these suboptimal techniques, such as MuSIC, occurs when two or more sources are highly correlated. Many variants of the MuSIC algorithm have been proposed to combat signal correlation. In addition, in the audio separation setup, one of the basic assumption is the statistical independence of the sources and hence, the sources are considered uncorrelated. Another problem of the MuSIC framework is to identify sources with great proximity. This case will not be examined in our analysis.

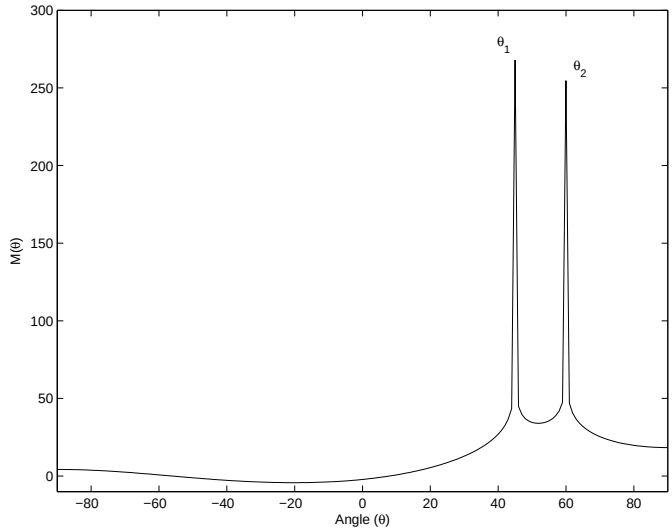

Figure 4.3: Example of the MuSIC algorithm in a 2 sources-3 sensors scenario. The two sources emitting at 45^o and 60^o. Two directions of arrival estimated by the MuSIC algorithm at $\theta_1 = 45^o$, $\theta_2 = 60^o$.

4.2.3 Beamforming - Separation

Having estimated the DOA, we have to estimate the coefficients of the beamforming filter, as the final objective is to unmix the sources. Assuming narrowband signals, one can unmix the source *deterministically* , using the pseudoinverse of the matrix $A = [\underline{a}(\theta_1)\ \underline{a}(\theta_1)\ \ldots\ \underline{a}(\theta_N)]$. However, when additive noise is present, we will still get post-processing additive noise using the pseudoinverse, as:

$$\underline{u}(n) = A^{+}\underline{x}(n) + A^{+}\underline{\epsilon}(n) \tag{4.15}$$

In order to deal with noise, one can unmix the sources using a *Wiener-type method*. Basically, we need to estimate the beamforming (unmixing) vector $\underline{w}_i$ that can separate the source coming from θ_i. From adaptive filter theory, one way to estimate the filter's coefficients is by minimising the mean square error, i.e. the following cost function:

$$\min_{\underline{w}_i} \mathcal{E}\{|\underline{w}_i^H \underline{x}|^2\} = \min_{\underline{w}_i} \underline{w}_i^H C_x^{-1} \underline{w}_i \tag{4.16}$$

$$\text{subject to } \underline{a}^H(\theta_i)\underline{w}_i = 1 \tag{4.17}$$

The analytical solution is given by the following formula:

$$\underline{w}_i = \frac{C_x^{-1}\underline{a}(\theta_i)}{\underline{a}^H(\theta_i)C_x^{-1}\underline{a}(\theta_i)} \tag{4.18}$$

4.2.4 Frequency Domain Beamforming

A *Frequency Domain beamformer* performs beamforming for each frequency bin. Instead of assuming a carrier frequency for narrow-band signals, we form $\underline{a}(\theta_i)$ for every frequency f. More specifically,

$$\underline{a}(\theta_i) = \begin{bmatrix} 1 \\ \alpha e^{-j2\pi fT} \\ \dots \\ \alpha e^{-j2\pi f(M-1)T)} \end{bmatrix} \tag{4.19}$$

As a result, we can estimate a beamformer $\underline{w}_i(f)$ for every frequency bin f and every source i.

Directivity patterns for Frequency Domain Beamformers

Having estimated the beamformers for each frequency f and source i, we can plot the directivity pattern for each frequency f. A more simplified expression for the directivity pattern follows:

$$F_i(f,\theta) = \sum_{k=1}^{N} w_{ik}^{ph}(f)e^{j2\pi f(k-1)d\sin\theta/c} \tag{4.20}$$

where $w_{ik}^{ph} = w_{ik}/|w_{ik}|$ and c is the velocity of sound. In figure 4.4, we can see the directivity patterns of a FD-beamformer. In this case, we simulated a single delay transfer function scenario between 2 sources and 2 sensors. The sensor spacing was $d = 1m$. We observed the following:

First of all, we can spot a consistent null along all frequencies that corresponds to the DOA of the source we want to remove from the auditory scene. This is something we expected. However, we notice that as the frequency f

increases, we get multiple nulls, instead of a single null. More specifically, around $f_{ripple} \approx c/2d$ we start getting multiple nulls in the directivity patterns. This is due to the periodicity of the function $F_i(f, \theta)$, now that the value of frequency f is increasing. This is also known as the *spatial aliasing condition* $d = \lambda/2$, where $\lambda = c/f$.

A *proof* for this threshold in the 2 sensors case follows. The directivity pattern is defined as:

$$F_i(f, \theta) = w_1 + w_2 e^{j2\pi f d \sin\theta/c} \tag{4.21}$$

where $w_1 = e^{j\phi_1}$ and $w_2 = e^{j\phi_2}$ are the beamforming filter's coefficients. We will examine the amplitude square of the function $F_i(f, \theta)$.

$$|F_i(f, \theta)|^2 = F_i(f, \theta) F_i^*(f, \theta) \tag{4.22}$$

$$= (w_1 + w_2 e^{j2\pi f d \sin\theta/c})(w_1^* + w_2^* e^{-j2\pi f d \sin\theta/c}) \tag{4.23}$$

$$= w_1 w_1^* + w_2 w_2^* + w_1 w_2^* e^{-j2\pi f d \sin\theta/c} + w_2 w_1^* e^{j2\pi f d \sin\theta/c} \tag{4.24}$$

As $w_1 w_1^* = w_2 w_2^* = 1$, we have

$$|F_i(f, \theta)|^2 = 2 + e^{j(\phi_1 - \phi_2)} e^{-j2\pi f d \sin\theta/c} + e^{-j(\phi_1 - \phi_2)} e^{j2\pi f d \sin\theta/c} \tag{4.25}$$

$$|F_i(f, \theta)|^2 = 2 + 2\cos(\phi_1 - \phi_2 - 2\pi f d \sin\theta/c) \tag{4.26}$$

We are interested in studying the periodicity of the above function. The periodicity is controlled by the term $2\pi f d \sin\theta/c$ as the term $\Delta\phi = \phi_1 - \phi_2$ represents the offset of the cosine. As $\theta \in [-\pi/2, \pi/2]$, it follows that $-1 \leq \sin\theta \leq 1$ and $-\pi \leq z = \pi \sin\theta \leq \pi$.

Then, if we want to have only one ripple in $|F_i|^2 = 2 + 2\cos(\Delta\phi + 2fdz/c)$ for all $z \in [-\pi, \pi]$, we have to ensure that the cosine remains in its first period, i.e. the argument of the cosine lies in the first period. More specifically, we have to ensure that $2fdz/c \in [0, 2\pi]$ or $2fdz/c \in [-\pi, \pi]$. As $z \in [-\pi, \pi]$ already, the condition for $|F_i|^2$ to have only one ripple is

$$2fd/c \leq 1 \Rightarrow \tag{4.27}$$

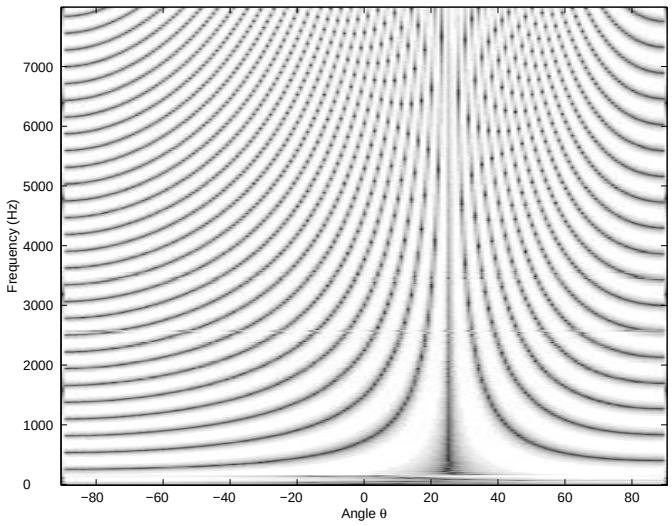

Figure 4.4: Directivity pattern along frequency for a single delay case.

$$f \leq c/2d \tag{4.28}$$

Even with multiple nulls, there will always be a null around the DOA. Directivity patterns can give us an efficient tool to estimate DOA in the case that we do not have more sensors than sources and can not use subspace methods. The null that appears in all frequencies should be a DOA. All directivity patterns of a broadband beamformer that is adjusted to separate a source with a specific DOA should feature a null around that DOA.

4.3 ICA as a Beamformer

Recently, the relationship between *convolutive blind source separation* and *beamforming* has been highlighted. In the context of frequency domain ICA, at a given frequency bin, the unmixing matrix can be interpreted as a *null-steering beamformer* that uses a *blind algorithm* (ICA) to place nulls on the interfering sources. However, we face the ICA permutation ambiguity of ordering the permutations along the frequency axis.

The source separation framework, as described so far, does not utilize any information concerning the geometry of the auditory scene (e.g. Directions

Of Arrival (DOA) of source signals, microphone array configuration). Inclusion of this additional information can help align the permutations using the sources estimated DOA to align the permutations along the frequency axis. Although in a real room recording, i.e. at a given frequency the "approved" DOA is only slightly perturbed, we assume that the main DOA comes from the direct path signal. This is equivalent to approximating the room's transfer function with a single delay. As we will see later on, this approximation is relatively true. In a similar manner to flipping solutions for the permutation problem, the permutations of the unmixing matrices are flipped so that the directivity pattern of each beamformer is approximately aligned.

So far in the ICA model, the sensors could have a totally arbitrary configuration. Hence, the idea of incorporating information about the sensors' arrangement can be interpreted as a *channel modelling* technique. The DOA information and directivity patterns are mainly channel modelling information to our system.

More specifically, having estimated the unmixing matrix $W(f)$ using a fast frequency-domain algorithm as proposed in [MD03], we want to permute the rows of $W(f)$, in order to align the permutations along the frequency axis. We form the following directivity pattern for each frequency bin f.

$$F_i(f,\theta) = \sum_{k=1}^{N} W_{ik}^{ph}(f) e^{j2\pi f(k-1)d\sin\theta/c} \tag{4.29}$$

where $W_{ik}^{ph}(f) = W_{ik}(f)/|W_{ik}(f)|$ is the phase of the unmixing filter coefficient between the k^{th} sensor and the i^{th} source at frequency f, d is the distance between the sensors and c is the velocity of sound in air.

However, in audio source separation we deal with signals that are recorded in a real room. This implies that the sensors capture more than a single delay and are modelled as *convolutive mixtures*. As a result, the directivity patterns of these channels are a bit different to those corresponding to single delay channels. In figure 4.5, we can present the directivity patterns of a real room transfer function, as measured in a university lecture room with a

two sensor one source setup. The sensor spacing was $d = 1m$ and the source was placed $2m$ from the origin of the array.

One can observe that in the real room case, the directivity pattern looks more "smeared". This is due to the fact that the DOA are slightly shifted along frequency. In a real room case with reverb, apart from the direct path signal that defines the "actual" DOA, we have other room's reflections that shift the "actual" DOA arbitrarily at each frequency (see figure 4.5).

On the other hand, the average shift of DOA along frequency is not so significant . As a result, we can spot a main DOA (around 22°). This implies that we can align the permutations in the source separation application, using the DOA.

At this point, we want to stress the reason why we want to use beamforming for permutation alignment only and not for separation. In section 4.2.3, we saw that if we know the DOA of the desired signal, we can form estimators to separate it and suppress the other sources. However, this assumes that we have a single delay scenario and therefore the DOA is consistent along frequency. In a real room scenario, where the DOA is shifted arbitrarily at each frequency, performing beamforming along an average DOA would give very poor quality separation. Instead, the slightly "shifted" DOA can help us identify the correct permutation of separated sources.

The questions that have to be answered are how we can get a good DOA estimate from this directivity pattern and how we can perform permutation alignment using DOA. In the next sections, we will try to address some solutions proposed by other people and the shortcomings we spotted in these problems, plus some novel ideas and a mechanism to apply subspace techniques for permutation alignment.

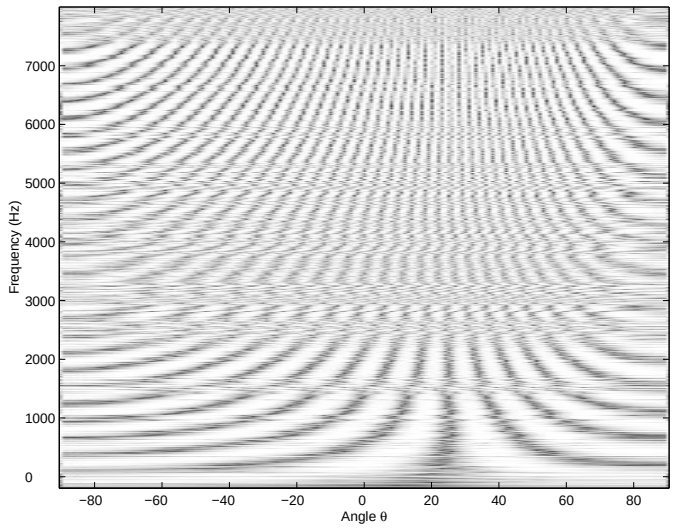

Figure 4.5: Directivity pattern along frequency for a real room transfer function. We can spot a main DOA along frequency, however, it seems to be slightly shifted due to the multipath of the real room transfer function.

4.4 Beamforming as a solution to the permutation ambiguity

4.4.1 DOA estimation ambiguity

In the case of more sensors than sources, DOA estimation is not a difficult task, as we can employ subspace techniques, such as MuSIC and ESPRIT and get very accurate DOA estimates. In a previous paragraph, we saw that exploiting the noise subspace, we can identify the number of sources and the directions of arrival.

Saruwatari et al [SKS01] estimated the DOA by taking the statistics with respect to the direction of the nulls in all frequency bins and then tried to align the permutations by grouping the nulls that exist in the same DOA neighbourhood. On the other hand, Ikram and Morgan [IM02] proposed to estimate the sources DOA in the lower frequencies, as they don't contain multiple nulls. Parra and Alvino [PA02] used more sensors than sources

along with *known* source locations and added this information as a geometric constraint to their unmixing algorithm.

A mechanism for DOA estimation

In figure 4.6, we plot the average beampatterns along a certain frequency range $\mathcal{F}$, assuming a two sensor setup, where $d = 1m$. More specifically, we plot the average beampatterns between $0 - 2KHz$, $2 - 4KHz$, $4 - 6KHz$ and $6 - 8KHz$. We can see that in the lower frequencies, we get clear peaks denoting the directions of arrival. However, in higher frequencies, we get peaks at the same angle, but also multiple peaks around the main DOA. Observing the higher frequencies, we can not really define which of the peaks is the actual DOA. As a result, we may want to use only the lower subband $(0 - 2KHz)$ for DOA estimation.

We can show that averaging beampatterns over the lower frequencies, we can get localised peaks around the DOA. Assume a two sensors setup. Following the analysis in section 4.2.4, we have an expression for $|F_i(f,\theta)|^2$ (4.26). Averaging (4.26) over a frequency range $\mathcal{F}$ that contains K_1 frequency bins, we get:

$$\frac{1}{K_1}\sum_{f\in\mathcal{F}}|F_i(f,\theta)|^2 = \frac{2}{K_1} + \frac{2}{K_1}\sum_{f\in\mathcal{F}}\cos(\Delta\phi_f - 2\pi f \sin\theta/c) \tag{4.30}$$

Assume that $\mathcal{F}$ represents the lower frequency band, where there are no multiple ripples. If the beamformers follow the correct permutations, then each of $|F_i(f,\theta)|^2$ will be cosines that feature a null around the DOA. Averaging over these cosine functions will emphasize the position of the average DOA. If we add $|F_1(f,\theta)|^2$ and $|F_2(f,\theta)|^2$, i.e. the beampatterns for the two sources, we will get two nulls, one for each source. Therefore, averaging $|F_1(f,\theta)|^2 + |F_2(f,\theta)|^2$ over the frequency band $\mathcal{F}$ will emphasize the position of the two nulls, i.e. the position of the two DOAs. In addition, this graph will be the same whether the permutations are sorted or not (due to the commutative property of addition). Hence, this mechanism can be used for DOA estimation, without sorting the permutations along frequency.

As a result, we propose the following mechanism for DOA estimation:

1. Unmix the sources using the ICA algorithm
2. For each frequency bin f and source i estimate the beamforming pattern $F_i(f, \theta)$.
3. Form the following expression for $\mathcal{F} = [0 - 2KHz]$

$$P(\theta) = \sum_{f \in \mathcal{F}} \sum_{i=1}^{N} |F_i(f, \theta)|^2 \tag{4.31}$$

   The minima of this expression will be an accurate estimate of the Directions of Arrival.

In figure 4.7, we can see that plotting (4.31) can give us an accurate estimate for the DOAs. The exact low-frequency range $\mathcal{F}$ we can use for DOA estimation is mainly dependent on the microphone spacing d. If we choose a small microphone spacing, the ripples will start to appear at higher frequencies, as $f_{ripple} \sim c/2d$. However, as the microphones will be closer, the signals that will be captured will be more similar. Thus, the source separation SNR will decrease considerably, as our setup will degenerate to the overcomplete case. Therefore, the choice of sensor spacing is a tradeoff between *separation quality* and *beamforming pattern clarity*.

4.4.2 Permutation alignment ambiguity

Once we have estimated the DOA, we want to align the permutations along the frequency axis to solve the permutation problem in frequency domain ICA. There is a slight problem with that. Basically, all nulls, as explained in an earlier section, are slightly drifted due to reverberation. As a result, the classification of the permutations may not be accurate.

One solution can be to look for nulls in a "neighbourhood" of the DOA. Then, we can do some classification, however, it is difficult to define the neighbourhood. Hu and Kobatake [HK03] observed that for a room impulse response around $300ms$, the drift from the real DOA maybe $1 - 3$ degrees

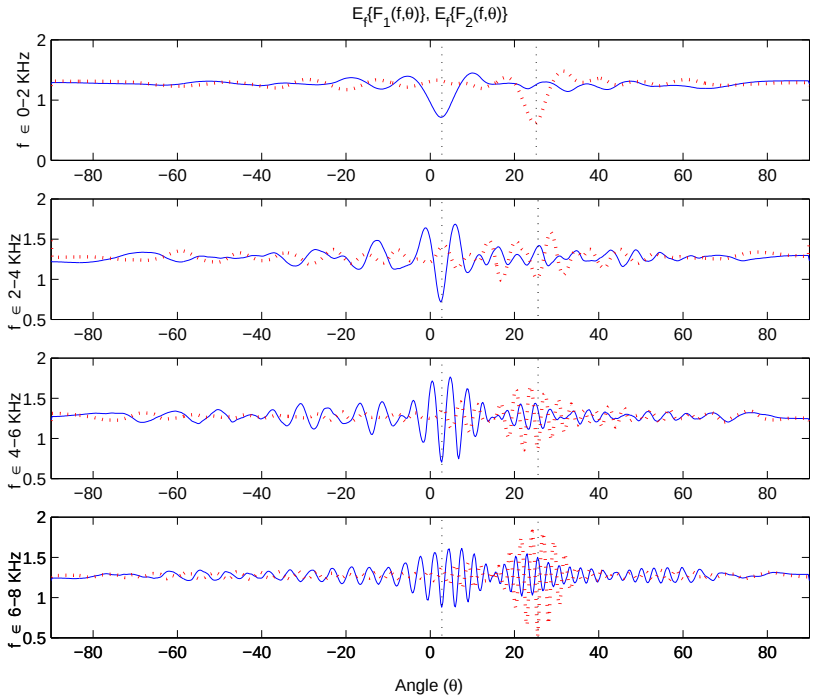

Figure 4.6: Average Beampatterns along certain frequency bands for both sources.

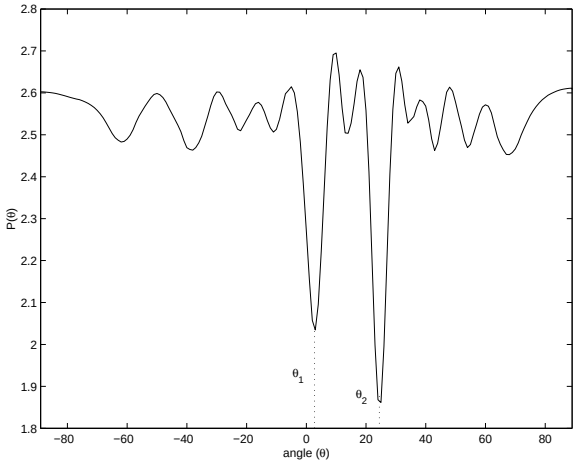

Figure 4.7: A plot of $P(\theta)$ as described in eq. 4.31 gives two distinct DOAs θ_1 and θ_2.

on average (this may be different at various frequencies). As a result, we can define the neighbourhood as 3 degrees around the DOA.

An additional problem is that even in such a small interval, in mid-higher frequencies there might be more than one null, making the classification even more difficult in these frequency bands.

A remedy to this problem might be to use beamforming (phase information) in lower-mid frequencies and the Likelihood Ratio (amplitude information) for mid-higher frequencies. However, we need a mechanism to tie the permutations between the lower and the higher frequency bands. This idea is under current investigation.

4.5 A novel method for permutation alignment using beamforming

Another idea is to introduce more efficient Directivity Patterns in our framework. The MuSIC algorithm is an alternative solution, used thoroughly in literature. The MuSIC algorithm is actually not immune to the multiple nulls ambiguity, however, the DOAs are more distinct and the permutation alignment should be more efficient. Although, in theory, we need to have more sensors than sources, it is possible to apply the MuSIC algorithm in the case of equal number of sources and sensors !

In Chapter 3, we saw that in order to rectify the *scale ambiguity*, we need to map the separated sources back to the microphones domain. Therefore, we have an observation of each source at each sensor, i.e. a more sensors than sources scenario. If we do not take any steps to solve the permutation problem, the ICA algorithm will unmix the sources at each frequency bin, however, the permutations will not be aligned along frequency. As we demonstrated in Chapter 3, mapping back to the observation space is not influenced by the permutation ambiguity. Hence, after mapping we will have observations of each source at each microphone, however, the order of sources will not be the same along frequency. Using the observations of all

microphones for each source, we can use MuSIC to find a more accurate estimation for the DOAs, using (4.14).

We can form "MuSIC directivity patterns" using $M(\theta)$ (4.14), instead of the original directivity patterns. To find the DOA estimates, we can form $P(\theta)$ as expressed in (4.31), using $M(\theta)$ instead of the original directivity pattern. Finally, we can use the DOAs to align the "sharper" "MuSIC directivity patterns".

The proposed algorithm can be summarised as follows:

1. Unmix the sources using the Fast Frequency Domain framework.
2. Map the sources back to the observation space, i.e. observe each source at each microphone.
3. Having observations of each source at each microphone, we apply the MuSIC algorithm to have more accurate DOA estimates along frequency.
4. Align permutations now, according to the DOAs estimated by MuSIC.

4.6 Experiments

In this section, we perform a couple of experiments to verify the ideas analysed so far in this chapter. We will use two basic experiment sets, as in Chapter 3. The first experiment will use artificial mixtures of a two sensor - two sources using only single delays. The second experiment will contain real room recordings of a two sensor - two sources setup.

4.6.1 Experiment 1 - Single Delay

In this section, we wanted to test our framework in terms of beamforming performance using single delays. Two speech signals are mixed artificially using delays between $6-5$ msecs at $16KHz$. In our experiments, we use the basic fast frequency domain framework, as described in Chapter 3. We test

the performance of the proposed solutions for the permutation problem, in terms of beamforming.

In figure 4.8, we test the fast frequency domain framework, where no solution for the permutation problem was used. We can see that even in the case of a single and small delay, the permutation ambiguity is clearly visible. Moreover, the ambiguity is also audible in the unmixed sources. We can see that a solution is definitely needed.

In figure 4.9, we can see the performance of the Likelihood Ratio (LR) solution. An amplitude-based solution seems to align the permutations along frequency in the case of a single delay.

In figure 4.10, we can see a plot of $P(\theta)$ (4.31) for this case of a single delay. We averaged the directivity patterns over the lower frequency band $(0-2KHz)$ and as a result we can see two clear Directions of Arrival. The estimated DOAs will be used to align the permutations along frequency. Since we are modeling a single delay, we will not allow any deviations from the estimated DOAs along frequency. As a result, we are going to align the permutations according to the existence of a null along the estimated DOAs. In figure 4.11, we can see the general performance of this scheme. We can spot some mistakes in the mid-higher frequencies, verifying that it might be difficult to align the permutations there.

In figure 4.12, we plot the MuSIC-generated Directivity Patterns for the case of no solution for the permutation ambiguity. The problem manifests itself very clearly in the figures. In addition, we emperically noticed that the peaks in the MuSIC algorithm tend to be more distinct, than those created by a normal directivity pattern.

Again in figure 4.13, we use the MuSIC Directivity Patterns to demonstrate the performance of the Likelihood Ratio solution.

In figure 4.14, we can see a plot of $P(\theta)$ (eq. 4.31) using the MuSIC algorithm. We averaged the MuSIC directivity patterns over the lower frequency band $(0-2KHz)$ and as a result we can see two clear Directions of Arrival. The difference with the graph in figure 4.10 is that the peaks indicating the

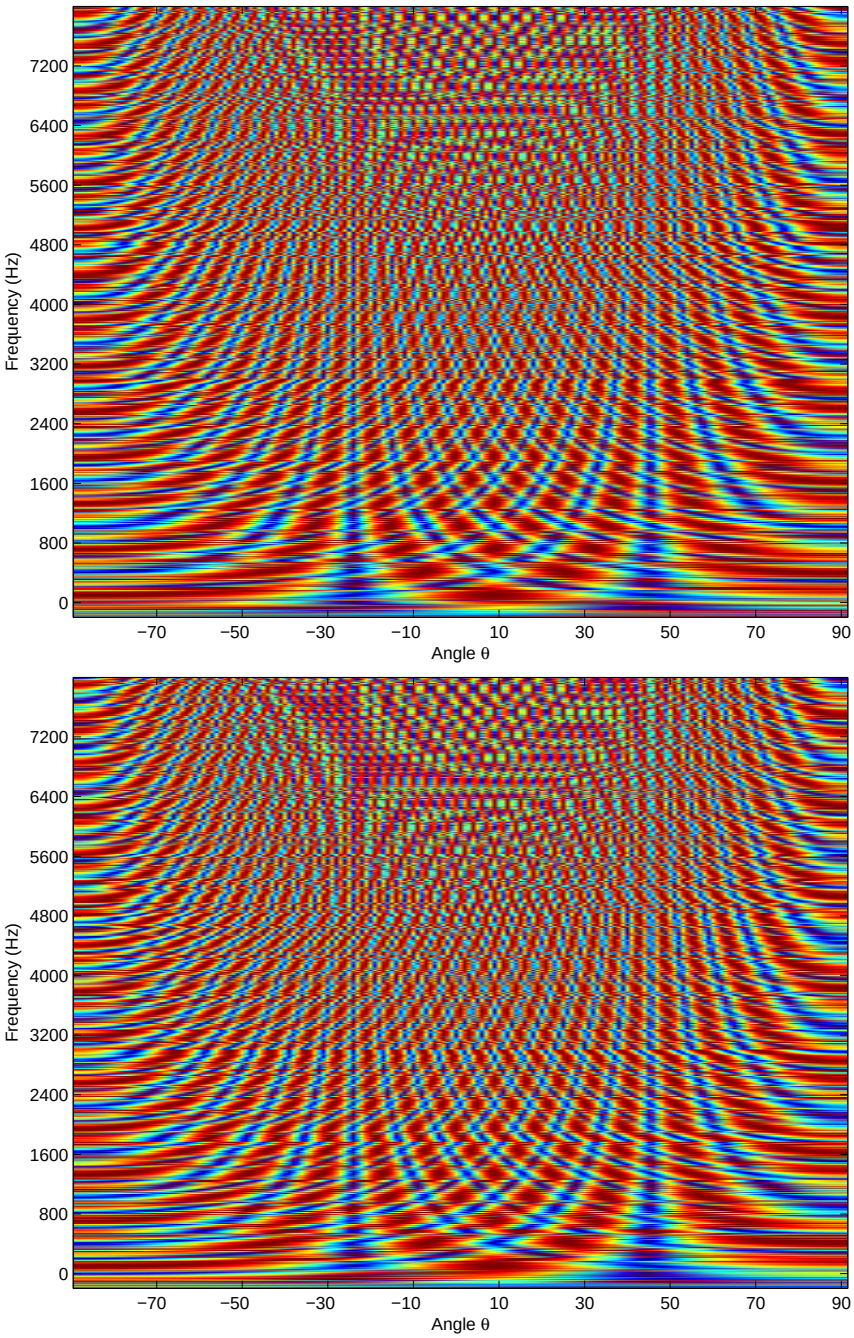

Figure 4.8: Directivity patterns for the two sources. Permutation problem exists even in the single delay case without any further steps.

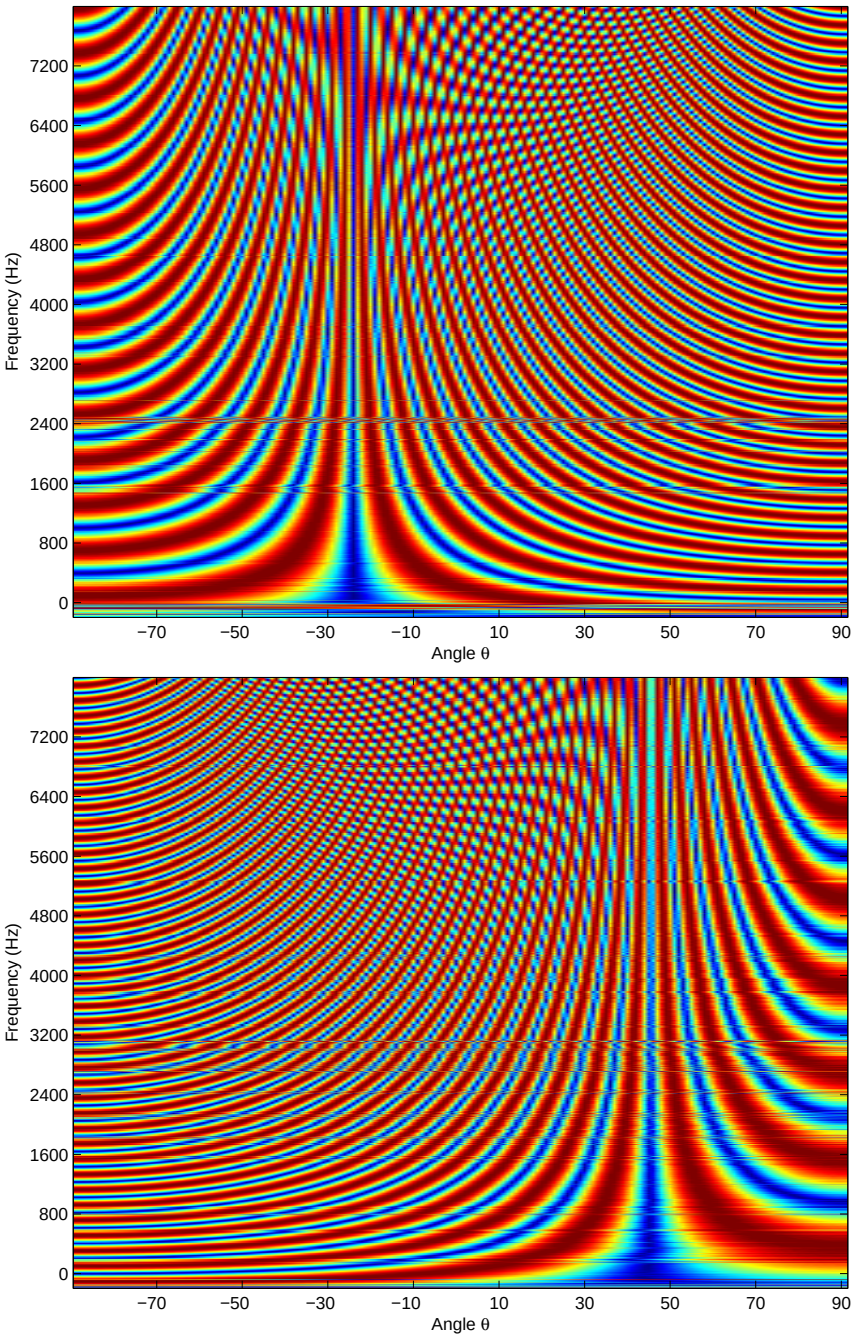

Figure 4.9: The Likelihood Ratio jump solution seems to align the permutations in the single delay case.

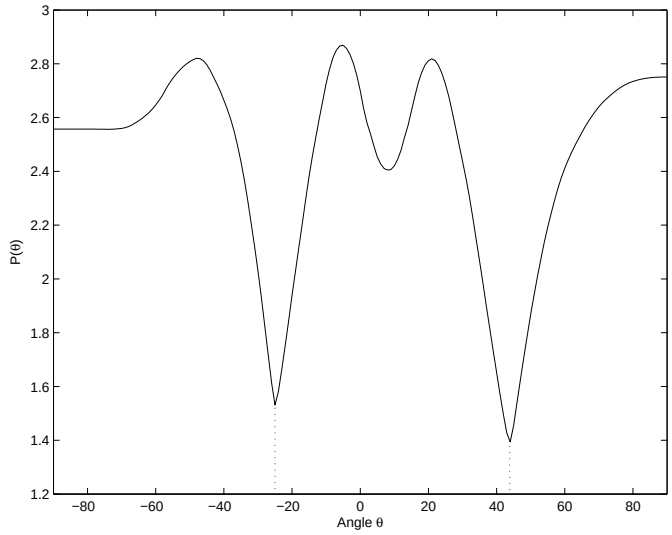

Figure 4.10: Plotting P(θ) (eq. 4.31) using the first 2KHz for the single delay case. Two distinct DOAs are visible.

Directions of Arrival are now a lot more "distinct". In figure 4.15, we can see that the permutations are correctly aligned using the MuSIC directivity plots. This seems to be due to the more "distinct" MuSIC directivity plots that seem to be more efficient for permutation alignment.

4.6.2 Experiment 2 - Real room recording

In this section, we discuss the use of beamforming for permutation alignment through a real world experiment. We used a university lecture room to record a 2 sources - 2 sensors experiment. We used two speakers (source 1 and source 2) and two cardioid microphones (mic 1 and mic 2), arranged as in figure 4.16. We investigate the nature of real room directivity patterns as well as explore the performance of the proposed schemes for permutation alignment.

We apply the Fast Frequency domain ICA algorithm, as described in Chapter 3, without any algorithm for the permutation problem. As a result, the sources will be unmixed, but randomly permuted along the fre-

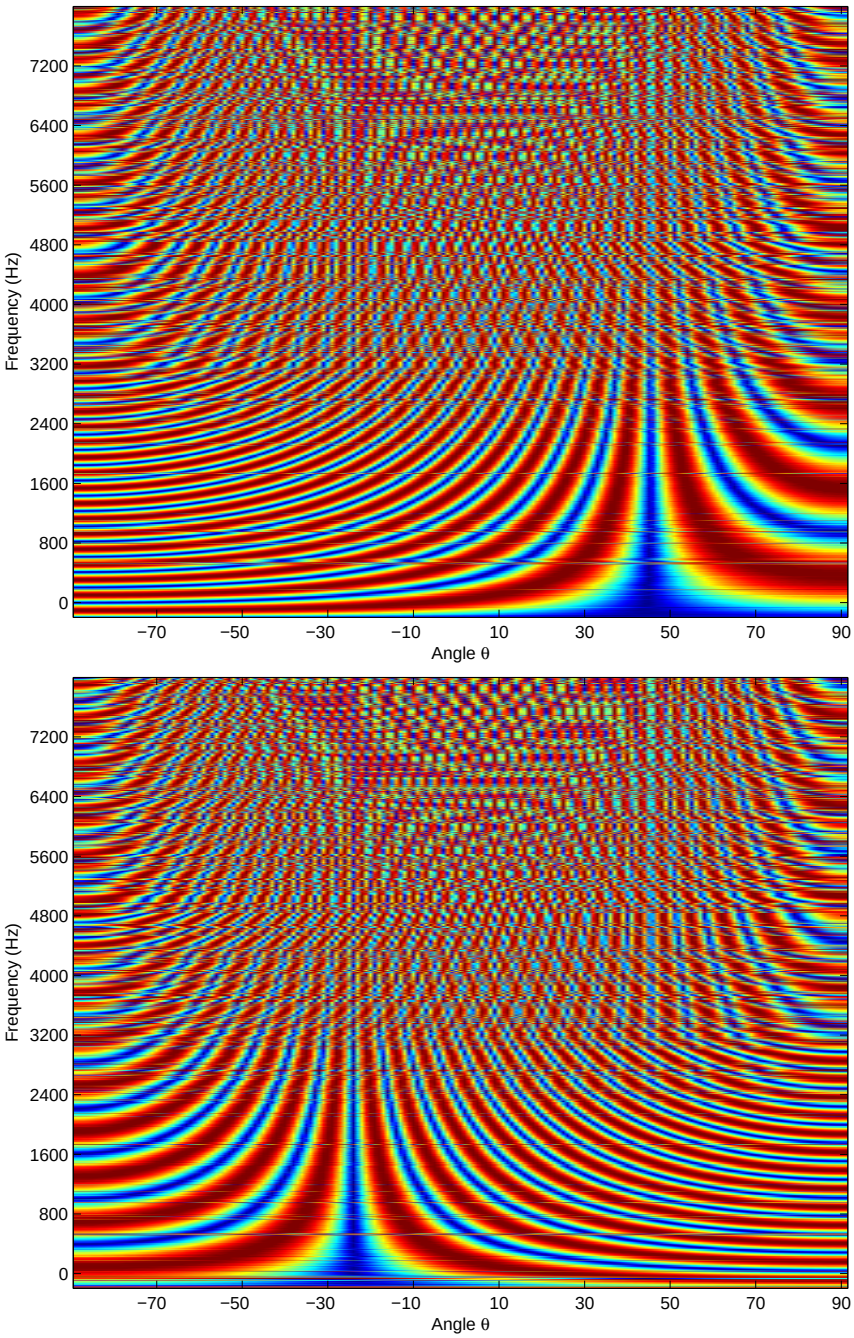

Figure 4.11: Permutations aligned using the Directivity Patterns in the single delay case. We can see some problems in the mid-higher frequencies.

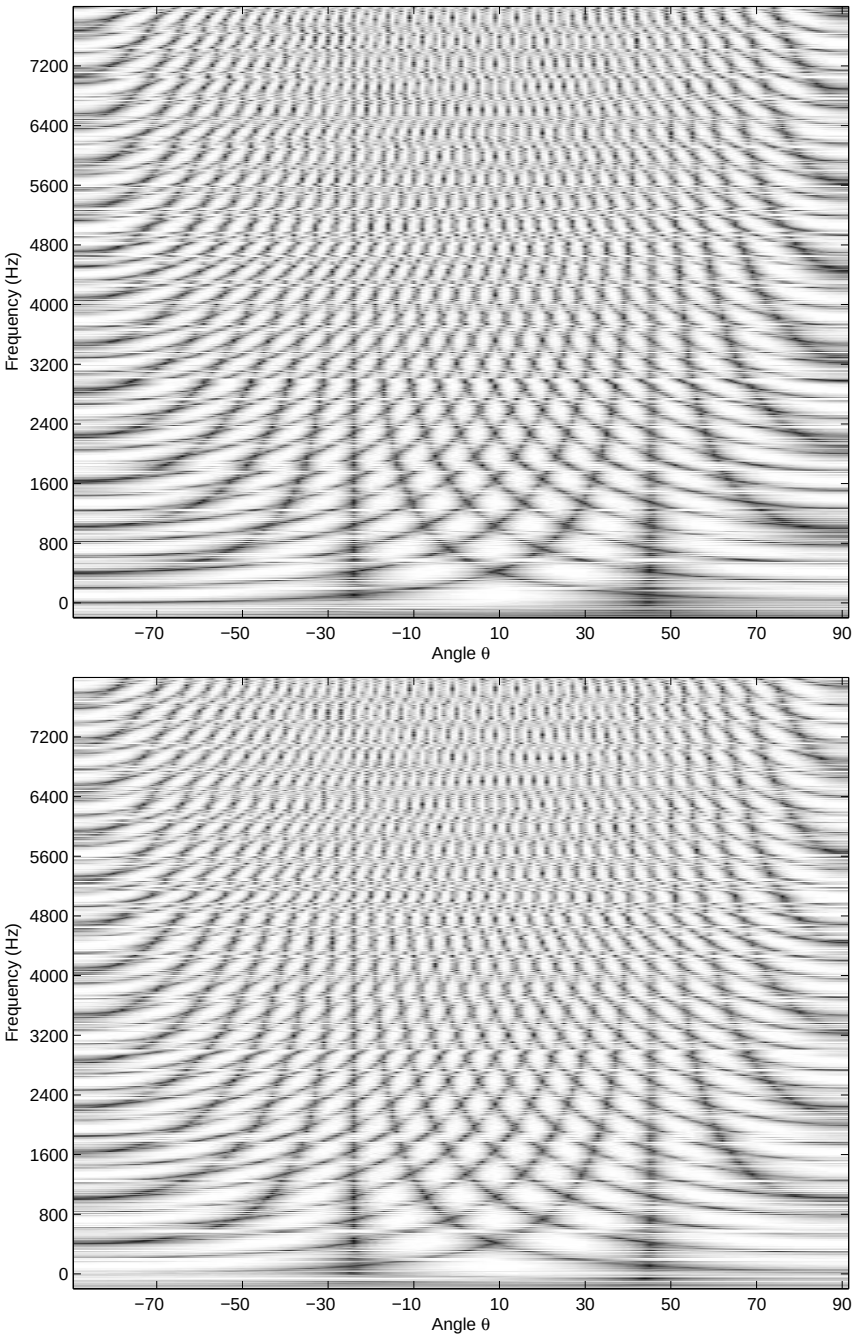

Figure 4.12: Using the MuSIC Directivity Patterns methodology for permutation alignment. The permutation problem is demonstrated here.

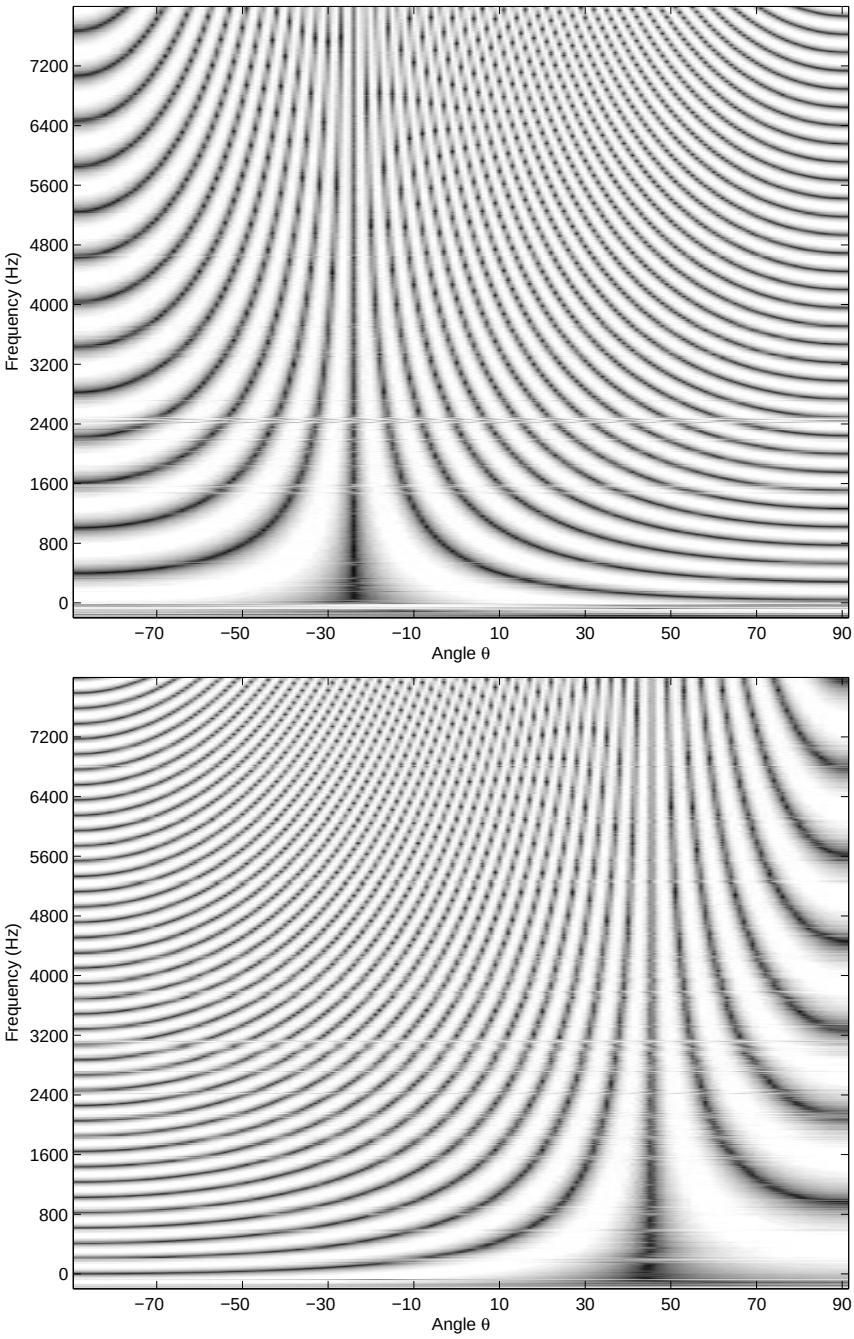

Figure 4.13: Plotting the MuSIC Directivity Patterns for the Likelihood Ratio solution for the single delay case.

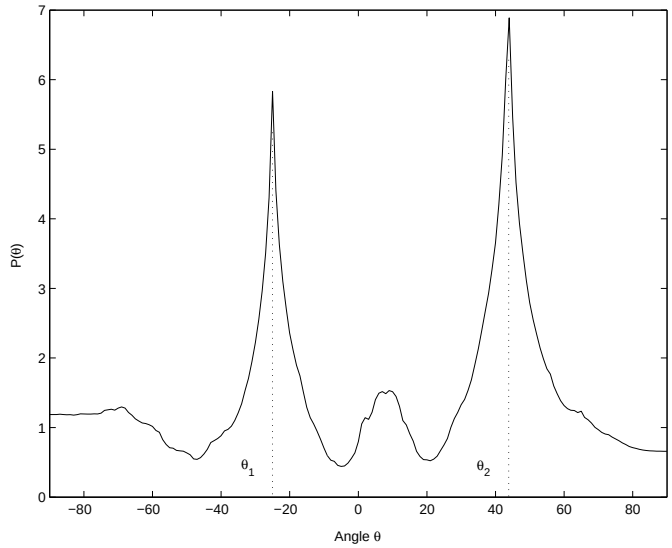

Figure 4.14: Accurate DOA estimates using the MuSIC algorithm.

quency axis. In figure 4.17, we can see the directivity patterns of the resulting unmixing matrices. We can see nothing comprehensible, as the mixed permutations create a scrambled image. The difficulty of the problem is obvious.

In figure 4.18, we can see the directivity patterns of the unmixing system, after applying the Likelihood Ratio Jump solution. We can see that an almost consistent alignment along frequency. Certain mistakes in several frequency bands are visible. Another observation is that although we have a clearly multipath environment, we can a spot a main Direction of Arrival, due to the strong direct path signal. The multipath environment mainly causes a slight drift of the main direction of arrival along frequency. However, the Likelihood Ratio is an amplitude criterion. In the next permutation alignment attempts using the directivity patterns, we shall allow a tolerance of $\pm 3^{\circ}$ for the actual position of the DOA, due to the multipath.

In figure 4.19, we can see a plot of $P(\theta)$ (4.31) for this case of real room recording. Averaging over the lower 2KHz, we seem to get a very clear image of where the main DOAs are, giving us an accurate measure

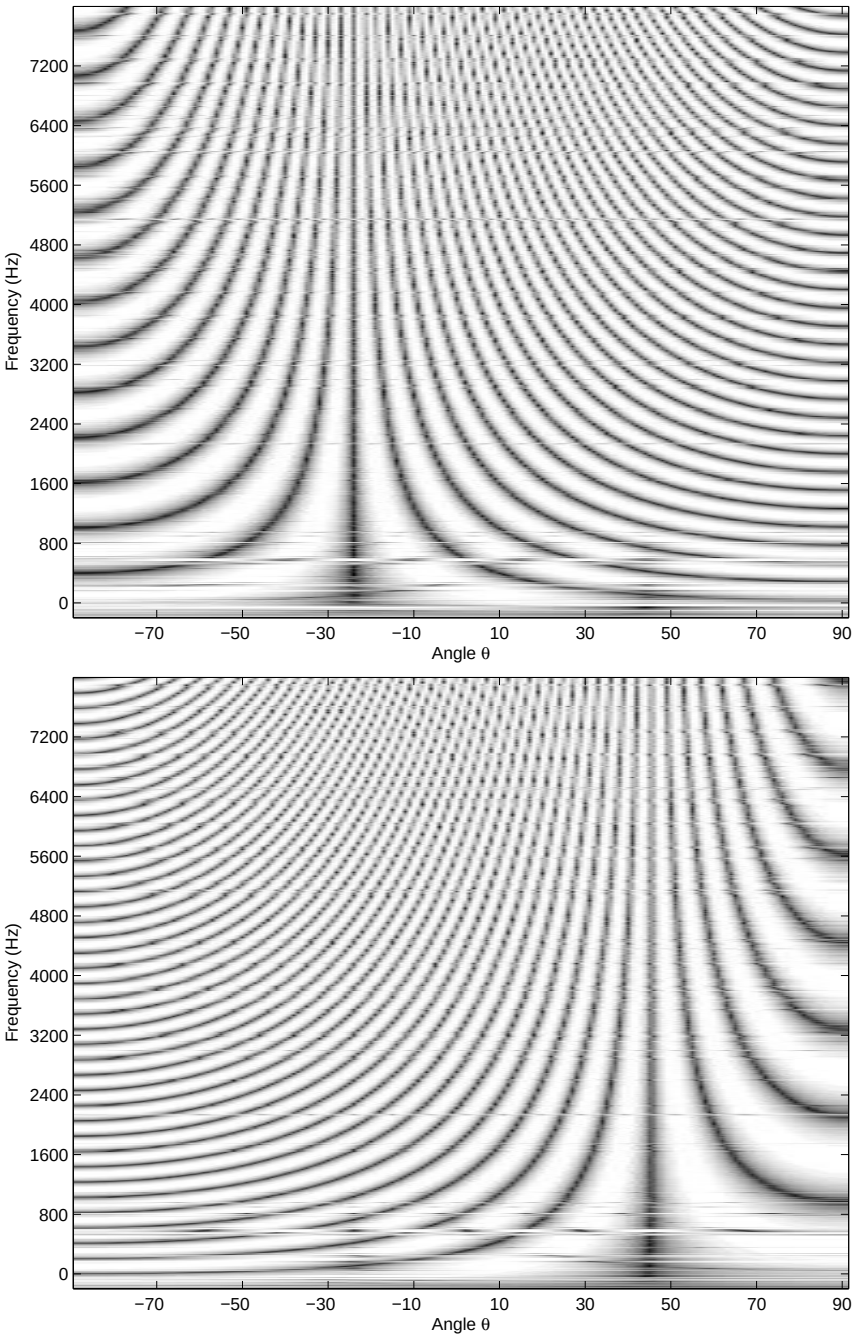

Figure 4.15: Accurate permutation alignment using the MuSIC directivity patterns.

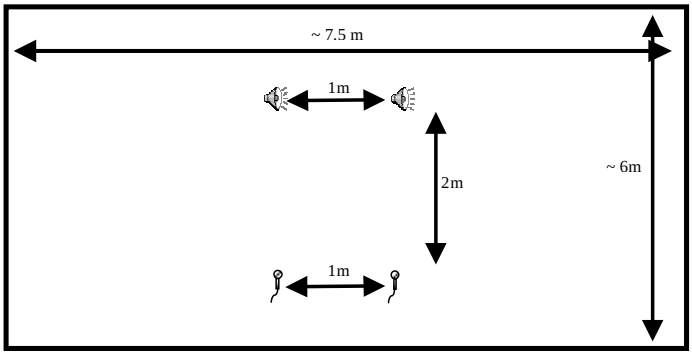

Figure 4.16: Experimental 2 sensor 2 sources setup in a real lecture room.

for this estimation task. We try to align to the permutations around the estimated DOAs allowing $\pm 3^{\circ}$ deviation. In figure 4.20, we see the results. We can spot that generally this scheme can perform robust permutation alignment in the lower frequencies, but considerable confusion exists in the higher frequencies, as expected from our theoretical analysis.

In figure 4.21, we plot the MuSIC-generated Directivity Patterns for the case of the unmixed sources without permutation alignment. The figures are more clear compared to figure 4.17 and we can still acknowledge the difficulty of the problem. In figure 4.22, the performance of the Likelihood Ratio solution is more clearly demonstrated using the MuSIC generated patterns.

In figure 4.23, we can see a plot of $P(\theta)$ (4.31), averaging the MuSIC directivity patterns over the lower frequency band $(0 - 2KHz)$. The two Directions of Arrival are clearly identified from this graph. In figure 4.24, we can see that the permutations are correctly aligned using MuSIC directivity plots. Again, the MuSIC directivity patterns seem to be more robust for permutation alignment.

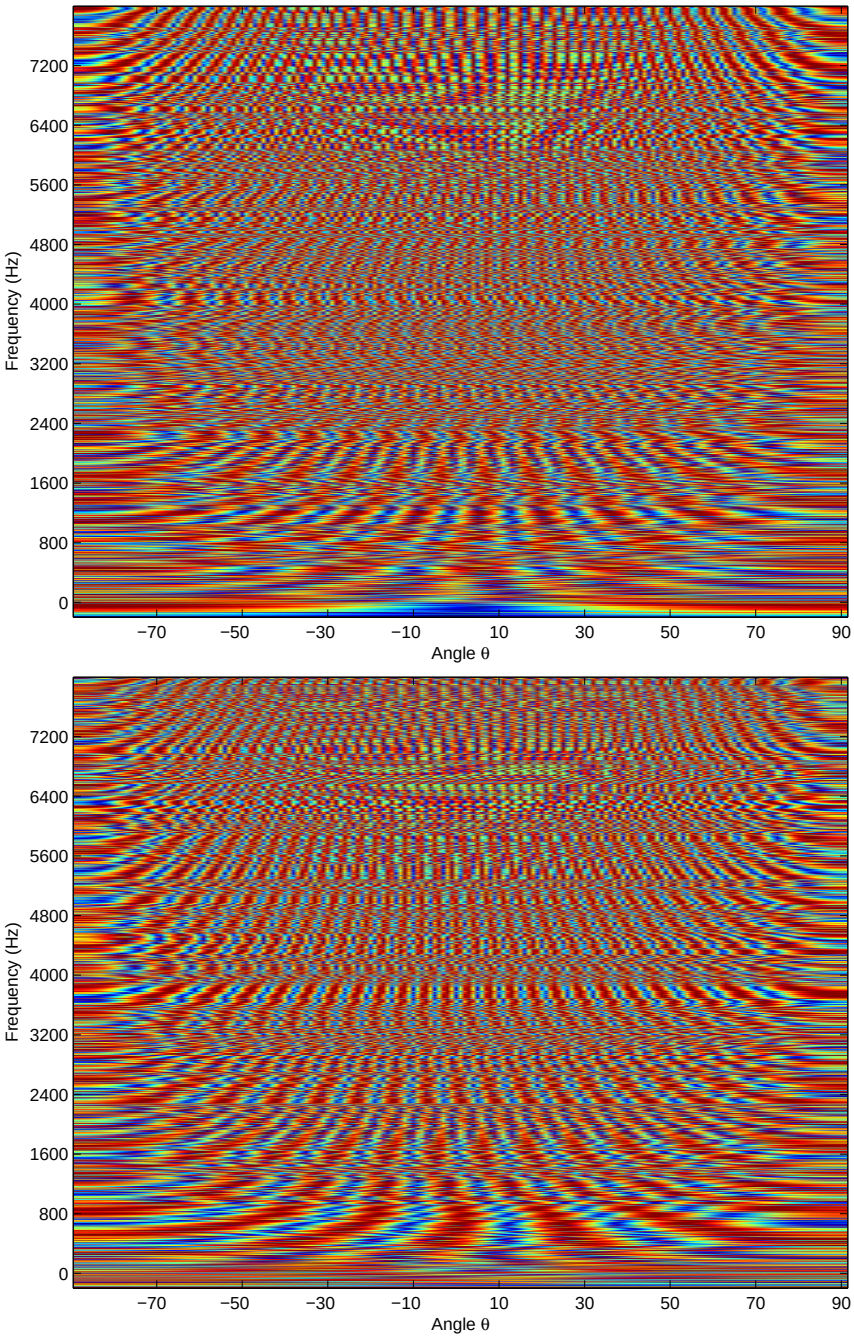

Figure 4.17: Directivity patterns for the two sources. Permutation problem exists in the real room case. No steps were taken for the permutation problem, resulting into nothing comprehensible.

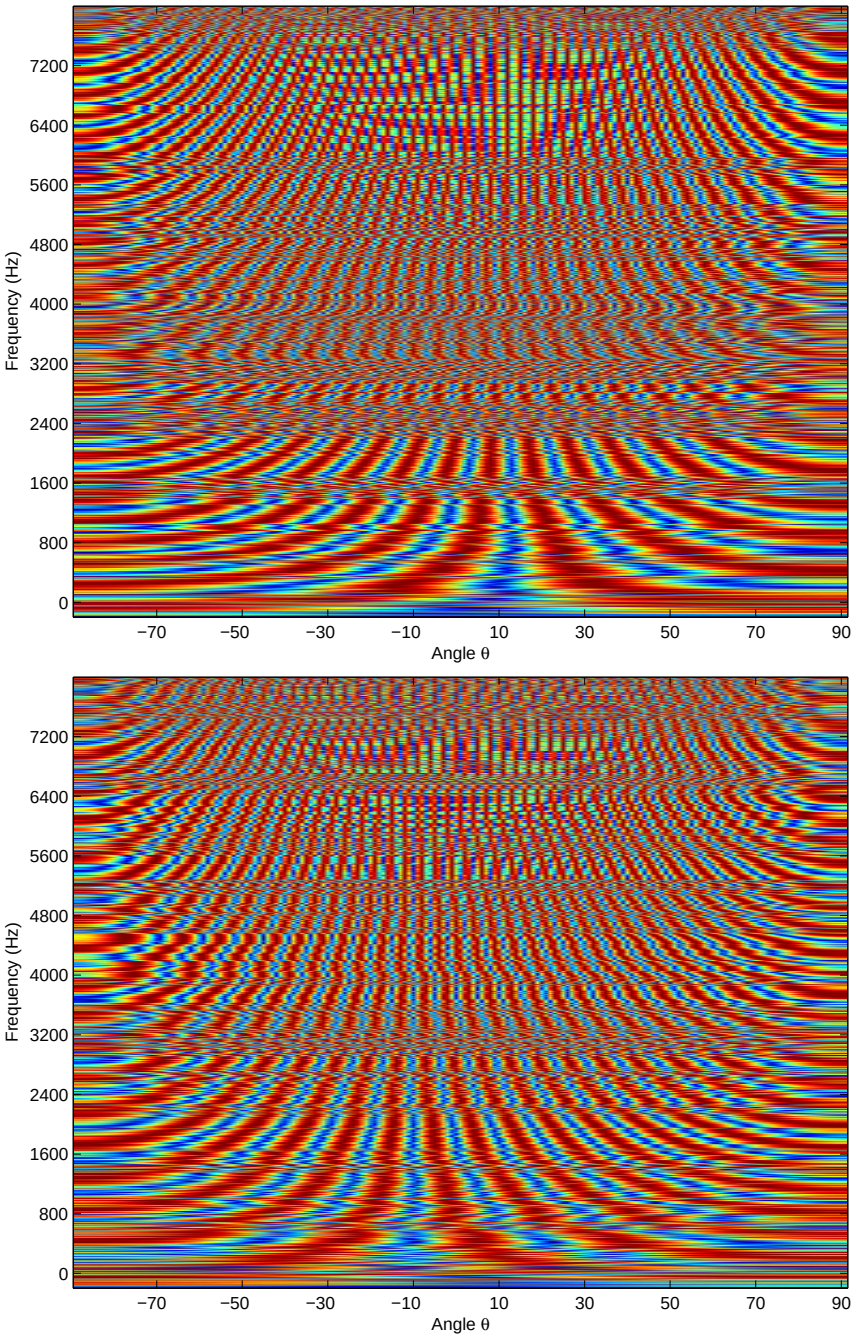

Figure 4.18: The Likelihood Ratio jump solution seems to align most of the permutations. Certain mistakes are visible, especially in the higher frequencies.

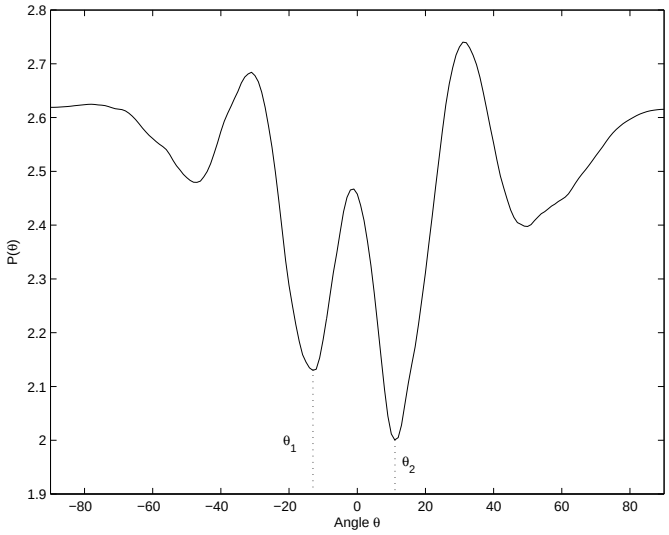

Figure 4.19: Plotting P(θ) (eq. 4.31) using the first 2KHz for the single delay case. Two distinct DOAs are visible for the real room case.

4.7 Sensitivity Analysis

All source separation algorithms assume stationary mixing. However, in some applications the sources tend to move in the auditory scene. Having described the source separation setup as a sort of adaptive beamforming, in this section we will try to explore a simple case of source movement. In order to test the sensitivity of our beamformer to movement, we recorded two setups in a real room (see figure 4.16). As a result, a two microphone - two speakers setup, as in figure 4.16, was initially recorded. Then, the left speaker was displaced by $50cm$ and the whole setup was recorded again. This source displacement attempts to simulate a small source movement. Comparing the two recordings, we tried to make some preliminary investigation into a) the beamformer sensitivity to movement and b) the distortion introduced due to movement.

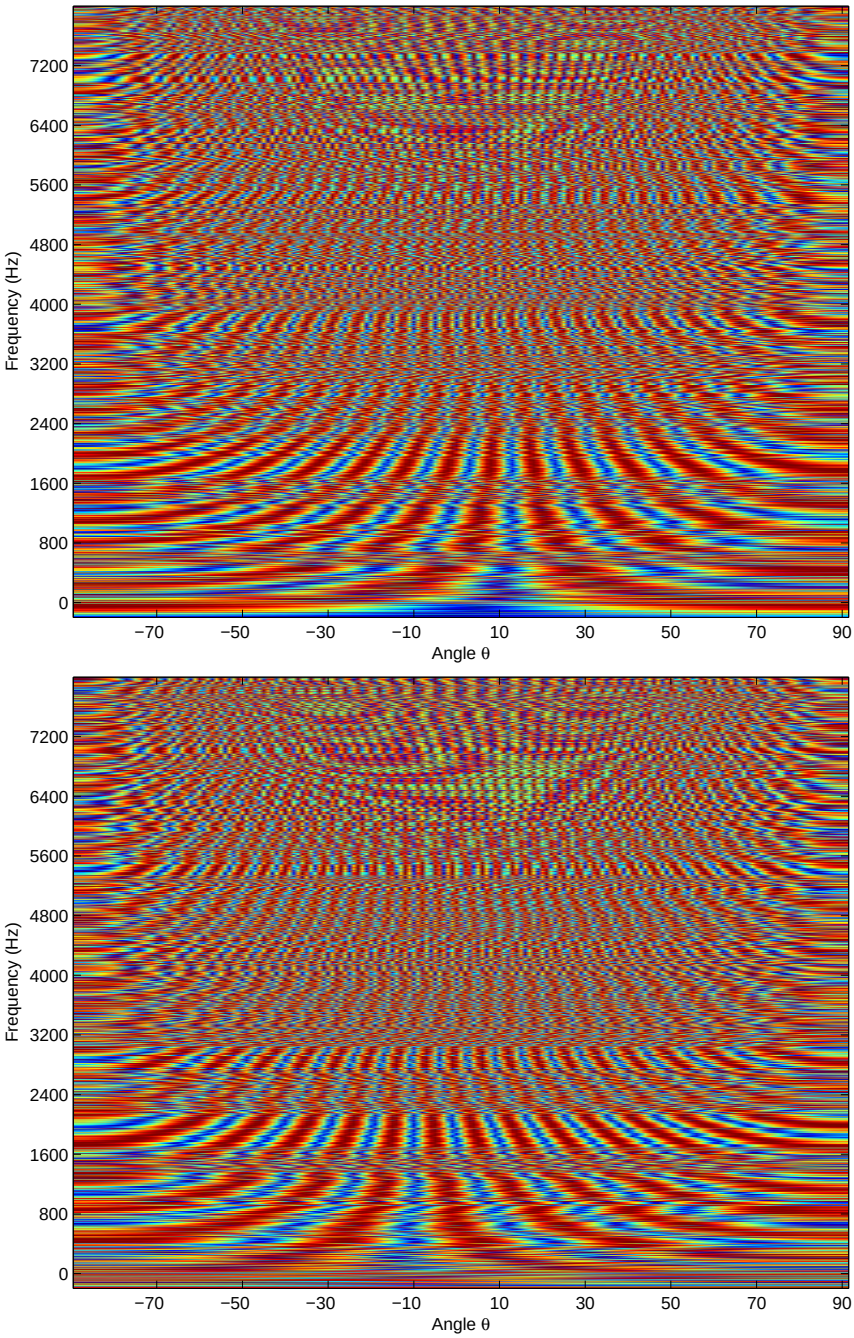

Figure 4.20: Permutations aligned using the Directivity Patterns in the real room case. We can see good performance in the lower frequencies but some inconsistencies in the mid-higher frequencies.

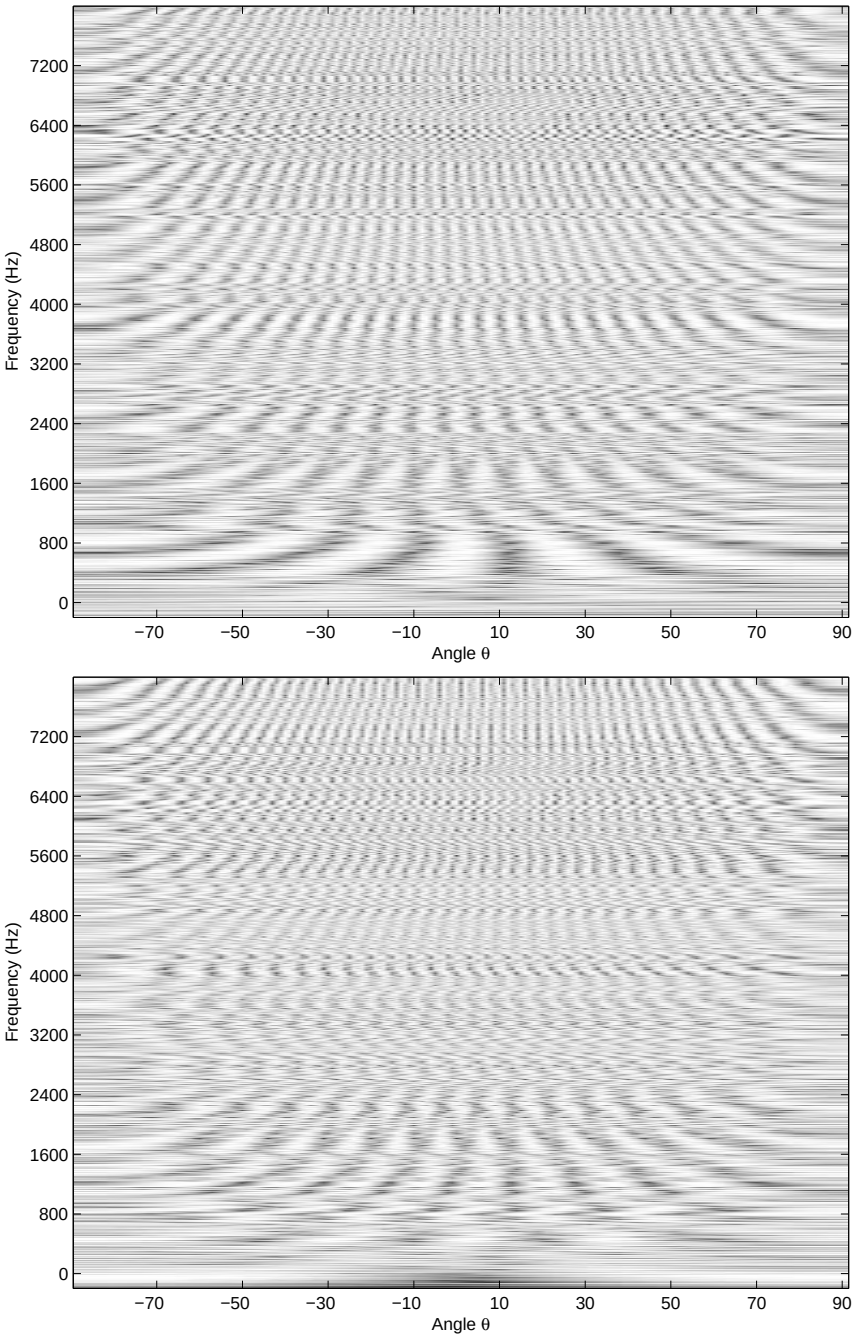

Figure 4.21: Using the MuSIC Directivity Patterns methodology for permutation alignment. No steps for the permutation problem are taken. The permutation problem is visible.

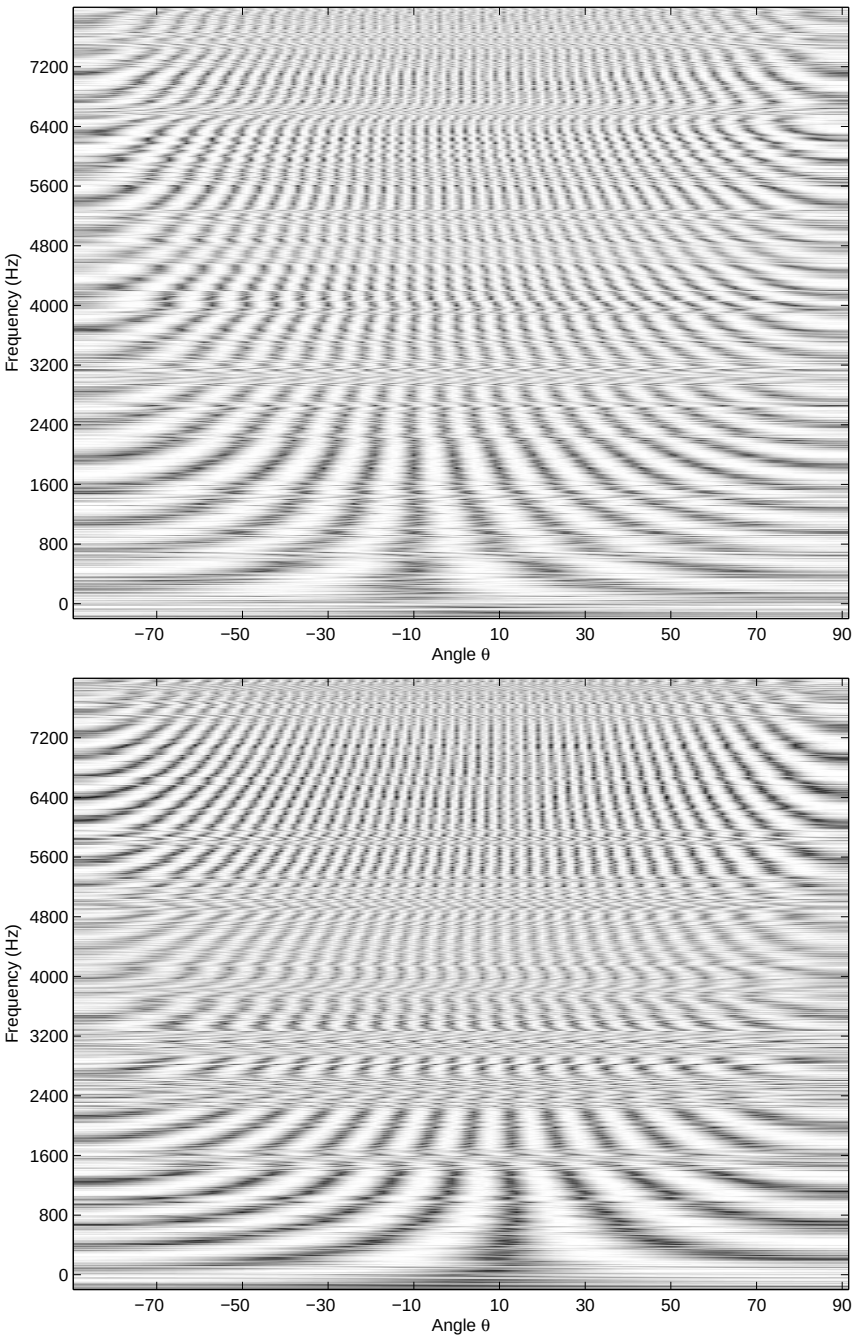

Figure 4.22: Plotting the MuSIC Directivity Patterns for the Likelihood Ratio solution for the real room case

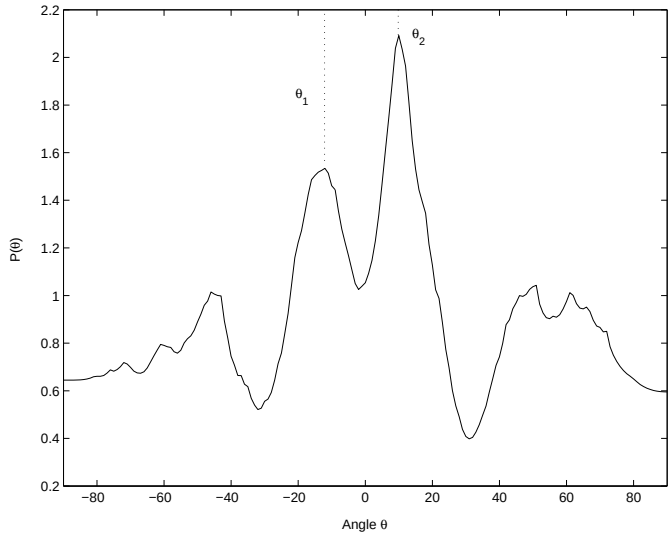

Figure 4.23: Accurate DOA estimates using the MuSIC algorithm in the real room case.

4.7.1 Beamformer's sensitivity to movement

We unmixed the two setups using the ICA algorithm and the LR algorithm. We calculated the beamformer patterns for the two cases and we tried to compare the two patterns to see the effect of misplacement. Comparing beamforming patterns along frequency, we see that the beamformers sensitivity to movement is a function of frequency. At low frequencies the beamformers null has been slightly shifted, due to movement. Figure 4.25(a) shows the directivity patterns at $160Hz$ for both the original and displaced experiments. Whilst there will be some degradation due to the misalignment, the original beamformer can still suppress the source at this frequency quite effectively.

In contrast to this, even at moderate frequencies the directivity pattern becomes more oscillatory, due to the shorter wavelength. Thus as the frequency increases the source separation algorithm is unable to suppress the interfering source. Figure 4.25(b) shows that, even at $750Hz$, we have a situation, where the null is almost replaced by a peak in the misaligned

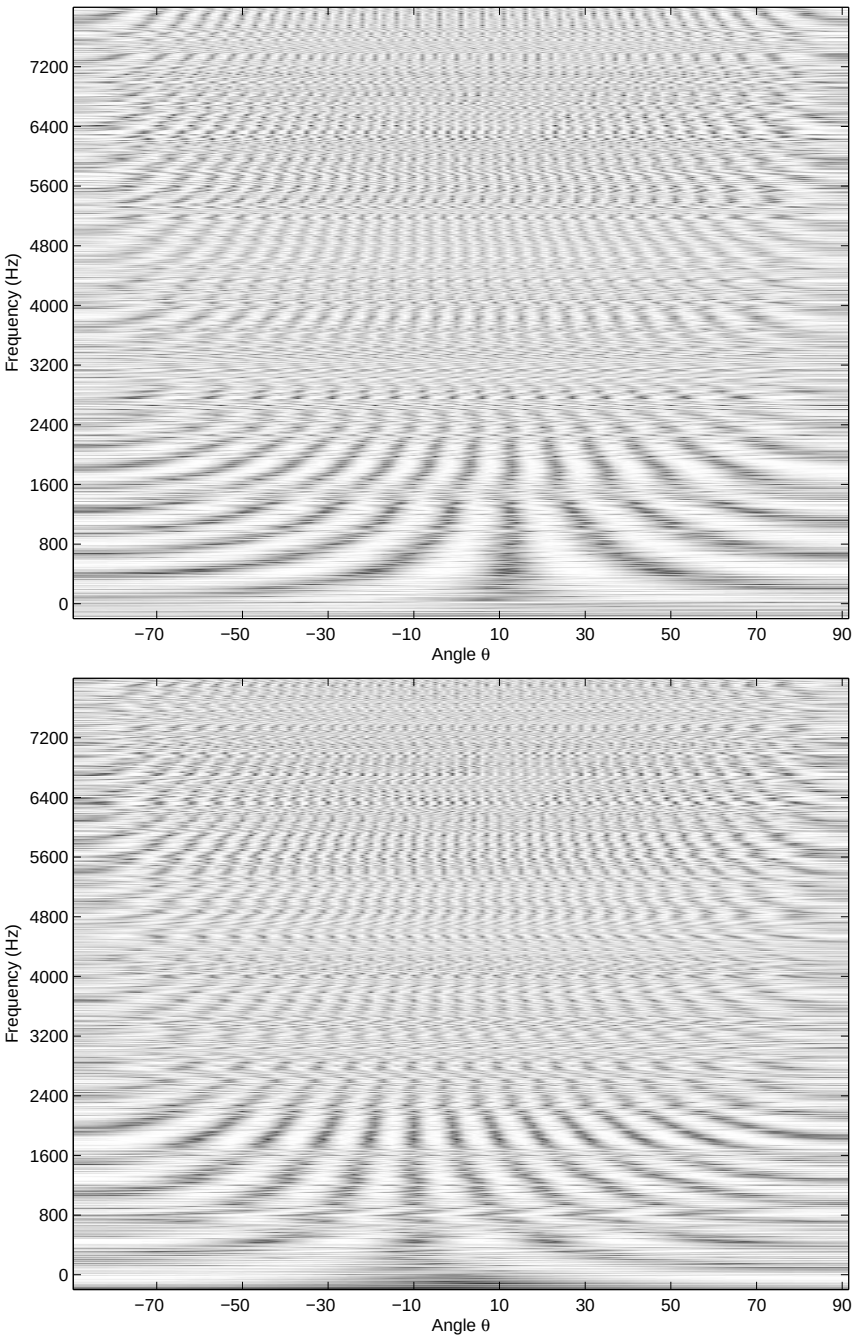

Figure 4.24: Permutation alignment using the MuSIC directivity patterns in the real room case.

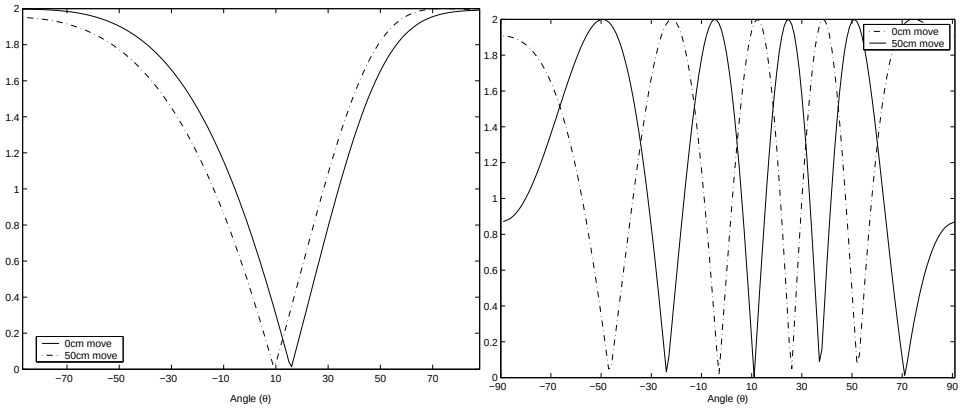

Figure 4.25: Comparing beamforming patterns at (a) 160Hz and (b) 750Hz.

patterns. In this situation, our beamformer is rendered useless. Hence, in mid-higher frequencies, possible source displacement can degrade the performance dramatically.

To quantify this, we evaluated the change in distortion due to the movement as a function of frequency. This is shown in figure 4.26 (we have used a log-frequency scale to highlight the behaviour at low frequencies). As predicted, distortion is not significantly affected at low frequencies. However, above about $200Hz$ the distortion increases by more than $5dB$. The result is that all practical source separation above about $300Hz$ has been lost.

4.7.2 Distortion introduced due to movement

We conclude this section by examining how the distortion, introduced by the displacement of one source, is manifested in the separated signals. From listening to the separated sources, it was clear that source 1 contained a considerable amount of crosstalk. Source 2, however, contained no crosstalk, but sounded substantially more "echoic". These observations can again be explained by considering the directivity patterns associated with the unmixing filters. From our above arguments, we see that displacing source 2 will introduce the observed crosstalk. However, since source 1 was not displaced the unmixing filters adapted to cancel this source will still place a correct

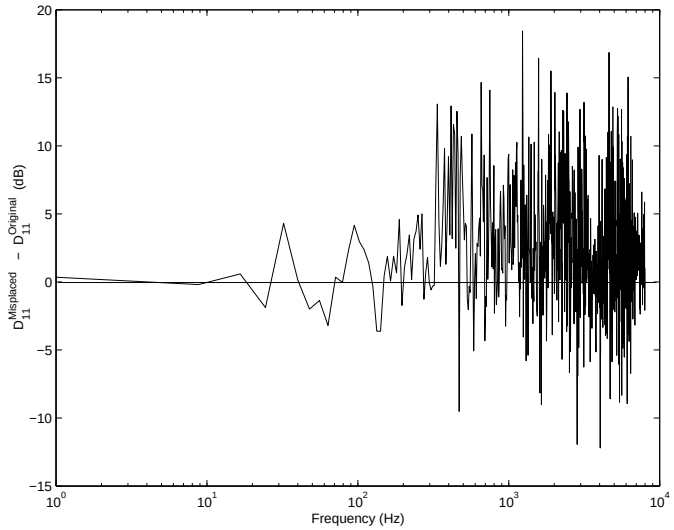

Figure 4.26: Distortion increases as a function of frequency in the case of a misaligned beamformer.

null in the direction of source 1. As a consequence, despite the fact that source 2 was moved, it is still correctly separated. The added reverberation in source 2 comes about, due to mapping the signals back to the microphone domain. This is more clearly illustrated in figure 4.27.

Specifically, if we assume that at a given frequency bin

$$\underline{X} = \begin{bmatrix} A_{11}A_{12} \\ A_{21}A_{22} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \tag{4.32}$$

represents the mixed signals. After separation, we map back to the microphone space, so we are trying to estimate the individual source signals observed at the microphones $\underline{X}_{s1}, \underline{X}_{s2}$:

$$\underline{X}_{s1} = \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} S_1, \underline{X}_{s2} = \begin{bmatrix} A_{12} \\ A_{22} \end{bmatrix} S_2 \tag{4.33}$$

this introduces the following constraint into our source estimates:

$$\underline{X} = \underline{X}_{s1} + \underline{X}_{s2} \tag{4.34}$$

However, due to misaligned beamforming, one source will get contamination from the other source. Therefore,

$$\underline{X}_{s1} = \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} S_1 + \epsilon \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} S_2 \tag{4.35}$$

where ϵ and G_1, G_2 model the error due to misaligned beamforming. Because (4.34) is a constraint to our reconstruction, this implies that the second source will get no contamination from source 1, but instead will be distorted, due to incorrect mapping to the observation space.

$$\underline{X}_{s2} = \left(\begin{bmatrix} H_{12} \\ H_{22} \end{bmatrix} - \epsilon \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \right) S_2 \tag{4.36}$$

4.8 Conclusion

In this chapter, we interpreted the Frequency-Domain audio source separation framework, described in Chapter 3, as a Frequency-Domain beamformer. The main motive was to discover more solutions for the permutation problem, as the solution proposed in Chapter 3 was based on *source modelling* or *amplitude information* in the frequency domain. Beamforming employs *phase information* or effectively the *channel information* to align the permutations.

We explored the directivity patterns produced by the ICA framework in the case of a real room. The directivity patterns seem to feature a main Direction of Arrival, however, it tends to *drift slightly along frequency* due to room reflections. We have reviewed some of the proposed methods for permutation alignment. This drift of the main DOA may hinder the quality of permutation alignment, especially in the mid-higher frequencies. We also saw that the distance between the microphones plays an important role in the effectiveness of these schemes. Large microphone spacing will result in multiple nulls appearing even in lower frequencies, making permutation alignment inaccurate. This problem can be rectified by using small micro-

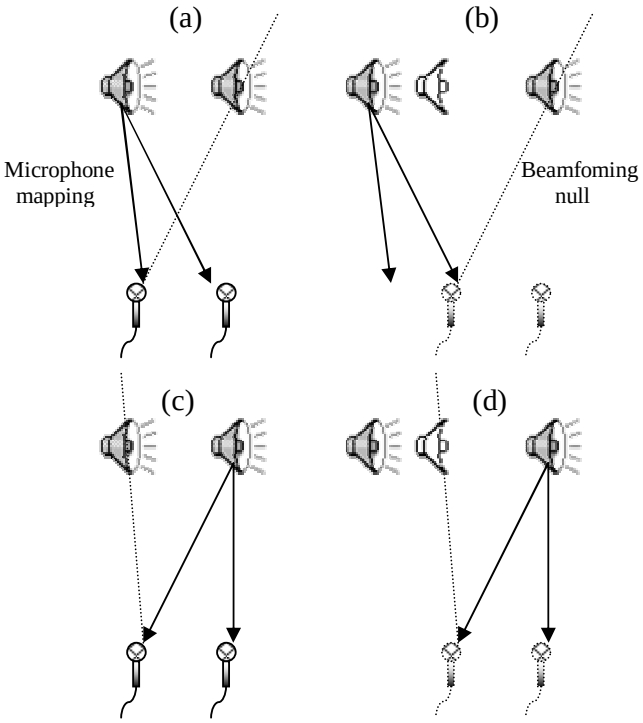

Figure 4.27: Distortion introduced due to movement.(a) Correct beamforming for right source and correct mapping for left source, (b) left source moves, correct beamforming for right source and incorrect mapping for left source, (c) correct beamforming for left source and correct mapping for right source, (d) left source moves, incorrect beamforming for left source and correct mapping for right source.

phone spacing, however, this might deteriorate the separation quality. A mechanism for DOA estimation using directivity patterns was proposed.

In addition, a novel mechanism to employ *subspace methods* for permutation alignment in the frequency domain source separation framework in the case of equal number of sources and sensors was proposed. In order to rectify the scale ambiguity, we need to map the separated sources back to the microphone domain. As a result, we have different observations of each source at each microphone, i.e. more sensor signals than sources. In this case, a subspace method, i.e. MuSIC, can be employed for permutation alignment. In our experiments, the MuSIC directivity patterns proved to be more efficient compared to the original directivity patterns for permutation alignment. In addition, such a scheme seems to be less computationally expensive in the general $N \times N$ case, compared to the Likelihood Ratio, as we do not have to work in pairs or even calculate the likelihood of all permutations of the N sources.

Finally, in a preliminary attempt to explore how robust is the source separation framework to movement, we observed that the separation algorithm seems to be more sensitive to movement in the higher frequencies than in the lower frequencies. In addition, we can demonstrate that in cases of a moving and a stationary source, we can always separate the moving source due to correct null placement on the stationary source. In contrast, the misaligned beamformer for the moving source will corrupt the stationary source with the other source.

Chapter 5

Intelligent Audio Source Separation

5.1 Introduction

Most source separation algorithms aim to separate all the sound objects present in the auditory scene. In fact, humans can not really separate all the sound objects that are present in the auditory scene simultaneously. Instead, we tend to focus on a specific source of interest and suppress all the other sources present in the auditory scene. We can hardly separate more than one source simultaneously. As a result, current source separation algorithms do not really try to emulate the human hearing system, but instead address a more difficult problem. A more realistic objective for the source separation scenario may be to separate *a specific source of interest*, especially in the overcomplete source separation case (*Blind Source Extraction*).

In this chapter, we discuss the idea of "*intelligent*" source separation, i.e. a more selective kind of algorithm. As blind source separation algorithms tend to focus more on the actual signals' statistics, we need to embed several tools to allow the source separation system to discriminate between sound objects. Previous research on *instrument recognition* can provide us with all the essential tools to perform *"intelligent" audio source separation* using

Independent Component Analysis.

5.2 Instrument Recognition

Automatic musical instrument recognition is the problem of automated identification of an instrument from a solo recording, using some previous knowledge of the instrument. Instrument recognition can be a useful tool in *Musical Information Retrieval* (MIR) applications (music indexing and music summarisation) [HABS00] and of course in *Automatic Music Transcription*. In addition, music compression algorithms may benefit from this kind of analysis, as instruments tend to have different bandwidth requirements, leading to a more appropriate compression scheme per instrument. Instrument recognition is also similar to *speaker recognition* or *verification*, where a person can be identified from his voice (*Biometrics*) [PBJ00].

An instrument/speaker recognition procedure is basically split into two phases:

- *Training phase.* During the training phase, some audio samples from the instrument or person are used to retrieve and store some information about the instrument in a statistical/dynamical model. The model is stored locally to form a database for each instrument.

- *Recognition phase.* During the recognition process, a smaller audio sample from the instrument is shown to the system. The system extracts from the sample the same type of information as in the training phase. The extracted information is compared with the information available in the database and finally the system makes an inference about the instrument or the person.

A general flow diagram for the training phase is depicted in figure 5.1 and the equivalent for the recognition phase in figure 5.3. We can see that the two flow diagrams have some blocks in common: *Preprocessing*, *Feature extraction*, *Instrument Modelling* and *Instrument Recognition* block. The function of these blocks is investigated below:

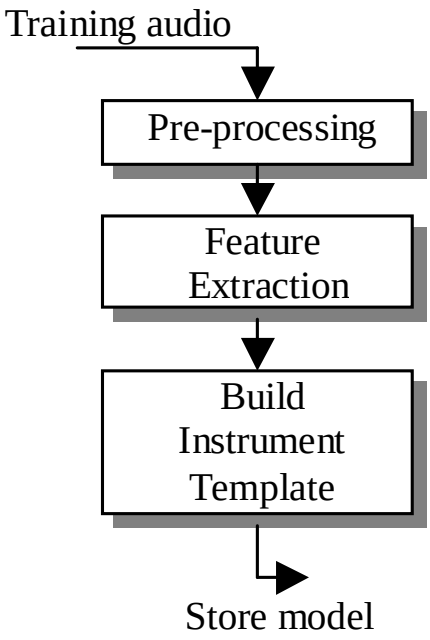

Figure 5.1: A general flow diagram for instrument recognition model training.

5.2.1 Preprocessing

The training/testing audio is usually passed through a pre-processing stage. This stage can have various steps, mainly depending on the application. Usually, possible DC bias is removed from the signal to cater for different recording sources. In addition, the signal is normalised to unit variance to cater for different input signal levels. Moreover, the silent parts are usually removed as they do not carry important information about the instrument and may introduce inaccuracy to the instrument modelling and recognition stage. Finally, the signal is *pre-emphasised*, using a first-order high-pass filter, such as the one in (5.1), increasing the relative energy of the high-frequency spectrum, emphasising formants and harmonics present in the higher frequencies. The pre-emphasis filter was introduced mainly in speech processing applications in order to cancel the glottal or lip radiation effects on speech production, especially when modelling speech with *Linear Predictive Models* [JPH93]. The effectiveness of pre-emphasis for musical signals is not so clear.

$$H(z) = 1 - 0.97z^{-1} \tag{5.1}$$

5.2.2 Feature Extraction

This is the most important part of the whole instrument recognition procedure. In this block, we extract several features from the signal that will be used to identify the instrument. Usually, the signal is segmented into overlapping, windowed frames and for each of these we calculate a set of parameters that constitute the *feature vector*. The performance of the recogniser is mainly determined by the feature set used in this step.

Many feature sets, capable of capturing the aural characteristics of an instrument, were proposed for instrument recognition [Ero01, Mar99]. We are going to outline some of the identifiers that are widely used in research. The effectiveness of each feature vector is usually observed through experiments without any theoretical justification of the superiority against the

other. Sometimes, feature construction follows the researchers' signal processing intuition without any theoretical proof and indeed prove to be effective [PZ03]. In addition, some coefficients seem to be more suitable for modelling certain kind of instruments. Finally, for the construction of the full feature vector, one can use a concatenation of various sets of coefficients. However, that will increase the complexity of the recogniser.

We will briefly present some of the features that can be used for instrument recognition. For a more detailed description on these set of coefficients, the reader can always refer to [Mar99, TC00].

- *Spectral envelopes*: A whole family of coefficients capture frequency envelopes. Among them, we encounter the *Linear Predictive coefficients*, the *Warped Linear Predictive coefficients*, the *Mel-Frequency Cepstral coefficients* (MFCC) and the *Perceptual Linear Predictive coefficients* capture signal envelopes in the frequency, warped log-frequency, mel-frequency and bark-frequency domain respectively. In addition, we can have the *Delta* and *Delta-Delta* versions of the above sets, representing the first and second order derivative of the coefficients.

- *Spectral features*: These include several other measurements that tend to model certain frequency domain characteristics of the signal such as *spectral centroid*, *crest factor* and *spectral flatness measure*.

- *Pitch, Vibrato and Tremolo Features*: these features try to capture basically some special characteristics of the instrument such as the pitch (*fundamental frequency*) range. Another clue for the instrument's identity is the variation of pitch (*vibrato*) introduced by the player and the intensity of such variation (*centroid modulation*).

- *Attack Features*: these features try to capture other temporal characteristics of the instrument, such as the *attack transient* of the signal. Measurements include duration of transient, slope and curvature of the energy profile and zero-crossing rate.

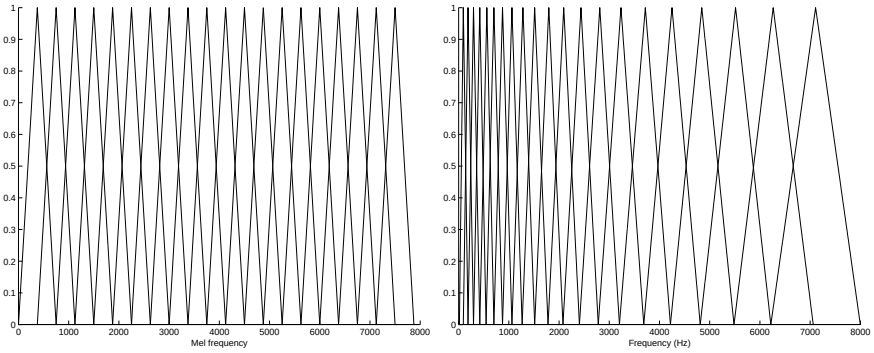

Figure 5.2: MFCC triangular filterbank in the Mel-frequency domain (left) and in the frequency domain (right)

The importance of all these features is discussed in [Ero01, Mar99, KC01, TC00]. In our further analysis, we will use the Mel Frequency Cepstrum Coefficients, as they featured robust performance in a study presented in [Ero01] and also generally in speaker verification [Mit00].

Mel Frequency Cepstrum Coefficients (MFCC)

In this section, we are going to describe briefly the extraction of the *Mel Frequency Cepstrum Coefficients (MFCC)* that will be used in our study, as presented by Rabiner and Juang [RJ93]. This is a standard procedure in feature extraction literature.

Basically, the MFCCs capture the signal energy levels in several frequency bands. To achieve that we construct a *filterbank* of triangular, equally-spaced and 50% overlapping in frequency filters (see figure 5.2). The important feature is that the filterbank is designed in the Mel-frequency domain rather than the frequency domain, in order to emulate the almost logarithmic human perception of frequency. Using the mapping between the Mel and the frequency domain, we map the filters to the frequency domain, where they are actually applied on the signal. The average energy of the filterbank outputs gives the signal energy levels along frequency.

The whole procedure is summarized as follows:

1. Divide the training/test signal into overlapping frames and take the FFT of these frames $x(f,t)$.

2. Choose the number of filters N_{fltr} for the filterbank. Design the triangular filter bank in the Mel-frequency domain and map it back to the frequency domain.

3. Filter the input signal frames with the filterbank and calculate the total energy of each filter's output. As a result, for each frame we have N_{fltr} values $mf(k,t)$ $(k = 1,\ldots,N_{fltr})$, representing the total frame energy that exists in each subband .

4. Finally, in order to compress the information conveyed by the MFCCs, we take the Discrete Cosine Transform (DCT) of the $\log mf(k,t)$, i.e. $MFCC(k,t) = DCT_k\{\log mf(k,t)\}$. Usually, the first component $(MFCC(1,t))$ is dropped and the feature vector is constructing the first $N_{co} < N_{fltr}$ coefficients.

5.2.3 Instrument Modelling

Finally, the feature vectors are used to build a model or reference template for the instrument/person. There are many techniques that have been used for instrument modelling in literature [Mar99]:

- *Vector Quantisers*,
- *Neural Network classifiers*
- *Hidden Markov Models* (HMM)
- *Gaussian Mixture Models* (GMM)

In the following analysis, we are going to use a *Gaussian Mixture Models* (GMM) recogniser, as presented by Reynolds and Rose [RR95]. The GMM will be used to describe the probability density function of the instrument's feature vectors, as a weighted sum of multivariate Gaussian distributions.

If $\underline{v}$ is a feature vector, then the probability model built by a GMM is given by the following equations:

$$p(\underline{v}|\lambda) = \sum_{i=1}^{M} p_i b_i(\underline{v}) \tag{5.2}$$

$$b_i(\underline{v}) = \frac{exp(-0.5(\underline{v} - \underline{m}_i)^T C_i^{-1}(\underline{v} - \underline{m}_i))}{\sqrt{(2\pi)^D |C_i|}} \tag{5.3}$$

where $p_i, \underline{m}_i, C_i$ are the weight, the mean vector and the covariance matrix of each Gaussian and M is the number of Gaussians used. A GMM model is usually described using the notation $\lambda = \{p_i, \underline{m}_i, C_i\}$, for $i = 1, .., M$.

In our analysis, we will assume that each Gaussian has its own covariance (*nodal covariance*), but each covariance matrix is diagonal, i.e. the elements of the feature vector are uncorrelated. This approximation is used to reduce the computational cost of the training procedure, as training full covariance matrices might be computationally expensive. The GMM model is usually trained using an *Expectation Maximisation* (EM) algorithm, as described in [RR95]. The parameters of the model λ_k for each instrument are stored in the database.

5.2.4 Instrument Recognition

A general flow-diagram for performing instrument recognition is shown in figure 5.3. Basically, the preprocessing and the feature extraction stages are identical to the ones used during training. Finally, in the instrument recognition block, we compare the features extracted from the test samples with the models stored in the database, in order to infer the type of instrument.

Suppose we have S instrument models λ_k stored and a set of T feature vectors for the instrument to be identified $V = \{\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_T\}$. The correct model should maximise the following probability:

$$\max_{1 \le k \le S} p(\lambda_k|V) = \max_{1 \le k \le S} p(V|\lambda_k) \tag{5.4}$$

In other words, the model maximising equation (5.2), given the data V, gives us the identity of the instrument. Usually, the log-probability is employed, i.e. we maximise the following function:

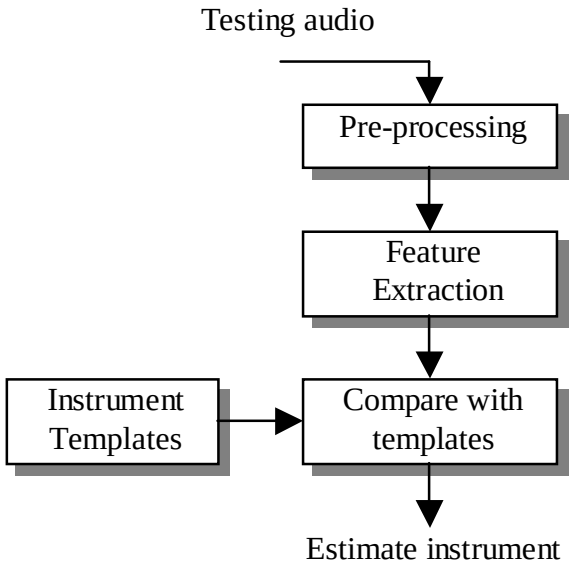

Figure 5.3: A general flow diagram for performing instrument recognition.

$$\max_{1\leq k\leq S} \log p(V|\lambda_k) = \max_{1\leq k\leq S} \sum_{i=1}^{T} \log p(\underline{v}_i|\lambda_k) \tag{5.5}$$

5.3 Intelligent ICA

In this study, we explore the possibility of combining the efficient probabilistic modelling performed by instrument/speaker verification in the ICA of instantaneous mixtures framework of equal number of sources and sensors, as described in Chapter 2. In other words, we impose high-level instrument source models in the current ICA framework, aiming to perform "intelligent" source separation. We propose two ways to perform intelligent separation:

- Combining nonGaussianity measurements and probabilistic inference from the model (Intelligent FastICA).
- Estimating the direction that maximises the posterior probability of the instrument's model (Bayesian approach).

One should point out that the instrument recognition problem that we are called to solve is slightly different to the one usually tackled in the literature before. In instrument recognition, we usually have an audio sample from an instrument/person and we compare the information acquired from this sample with the templates in the database. In the Intelligent ICA case, the problem is quite the opposite. *We know the identity of the instrument/person and we want to identify the audio source that is best represented by the instrument/person's model.* Mathematically speaking, this can be formulated as follows.

Suppose we have N series of feature vectors $\underline{V}_k$, belonging to different audio sources and the desired instrument model λ_{des}. The correct audio source should maximise the following likelihood:

$$\max_{1 \leq k \leq N} p(V_k|\lambda_{des}) \tag{5.6}$$

5.3.1 Intelligent FastICA

In this section, we propose a method to separate the desired source using the kurtosis-based one unit learning law as presented in (2.47) and the GMM model λ that was trained for the specific instrument. We also assume the noiseless, rectangular, instantaneous mixtures model as described in section 2.2.1.

$$\underline{x}(n) = A\underline{s}(n) \tag{5.7}$$

First of all, we prewhiten the observation signals $\underline{x}(n)$, i.e. perform *Principal Component Analysis*. After this step, the sources become uncorrelated, i.e. orthogonal to each other in the N-dimensional space. Assuming that V is the prewhitening matrix, then

$$\underline{z}(n) = V\underline{x}(n) \tag{5.8}$$

The next step is to randomly initiate a one-unit learning rule based on nonGaussianity, for example Hyvärinen's kurtosis-based algorithm, as

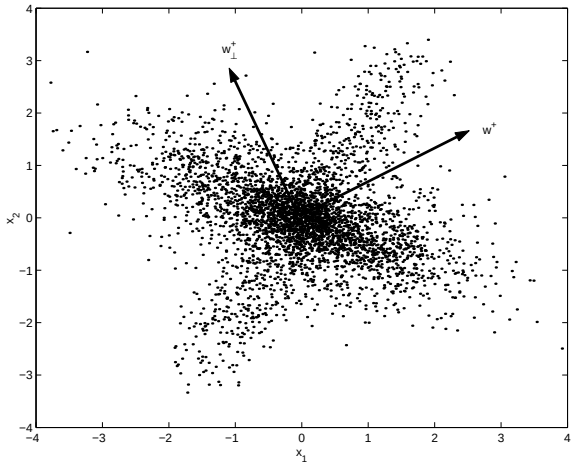

Figure 5.4: A scatter plot of the two sources, two sensors case. Getting an estimate of the most nonGaussian component can give an estimate of the other component in the prewhitened 2D space.

described in [HO97].

$$\underline{w}^+ \leftarrow \mathcal{E}\{\underline{z}(\underline{w}^T\underline{z})^3\} - 3\underline{w} \tag{5.9}$$

$$\underline{w}^+ \leftarrow \underline{w}^+/||\underline{w}^+|| \tag{5.10}$$

Consequently, we need to get N orthogonal estimates $\underline{w}_i^+$ towards the N nonGaussian components in the mixture. This is performed by randomly initialising N learning rules (5.9) and keeping the new estimates orthogonal to each other, forming an orthonormal basis (see section 2.2.7). An example of the 2×2 case is illustrated in figure 5.4. The first estimate given by the ICA algorithm is noted by $\underline{w}^+$. The direction of the orthogonal vector will then be $\underline{w}_\perp^+ \leftarrow \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \underline{w}^+$.

The next step is to unmix the estimated sources at each direction and perform *instrument verification*. In other words, extract the feature vectors $\underline{v}_i$ from each of the estimated signals and then calculate the probability $p(\underline{v}_i|\lambda_{des})$ given the desired instrument model λ_{des}, as described by (5.2), (5.3). The direction that maximises this probability is the best estimate of the direction of the desired source.

The same procedure is repeated until convergence. The *likelihood comparison step* enables us to choose the desired local maximum of kurtosis. This method works in *batch mode*, i.e. processing all, or at least big blocks of, the available data set. This is essential in order to perform fast estimation of the nonGaussian components and accurate instrument recognition. A stochastic gradient approach will not be suitable for instrument recognition.

The algorithm is summarised as follows:

1. Prewhiten the data, i.e. decorrelate the sources.
2. Randomly initialise the one-unit algorithm, getting N orthogonal estimates of nonGaussian components.
3. Extract features $\underline{v}_i$ from each estimate and choose the one that maximises $p(\underline{v}_i|\lambda_{des})$, where λ_{des} is the desired instrument model.
4. Repeat until convergence

5.3.2 Bayesian Approach

In this section, we investigate whether the trained instrument probability model can itself give us an efficient tool for performing "intelligent" source separation. More specifically, we will maximise the posterior probability of the model to form a *Maximum Likelihood* (ML) estimate of the unmixing vector $\underline{w}$. For our analysis, we will assume that the observed signals are again prewhitened, to allow for orthogonal projections. The optimisation problem is set as follows:

$$\max_{\underline{w}} G(\underline{w}) \tag{5.11}$$

where $G(\underline{w}) = \log p(\underline{v}|\underline{x}, \lambda)$ is defined by equation (5.2, 5.3).

We can form a *gradient ascent* solution to this problem, which is given by the following law:

$$\underline{w}^+ \leftarrow \underline{w} + \eta E\{\frac{\partial G}{\partial \underline{w}}\} \tag{5.12}$$

$$\underline{w}^+ \leftarrow \underline{w}^+ / ||\underline{w}^+|| \tag{5.13}$$

where η is the learning rate of the gradient ascent. Therefore, we have to calculate the $\partial G/\partial \underline{w}$. Forming an expression connecting $G(\underline{w})$ with $\underline{w}^T\underline{x}$ is not so straightforward, as it is not easy to represent feature extraction with a function f, i.e. $\underline{v} = f(\underline{w}^T\underline{x})$. However, we can split the derivative in the following parts:

$$\frac{\partial G}{\partial \underline{w}} = \frac{1}{p(\underline{v})}\frac{\partial p(\underline{v})}{\partial \underline{v}}\frac{\partial \underline{v}}{\partial \underline{w}} \tag{5.14}$$

where

$$\frac{\partial p(\underline{v})}{\partial \underline{v}} = -\sum_{i=1}^{M} p_i b_i(\underline{v}) C_i^{-1}(\underline{v} - \underline{m}_i) \tag{5.15}$$

The term $\partial \underline{v}/\partial \underline{w}$ is hard to define. In our analysis, we performed numerical calculation of this derivative. Another approach can be to perform numerical calculation of the whole derivative.

However, a Maximum Likehood estimate may not always be accurate. In the following paragraph, we demonstrate that the Gaussian Mixtures estimator can be sensitive to additive noise plus other perturbations in the time domain. We generated an instrument recognition system using 18 MFCCs and 16 Gaussian Mixtures. We trained models λ for three instruments: violin, piano and acoustic guitar. We measured the log-likelihood $p(\underline{v}|\lambda)$ (5.2), (5.3) for various test waveforms given the three trained models. The results are summarised in table 5.1. First of all, we wanted to test the instrument recognition performance of this setup. In the first two lines, we compare the likelihood of samples from the trained instruments with the likelihood of accordeon samples given the corresponding instrument model. We can see that in all cases, the estimator prefers the correct instrument from the accordeon, i.e. performs correct instrument recognition. Then, we examine the effect of additive Gaussian noise in the time domain on the estimator. We compare the likelihood of the correct signal given the correct model with the likelihood of the correct signal plus Gaussian noise of zero mean and variance 0.01 (assuming that the instrument signals are scaled to unit variance). We observe that in all cases, the estimator seems to prefer the noise-corrupted version rather than the original version. We will see

that this can have an effect on the Intelligent ICA approach, when we have mixtures of different instruments. In line 4 of table 5.1, we calculate the probability of a linear mixture containing 90% of the correct signal and 10% of accordeon samples. We can see that the estimator, in almost all cases prefers the mixture rather than the correct source. Note, however, that this does not imply that the model prefers accordeon type features, e.g. final row Table 5.1. In all cases, the estimator prefers the correct signal than the mixture, which implies that we can use this criterion for source separation. However, steps must be taken to control this inherent sensitivity to noise.

Table 5.1: Inaccuracy of Maximum Likelihood estimation in instrument recognition. We also demonstrate the models' performance in instrument recognition and in presence of additive Gaussian noise and linear mixtures of the three instruments individually with accordeon. All results scaled by 10^3 and v_{acc} represents feature vectors from accordeon samples.

$\underline{v}, \lambda$	Violin	Piano	Acoustic guitar
$\log p(\underline{v}\|\lambda)$	-4.80	-7.33	-5.36
$\log p(\underline{v}_{acc}\|\lambda)$	-6.13	-9.18	-6.02
$\log p(\underline{v} + \mathcal{N}(0, 0.01)\|\lambda)$	-4.61	-7.24	-4.94
$\log p(0.9\underline{v} + 0.1\underline{v}_{acc}\|\lambda)$	-4.75	-6.24	-5.26
$\log p(0.1\underline{v} + 0.9\underline{v}_{acc}\|\lambda)$	-5.76	-8.47	-5.59

It turns out that this problem is not new and was also observed in many speaker verification approaches [CLY96, SZ97]. In order to deal with misclassification errors, the log-probability of the user claiming identification is normalised to the mean of the log-probabilities of all the *cohort* speakers, i.e. the speakers whose models are very similar (i.e. scoring equally well). This operation similar to median filtering is usually known as *cohort normalisation* [CLY96, SZ97]. Therefore, it would be beneficial for the Bayesian Intelligent ICA to optimise the following cost function, in order to improve

the performance.

$$G(\underline{w}) = \log P(\underline{v}|\underline{x}, \lambda_{des}) - \frac{1}{N-1} \sum_{\substack{i=1 \\ i \neq des}}^{N} \log P(\underline{v}|\underline{x}, \lambda_i) \tag{5.16}$$

where λ_{des} is the model of the desired instrument and λ_i are the models of the other instruments present in the mixture. In other words, we try to maximise the difference between the desired instrument and all other instruments present in the mixture. In fact, (5.16) is not restricted to "cohort" models, as previously mentioned. As the number of sources and sensors N used is small, it would be realistic to use all available models. In cases of large N, we may have to restrict our search to the "cohort" models to reduce computational complexity.

This function is difficult to optimise, therefore, we calculated numerically the derivative of G. However, in the following section we demonstrate that it is possible to perform intelligent blind separation by using the posterior likelihood of the instrument/person model.

5.4 Experiments

In this section, we are going to test the proposed schemes using some basic experiments. First of all, we had to configure the instrument identification system. We tested several feature vectors - number of Gaussians configurations. However, as the aim was to build a fairly simple, fast and robust system, we ended up using a combination of 18 MFCCs and 16 Gaussian Mixtures ($M = 16$). The MFCCs performed reasonably well in our study, as well as in [Ero01]. One can argue in favour of other coefficients, depending on the difficulty of the instrument recognition problem. We chose to discriminate between 5 different family instruments, such as: *violin*, *accordeon*, *acoustic guitar*, *piano* and *tenor saxophone*, in order to verify the idea of intelligent source separation. A general system would possibly require more Gaussians or more appropriate features for intra-instrument family discrimination [Ero01].

We trained our system using ~ 6 minutes of solo recordings from each instrument. The analysis frame size was $16msecs$. The model was trained using 18 MFCCs and 16 Gaussian Mixtures ($M = 16$) and the model was stored locally. For the model training, we used the corresponding function from VOICEBOX [Bro]. For the recognition process, we used around 30 seconds of different solo instrument recordings. Random instantaneous mixtures were created to test the intelligent ICA algorithms.

5.4.1 Intelligent FastICA

The Intelligent FastICA framework provided very fast, robust separation of the desired source (see figure 5.5). Tests using all pairs of the trained instruments were successful. Minor misclassification errors were due to the recogniser's inefficiency. A more complex and competent instrument recognition system could rectify these mistakes. In addition, due to the good convergence speed of the FastICA, we managed to demonstrate the effectiveness of Intelligent FastICA. The algorithm would pick the desired source in a couple of iterations ($\sim 2-3secs$ for an average Pentium III computer).

One of the advantages of this proposed framework is that it is *modular*. The *instrument recognition* module is totally independent to the *source separation* module. Any instrument recognition setup will work in the proposed framework without any modification. Therefore, we may change the recogniser according to our accuracy needs. In addition, any source separation module can be used, according to our needs.

One of the drawbacks is that the proposed framework is effectively a post-separation recognition task. This implies that in the search of an individual source, we would have to make an estimate of all the present sources and test them all using the instrument classifier. Although this might not be a problem for $3-4$ sources, it can be quite computationally expensive and not very efficient in the case of many sources.

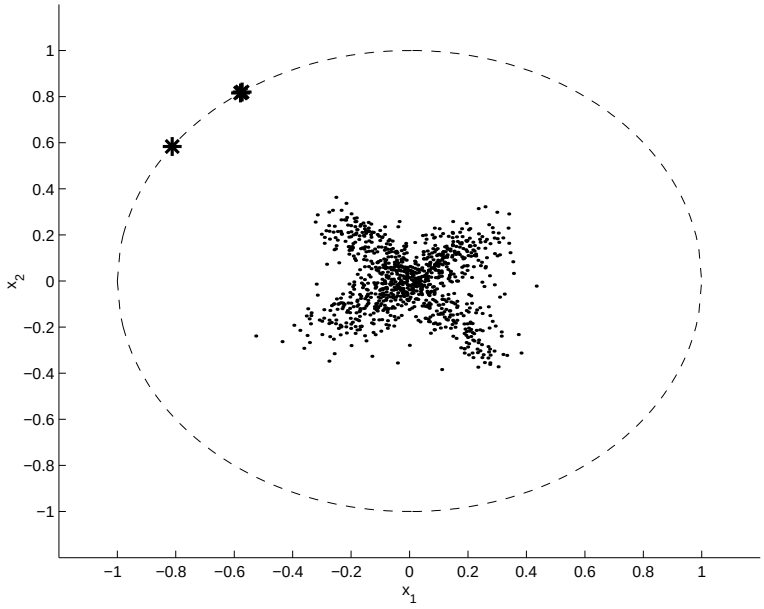

Figure 5.5: Fast convergence of the intelligent FastICA scheme. The stars on the unit circle denote the algorithm's iterations.

5.4.2 Bayesian Approach

Our first task for the Bayesian approach was to verify the principle that one can perform intelligent source separation optimising the posterior likelihood of the observed signal given the desired model. We will formulate 2×2 and 3×3 examples to evaluate the nature of the likelihood function and the validity of the arguments above, therefore, the general $N \times N$ case will not be examined.

We used the two sources - two sensors scenario to demonstrate this principle. The observations are initially prewhitened. Once prewhitened, all our solutions $\underline{w}$ lie on the unit circle and as a result the desired signal can be separated using an orthogonal projection in the form:

$$u_{des} = \underline{w}^T \underline{z} = \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix} \underline{z} \tag{5.17}$$

where θ represents the "direction" of the desired source. Hence, in the 2

sources 2 sensors case, we can express $G(\underline{w}) = G(\theta)$ (see (5.14)) and optimise it in terms of θ rather than $\underline{w}$. As a result, the problem has dropped to a one-dimensional optimisation problem.

In figure 5.6, we plot the function $G(\theta) = \log p(\underline{v}|\underline{x}, \lambda)$ for two trained instruments (accordeon, acoustic guitar) that are present in the auditory scene for various values of $\theta \in [0, \pi]$. The following observations were made:

First of all, $G(\theta)$ is a smooth function that will be easy to optimise, even using numerical optimisation, as described earlier. This might be due to the smoothness of the Gaussian Mixtures model and the instantaneous mixing of the sources.

The "direction" of the sources can easily be depicted from the graph and they are orthogonal to each other, as expected due to prewhitening. However, we should expect to see a clear peak at the direction of the desired instrument and at the same time a minimum at the direction of the unwanted instrument. In figure 5.6, we can see that this is true for the case of one instrument (i.e. $G_1(\theta) = \log p(\underline{v}|\underline{x}, \lambda_1)$), however, we do not get a clear peak in the case of the second instrument (i.e. $G_2(\theta) = \log p(\underline{v}|\underline{x}, \lambda_2)$). This is mainly due to the recogniser's sensitivity to noise, as described in the previous section. As a result, the recogniser may consider a linear mixture of the desired source with slight contamination from the other sources more probable than the original source. This can cause inaccuracy in the estimation procedure.

In addition, this is an attempt to identify a specific source in a mixture, using a model only for the source of interest and not for the other sources. This is a difficult task as the estimator does not have any profile for the sources it needs to reject. This may also explain the interdeterminacies shown in figure 5.6. Comparing information about the other instruments in the auditory scene might enhance performance.

To rectify this inaccuracy, we can use cohort normalisation. In figure 5.7, we plot the cohort normalised function $G(\theta)$, as described in eq. 5.16. In this case, we spot clear peaks and nulls for each instrument in either case.

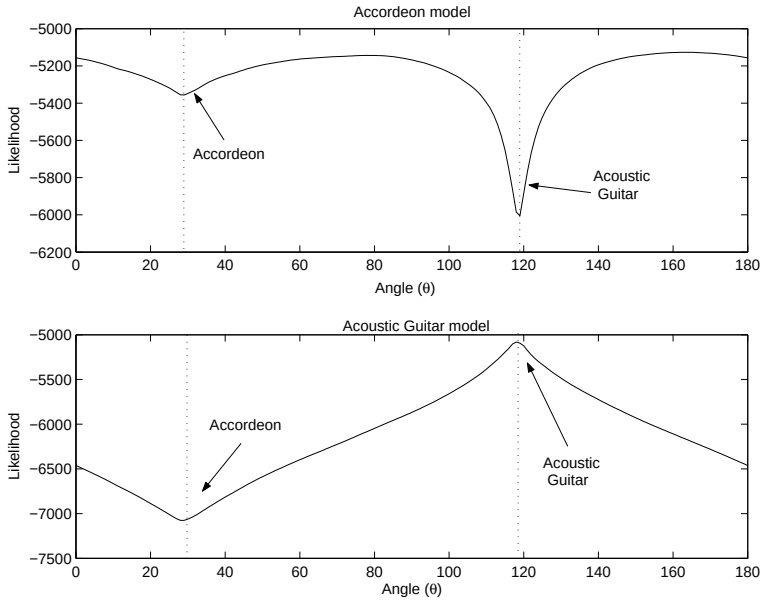

Figure 5.6: Plot of $G(w)$ (eq. 5.11), as a function of θ for the two instruments (accordeon (up) and acoustic guitar (bottom)).

As a result, cohort normalisation can indeed rectify some misclassifications, introduced by Maximum Likelihood estimation.

The numerical optimisation of the cohort normalised function can achieve successful intelligent separation of the sources, due to the smoothness of the cost function. However, the speed of the optimisation varies due to the numerical calculation of the gradient and the learning rate choice. The algorithm's convergence was tested for any of the trained instruments with success. In figure 5.10, we can see the algorithm's slow convergence, optimising the cohort normalised likelihood for the accordeon.

The effectivess of cohort normalisation can also be observed in figures 5.8, 5.9, where we can perform "intelligent ICA" between *piano* and *violin*. Again, we can see the multiple peaks in the case of the single model plot, whereas the cohort normalised plot can give global solutions.

Another point that needs to be clarified is the following: In the Intelligent FastICA approach, we used the nonGaussianity along with inference from

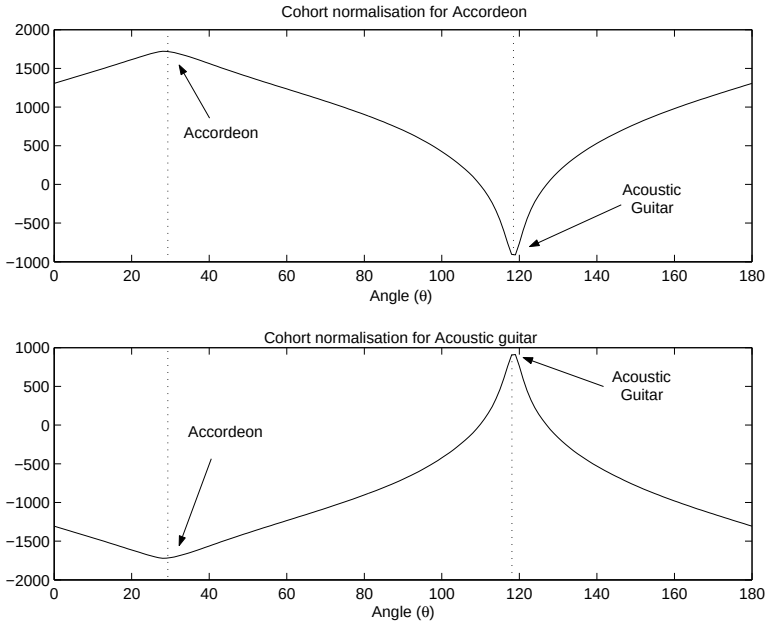

Figure 5.7: Plot of cohort normalised $G(w)$ (eq. 5.16), as a function of θ for the two instruments (accordeon and acoustic guitar).

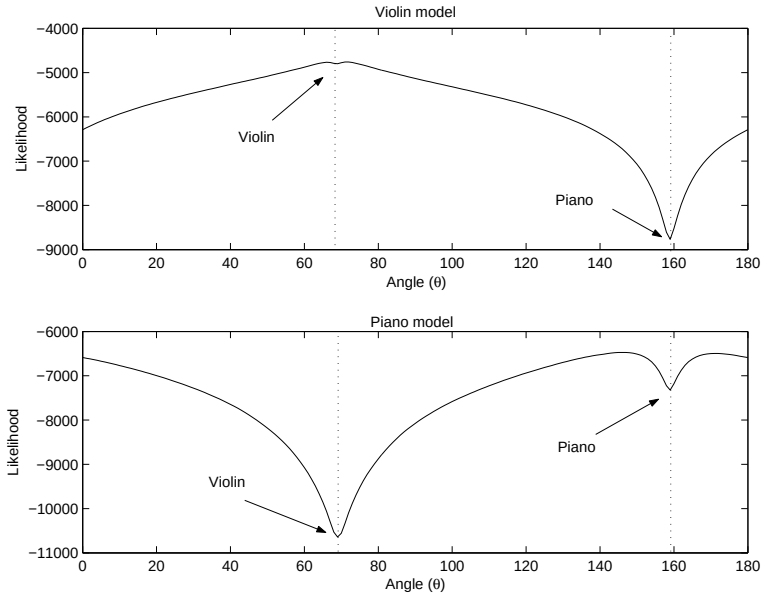

Figure 5.8: Plot of $G(w)$ (eq. 5.11), as a function of θ for the two instruments (violin (up) and piano (below)).

$G(\underline{w}) = \log p(\underline{v}|\underline{x}, \lambda)$ (non-normalised version). However, we do not get the same interdeterminancy in that case. This is because we always compare the likelihood of estimates orthogonal to each other. In the 2×2 case, this implies that we will compare $G(\theta)$ with $G(\theta + 90^\circ)$, as we optimise θ with the nonGaussianity algorithm. Observing figure 5.6, we can see that this comparison can always direct us to the desired source. The nonGaussianity algorithm will then ensure the accurate estimation of the desired source.

We also performed some tests in the 3×3 case, using accordeon, acoustic guitar and violin. Once prewhitened, the desired signal can be separated using an orthogonal projection in the following form, as all solutions lie in the 3-D unit sphere. This transforms the problem to a two-dimensional optimisation problem $G(\theta_1, \theta_2)$.

$$u_{des} = \underline{w}^T \underline{z} = \left[\begin{array}{ccc} cos\theta_1 \cos\theta_2 & \cos\theta_1 \sin\theta_2 & \sin\theta_1 \end{array} \right] \underline{z} \tag{5.18}$$

In figures 5.11, 5.12, 5.13, we can see the cohort normalised version of the

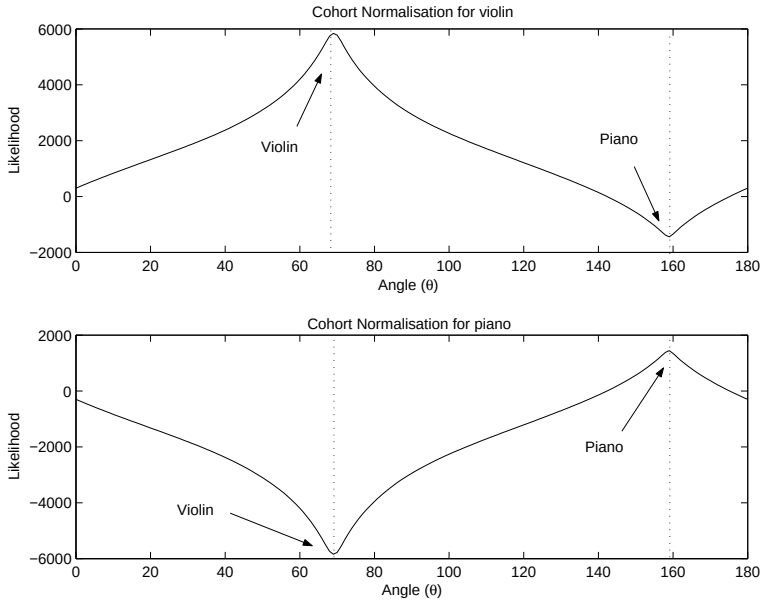

Figure 5.9: Plot of cohort normalised $G(w)$ (eq. 5.16), as a function of θ for the two instruments (violin and piano).

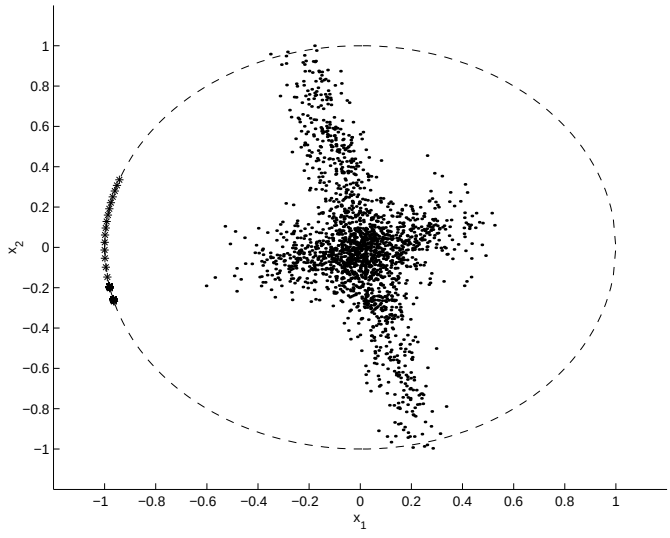

Figure 5.10: Slow convergence of the numerical optimisation of the cohort normalised likelihood. The stars on the unit circle denote the algorithm's iterations.

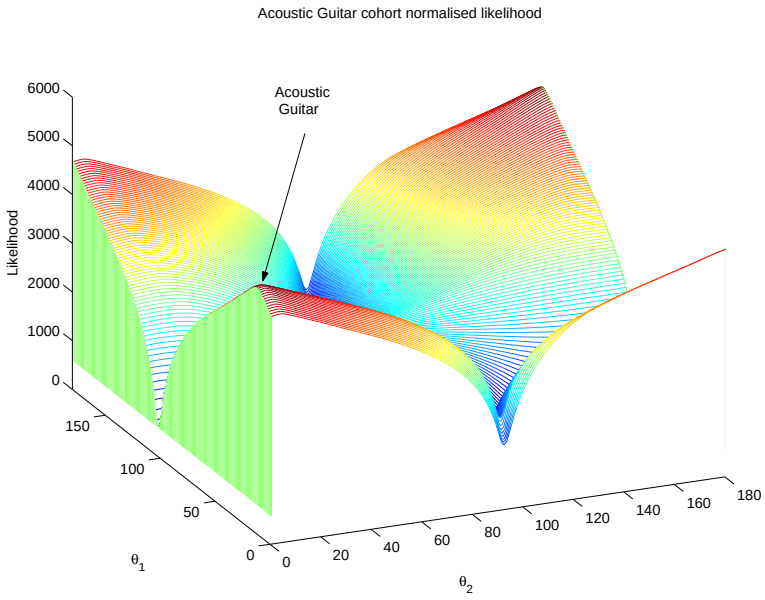

Figure 5.11: Plot of the cohort normalised $G(w)$ (eq. 5.11), as a function of θ_1, θ_2 in the 3×3 case for the acoustic guitar case.

$G(\theta_1, \theta_2)$ for the acoustic guitar, accordeon and violin respectively. We can see that generally the cohort normalised likelihood can give robust global solutions for the source separation problem, even in higher dimensional cases, however, problems similar to the unnormalised likelihood have been seen to occur (see figure 5.12).

In contrast to the Intelligent FastICA approach, another advantage of the proposed Bayesian framework is that we estimate *only the source of interest* and not all the sources that are present in the auditory scene. Instead, in the proposed Bayesian framework, we use models of the other sources, rather than actual sources' estimates. This might be a benefit in the case of many sources.

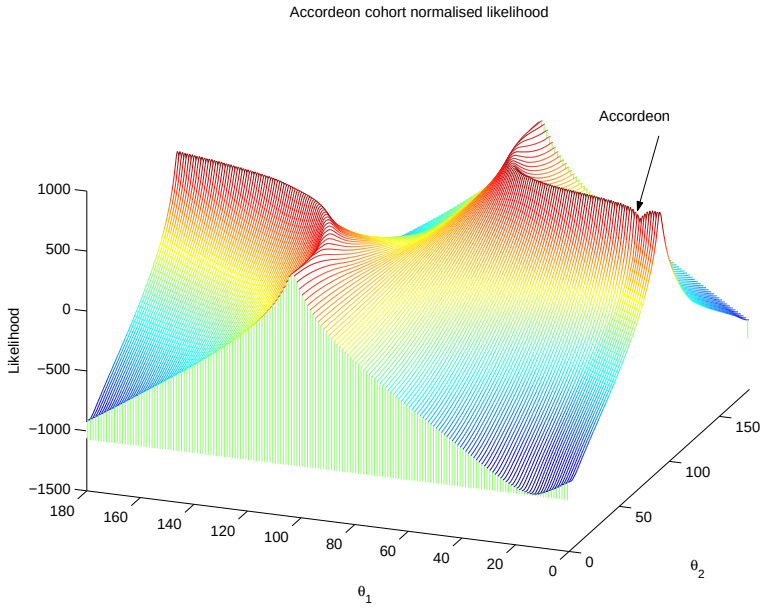

Figure 5.12: Plot of the cohort normalised $G(w)$ (eq. 5.11), as a function of θ_1, θ_2 in the 3×3 case for the accordeon case.

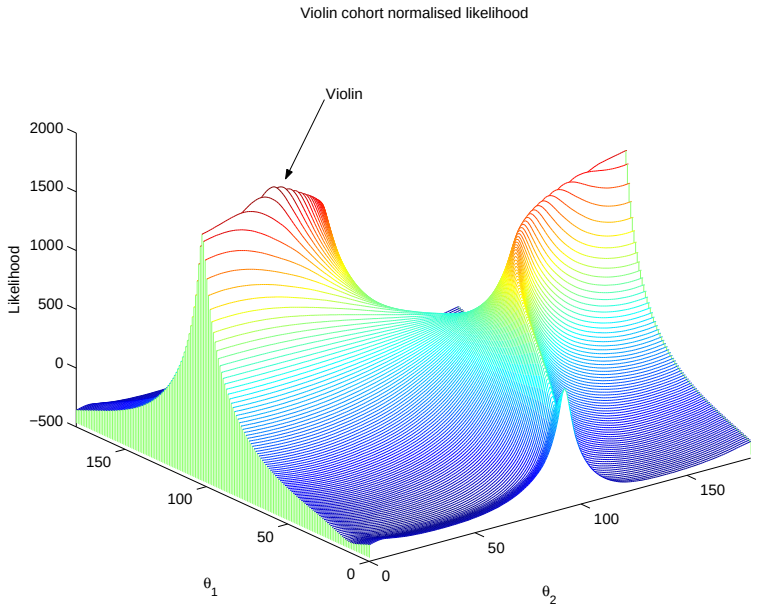

Figure 5.13: Plot of the cohort normalised $G(w)$ (eq. 5.11), as a function of θ_1, θ_2 in the 3×3 case for the violin case.

5.5 Conclusion

In this chapter, we explored the idea of performing "intelligent" source separation. In other words, we investigated the idea of separating a desired source of interest from the auditory scene. We explored the problem in the instantaneous mixtures case, combining current developments in the area of *instrument recognition* and *source separation*. A feasibility study was conducted to demonstrate that "intelligent" source separation is possible. Two methods were proposed to support this argument. One combining a nonGaussianity ICA algorithm and inference from a simple instrument recognition system. Another approach to intelligent source separation is by maximising of the *cohort-normalised posterior likelihood* with good results but difficult optimisation.

An improvement to the cohort normalisation scheme might be to train a *global "noise" model*, using samples from all the instruments. Hence, in the cohort normalisation step we would have to compare the likelihood of the desired instrument model against the likelihood of the "noise" model, instead of averaging over the likelihoods of the other instrument models. This might reduce the computational complexity as well.

The results presented in this chapter highlight a fundamental weakness of these traditional instrument/speaker recognisers. So far, all modelling efforts have concentrated on optimising the performance of various feature sets and statistical models for instrument/speaker recognisers. In this chapter, we extended the use of these models in the slightly different framework of source separation and we observed that these models can be sensitive to noise and linear mixtures with other sources. In other words, the robust performance of these models is limited outside the area they were designed for. As a result, we need better models that will be able to characterise and recognise instruments/speakers despite the presence of noise or other instruments.

Another improvement might be to replace Gaussian Mixtures modelling with Hidden Markov modelling as proposed recently by Reyes et al [RRE03]

for source separation. Hidden Markov Models have been widely used in the speaker/instrument recognition community with great success [CLY96], the only disadvantage being their computational complexity.

A possible extension of the ideas presented in this chapter will be to adapt the concept of "intelligent" source separation to the overcomplete ICA case. The knowledge of an instrument model might be an extra tool for the case of more sources than sensors. Although, the presence of reverb will deteriorate the performance of an instrument recognition system, the idea of "intelligent" source separation should be expanded to the case of convolutive mixtures.

Chapter 6

Conclusions-Future Work

6.1 Summary and Conclusions

In this text, we have covered many aspects in the field of *Audio Source Separation* using *Independent Component Analysis* (ICA). This section will summarise the main issues of the problem, giving specific emphasis on the observations and improvements introduced in this text.

Audio source separation is the problem of decomposing an auditory scene into its audio components (objects). This is a natural task for humans, however, many issues need to be addressed when it comes to implement such a system using a computer and a set of sensors, capturing the auditory scene. There are methods, which try to emulate human perception, using a set of grouping rules for several features of the spectrogram, in order to perform separation (Computational Auditory Scene Analysis). We mainly investigated methods that observe the signals' statistics and separate the sources looking for components with specific statistical profile. The directivity of these audio signals towards the sensor array was also employed.

We decomposed the source separation problem into three subproblems, each of them trying to deal with a specific aspect of the problem. Altogether, they can form a consistent model for the real audio source separation problem. More specifically, we firstly reviewed current solutions in source

separation of *instantaneous mixtures* of equal number of sources and sensors. The problem consists of estimating the unmixing matrix. The general ICA framework was introduced assuming *statistical independence* between the audio components. Many possible ways of interpreting statistical independence produced a couple of ICA algorithms to estimate the unmixing matrix. We examined some of them, such as the Bell and Sejnowski, the natural gradient, the FastICA and the JADE algorithm. All these algorithm have very good performance and tend to have similar update rules, although they were derived from different principles. The subproblem of more sources than sensors was then examined. The *overcomplete ICA* problem is slightly different, as there are two outstanding issues: a) estimate the mixing matrix and b) estimate the components given the estimated mixing matrix. Many solutions proposed based on a Bayesian framework (Lewicki, Attias) and some others based on clustering approaches (Hyvärinen, Zibulevski). As this is an "ill-determined problem", the separated outputs usually feature no crosstalk between the sources, but they seem to be slightly distorted, compared to the original sources. Finally, the subproblem of source separation of real world recordings was investigated in the case of equal number of sources and sensors. We introduced the "*convolutive mixtures*" model, where the sensors capture convolutions of each source with FIR filters that model the room transfer function between each sensor and source. Based on the general ICA framework, a couple of *time-domain* methods and *frequency-domain* methods were discussed, estimating the unmixing filter or the audio sources either in the time or in the frequency domain (Lee, Smaragdis, Parra etc). Choosing the domain to perform these tasks can have several advantages and disadvantages.

Estimating the unmixing filter in the time-domain can have limited performance due to the computational cost of the convolution and the speed of the adaptation. As a result, unmixing in the frequency domain seems to be an obvious choice. For each frequency bin, we can apply an instantaneous mixtures ICA algorithm to perform separation. The question is where to ap-

ply the source model for the ICA algorithm. Modelling in the time-domain has the advantage that the inherent permutation ambiguity does not seem to exist, however, we need to perform continuous mappings to and from the frequency domain. To avoid this, we can model the sources in the frequency domain. We can run independent instantaneous mixtures ICA algorithms for each frequency bin, assuming that the source models in each case are statistically independent. This causes a source ordering ambiguity between the frequency bins, known as the *permutation ambiguity*. Along with the permutation ambiguity, the frequency-domain framework features a *scale ambiguity*, that is inherent to the ICA algorithm.

To deal with the scale ambiguity, we proposed to *map the separated sources, back to the observation space*. As Cardoso initially pointed out, this can rectify any arbitrary scaling, performed by the ICA algorithm. We showed that the scale ambiguity can be removed even with the permutation ambiguity existing. A number of solutions were proposed for the permutation ambiguity. Smaragdis proposed to couple the filters of neighbouring bins and Parra proposed to impose a smooth filter constraint (channel modelling approaches). Lee proposed to model the signals in the time-domain (source modelling approach). To solve the permutation problem, we imposed a time-frequency model that could be used in any Maximum Likelihood ICA approach. Together with a Likelihood ratio jump, we can couple the frequency bins that bear the same energy bursts along time (average energy envelop along time). This solution seemed to be robust, even in the case of real room recordings. One concern is that it can be computationally expensive in the case of more than two sources.

All proposed frequency-domain ICA frameworks used the natural gradient algorithm to perform separation. Gradient-type algorithms may be generally robust for optimisation, however, they have certain drawbacks: a) they *converge relatively slowly*, b) their *stability* depends on the *choice of the learning rate*. As a result, we replaced the natural gradient algorithm with a faster and more robust algorithm. Based on Hyvärinen's work on

Maximum Likelihood FastICA algorithms for instantaneous mixtures, we adapted these algorithms into the frequency-domain ICA framework, fitting the time-frequency model plus Likelihood Ratio solution for the permutation problem. The result was a *Fast Frequency-domain framework*, which featured robust performance even with real room recordings.

We also explored several general aspects of the proposed frequency-domain framework. One can interpret the Short-Time Fourier transform as a filterbank with poor performance. As a result, the proposed unmixing framework introduces aliasing between the neighbouring frequency bins. We demonstrated that this aliasing can have minor effects using *oversampling*, i.e. greater overlap ratio. Another solution might be the substitution of the FFT with a more efficient filterbank. Then, we revised the benefits of source modelling in a sparser domain than the time-domain. However, we explored the effects of the frame size in the frequency domain framework for real room acoustics. We demonstrated that even in the frequency domain, the nonGaussianity of the signal can drop in the case of long frame sizes, that are used to model real room acoustics. As a result, the performance of the source separation estimator deteriorates.

In the next section, we explored some channel modelling solutions for the permutation problem using *beamforming*. The ICA setup can be regarded as an array and therefore knowledge from array signal processing can be used as a channel modelling approach to solve the permutation problem. Assuming that there is a strong direct path in the room transfer functions, one can align the permutations matching the main Directions of Arrival (due to the direct path) of the signals along frequency (Saruwatari et al, Ikram and Morgan, Parra and Alvino). We explored the possibility of using array signal processing to solve the permutation ambiguity. We also saw that the multipath environment causes a slight drift of $\pm 3^{\circ}$ degrees around the main DOA (introduced by the direct path) along frequency. We saw that the directivity patterns tend to feature multiple nulls increasing with frequency. We showed that the frequency that the multiple nulls start to appear at is a

function of the distance between the microphones. The multiple nulls hinder the alignment of the beampatterns as the frequency increases. Keeping the distance between the microphones small, we can shift this phenomenon to the higher frequencies, however, we deteriorate the quality of the separation, as the sensor signals become more correlated and the Signal-Noise Ratio will drop. We proposed a mechanism for DOA estimation, averaging over the directivity patterns of the lower frequencies. We then tried to align the permutations around the estimated DOA with relative success. One improvement over this scheme was to use directivity patterns created with the MuSIC algorithm. As the MuSIC algorithm implies a more sensors than sources setup, it is not applicable in our case. However, having separated the signals using an ICA algorithm, we can map the sources back to the sensor space, and have an observation of each source at each microphone. This will enable us to use the MuSIC algorithm for more efficient DOA estimation and permutation alignment along frequency. Last but not least, we made a preliminary investigation of the sensitivity of the proposed frequency domain framework to movement. We observed that a slight movement of one source will not greatly affect the lower frequencies, however the mid-higher frequencies may render the old unmixing filters useless. As a result, the distortion introduced by movement is a function of frequency. We also demonstrated with a real room example that in the case of one source moving, the old unmixing filters will separate the moving source with a little more reverb. On the other hand, they will not separate the source that did not move, due to incorrect beamforming of the array.

Finally, we introduced the idea of *"intelligent" Independent Component Analysis*. Effectively, our brain does not separate all the sources at the same time, but instead focuses only on the one, we are interested in. Consequently, 'intelligent" ICA tried to separating a specific source of interest out of the auditory scene. This can be achieved by incorporating knowledge from instrument recognition. Instrument models are trained before separation, and we demonstrated that the probabilistic inference from the

instrument model can separate the desired instrument. We used a simple instrument recognition setup consisting of 16 Gaussian Mixtures and 18 Mel-Cepstrum coefficients and instantaneous mixtures. We demonstrated that "intelligent ICA" can be performed either as a post-processing step after a common ICA rule, or simply by optimising the posterior likelihood of the model. Optimising the difference between the probability of the desired model and that of the unwanted models seemed to get more robust separation. In addition, we highlighted a fundamental weakness of traditional recognisers to additive noise and small perturbations of sources.

6.2 Open problems

In this thesis, we have discussed several aspects of the general audio source separation problem and proposed some solutions on open problems in this field or some thoughts and considerations on some other problems. However, there is a large number of open problems in the field that we did not have time to address in this text. To conclude the overview of the audio source separation problem, we would like to use this last section to briefly outline some of the most important outstanding issues in the audio source separation problem. Research in these issues might enhance the performance of current source separation systems in the future.

6.2.1 Additive noise

A very small number of the present approaches for convolutive mixtures tend to remove possible additive noise, during the separation. The additive noise is still a very difficult and open problem for the Blind Source Separation framework. The impact of noise on the estimator's performance depends on the type and level of noise. Cardoso [Car98a] has pointed out that the benefits of noise modelling for Blind Source Separation are not so clear. In cases of high SNR, the bias on estimates for A are small and some noise reduction can be achieved using robust denoising techniques as a pre- or

post-separation task. In cases of low SNR, the problem is very difficult to solve anyway.

The main argument is that denoising can always be a post-processing task, as the fourth-order statistics usually employed by the ICA algorithm are theoretically immune to possible additive Gaussian noise, as for example $kurt\{A\underline{s} + \underline{\epsilon}\} = kurt\{A\underline{s}\}$, where $\underline{\epsilon}$ models the noise. For other fourth-order measures, there are some bias removal techniques proposed by Hyvärinen [Hyv98, Hyv99b]. Therefore, the estimator of the mixing matrix A is effectively not influenced by the additive noise. However, the estimates for the sources contain some additive noise, as you can always formulate the mixtures, as follows:

$$\underline{x} = A\underline{s} + \underline{\epsilon} = A(\underline{s} + \underline{\epsilon}_1) \tag{6.1}$$

As a result, we can always perform post-denoising of the noisy estimated components, using common denoising techniques, as explained by Godsill et al [GRC98], or even perform *sparse code shrinkage*, as proposed by Hyvärinen [Hyv98, Hyv99b].

In addition, there are some overcomplete ICA approaches that perform denoising along with the separation [Att99, DM04]. It would be nice to have a denoising module in the audio source separation framework, making it a general audio enhancement tool as well.

6.2.2 Dereverberation

All convolutive source separation algorithms aim to estimate the independent components present in the auditory scene. Although most of these algorithms are inspired by *blind deconvolution*, they do not aim to dereverberate (deconvolve) the input data but simply identify the audio sources present in the auditory scene. The main reason behind that is that the cost function we optimise for convolutive source separation tries to isolate the independent components rather than dereverb the actual signals. As a result, after separation we get the independent sources , as they would be captured by the sensors, if they were alone in the room.

Again, dereverberation can be a post- or even a pre- processing task. There are many blind deconvolution methods that can enhance the separated audio objects [GFM01]. However, if we perform deverberation along with the source separation task, we might enhance the performance of the source separating system. In section 3.7, we explored the effect of the frame size plus the effect of reverb on the fourth-order statistics of the signal and more specifically on kurtosis. We saw that reverberation renders the signal more Gaussian, even in the frequency domain. However, we know that the Cramer-Rao bound of the source separation estimator depends mainly on the nonGaussianity of the signal. As a result, if we perform deverberation alongside source separation, we will make the signals more nonGaussian to the benefit of the source separation algorithm.

A possible deverberation approach that might be appropriate for the source separation environment is using *Linear Predictive modeling* as explored by Gillespie et al [GFM01] and Kokkinakis et al [KZN03]. One of the benefits is that Linear Predictive analysis will not be too computationally expensive for the audio source separation framework.

6.2.3 More sources than sensors in a convolutive environment

A possible extension of the proposed framework in this thesis is to adapt strategies for the case of more sources than sensors in the convolutive case. In Chapter 2, we have reviewed a couple of techniques for tackling the "overcomplete" instantaneous mixtures problem. However, there is not so much work done on the "overcomplete" convolutive mixtures problem.

The best known example of such a method is the DUET algorithm, as presented by Rickard et al [JRY00, RBR01]. The DUET algorithm assumes a specific delayed model, that is mainly applicable in audio source separation cases with small delays, such as hearing aids etc. Using a two-sensor model, the DUET algorithm can separate the sources, by calculating *amplitude differences* (AD) and *phase differences* (PD) between the sensors. The

sources tend to be fairly localised in a AD vs PD plot, thus enabling efficient separation. Several improvements on the DUET algorithm have been proposed to enhance its performance [VE03, BZ03]. However, such systems are limited to work in a "friendly" environment. Their performance in real reverbant rooms is very limited.

Ideally, we would have to convert some of the overcomplete strategies to work into the frequency domain for convolutive mixtures. A proper Maximum Likelihood solution, as proposed by e.g. Lewicki [LS98], in the frequency domain together with a solution for the permutation problem would be computationally heavy. Based on Attias' general Gaussian Mixtures strategy [Att99], Davies and Mitianoudis [DM04] proposed a simplified two-state Gaussian mixtures model together with an Expectation-Maximization algorithm as a fast separation and denoising solution for the instantaneous overcomplete problem. Perhaps, a generalisation of this idea in the complex domain might give a viable solution for the overcomplete convolutive problem.

6.2.4 Real-time implementation

One of the concerns when proposing an algorithm or a framework is whether this algorithm can work online (*real-time implementation*). However, as the processing power of computers and DSP chips keeps increasing, it might be possible that very computationally expensive algorithms will be implemented in real-time in a couple of years.

Aside from this fact, a vital research point for all the proposed audio source separation algorithms is whether they can be implemented in real-time. The *stochastic gradient* algorithm, as proposed by Bell and Sejnowski [BS95], is an online version of the natural gradient algorithm for instantaneous mixtures, aiming to follow the gradient while getting new data. Parra and Spence [PS00a] proposed an online version of their non-stationarity frequency-domain algorithm for convolutive mixtures, by employing the stochastic gradient idea, with promising results. In addition, the

DUET algorithm was implemented real time with a lot of variations [RBR01, BZ03].

Another approach is to work in a mixed block-based and real-time approach (i.e. a block LMS-type structure), where some data are stored in a local buffer for processing and the result is output in blocks. The problem is to find an optimal size for this buffer, so that the source separation algorithm has enough data to accurately estimate the signals statistics and the real-time processor is able of handling the computational cost without audible interruption. The systems we have tested in this thesis work efficiently with around $\sim 9-10$ secs of input data, which is far from real-time implementation. Of course, we refer to real room recordings. If we assume that the room is not so echoic, then we might be able to use smaller windows and therefore less data will be needed for efficient separation. Optimising these systems for real-time operation would increase the number of applications for these algorithms. Perhaps, adapting the system to work in a stochastic gradient framework might enable real-time implementation.

6.2.5 Non-stationary mixing

Most of the systems proposed hitherto for audio source separation assume that the *mixing environment is stationary*, i.e. the sources do not move in the auditory scene. This might be valid in applications like source separation from concert recordings, however, in all other cases there is bound to be some kind of source movement. Hence, in order to approach the real audio source separation, one may have to come up with a solution for non-stationary mixing.

In section 4.7.2, we performed some preliminary research on how robust a source separation system can be to movement and got an idea of current systems' limitations. There is not so much work done on this field for source separation, however, there is considerable amount of research performed in array signal processing, in terms of tracking moving audio objects [HBE01]. Perhaps, one could borrow ideas and import techniques from array signal

processing to deal with the problem in the context of audio source separation.

A possible approach might be to assume that the sources change position after specific intervals. During these intervals, we can assume that the mixing is stationary and therefore we can apply the present techniques to perform separation. Again, we have to define the length of this interval. It should be long enough for the source separation to have enough data and short enough for the mixing environment to be considered stationary. This is a similar problem to the real-time implementation using blocks of data. Therefore, finding an optimal block length might solve both problems at once. In addition, a stochastic gradient approach might be able to adapt to non-stationary mixing, however, current experience shows that its adaptation is too slow.

Bibliography

[ACY96] S.-I. Amari, A. Cichocki, and H.H. Yang. A new learning algorithm for blind source separation. In *Advances in Neural Information Processing Systems 8*, pages 757–763. MIT Press, 1996.

[AHJ85] B. Ans, J. Hérault, and C. Jutten. Adaptive neural architectures: detection of primitives. In *Proc. of COGNITIVA'85*, pages 593–597, Paris, France, 1985.

[AMNS01] S. Araki, S. Makino, T. Nishikawa, and H. Saruwatari. Fundamental limitation of frequency domain blind source separation for convolutive mixture of speech. In *ICASSP*, pages 2737–2740, 2001.

[Att99] H. Attias. Independent factor analysis. *Neural Computation*, 11(4):803–851, 1999.

[Aud] MPEG-4 Structured Audio. http://sound.media.mit.edu/mpeg4/.

[BC94] G. J. Brown and M. Cooke. Computational auditory scene analysis. *Comput. Speech Lang.*, 8(4):297–336, 1994.

[BH00] E. Bingham and A. Hyvärinen. A fast fixed-point algorithm for independent component analysis of complex-valued signals. *Int. J. of Neural Systems*, 10(1):1–8, 2000.

[Bre99] A.S. Bregman. *Auditory Scene Analysis: The perceptual organisation of sound*. MIT press, 2nd edition, 1999.

[Bro] M. Brookes. Voicebox - speech processing toolbox for MATLAB, available from http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html.

[BS95] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[BS98] R. Brennan and T. Schneider. A flexible filterbank structure for extensive signal manipulations in digital hearing aids. In *Proc. Int. Symposium on Circuits and Systems*, Monterey, California, 1998.

[BZ03] M. Baeck and U. Zölzer. Real-time implementation of a source separation algorithm. In *Proc. Digital Audio Effects (DAFx)*, London, UK, 2003.

[Car90] J.-F. Cardoso. Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'90)*, pages 2655–2658, Albuquerque, New Mexico, 1990.

[Car97] J.-F. Cardoso. Infomax and maximum likelihood for source separation. *IEEE Letters on Signal Processing*, 4:112–114, 1997.

[Car98a] J.-F. Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 9(10):2009–2025, 1998.

[Car98b] J.-F. Cardoso. Multidimensional independent component analysis. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'98)*, Seattle, WA, 1998.

[CC01] L.-W. Chan and S.-M. Cha. Selection of independent factor model in finance. In *Proc. Int. Workshop on Independent Com-*

ponent Analysis and Blind Signal Separation (ICA2001), San Diego, California, 2001.

[Chr92] K.B. Christensen. The application of digital signal processing to large-scale simulation of room acoustics: Frequency response modelling and optimization software for a multichannel DSP engine. *Audio Engineering Society*, 40:260–276, 1992.

[CLY96] C.W. Che, Q. Lin, and D.S. Yuk. An HMM approach to text-prompted speaker verification. In *Proc. of ICASSP*, pages 673–676, Atlanta, Georgia, 1996.

[Com94] P. Comon. Independent component analysis—a new concept? *Signal Processing*, 36:287–314, 1994.

[CS93] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEE Proceedings-F*, 140(6):362–370, 1993.

[CT91] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[CUMR94] A. Cichocki, R. Unbehauen, L. Moszczynski, and E. Rummert. A new on-line adaptive algorithm for blind separation of source signals. In *Proc. Int. Symposium on Artificial Neural Networks ISANN-94*, pages 406–411, Tainan, Taiwan, 1994.

[Dav00] M. Davies. Audio source separation. *Mathematics in Signal Processing*, V, 2000.

[DM04] M. Davies and N. Mitianoudis. A simple mixture model for sparse overcomplete ICA. *IEE proceedings in Vision, Image and Signal Processing*, 151(1):35–43, 2004.

[DS03] L. Daudet and M. Sandler. MDCT analysis of sinusoids and applications to coding artifacts reduction. In *Proc. of the AES 114th convention*, Amsterdam, 2003.

[EK03] J. Eriksson and V. Koivunen. Identifiability and separability of linear ICA models revisited. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 23–27, Nara, Japan, 2003.

[Ell96] D.P.W. Ellis. *Prediction-driven computational auditory scene analysis*. PhD thesis, M.I.T., 1996.

[Ero01] A. Eronen. Comparison of features for musical instrument recognition. In *Proc. Int. Workshop on Applications of Signal Processing on Audio and Acoustics*, New Paltz, New York, 2001.

[FMF92] K. Farrell, R. J. Mammone, and J. L. Flanagan. Beamforming microphone arrays for speech enhancement. In *Proc. ICASSP'92*, pages 285 – 288, San Francisco, CA, 1992.

[GFM01] B.W. Gillespie, D.A.F. Floréncio, and H.S. Malvar. Speech dereverberation via maximum-kurtosis subband adaptive filtering. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3701–3704, Salt Lake City, UTAH, 2001.

[GJ82] L.J. Griffiths and C.W. Jim. An alternative approach to linearly constrained adaptive beamforming. *IEEE Trans. on Antennas and propagation*, 30:27–34, 1982.

[GRC98] S.J. Godsill, P.J.W. Rayner, and O. Cappe. Digital audio restoration. *Applications of Digital Signal Processing to Audio and Acoustics*, pages 133–193, 1998.

[GV92] A. Gilloire and M. Vetterli. Adaptive filtering in subbands with applications to acoustic echo cancellation. *IEEE Trans. Signal Processing*, 40(8):1862–1875, 1992.

[HABS00] P. Herrera, X. Amatriain, E. Batlle, and X. Serra. Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In *Proc. of the ISMIR*, Plymouth, Massachusetts, 2000.

[Hay96] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 3rd edition, 1996.

[HBE01] Y. Huang, J. Benesty, and G.W. Elko. An efficient linear-correction least-squares approach to source localization. In *Proc. Int. Workshop on Applications of Signal Processing on Audio and Acoustics*, New Paltz, New York, 2001.

[HCB95] J.H. Husøy, J.E. Christiansen, and F. Barstad. Adaptive filtering in subbands: A tutorial overview. In *Proc. Norwegian Signal Processing Symposium NORSIG-95*, Stavanger, Norway, 1995.

[HCO99] A. Hyvärinen, R. Cristescu, and E. Oja. A fast algorithm for estimating overcomplete ICA bases for image windows. In *Proc. Int. Joint Conf. on Neural Networks*, pages 894–899, Washington, D.C., 1999.

[HH00] A. Hyvärinen and P. O. Hoyer. TopographicICA as a model of V1 receptive fields and topography. In *Proc. Int. Conf. on Neural Information Processing (ICONIP'00)*, Taejon, Korea, 2000.

[HK03] X. Hu and H. Kobatake. Blind source separation using ica and beamforming. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 597–602, Nara, Japan, 2003.

[HKO01] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley, New York, 2001.

[HO97] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[Hyv98] A. Hyvärinen. Independent Component Analysis in the presence of Gaussian noise by maximizing joint likelihood. *Neurocomputing*, 22:49–67, 1998.

[Hyv99a] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 10(3):626–634, 1999.

[Hyv99b] A. Hyvärinen. Fast Independent Component Analysis with noisy data using Gaussian Moments. In *Proc. Int. Symp. on Circuits and Systems*, pages V57–V61, Orlando, Florida, 1999.

[Hyv99c] A. Hyvärinen. The fixed-point algorithm and maximum likelihood estimation for independent component analysis. *Neural Processing Letters*, 10(1):1–5, 1999.

[Hyv99d] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[IM99] S. Ikeda and N. Murata. A method of ICA in time-frequency domain. In *Proc. Int. Workshop on ICA and Signal Separation*, pages 365–370, Aussois, France, 1999.

[IM00] M.Z. Ikram and D.R. Morgan. Exploring permutation inconsistency in blind separation of signals in a reverberant environment. In *ICASSP'00*, pages 1041–1044, 2000.

[IM02] M.Z. Ikram and D.R. Morgan. A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation. In *ICASSP*, pages 881–884, 2002.

[JPH93] J.R. Deller Jr, J.G. Proakis, and J.H.L. Hansen. *Discrete-Time Processing of Speech Signals*. MacMillan, 3rd edition, 1993.

[JRY00] A. Jourjine, S. Rickard, and O. Yilmaz. Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures. In *Proc. ICASSP'00*, pages 2985–2988, Istanbul, Turkey, 2000.

[KC01] B. Kostek and A. Czyewski. Representing instrument sounds for their automatic classification. *Audio Engineering Society*, 49, 2001.

[KZN03] K. Kokkinakis, V. Zarzoso, and A.K. Nandi. Blind separation of acoustic mixtures based on linear prediction analysis. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 343–348, Nara, Japan, 2003.

[Lam96] R. H. Lambert. *Multichannel Blind Deconvolution: FIR Matrix Algebra and Separation of Multipath Mixtures*. PhD thesis, Univ. of Southern California, 1996.

[LBL97] T.-W. Lee, A. J. Bell, and R. Lambert. Blind separation of delayed and convolved sources. In *Advances in Neural Information Processing Systems*, volume 9, pages 758–764. MIT Press, 1997.

[Lee98] T.-W. Lee. *Independent Component Analysis - Theory and Applications*. Kluwer, 1998.

[LH99] L. Liu and J. He. On the use of orthogonal GMM in speaker recognition. In *ICASSP'99*, pages 845–848, 1999.

[LLYG03] W.C. Lee, C.M. Liu, C.H. Yang, and J.I. Guo. Fast perceptual convolution for room reverberation. In *Proc. Digital Audio Effects (DAFx)*, London, UK, 2003.

[LS98] M. Lewicki and T. J. Sejnowski. Learing overcomplete representations. In *Advances in Neural Information Processing Systems 10*, pages 556–562. MIT Press, 1998.

[Mac02] D.J.C. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. Technical report, Draft paper available from *http://www.inference.phy.cam.ac.uk/mackay/*, 2002.

[Mar99] K.D. Martin. *Sound-Source Recognition: A Theory and Computational Model*. PhD thesis, M.I.T., 1999.

[MBJS96] S. Makeig, A. J. Bell, T.-P. Jung, and T. Sejnowski. Independent component analysis of electroencephalographic data. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, Cambridge MA, 1996. MIT Press.

[MD03] N. Mitianoudis and M. Davies. Audio source separation of convolutive mixtures. *Trans. Audio and Speech Processing*, 11(5):489 –497, 2003.

[Mit98] S. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, 1998.

[Mit00] N. Mitianoudis. *A graphical framework for the evaluation of speaker verification systems*. Imperial College, MSc thesis, 2000.

[MS00] T.K. Moon and W.C. Stirling. *Mathematical Methods and algorithms for signal processing*. Prentice Hall, Upper Saddle River, N.J., 2000.

[NGTC01] S.E. Nordholm, N. Grbic, X.J. Tao, and I. Clesson. Blind signal separation using overcomplete subband representation. *IEEE Trans. Audio and Speech Processing*, 9(5):524–533, 2001.

[OS89] A. Oppenheim and R. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, 1989.

[PA02] L. Parra and C. Alvino. Geometric source separation: Merging convolutive source separation with geometric beamforming. *IEEE Transactions on Speech and Audio Processing*, 10(6):352–362, 2002.

[PBJ00] S. Pankanti, R.M. Bolle, and A. Jain. Biometrics: The future of identification. *IEEE Computer*, pages 46–49, 2000.

[PP97] B. A. Pearlmutter and L. C. Parra. Maximum likelihood blind source separation: A context-sensitive generalization of ICA. In *Advances in Neural Information Processing Systems*, volume 9, pages 613–619, 1997.

[PS00a] L. Parra and C. Spence. On-line convolutive source separation of non-stationary signals. *J. of VLSI Signal Processing*, 26(1-2), August 2000.

[PS00b] L. Parra and C. Spence. Convolutive blind separation of non-stationary sources. *IEEE Trans. on Speech and Audio Processing*, 8(3):320–327, March 2000.

[PZ03] F. Pachet and A. Zils. Evolving automatically high-level music descriptors from acoustic signals. Submitted for publication to IJC, Sony CSL, 2003.

[RBR01] S. Rickard, R. Balan, and J. Rosca. Real-time time-frequency based blind source separation. In *Proc. ICA2003*, San Diego, CA, 2001.

[RJ93] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[RMS01] J. Reiss, N. Mitianoudis, and M. Sandler. Computation of generalized mutual information from multichannel audio data. In *110th Audio Engineering Society International Conference*, Amsterdam, The Netherlands, 2001.

[RR95] D.A. Reynolds and R.C. Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Trans. on Speech and Audio Processing*, 3(1), 1995.

[RRE03] M. Reyes, B. Raj, and D. Ellis. Multi-channel source separation by factorial HMMs. In *Proc. ICASSP*, Hong Kong, 2003.

[Sch86] R.O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Trans. on Antennas and propagation*, AP-34:276–280, 1986.

[SD01] X. Sun and S. Douglas. A natural gradient convolutive blind source separation algorithm for speech mixtures. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001)*, San Diego, California, 2001.

[SKS01] H. Saruwatari, T. Kawamura, and K. Shikano. Fast-convergence algorithm for ICA-based blind source separation using array signal processing. In *Proc. Int. IEEE Workshop on Application of signal processing to audio and acoustics*, pages 91–94, New Paltz, New York, 2001.

[Sma97] P. Smaragdis. *Information theoretic approaches to Source Separation.* Media Lab, M.I.T., Masters thesis, 1997.

[Sma98] P. Smaragdis. Blind separation of convolved mixtures in the frequency domain. *Neurocomputing*, 22:21–34, 1998.

[Sma01] P. Smaragdis. *Redundancy reduction for computational audition, a unifying approach.* PhD thesis, M.I.T., 2001.

[STS99] D. Schobben, K. Torkolla, and P. Smaragdis. Evaluation of blind signal separation methods. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA1999)*, Aussois, France, 1999.

[SZ97] S. Sarma and V. Zue. A segment-based speaker verification system using SUMMIT. In *Proc. Eurospeech 97*, Rhodes, Greece, 1997.

[TC00] G. Tzanetakis and P. Cook. Audio information retrieval (AIR) tools. In *Proc. Int. Symposium On Music Information Retrieval*, Plymouth, Massachusetts, 2000.

[Tor96] K. Torkkola. Blind separation of convolved sources based on information maximization. In *Proc. IEEE Workshop on Neural Networks and Signal Processing (NNSP'96)*, pages 423–432, Kyoto, Japan, 1996.

[vdKWB01] A.J.W. van der Kouwe, D.L. Wang, and G.J. Brown. A comparison of auditory and blind separation techniques for speech segregation. *IEEE Transactions on Speech and Audio Processing*, 9:189–195, 2001.

[VE03] H. Viste and G. Evangelista. On the use of spatial cues to improve binaural source separation. In *Proc. Digital Audio Effects (DAFx)*, London, UK, 2003.

[VK96] M. Viberg and H. Krim. Two decades of array signal processing - the parametric approach. *IEEE Signal Processing magazine*, 13(4):67–94, 1996.

[vVB88] B.D. van Veen and K.M. Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Magazine*, 5:4–24, 1988.

[Wes] A. Westner. http://www.media.mit.edu/∼ westner.

[WJSC03] W. Wang, M.G. Jafari, S. Sanei, and J.A. Chambers. Blind separation of convolutive mixtures of cyclostationary sources using an extended natural gradient. In *Proc. Int. Symposium on Signal Processing and its applications*, Paris, France, 2003.

[WP02] S. Weiss and I. Proudler. Comparing efficient broadband beamforming architectures and their performance trade-offs. In *DSP conference*, 2002.

[ZKZP02] M. Zibulevsky, P. Kisilev, Y.Y. Zeevi, and B.A. Pearlmutter. Blind source separation via multinode sparse representation. *Advances in Neural Information Processing Systems*, 14:1049–1056, 2002.