Information Theoretic Approaches to Source Separation

Paris J. Smaragdis

Bachelor of Music
Berklee College of Music, Boston 1995

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Technology
at the Massachusetts Institute of Technology

June 1997

Author
Paris J. Smaragdis
Program in Media Arts and Sciences
May 8, 1997

Certified by
Barry L. Vercoe
Professor, Program in Media Arts and Sciences
Thesis Supervisor

Accepted by
Stephen A. Benton
Chair, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Information Theoretic Approaches to Source Separation

by

Paris J. Smaragdis

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on May 8, 1997
in partial fulfillment of the requirements for the degree of
Master of Science in Media Technology

Abstract

The problem of extracting sources from a mix of these has been extensively addressed by a lot of research. In this thesis the problem is being considered in the sonic domain using algorithms that employ information theoretic techniques. Basic source separation algorithms that deal with instantaneous mixtures are introduced and are enhanced to deal with the real-world problem of convolved mixtures. The proposed approach in this thesis is an adaptive algorithm that operates in the frequency domain in order to improve efficiency and convergence behavior.

In addition to developing a new algorithm there is also an effort to point important similarities between human perception and information theoretic computing approaches. Source separation algorithms developed by the auditory perception community are shown to have strong connections with the basic principles of the strictly engineering algorithms that are introduced, and proposals for future extensions are made.

Thesis Supervisor:
Barry Vercoe
Professor, Program in Media Arts and Sciences

Information Theoretic Approaches to Source Separation

Paris J. Smaragdis

Thesis Reader

V. Michael Bove
Associate Professor
Program in Media Arts and Sciences

Thesis Reader

Kari Torkkola
Principal Staff Scientist
Motorola, Inc., Phoenix Corporate Research Laboratories

Acknowledgments

The acknowledgments page in this thesis is probably the lowest stress part I have to write, given that it will not be judged or edited by my wonderful thesis readers or my excellent advisor. Their invaluable comments and support during the making of this thesis are deeply appreciated. In particular I must thank Kari Torkkola for his influential papers and helpful comments; Mike Bove for sharing his expertise on everything I asked him; and Barry Vercoe for being the coolest advisor I can think of.

And if the acknowledgment page writing is relaxed, it fades in comparison to the procrastination part that escorts the writing of a thesis; courtesy of the miraculously idle and lounging group of κωλοβαρες@mit and the Greek campus mafia, especially their principal components: AA (my dear racing car co-driver), little ‘o’, the Μουκηδες for their most enjoyable meals (thanks Nat!) and company and the presidential couple (yes you two!) for the fun and the rides!

However amidst of all this procrastination, I did have to write a thesis, something that wouldn’t happen without the great support of my group. Individual thanks go to Mike for being a very cool officemate who engages in wonderful and deep discussions; Bill for withstanding my frequent bombardment of questions and continuously pointing me to the right way; Keith for enlightening my DSP and C++ world; Eric, the master of written and spoken word (in addition to music cognition) who proofread my thesis and helped clean up my Greek-grammar-oriented English dialect; DAn the man who must be having nightmares of me asking him questions all the time (thanks DAn!!); Adam for helping me stay sane and feel better; and Jonathan for making me feel more of a senior member! (as well as being a great “You don’t know Jack” competitor!!) I can only hope to be as helpful to them too. Of course it would be a great omission to forget my ace mathematical advisor and dear friend, Elias and my UROP, Petros for keeping me on my toes. I must also thank Johann-Sebastian for keeping me company during these looong coding/debugging/thinking/deriving/hair-pulling night sessions ...

And last (but in no way least), my deepest thanks to Richard Boulanger, without whom I’d be probably be carrying cables right now in a music studio (in some Greek army camp!); and my parents for their bold support and trust on me, without which I’d be a miserable unemployed high-school physics teacher who always wanted to be a musician!

Anti-acknowledgments

I would not like to thank, the Boston weather, without which we could have an ordinary spring. I do not want to thank the IRS for making sure that tax forms are due around the same time my thesis and conference papers are! And I really don’t want to thank my physics teacher at high-school for giving me a bad term grade since I didn’t know what entropy was (I know now, really I do ... Appendix A!).

Table of Contents

Chapter 1. Introduction and Overview 15

1 . 1 Introduction 15

1 . 1 . 1 Cocktail Parties and Passing Cars 15

1 . 1 . 2 The Source of the Problem 15

1 . 1 . 3 Traditional Approaches 16

1 . 1 . 4 ... and the State of the Art 16

1 . 1 . 5 Why the Information Theoretic Approach? 16

1 . 2 Thesis Overview 17

Chapter 2. Background. 19

2 . 1 Overview 19

2 . 2 Classification of Separation Algorithms 19

2 . 3 Single Input Approaches 20

2 . 4 Multi Input Approaches 21

2 . 4 . 1 Early Approaches 23

a. Pierre Comon, ICA Approach 23

b. Herault & Jutten, Neuromimetic Approach 23

2 . 4 . 2 Modern Extensions 25

a. Anthony Bell, Information Maximization 25

b. Shun-ichi Amari, Natural Gradient 27

2 . 5 Choosing an Approach 29

2 . 6 Conclusions 30

Chapter 3. Convolved Mixtures 31

3 . 1 Overview 31

3 . 2 Convolved Mixtures 31

3 . 3 The Feedforward Network Approach 35

3 . 4 The Feedback Network Approach 37

3 . 5 Conclusions 38

Chapter 4. Frequency Domain Extensions. 39

4 . 1 Overview 39

4 . 2 Why Move to Another Domain? 39

4 . 2 . 1 Efficiency Problems 40

4 . 2 . 2 Convergence Considerations 40

4 . 3 Planning the Algorithm 41

4 . 3 . 1 Moving to the Frequency Domain 41

4 . 4 Neural Networks in the Complex Domain 42

4 . 4 . 1 Complex Units 42

4 . 4 . 2 The Activation Function 43

4 . 4 . 3 The Learning Functions 45

4 . 5 The Frequency Domain Algorithm 45

4 . 5 . 1 Implementation Specifics 47

a. The Frequency Transform 47

b. The CANN Bank 47

c. The Time Transform 48

4 . 6 Improvements over Time Domain Algorithms 48

4 . 6 . 1 Efficiency 48

4 . 6 . 2 Convergence Properties 49

4 . 7 Conclusions 49

Chapter 5. Conclusions and Future Work 51

5 . 1 Overview 51

5 . 2 Performance Results 51

5 . 2 . 1 Instantaneous mixtures 52

5 . 2 . 2 Delayed Mixtures 53

5 . 2 . 3 Convolved mixtures 55

5 . 3 Future Work 58

5 . 3 . 1 Short term 58

5 . 3 . 2 Long term 58

5 . 4 Conclusions 59

Appendix A. Information Theory Basics 61

A . 1 Introduction 61

A . 2 Definition of Entropy 61

A . 3 Joint and Conditional Entropy 63

A . 4 Kullback-Leibler Entropy 63

A . 5 Mutual Information 64

Appendix B. Derivation of Bell's Rule 67

B . 1 Overview 67

B . 2 1 by 1 Case 67

B . 3 N by N Case 69

Appendix C. Derivation of Amari's Rule 71

C . 1 Overview 71

C . 2 Learning Rule Derivation 71

C . 3 Natural Gradient 73

Appendix D. Derivation for Convolved Mixture Time Domain Algorithms 75

D . 1 Overview 75

D . 2 Feedforward Architecture 75

D . 3 Feedback Architecture 77

Bibliography. 79

List Of Figures

Figure 1 Grouping in the frequency domain 20

Figure 2 The Herault-Jutten Network 24

Figure 3 Bell's Network 26

Figure 4 'Flattening' the input PDFs 28

Figure 5 A case of convolved mixtures 32

Figure 6 The feedforward solution for convolved mixtures 35

Figure 7 The feedback solution for convolved mixtures 37

Figure 8 A complex neuron 42

Figure 9 The tanh function in the complex domain 43

Figure 10 A proper complex activation function 44

Figure 11 Flow graph of the algorithm 46

Figure 12 Comparison between time and frequency domain algorithms 47

Figure 13 Instantaneous unmixing results in the time domain 52

Figure 14 Instantaneous mixing results in the frequency domain 53

Figure 15 Delayed mixing results in the time domain 54

Figure 16 Delayed mixing results in the frequency domain 55

Figure 17 Cauchy noise 56

Figure 18 Convolved mixing results in the time domain 56

Figure 19 Convolved mixing results in the frequency domain 57

Figure 20 Convolved mixing results in the frequency domain using parameter fine tuning . 57

Chapter 1. Introduction and Overview

1.1 Introduction

1.1.1 Cocktail Parties and Passing Cars

The *cocktail party problem* is a classic example demonstrating the need for robust audio separation algorithms. Imagine being in a loud party, trying to talk with a friend. During your conversation the sounds that reach your ears are a complicated mix of music, other people talking, glasses tinkling and so on. Even though all of these sounds arrive at the ear as a single waveform, you are able to understand your friend and still have time to enjoy the music and keep alert of your surroundings by audibly checking the scene. Machines, on the other hand, get confused when even a distant and faint sound such as a passing car goes by as you talk to their microphones.

1.1.2 The Source of the Problem

An unfortunate fact in the field of machine listening is that people tend to think in terms of isolated sounds. This has led most audio related research to focus on the problem of analyzing single sounds. There are myriad algorithms that can extract valuable information from audio and an enormous bibliography on isolated sound analysis. However, even though these algorithms perform very well using single sound input, in the real world there is bound to be external interference. As a result, robust 'isolated sound' applications such as speech recognizers cannot operate in a noisy street, pitch trackers give up when presented with polyphonic music, etc.

1.1.3 Traditional Approaches ...

Given the number of existing algorithms that operate on isolated sounds and the algorithmic complexity required to deal with sounds in a human-like way, a need for algorithms that separate mixed sounds arises. The problem is simple to state: given a mix (or several mixes) of sounds how can we isolate the parts that we are interested in? As the solution of such a problem could boost audio related applications to a new era, much attention has been given to this problem of 'source separation'. According to Bodden (1993) there are two classes of algorithms, *single-input* and *multi-input.* Most of the single-input algorithms have been the product of auditory perception research and are mostly based on frequency domain grouping principles. Multi-input algorithms are mainly developed by electronic engineers for applications such as radar and sonar scene analysis as well as in communications

1.1.4 ... and the State of the Art

Even though there has been an extraordinary amount of work done in this field using the above techniques, it is obvious that there are still problems to consider. Single channel methods are usually non-causal, introduce audible artifacts in the extracted sounds and seriously damage their quality. Multi-channel algorithms are computationally very intensive, work under constrained settings and often perform poorly. However a new family of algorithms has been recently introduced (Bell and Sejnowski 1995, Amari *et al.* 1996) which perform with considerable success. These algorithms are multi-channel but instead of using the traditional correlation methods they are based on information theoretic formulations. Instead of using heuristics, as the frequency grouping algorithms do, or limiting the algorithms by using low order statistics, as most multi-input algorithms do; this approach gets around these pitfalls by employing information theory.

1.1.5 Why the Information Theoretic Approach?

Since there are so many different viewpoints and different classes of algorithms to attack this problem why choose this approach? There are two main reasons, one lying purely in the algorithmic domain and the other in the perceptual sciences.

First and foremost, the performance of this approach is exceptionally good. Unlike other approaches the solution is well put and mathematically justified. There is an optimal solution and minimal use of assumptions and approximations. In addition to these arguments, implementations of this approach can be significantly faster than their counterparts.

An additional, attractive feature of this approach is the link to theories of perception. During the mid-50's new theories relating information theory to perception appeared (Attneave 1954, Barlow 1959, 1961) and recently there has been development of new algorithms that support them (Linsker 1988, Atick and Redlich 1990, Redlich 1993). The main idea behind this work is that by maximization of sensory information it is possible to make systems that behave very much like we do; by self-organization through observation. These ideas have been applied to the modalities of vision (Bell 1996) and olfaction (Hopfield 1991) but not to the field of auditory perception. The results from vision research are very encouraging since it has been shown that a lot of knowledge normally provided to vision systems (e.g. edge filters) can be developed within the system itself. It is very possible that developing such information theoretic ideas even more could result in a more general and accurate theory on auditory perception.

1.2 Thesis Overview

This thesis is split in three main parts. The first is the background chapter that covers the mathematical definition of the source separation problem and presents some of the early approaches to solving this problem as well as the fundamental structures and ideas used throughout this thesis. Even though we will be mostly occupied with multi-input algorithms, a brief introduction to single-input approaches is also given and interpreted with respect to the same information theoretic principles that we use for multi-input approaches.

In the following chapter we extend the coverage of source separation algorithms to convolved mixtures. The pure definition of source separation, as shown in the first part, implies instantaneous mixtures and this is proven to be inadequate for attempting to separate sounds recorded in a real environment. The existence of sound propagation delays, different sensor responses and room imposed transfer functions, causes filters to transform every source before the mixing procedure, rendering the instantaneous unmixing procedure useless. In order to deal with this problem in a way so as not to change the already existing notation and theory, the idea of the FIR linear algebra is introduced. Using the FIR linear algebra notation we can use the same mixing and unmixing formulas as in the instantaneous mixing case and avoid cumbersome notation and derivations.

In the third and final part, shortcomings of the previous approaches are revealed and an alternative is proposed. It is shown that by operating in the frequency domain rather than the time domain can be much more efficient and easier in terms of convergence. Based on properties of the FIR linear algebra it is easy to transfer the already existing time domain rules in the frequency domain and formulate a robust algorithm.

Chapter 2. Background

2.1 Overview

This chapter presents the problem of source separation as seen by different viewpoints and some of the proposed solutions so far. The mathematical groundwork needed for further work is also laid out, and linked to popular heuristic approaches that have been proposed.

2.2 Classification of Separation Algorithms

The problem of source separation in the auditory domain has received a lot of attention and numerous approaches have been developed. In general these approaches are divided into two parts, single input and multi input algorithms. For reasons explained further on, this thesis will be primarily occupied with multi input approaches. However a short introduction of single input methods will be presented since later on it will be shown that the same underlying principles apply to both approaches.

2.3 Single Input Approaches

Single input algorithms have been the older approaches to source separation. This problem is often encountered in engineering, especially in radar and sonar theory, as well as in the auditory perception community. Due to resource constrains or philosophy reasons these approaches were developed using a single input. Particularly in the auditory domain these approaches are preferred since they are closer to the way we listen than multi input algorithms. Most of the single input algorithms attempt to separate sounds in the frequency domain. By using heuristic grouping rules, frequency components that seem to belong to the same source are extracted and then used to resynthesize the outputs. These grouping rules are looking to group frequency bins with common onsets, common amplitude and frequency modulations, harmonic ratios, etc. (Figure 1)

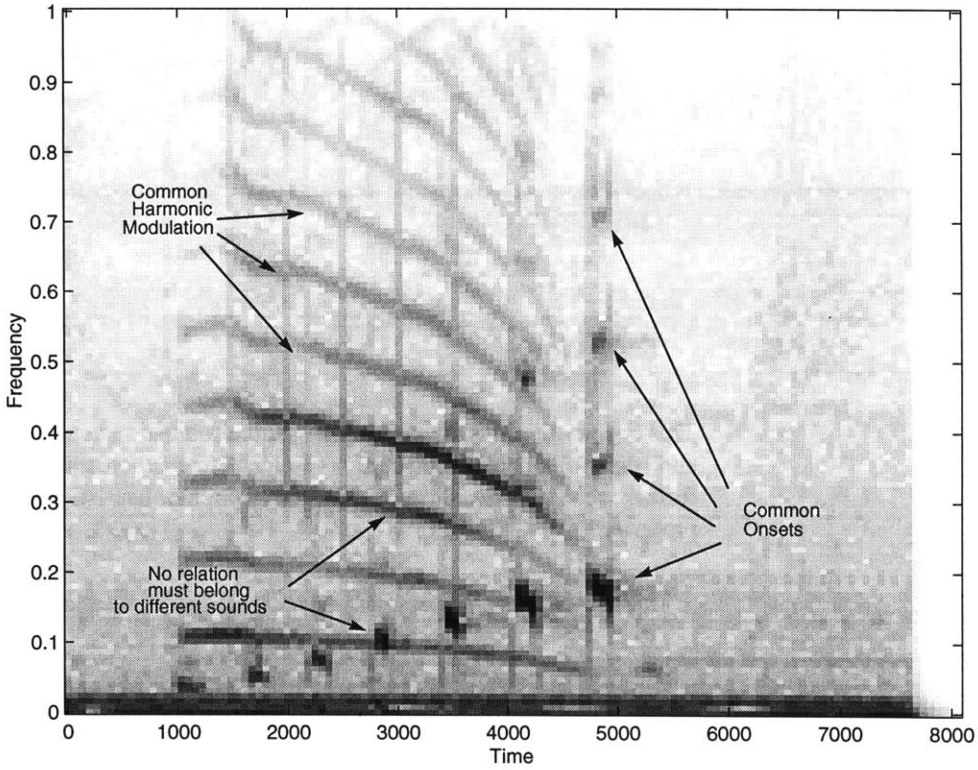

Figure 1 **Grouping in the frequency domain**

Early work on audio separation was done by Stockham (1975), who used homomorphic signal processing to separate the singer Caruso's voice from background noise and accompaniment in recordings made in 1908; Parsons (1976), who performed FFT analysis and grouped harmonically related frequency bins to obtain the spectra of the original sounds; Weintraub (1985), who used autocorrelation to group channels from cochlear filterbanks; Nakatami *et al.* (1994), who used different grouping strategies on frequency representations; and finally Cooke and Brown (Cooke, 1991, Brown, 1992) and Ellis (1992), who used gestalt grouping principles for picking frequency components that seem to belong to the same sound.

Unfortunately the single input approaches come with inherent problems. Tampering with the frequency domain representation of a sound and then going back to the time domain is often a highly degrading process. In addition, by using heuristic rules a lot of data is left ungrouped and discarded before the time transformation. These factors usually contribute to bad-sounding resyntheses and prohibit such algorithms from being used in high quality implementations.

2.4 Multi Input Approaches

Multi input separation techniques pose a considerably different problem from single input techniques. Separation is attempted upon presentation of a set of mixtures instead of a single mixture. Such a case is when we have a number of microphones in a room. Each microphone will record a different mix of the sources and by cross-cancellation it is possible to suppress unwanted sources to inaudible levels. An additional difference in these techniques is that we have no prior knowledge about the nature of the sources. Because of this, the problem is also referred to as *blind source separation*. The formal definition and the notation used is as follows.

Assume that we have N sources, s_i, which transmit signals that after propagation in an arbitrary medium are measured by M sensors. The signals that are measured by these sensors will be called x_i. The mapping from s_i to x_i is an unknown function f_i so that:

$$x_i = f_i(s_1, \ldots, s_N) \qquad (1)$$

In acoustics this function is usually a linear superimposition, so using linear algebra notation we can rewrite the above equation in a more elegant form as:

$$\mathbf{x}(t) = \mathbf{A} \cdot \mathbf{s}(t) \qquad (2)$$

where:

$$\mathbf{x}^T(t) = \left[x_1(t) \ \ldots \ x_M(t)\right], \qquad (3)$$

$$\mathbf{s}^T(t) = \left[s_1(t) \; \dots \; s_N(t)\right] \tag{4}$$

and $\mathbf{A} \in \Re^{M \times N}$ is an unknown invertible matrix which we will call the mixing matrix[†].

The task is to recover the s_i signals given only the $\mathbf{x}_i(t)$ vectors (which are our observables). Even though this seems like an ill-defined problem it is possible to obtain a solution given only two constraints. We cannot recover the s_i signals in the same order they came in (meaning that the resulting signals will have their subscript index permuted in relation to the inputs) and we cannot get the output signals in their original amplitude. Neither of these problems are serious with acoustical data since we care for the separated sounds no matter what the order is and since we can easily scale the data to fit our needs.

In all of the approaches to be introduced the objective is to find the inverse (or pseudo-inverse) of the mixing matrix **A**. Once $\mathbf{A}^{-1}$ is computed, it is easy to see that by:

$$\mathbf{s}(t) = \mathbf{A}^{-1} \cdot \mathbf{x}(t) \tag{5}$$

we can recover the original signals. As mentioned above we are not able to estimate the exact $\mathbf{A}^{-1}$ but rather the matrix $\mathbf{W} \in \Re^{N \times M}$:

$$\mathbf{W} = \mathbf{P} \cdot \mathbf{A}^{-1} \tag{6}$$

where $\mathbf{P} \in \Re^{M \times M}$ is a scaling and permutation matrix with one non-zero element in every row and column.

The way that the inverse mixing matrix is usually estimated is by assuming that the input sources are mutually independent (more intuitively stated as: a signal s_i will not be influenced by another signal s_j). This might seem like an aggressive assumption but in fact, real world signals from different sources do not have statistical dependencies between them. This also explains why the limitation expressed in Equation (6) exists, because statistical dependence is independent of permutation and scaling.

The unmixing equation that will give us the clean outputs is:

$$\mathbf{u}(t) = \mathbf{W} \cdot \mathbf{x}(t) \tag{7}$$

[†] The assumption that this matrix is invertible needs to be made in order to be able to recover the inputs. In general non-invertible mixing matrices exist if all sources originate from the same position as well as all the sensors used to observe them. Since it is impossible to have microphones and speakers at the same physical location, this assumption is fair.

The outputs will be notated as u_i. For on-line algorithms they will not be equal to the inputs until convergence is achieved. Once $\mathbf{W} \approx \mathbf{P} \cdot \mathbf{A}^{-1}$ then we would have $\mathbf{u} \approx \mathbf{s}$. The notation used in this section will be adopted for the remaining of the chapter.

2.4.1 Early Approaches

In this section we examine two of the original algorithms that were proposed for solving this problem. They provide a good introduction to the state of the art algorithms and briefly introduce the concept of *Independent Components Analysis* (ICA) which is very closely related to blind source separation.

a. Pierre Comon, ICA Approach

Pierre Comon (1989) was the first to introduce the notion of ICA in a rigorous and mathematical fashion. Comon's work was not specifically on source separation but on finding domain decompositions that yield basis sets in which bases are statistically maximally independent. This can be seen as an extension to the well known Principal Components Analysis (PCA) where the bases are independent up to the second order statistics. In addition to that, ICA also ensures independence of higher order statistics.

The relation to source separation comes from the fact that the observations we have are linear combinations of some statistically independent signals. If we consider these signals as the set of bases that construct the observations, we can use an ICA algorithm to estimate them.

Comon's approach uses PCA first, to achieve independence up to second order statistics. Afterwards a more involved algorithm is used for higher order independence. For computational reasons, Comon makes use of only the third and fourth order cumulants in this optimization scheme.

Comon's algorithm has good behavior but requires estimation of some higher order statistics which is computationally expensive. This work served as a springboard for further research which resulted in more efficient forms of ICA and source separation.

b. Herault & Jutten, Neuromimetic Approach

Herault and Jutten (1991) introduced a different approach to the blind separation problem using what they call a 'neuromimetic' approach. They implement an unmixing equation using the structure shown in Figure 2 for the 2 by 2 case:

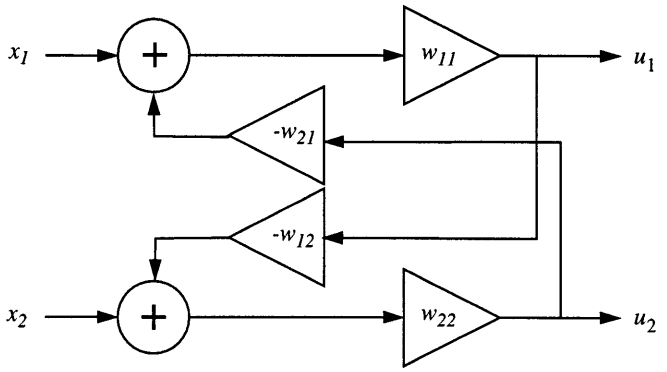

Figure 2 **The Herault-Jutten Network**

An interesting point here is that the Herault-Jutten network has the w_{ii} weights set to unity throughout training and performs an on-line estimation on the remaining weights. Because of the recurrent elements this algorithm cannot be described by the unmixing equation (Equation (7)). Instead the unmixing equation used here is[†]:

$$u_i(t) = x_i(t) - \sum_{j=1, j \neq i}^{N} w_{ij} \cdot s_j(t) \tag{8}$$

or by using the more elegant linear algebra form:

$$\mathbf{u}(t) = \mathbf{x}(t) - \mathbf{W} \cdot \mathbf{u}(t) \tag{9}$$

It should also be noted that this algorithm requires N inputs for N outputs; in other words the mixing matrix has to be square. In addition to this Herault and Jutten impose yet another restriction which requires the input signals to have zero mean.

In order to find the unmixing matrix Herault and Jutten used gradient descent with $u_i^2(t)$ as the cost function[‡]. By computing the gradient w.r.t. the weights we obtain the following learning rule:

[†] A fine point in this equation is that we assume zero delay connections. This is not possible and instead the creators of this algorithm rely on the fact that most sources change slowly in time so that $s_i(t) \approx s_i(t-1)$.

[‡] It is shown in their paper (Herault and Jutten 1991) that this quantity relates to statistical independence of the outputs

$$\Delta w_{ij} \propto f(s_i(t)) \cdot g(s_j(t)) \tag{10}$$

where $f(x)$ and $g(x)$ are two different, odd, non-linear functions constant throughout training and $j \neq i$.

Additional early work on this problem has been done by Cardoso (1996, 1993), Burel (1992) who employed multi-layer perceptrons, Matsuoka (1995) who used an algorithm similar to the Herault-Jutten network and Molgedey (1994) who used time correlations for the same structure.

2.4.2 Modern Extensions

Using principles laid out by Comon, and by Herault and Jutten, a new family of algorithms for the blind separation problem has appeared recently. These algorithms employ the information theory approach as pointed by Comon and fast computational structures inspired from Herault and Jutten. These algorithms have proven to be very efficient and accurate for use in real world situations.

The main drawback of the older algorithms is that they rely on the minimizations of higher order cross-cumulants or cross-moments. Unfortunately the estimation of these measures requires significant computation and in addition to this, there is an infinite number of them, making the problem intractable. Many algorithms bypass this by ensuring independence up to the fourth order statistics and assuming gaussian characteristics for the inputs[†]. However these are severe approximations that often result in poor performance.

a. Anthony Bell, Information Maximization

An alternative approach was introduced by Bell and Sejnowsky (1996) who used information as a cost function. By doing so they bypass the cumulant or moment estimations and produced an algorithm that efficiently optimized in all orders of statistics. Instead of cross-statistics the measure of mutual information was used. Mutual information is a measure that is described by all of the higher order cross-statistics so that optimizing it results in a complete cross-statistics optimization. The structure used is shown for the 2 by 2 case in Figure 3. This structure implements the unmixing equation (Equation (7)).

[†] Gaussian random variables have zero statistics from the fifth order and up

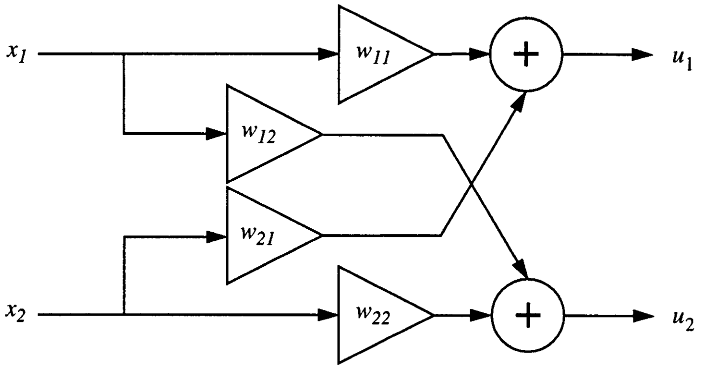

Figure 3 **Bell's Network**

The method that Bell describes, is the maximization of mutual information between the input and the output of the network. This is done by maximizing the information flow of every output. For the learning procedure Bell applies a monotonic sigmoid function to the outputs of the separation structure so that:

$$\mathbf{y}(t) = \tanh(\mathbf{W} \cdot \mathbf{x}(t)) \qquad (11)$$

By doing so we can express the probability density functions (PDFs) of **y** with respect to the input PDFs using the formula:

$$P_y(\mathbf{y}) = \frac{P_x(\mathbf{x})}{|\mathbf{J}|} \qquad (12)$$

where **J** is the Jacobian of the transformation (Equation (11)). Bell, assuming Gaussian-like inputs, shows that if we maximize the Jacobian we will have a maximally flat output distribution, and that results in information flow maximization. If we use the tanh(x) function for the sigmoid and assume zero mean inputs[†] the resulting learning rule is (see Appendix B for full derivation):

$$\Delta\mathbf{W} \propto [\mathbf{W}^T]^{-1} - 2 \cdot \tanh(\mathbf{u}) \cdot \mathbf{x}^T \qquad (13)$$

[†] The network in Figure 3 can be set up with additional unit inputs to the summation nodes. By doing so we can deal with biased inputs. However the additional mathematical complexity is avoided for simplicity reasons.

Bell used this algorithm successfully for separating up to ten sources. This was the first of a series of very successful algorithms.

b. Shun-ichi Amari, Natural Gradient

Amari *et al.* (1996) used a different approach which resulted in the same learning rule and then it was further extended by performing descent on the *natural gradient.* The derivation given by Amari uses the Kullback-Leibler distance (measure of PDF similarity) as the starting point[†]. Since we want our outputs to be independent we want the joint entropy of the outputs to be the same as the product of the individual entropies[‡]. One way to do so is to minimize the Kullback-Leibler distance between the PDF of the output vector and the product of the PDFs of the individual outputs with respect to the unmixing matrix. If we do so we ensure that the outputs are statistically independent. The formula for this distance is:

$$K(\mathbf{W}) = \int P(\mathbf{u}) \cdot \log \frac{P(\mathbf{u})}{\prod_{i=1}^{N} P(u_i)} d\mathbf{u} \tag{14}$$

Where $P(x)$ is the PDF of x. Using a Gram-Charlier cumulant expansion, the PDFs in the right hand side of the above equation can be approximated and the resulting learning rule from that equation is (see Appendix C for full derivation):

$$\Delta\mathbf{W} \propto [\mathbf{W}^T]^{-1} - f(\mathbf{u}) \cdot \mathbf{u}^T \tag{15}$$

where $f(x) = \frac{3}{4}x^{11} + \frac{25}{4}x^9 - \frac{14}{3}x^7 - \frac{47}{4}x^5 + \frac{29}{4}x^3$. It is easy to see that this learning rule is similar to the rule that Bell derived. This learning rule also works with other types of activation functions as well. Because the activation function that Amari derived is not bounded it is numerically safer to use the hyperbolic tangent instead.

An additional observation that Amari makes is that the space we are optimizing in is a Riemannian space. Because of this we can alter the learning rule to fit the space we are optimizing in (make use of the space's natural gradient). Cardoso and Laheld (1996) and Amari (1997) show that in this particular problem we can make up for this with a right multiplication (Appendix C) by $\mathbf{W}^T\mathbf{W}$, which gives:

$$\Delta\mathbf{W} \propto [\mathbf{I} - f(\mathbf{u}) \cdot \mathbf{u}^T] \cdot \mathbf{W} \tag{16}$$

[†] Refer to Appendix A for the mathematical definition of the Kullback-Leibler distance

[‡] A consequence of: If A and B are independent then $P(A, B) = P(A) \cdot P(B)$

By performing steepest descent using the natural gradient, convergence is significantly faster and more stable. In addition to good convergence behavior, there is also increased efficiency since the learning rule does not include a matrix inversion anymore.

Complicated as these two algorithms are, they can be seen from a more intuitive point. Each in their own way they are resulting in maximization of the entropies of the individual outputs. The basic assumption is that the inputs have Gaussian-like PDFs. If these input PDFs are properly aligned and passed through a sigmoid resembling their CDFs the output will have a uniform distribution (which is the PDF that has the maximum possible entropy). The sigmoid we use is fixed, but the unmixing weight can scale the input data so that the output has a flat PDF. A one-input one-output example is illustrated in Figure 4:

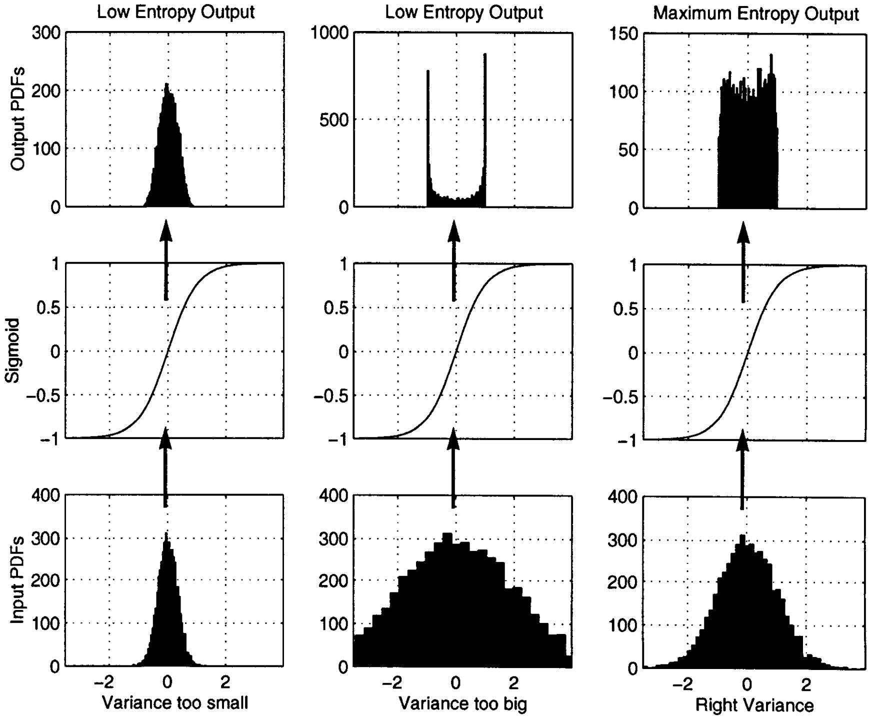

Figure 4 'Flattening' the input PDFs

In the left case the input has small variance. Once through the sigmoid the output is still a Gaussian-like PDF which has low entropy. In the middle graph we have the opposite case where a big variance results in a very 'peaky' PDF with low entropy. In the right case we have the ideal case where the input PDF is well aligned with the sigmoid and it produces a uniform PDF output. In this case the outputs have maximal entropies, thus independence.

Additional approaches have been developed which resulted in the same learning rules. The Euclidean form of the gradient (Equation (13)) was also be derived using maximum likelihood as shown by McKay (1996), who also extended the learning rule to the proper space. Cardoso (1997) proved that the information maximization principle is equivalent to maximum likelihood.

There have also been other approaches for multi-input separation which made use of phase-array detectors. In general these approaches haven't been as popular since they require a well defined sound field and microphone positioning, stationary sound sources or unpractically large microphone arrangements. In the aforementioned techniques the only restrictions are that we need as many, or more, microphones as sources and that for now we need them to be identical. An additional problem with some of the phase-array approaches is that they make use of only second order statistics which are not a strong criterion of independence.

2.5 Choosing an Approach

Even though single input approaches are very different from multi input approaches the underlying assumptions and principles are still the same. The grouping rules that are used for the single input frequency domain algorithms are based on the Gestalt principles, which are just heuristics for achieving minimum length descriptions. These minimum length description groups of frequency components, once combined, form a minimum joint entropy description. Minimum joint entropy implies statistical independence between these groups which is the criterion we are optimizing for in the multi channel approaches. In fact there have been papers on the relation of perceptual grouping and information theoretic measures (Barlow 1959, 1961, 1989,Attneave 1954). Optimizing with respect to information theoretic measures will obviously be more efficient, complete, and accurate than performing combinatorics with heuristic rules.

An additional advantage of the multi channel approaches is the existence of an optimal solution. With single channel approach there are no optimal solutions and the extractions methods that are employed are seriously degrading sound quality. An additional problem with single channel separation is that there are no mathematical operations that can separate mixed sequences given only one mix. Multi input approaches on the other hand, offer a correct solution by solving a system of equations. That way there are no losses during recovery and the sound quality remains intact.

For these two reasons this thesis will be occupied with multi input algorithms. In the last section of the thesis I'll be proposing future extensions of the single input algorithms that make use of more robust grouping rules which are mathematically correct and complete.

2.6 Conclusions

In this chapter we have seen various approaches to the problem of source separation. One important observation that was made is that most of these techniques share the same underlying principle, that of maximizing statistical independence. It has been shown that multi input approaches are better defined problems than single input approaches since they provide an optimal solution and no quality degradation.

Chapter 3. Convolved Mixtures

3 . 1 Overview

The approaches that were presented in the previous chapter are robust enough to use in the real world, however we rarely encounter perfectly instantaneous mixtures of sounds. In this chapter we'll introduce the problem of convolved mixtures and some of the proposed extensions to the unmixing algorithms to deal with this problem. In order to cope with the increasing mathematical complexity we introduce the FIR matrix algebra which allows us to use the already established notation to express the new problem.

3 . 2 Convolved Mixtures

In the real world it is very rare to find instantaneous mixtures of audio signals. Most acoustic environments impose their impulse responses to sources recorded within them, and the measurements are impossible to model with a memoryless equation. To illustrate, consider a case where we have two sources (s_1 and s_2) recorded by two microphones (M_1 and M_2) in a room (Figure 5). As is the case in the real world, the microphones will not only record the direct sound from the sources but also their reflections from the room walls (only first order reflections are shown in the figure, in theory there are infinite reflections[†]). Even if the recording was taking place in an anechoic chamber, where reflections are suppressed, each sound would reach every microphone

[†] Even though a room does have infinite reflections, it is customary to ignore reflections that are 60 dB below the original signal. For practical purposes room responses are considered to be FIR rather than IIR.

at a different time due to propagation delays. This is also considered as a case of convolved mixtures where the mixing filters are just delays. Finally, the microphones will most likely have slightly different frequency responses, which introduces another layer of filtering in the problem. Any one of the above reasons is enough to make the problem ill-defined for instantaneous mixture solutions, considering that all three apply in a real world recording, we find a need for a new separation algorithm.

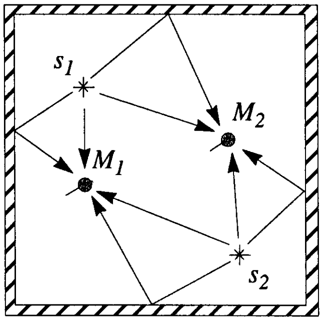

Figure 5 **A case of convolved mixtures**

The equation that describes this convolved mixing process is:

$$x_i(t) = \sum_{j=1}^{N} \sum_{k=0}^{M} s_j(t-k) a_{ij}(k) \tag{17}$$

where s_j are the N original sources, a_{ij} are the order M mixing filters that describe the acoustic environment and x_i are the measured signals. It is obvious that recovering the s_j terms is impossible by just a matrix multiplication with the x_i terms, such as in the previous chapter.

In order to make a connection to the previous chapter we will proceed by expressing this mixing process in FIR matrix algebra terms (Lambert 1996). In FIR matrix algebra, matrices can contain time series as elements, in addition to scalars. In that case element multiplications are substituted by element convolutions:

$$\begin{bmatrix} \underline{\mathbf{a}}_{11} & \underline{\mathbf{a}}_{12} \\ \underline{\mathbf{a}}_{21} & \underline{\mathbf{a}}_{22} \end{bmatrix} \cdot \begin{bmatrix} \underline{\mathbf{s}}_1 \\ \underline{\mathbf{s}}_2 \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{a}}_{11} * \underline{\mathbf{s}}_1 + \underline{\mathbf{a}}_{12} * \underline{\mathbf{s}}_2 \\ \underline{\mathbf{a}}_{21} * \underline{\mathbf{s}}_1 + \underline{\mathbf{a}}_{22} * \underline{\mathbf{s}}_2 \end{bmatrix} \tag{18}$$

We can also express FIR matrix operations in the frequency domain. In that case the time series become polynomials and the convolutions transform to element-wise multiplications (note that the elements are now complex number sequences and we need to perform complex multiplication):

$$\begin{bmatrix} \hat{\underline{\mathbf{a}}}_{11} & \hat{\underline{\mathbf{a}}}_{12} \\ \hat{\underline{\mathbf{a}}}_{21} & \hat{\underline{\mathbf{a}}}_{22} \end{bmatrix} \cdot \begin{bmatrix} \hat{\underline{\mathbf{s}}}_1 \\ \hat{\underline{\mathbf{s}}}_2 \end{bmatrix} = \begin{bmatrix} \hat{\underline{\mathbf{a}}}_{11} \cdot \hat{\underline{\mathbf{s}}}_1 + \hat{\underline{\mathbf{a}}}_{12} \cdot \hat{\underline{\mathbf{s}}}_2 \\ \hat{\underline{\mathbf{a}}}_{21} \cdot \hat{\underline{\mathbf{s}}}_1 + \hat{\underline{\mathbf{a}}}_{22} \cdot \hat{\underline{\mathbf{s}}}_2 \end{bmatrix} \tag{19}$$

The time series elements of a FIR matrix are notated as underlined vectors (lowercase bold, e.g. ($\underline{\mathbf{a}}$), the FIR matrixes are notated as an underlined matrix (uppercase bold, e.g. $\underline{\mathbf{A}}$) and their respective frequency transforms are denoted using the hat mark (e.g. $\hat{\underline{\mathbf{a}}}$ or $\hat{\underline{\mathbf{A}}}$).

Using the above rules, Equation (17) in FIR matrix algebra terms becomes the familiar mixing expression:

$$\underline{\mathbf{X}} = \underline{\mathbf{A}} \cdot \underline{\mathbf{S}} \tag{20}$$

where the mixing matrix $\underline{\mathbf{A}}$ consists of the FIR filters $\underline{\mathbf{a}}_{ij}$, whose elements are the coefficients, of the order M mixing filters[†]:

$$\underline{\mathbf{A}} = \begin{bmatrix} \underline{\mathbf{a}}_{11}^{T} & \cdots & \underline{\mathbf{a}}_{1n}^{T} \\ \vdots & \ddots & \vdots \\ \underline{\mathbf{a}}_{n1}^{T} & \cdots & \underline{\mathbf{a}}_{nn}^{T} \end{bmatrix} \tag{21}$$

and $\underline{\mathbf{S}}$ is:

$$\underline{\mathbf{S}} = \begin{bmatrix} \underline{\mathbf{s}}_1 \\ \vdots \\ \underline{\mathbf{s}}_n \end{bmatrix} \tag{22}$$

where the $\underline{\mathbf{s}}_i$ are the original source vectors.

[†] The mixing filters aren't necessarily the same order, however by zero padding the shorter ones we can make sure that all filters have the same length.

Using similar reasoning as before we want to estimate the inverse of $\underline{\mathbf{A}}$. So that we can recover the original sources by:

$$\underline{\mathbf{S}} = \underline{\mathbf{A}}^{-1} \cdot \underline{\mathbf{X}} \tag{23}$$

The inverse of a FIR matrix $\underline{\mathbf{A}}$ is defined as:

$$\underline{\mathbf{A}}^{-1} = \frac{1}{det(\underline{\mathbf{A}})} \cdot \underline{\mathbf{G}} \tag{24}$$

det($\underline{\mathbf{A}}$ *)* is:

$$det(\underline{\mathbf{A}}_{n \times n}) = \sum_{j=1}^{n} \underline{\mathbf{a}}_{1j} \cdot \underline{\mathbf{A}}_{1j} \tag{25}$$

where $\underline{\mathbf{a}}_{1j}$ is the time series in the *j*th column of the first row and $\underline{\mathbf{A}}_{ij}$ is a cofactor of $\underline{\mathbf{A}}$.

$\underline{\mathbf{G}}$ is defined as:

$$\underline{\mathbf{G}} = \begin{bmatrix} det(\underline{\mathbf{A}}_{11}) & \dots & det(\underline{\mathbf{A}}_{n1}) \\ \vdots & \ddots & \vdots \\ det(\underline{\mathbf{A}}_{1n}) & \dots & det(\underline{\mathbf{A}}_{nn}) \end{bmatrix} \tag{26}$$

Alternatively we can express the above expressions in the frequency domain to obtain an identical looking equation, in which the matrix elements are the frequency transforms of the filters and element-wise multiplication is used instead of convolution.

As we'll see in the following sections there are two ways of implementing this matrix inversion, each one with its own learning rule.

3.3 The Feedforward Network Approach

The first solution for the convolved mixtures problem was proposed by Bell and Sejnowski (1996). Their algorithm extended the instantaneous mixture approach presented in the same paper. Their proposed structure, for two inputs, is shown in Figure 6:

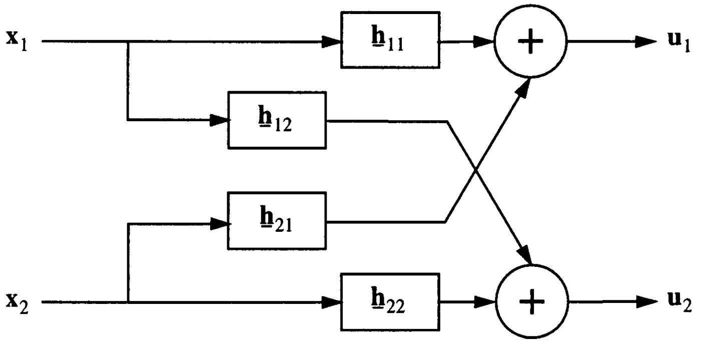

Figure 6 **The feedforward solution for convolved mixtures**

The boxes in the figure represent the unmixing filters. For simplicity for the remaining of the chapter we'll consider the case where we have two sources. The function of the network in Figure 6 is:

$$\underline{U}_{2\times 1} = \underline{H}_{2\times 2} \cdot \underline{X}_{2\times 1} \tag{27}$$

where $\underline{X}$ is a matrix containing the observed mixtures, $\underline{H}$ is our estimate of the unmixing matrix (with elements $\underline{h}_{ij}$)and $\underline{U}$ the outputs of the network.

By expanding the matrix operation from Equation (27) we have:

$$\underline{u}_1 = \underline{h}_{11} * \underline{x}_1 + \underline{h}_{12} * \underline{x}_2 \tag{28}$$

$$\underline{u}_2 = \underline{h}_{21} * \underline{x}_1 + \underline{h}_{22} * \underline{x}_2 \tag{29}$$

Our goal is the same as in the previous chapter, to maximize the independence between the outputs $\underline{u}_i$. The $\underline{H}$ matrix that will satisfy this condition is:

$$\underline{\underline{\mathbf{H}}} = \frac{1}{\underline{\mathbf{a}}_{11}{}^{*}\ \underline{\mathbf{a}}_{22} - \underline{\mathbf{a}}_{12}{}^{*}\ \underline{\mathbf{a}}_{22}} \cdot \begin{bmatrix} \underline{\mathbf{a}}_{22} & -\underline{\mathbf{a}}_{21} \\ -\underline{\mathbf{a}}_{12} & \underline{\mathbf{a}}_{11} \end{bmatrix} \tag{30}$$

where $\underline{\mathbf{a}}_{ij}$ are the elements of the mixing matrix.

The procedure used is the same as the one in Bell's algorithm in the previous chapter, where the goal is to maximize the information flow from input to output. The optimization used was maximization of $\langle \ln|\mathbf{J}| \rangle$, where $\mathbf{J}$ is the Jacobian of the network after a sigmoid has been applied to the outputs and $\langle\ \rangle$ denotes expectation. By performing the derivation shown in Appendix D we obtain the following learning rules:

$$\Delta\underline{\underline{\mathbf{H}}}_0 \propto [\underline{\underline{\mathbf{H}}}_0^T]^{-1} - 2 \cdot \tanh(\underline{\mathbf{u}}_0) \cdot \underline{\mathbf{x}}_0{}^T \tag{31}$$

and

$$\Delta\underline{\underline{\mathbf{H}}}_k \propto -2 \cdot \tanh(\underline{\mathbf{u}}_0) \cdot \underline{\mathbf{x}}_k{}^T,\ k > 0 \tag{32}$$

where the subscript by the FIR matrices and vectors denotes time index. Note that this operation returns a matrix or a vector and that Equation (31) and Equation (32) are using conventional linear algebra rules. So for the first weight we use the same learning rule as in the linear mixtures while for the remaining weights there is a slightly different update.

For practical reasons, we can also ignore the direct filters ($\underline{\mathbf{h}}_{ii}$) and use a scaling parameter in their place. This will make the implementation more efficient since we compute less convolutions than we would otherwise. In this case the matrix $\underline{\underline{\mathbf{H}}}$ would be:

$$\underline{\underline{\mathbf{H}}} = \begin{bmatrix} h_1 & \underline{\mathbf{h}}_{12} & \cdots & \underline{\mathbf{h}}_{1n} \\ \underline{\mathbf{h}}_{21} & h_2 & & \mathbf{h}_{2n} \\ \vdots & & \ddots & \vdots \\ \underline{\mathbf{h}}_{n1} & \mathbf{h}_{n2} & \cdots & h_n \end{bmatrix} \tag{33}$$

where h_i are real scalars. However strong minima are introduced this way and convergence is never achieved (instead the inputs are whitened).

In order to improve performance of this algorithm, it is also be possible to perform natural gradient descent instead of plain gradient descent (Amari *et al.* 1997).

3.4 The Feedback Network Approach

One of the problems with the feedforward approach is the fact that by maximizing the output entropy we also force the filters to whiten the data. Whitening of data tends to form highpass filters that distort the spectral quality of the inputs.

A more practical alternative was proposed by Torkkola (1996a), who employed a feedback network architecture (Figure 7), similar to an older approach by Platt and Faggin (1992).

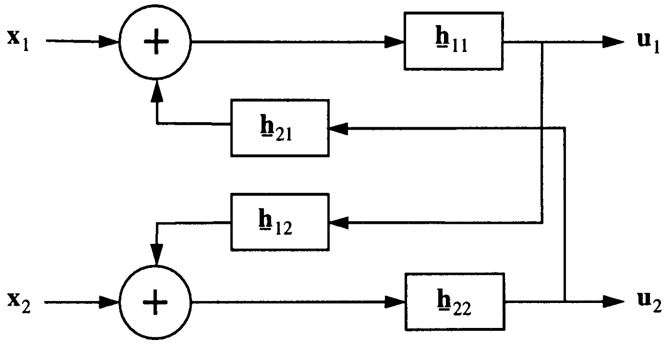

Figure 7 **The feedback solution for convolved mixtures**

The approach was similar as before. The goal is to maximize the $\langle \ln|\mathbf{J}|\rangle$ after a sigmoid is applied to the outputs, in which case the resulting learning rules (see Appendix D for derivation) are:

$$\Delta\underline{h}_{ii}(0) \propto 2 \cdot \tanh(\underline{u}_i(t)) \cdot \underline{x}_i(t) + \frac{1}{\underline{h}_{ii}(0)} \quad (34)$$

$$\Delta\underline{h}_{ii}(k) \propto 2 \cdot \tanh(\underline{u}_i(t)) \cdot \underline{x}_i(t-k),\ k > 0 \quad (35)$$

and

$$\Delta\underline{h}_{ij}(k) \propto 2 \cdot \tanh(\underline{u}_i(t)) \cdot \underline{u}_j(t-k) \quad (36)$$

Just as in the feedforward case, we can again force the diagonal of the unmixing FIR matrix to be composed of scalars.

The feedback approach is interesting in the convolved case since it exhibits better performance than the feedforward structure. The reason for this, is that the feedforward structure attempts to maximize the entropy of the output. Entropy maximization though implies independence not only between the sources but also between consecutive samples. Since in this structure we have filters which can alter time dependencies the feedforward algorithm spends a lot of its resources removing time dependencies in the same signal instead of cross-source dependencies, i.e. whitening it. As a result the outputs are separated but severely filtered.

The feedback structure feeds each filtered input to all of the other outputs. By having this cross filtering structure the filters are forced to deal with the cross-source dependency instead of concentrating on each source.

3.5 Conclusions

In this chapter we described some algorithms that have been developed in order to solve the problem of convolved mixture separation. In order to be consistent with previous notation we also presented them using the FIR matrix algebra, which proved to be an elegant way to avoid complicated equations and will prove to be invaluable in the derivation of the frequency domain algorithms in the next chapter.

Chapter 4. Frequency Domain Extensions

4.1 Overview

In this chapter a new algorithm for separating convolved mixtures of sounds will be presented. Unlike the algorithms in the previous chapter this algorithm operates in the frequency domain. As is the case with adaptive filter theory, it is shown that by operating in the frequency domain there is considerable improvement in convergence, speed of learning and efficiency. Because of the nature of this approach, an introduction to neural networks that work in the complex numbers domain will be also presented.

4.2 Why Move to Another Domain?

The time domain algorithms presented in the previous chapter, even though functional, have some problems. The more prominent ones being lack of efficiency and some convergence rate properties. It is well known in adaptive filter theory (Haykin 1996) that these two problems are present in the time domain and that they are best bypassed by performing adaptation in the frequency domain.

4.2.1 Efficiency Problems

Time domain algorithms are fine for small mixing filters, but when it comes for real time implementations with realistically long filters, they can be unrealizable because of computational requirements. To illustrate, real time separation of two sources sampled at 44.1 kHz in a small room (impulse response of 0.25 sec) would require approximately 16 GigaFlops which is a very demanding workload.

Some performance improvement can be obtained by the use of IIR filters instead of FIR. That way smaller length filters can be used, but at the added risk of numerical instability and the inability to invert non-minimum phase mixing filters. Even then the computational requirements are still very demanding for a real time implementation.

A move towards the frequency domain will be beneficial since there are efficient algorithms of performing these convolutions in that domain with significantly faster performance.

4.2.2 Convergence Considerations

Time domain algorithms for convolved mixture separation are using the maximum entropy cost function. Even though this function was the right one to use for the instantaneous mixtures it is problematic in the case of convolved mixtures. By maximizing the entropy of the outputs we are not only removing statistical dependencies between them but also between consecutive samples in the same signal (spectral whitening). This is because there is optimization credit for decorrelating contiguous samples as well as separating individual outputs. Since filter taps are capable of whitening a signal in many more configurations than they can separate the inputs, it is very easy to waste a lot of resources doing just that. Since most natural sounds have strong time dependencies it is very common for such algorithms to start forming high-pass filter, which results in separated yet whitened outputs.

Remedies for this solution were proposed by Torkkola (1996a, 1997a), who used an algorithm to detect the initial delays of the sources to assist convergence. However this estimation introduces additional complications and convergence considerations (Torkkola (1996b)). The use of a feedback architecture was also made by Torkkola (1996a) which eliminated the whitening problems and exhibited better quality performance.

Finally time domain adaptations have the disadvantage of increased complexity as the length of the mixing filters increases. That is because the update of a filter tap will influence the learning of the ones succeeding it. So update for a filter tap, computed at the time of estimation, will be different from the optimal update which will account for the already updated preceeding filter taps. This leads convergence with very long filters to be potentially problematic. It is desirable to update the filter parameters in a space where they can be independent of each other. One such space is the frequency domain where the all the parameters are orthonormal.

4.3 Planning the Algorithm

4.3.1 Moving to the Frequency Domain

As shown in the previous chapter a mixture of convolved sources can be modeled by the following FIR linear algebra equation:

$$\underline{\mathbf{X}} = \underline{\mathbf{A}} \cdot \underline{\mathbf{S}} \tag{37}$$

We also know that the above expression is equivalent to:

$$\underline{\hat{\mathbf{X}}} = \underline{\hat{\mathbf{A}}} \cdot \underline{\hat{\mathbf{S}}} \tag{38}$$

where the hat mark denotes frequency transform of the FIR filters. If we unfold this matrix product, (for the 2 by 2 case) we get:

$$\begin{bmatrix} \underline{\hat{\mathbf{a}}}_{11} & \underline{\hat{\mathbf{a}}}_{12} \\ \underline{\hat{\mathbf{a}}}_{21} & \underline{\hat{\mathbf{a}}}_{22} \end{bmatrix} \cdot \begin{bmatrix} \underline{\hat{\mathbf{s}}}_1 \\ \underline{\hat{\mathbf{s}}}_2 \end{bmatrix} = \begin{bmatrix} \underline{\hat{\mathbf{a}}}_{11} \cdot \underline{\hat{\mathbf{s}}}_1 + \underline{\hat{\mathbf{a}}}_{12} \cdot \underline{\hat{\mathbf{s}}}_2 \\ \underline{\hat{\mathbf{a}}}_{21} \cdot \underline{\hat{\mathbf{s}}}_1 + \underline{\hat{\mathbf{a}}}_{22} \cdot \underline{\hat{\mathbf{s}}}_2 \end{bmatrix} \tag{39}$$

where the $\cdot$ operator is element-wise multiplication. If we look at this result at element index *i*, we have:

$$\begin{bmatrix} \hat{a}(i)_{11} \cdot \hat{s}(i)_1 + \hat{a}(i)_{12} \cdot \hat{s}(i)_2 \\ \hat{a}(i)_{21} \cdot \hat{s}(i)_1 + \hat{a}(i)_{22} \cdot \hat{s}(i)_2 \end{bmatrix} \tag{40}$$

where every element is a complex number. This can be rewritten as a matrix multiplication which would make Equation (38) equivalent to:

$$\hat{\mathbf{X}}_i = \hat{\mathbf{A}}_i \cdot \hat{\mathbf{S}}_i, \qquad i = 1, \ldots, N \tag{41}$$

By closer examination of the above equation, we can see that it describes a set of cases of simple mixtures[†]. This leads us back to the original problem of instantaneous mixtures as described in chapter 2. The only difference now is that the input signals are complex number sequences instead of real. In order to continue the development of the frequency domain approach we need to formulate the algorithm in the complex number domain and introduce neural networks in the complex domain.

[†] In other words "Convolution in the time domain is multiplication in the frequency domain"

4.4 Neural Networks in the Complex Domain

Neural networks are traditionally implemented for real number inputs and outputs. However it is perfectly possible to use neural networks with complex numbers (referred to from now on as CANN - Complex Artificial Neural Networks). In cases where there is processing in the complex domain, CANNs are often superior to the more traditional Artificial Neural Networks (ANN). Even though ANNs can be used for complex domain problems by using an ANN for the real part and another for the imaginary, CANNs will almost always train faster, are more resistant to becoming trapped in local minima and generalize much better. In addition the values of the CANN networks can be easily interpreted and related to statistical techniques. The use of ANNs introduces the extra problem of hard-to-interpret weights.

Especially where we have frequency domain processing, CANNs are a superior choice because their weights will be easy to interpret and relate to the separating filters we are trying to estimate.

4.4.1 Complex Units

CANNs are like regular ANNs except that their weights, inputs and outputs are complex numbers instead of real. So, for example, a complex CANN unit looks like the one shown in Figure 8,

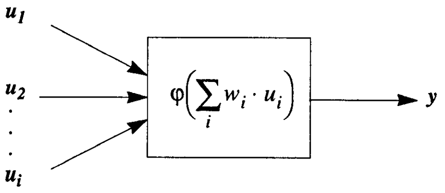

Figure 8 A complex neuron

where u_i are the net inputs, w_i the unit weights and y is the unit output each complex values. The activation function $\varphi(\cdot)$ is a nonlinear complex activation function which must satisfy certain properties described in the following section. CANNs are constructed by interconnecting complex units such as the one in Figure 8.

In mathematical notation there is no distinction between the CANN and ANN formulas for the forward pass. So the implementation of the separation network would still be a matrix multiplication and a pass through the activation function. However for the backward pass there can be significant differences that vary with the type of algorithm.

4.4.2 The Activation Function

One of the most subtle points in designing CANNs is the activation function. For the ANNs used in previous chapters we used the hyperbolic tangent. For CANNs this function will not work for a variety of reasons. Recall that for complex numbers the hyperbolic tangent is defined as:

$$\tanh(z) = \frac{e^z + e^{-z}}{e^z - e^{-z}} \tag{42}$$

It is clear that for values of $z = \left(k + \frac{1}{2}\right)\pi i$ this function (shown in Figure 9) is not defined.

Such singularities will cause stability problems in software and hardware implementations and will hinder the learning process. This problem has been addressed by Georgiou and Koutsogeras (1992) who developed the following set of properties that a complex activation function must have in order to be used for a CANN:

- The activation function must be non-linear in both the real and the imaginary domains. Otherwise we lose the desired properties that nonlinear networks have.
- The activation function should be bounded otherwise implementation of the forward pass is not feasible due to numerical overflows.

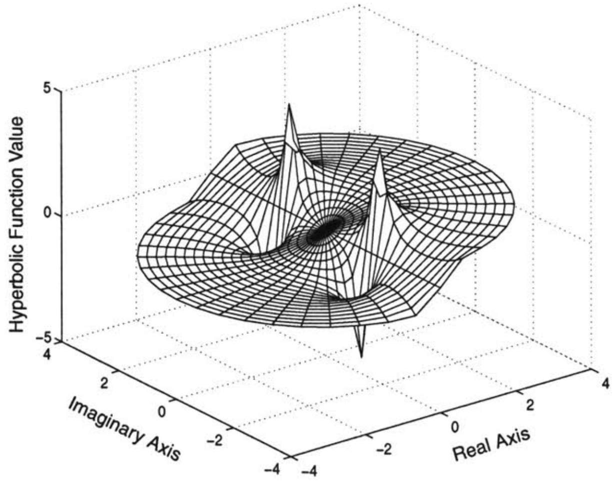

Figure 9 The tanh function in the complex domain

- The partial derivatives of the activation function must exist and must also be bounded in order to be able to implement the backward pass.
- The activation function must not be an *entire function*[†]. By Liouville's theorem we know that bounded entire functions are constant. This would conflict with the first desired property.
- And finally the sum of the partial derivatives of the cost function w.r.t. the weights should be different from zero. If not then the gradient of the error could be zero, which would stop the learning procedure and disable learning.

The function that will be used for this implementation is a variant of the activation function proposed by Benvenuto and Piazza (1992) tailored for this problem. This function fulfills the above requirements and is defined as:

$$\varphi(z) = \tanh(z_R) + i \cdot \tanh(z_I) \tag{43}$$

where z_R is the real part of z and z_I its imaginary part. It is fortunate in this case that $\frac{\partial}{\partial z}\varphi(z) = -2 \cdot z$, which is the same derivative used in the real domain derivations. The function is shown in Figure 10 and it is easy to see that it complies with the complex activation function requirements.

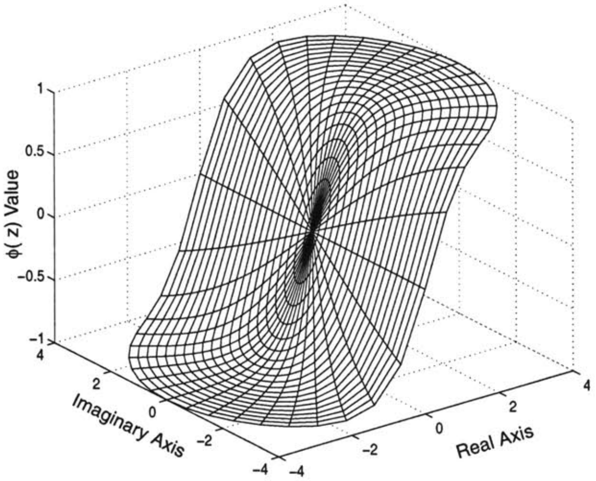

Figure 10 A proper complex activation function

[†]. Entire functions are complex functions which are analytic everywhere in the complex domain

Referring to the intuitive explanation of information maximization in Chapter 2, the reason why this function was chosen is because it will be a better fit to the PDFs of the frequency bin sequences which can be roughly seen as 2 dimensional Gaussians with means at 0,0.

4.4.3 The Learning Functions

Changing the learning rules to work in the complex domain is a simple procedure in this case. Since the derivative is notationally the same, we only need to change the matrix transpositions to Hermitian transpositions. So for Bell's rule we have:

$$\Delta\mathbf{W} \propto [\mathbf{W}^T]^{-1} + \mathbf{Y}\cdot\mathbf{X}^T \rightarrow \Delta\mathbf{W} \propto [\mathbf{W}^H]^{-1} + \mathbf{Y}\cdot\mathbf{X}^H \tag{44}$$

and for Amari's natural gradient rule:

$$\Delta\mathbf{W} \propto (\mathbf{I} - \mathbf{Y}\cdot\mathbf{U}^T)\cdot\mathbf{W} \rightarrow \Delta\mathbf{W} \propto (\mathbf{I} - \mathbf{Y}\cdot\mathbf{U}^H)\cdot\mathbf{W} \tag{45}$$

Because of the better convergence properties that Amari's rule exhibits, we use it in this implementation.

4.5 The Frequency Domain Algorithm

Now that the complex network rules have been set up we can continue with the construction of the algorithm. Our goal is to separate a group of instantaneous mixtures of complex number sequences. In order to do that we can apply one instance of the instantaneous separation rule to the frequency bin tracks. A flow graph of this procedure is shown in Figure 11. The inputs are transformed to the frequency domain where we can use a bank of separation networks, to separate the Fourier coefficients, now linearly mixed. The separated bins are then transformed back to the time domain to give back the original sources.

It is clear that this implementation is very close to a fast convolution algorithm. The elements of the weight matrices of the CANNs are actually the frequency response values of the separation filters. To clarify consider the 2x2 case in Figure 12. As shown in the equivalent time domain implementation we would have four separating filters (two direct and two cross filters, denoted by h_1, h_2, h_3 and h_4). In the frequency domain every frequency bin will have a CANN with a 2x2 complex weight matrix. The sequence formed by the corresponding elements of every matrix will be the frequency response of one of the four filters. In the depicted case we would have:

$$DFT(h_i, f) = H_i^{(f)} \tag{46}$$

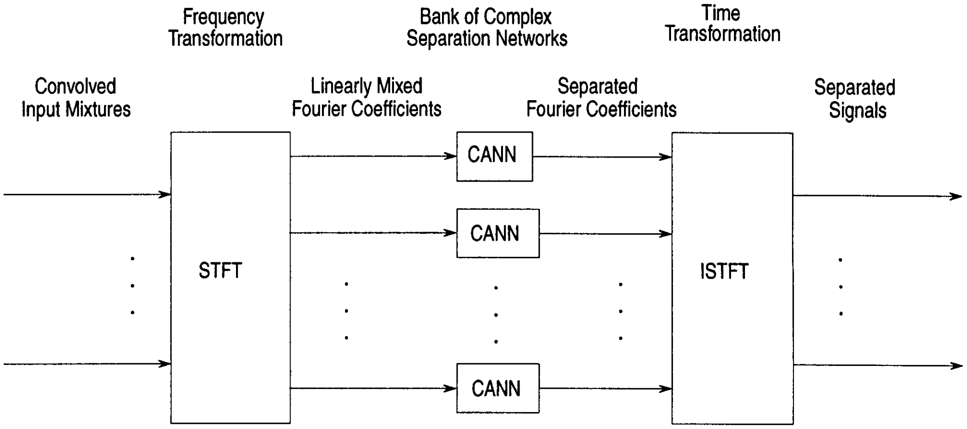

Figure 11 **Flow graph of the algorithm**

If we want to numerically evaluate the results, this is how we obtain the separation filters. It is easy to see the value of using CANNs instead of ANNs. Ignoring convergence considerations, an ANN implementation would provide us with hard to interpret weights in the form of a 2 by 2 matrix for every complex number The fact that we use CANNs makes the weights meaningful and their values useful for evaluating the performance of the algorithm.

By the above it is evident that in the diagram in Figure 11 we are separating with FIR filters. If the mixing filters are IIR it is theoretically possible to achieve perfect separation (since the inverse of an IIR filter is an FIR filter). However most rooms tend to have FIR-like responses (especially in the early stages of reverberation) which means that, given sufficiently long separation filters, the separation will be approximate[†]. Even though this may sound like a problem it is not so serious since by using FIR filters we can invert non-minimum phase mixing filters (which most rooms are) and we avoid instabilities that IIR filters are infamous for. Furthermore the fast convolution algorithm is more efficient than the IIR time domain algorithm.

If we want to directly resynthesize the unmixed sources from this algorithm there is one minor complication we need to deal with. We shouldn't use one CANN for every frequency bin. This would implement unmixing filters equal in length to the frequency transformation which will result in circular convolution. This will introduce audible clicks to the outputs that will mask over the separated sounds and result in poor audio quality. The solution for this problem is to adapt on every other bin and then estimate the in-between coefficients by interpolation. Since interpolating in the frequency domain corresponds to zero padding in the time domain, this transformation will yield zero padded filters which give us convolution of a safe length.

[†] In theory, room responses are IIR but in reality they are measured as FIR filters.

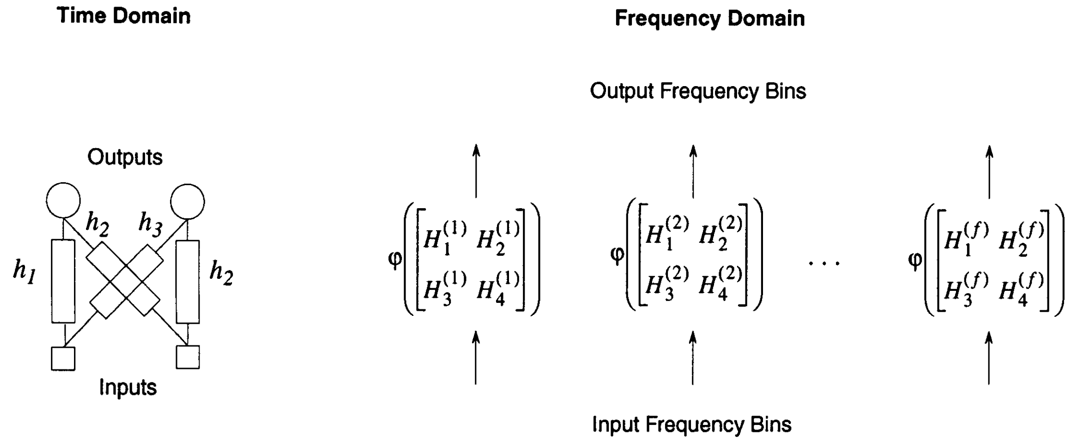
Time Domain
Outputs
h_2 h_3
h_l
h_2
Inputs
Frequency Domain
Output Frequency Bins
$\varphi\left(\begin{bmatrix} H_1^{(1)} & H_2^{(1)} \\ H_3^{(1)} & H_4^{(1)} \end{bmatrix}\right)$
$\varphi\left(\begin{bmatrix} H_1^{(2)} & H_2^{(2)} \\ H_3^{(2)} & H_4^{(2)} \end{bmatrix}\right)$
. . .
$\varphi\left(\begin{bmatrix} H_1^{(f)} & H_2^{(f)} \\ H_3^{(f)} & H_4^{(f)} \end{bmatrix}\right)$
Input Frequency Bins

Figure 12 **Comparison between time and frequency domain algorithms**

4.5.1 Implementation Specifics

a. The Frequency Transform

The frequency transformation is an optimized implementation of the Short Time Fourier Transform. Since the input signals are real, the Fast Hartley Transform (FHT) is being used for the transformation. The FHT is optimized for real signals and offers a performance superior to real FFT and trigonometric recombination techniques.

The size of the FHT determines the length of the deconvolving filters. In order to present more data to the CANNs, so as to help training, the FHTs can overlap with each other. Finally because we perform a convolution we need to zero pad the FHT inputs to avoid time aliasing once we go back to the time domain.

b. The CANN Bank

The CANN bank is the heart of the algorithm. For every frequency bin we assign a CANN implementing Amari's rule. Training is performed on-line and the CANNs are given an input every time a new FHT is computed. By the matrix multiplication that the CANNs perform we also implement a set of convolutions with the estimated separating filters. The outputs of the CANNs are then being given to the time domain transformation algorithm to obtain the original sources.

There are two issues that arise with this approach. The separation matrices of every CANN are not guaranteed to have the same scaling and permutation. In order to get around the scaling problem we can multiply every separation matrix M_i with $det(M_i)^{-\frac{1}{n}}$ which will then result to a matrix $M_i \cdot det(M_i)^{-\frac{1}{n}}$ which has a determinant of 1. If all of the CANNs have a determinant of 1 they will be volume conserving and that will not alter the spectral envelope of the signal in disproportionate ways.

For the permutation there are two schemes. We can compute the separating filters offline, in which case we estimate the separating matrix of every bin throughout the input and then proceed to the next frequency bin. We then force every CANN to use as initial weights the weights of the CANN assigned to the previous bin. Since we are zero padding to avoid time aliasing the spectrum will be interpolated by at least a factor of two which will ensure a smoother spectral envelope. Since the neighboring frequency values should be somewhat close to the one we are training, we make sure that the permutation will not change from bin to bin.

An alternative scheme is the on-line algorithm which is more valuable since it is memory efficient and can be applied for real-time implementations. In this scheme all frequency bins are trained in parallel and there is no interaction between them. There are few guarantees that all unmixing matrices will converge to the same permutation and this can cause adaptation problems. In general, given the relations of frequency tracks from the same source the gradient paths are somewhat similar for all bins and by careful use of the adaptation parameters this problem can be avoided (but at the cost of slower convergence).

c. The Time Transform

The time transformation is implemented using the Inverse Fast Hartley Transform (IFHT). Since this is a convolution algorithm we use the overlap add technique to go back in the time domain. Given that efficiency is important to this application we do not need to use all of the overlapping FHTs from the analysis to go back to the time domain (recall that we use a high overlap factor to increase the number of presentations and help convergence).

4.6 Improvements over Time Domain Algorithms

At this point the problems that we have encountered in the time domain algorithms have been solved by the transition to the frequency domain.

4.6.1 Efficiency

Unlike the time domain algorithms, this approach is much more efficient since it is a $O(N \cdot \log N)$ type algorithm versus the $O(N^2)$ time domain approaches. For the example in Section 4 . 2 . 1 that required 16 GigaFlops this algorithms requires only 75 MegaFlops. A considerable improvement! Of course, as is the case with fast convolution, for very small filter lengths this algorithm will be slower. However, given that we attempt to invert FIR like mixing with FIR unmixing we need to ensure that the separation filters are adequately long which is beyond the threshold where this algorithm performs best.

4 . 6 . 2 Convergence Properties

As mentioned before, the main problem with time domain adaptation is that convergence can be hindered by the statistical dependencies between the filter taps. Using the CANN bank we are decomposing the problem into many simpler and independent problems. The representation of the unmixing filters in the frequency domain provides a set of orthogonal coefficients to train on, so updates on one coefficient will not influence the training of the others. In general this ensures a faster convergence.

In addition to the above feature, convergence in the frequency domain is also invariant to filter length. The length of the mixing and separating filters does not complicate training. It will increase the length of the FHTs and the number of CANNs required, but since the CANNs are independent of each other this will only introduce extra numeric computation and no additional complexity in training.

4 . 7 Conclusions

In this chapter I've presented a new algorithm, formulated in the frequency domain, that has some definite improvements from other approaches. There is a faster convergence rate and fewer minima due to the orthogonality of the frequency coefficients. Also the move to the frequency domain allows us to implement the separation filters using fast convolution. The resulting algorithm features a dramatic increase in performance. In order to implement the algorithm it was also necessary to use complex domain neural networks, which are a superior choice to traditional neural networks since they have better generalization abilities and we obtain interpretable weights using them.

Chapter 5. Conclusions and Future Work

5.1 Overview

In this chapter, results from different classes of mixing problems will be presented and interpreted. After these observations, extensions that could take place as future work will be introduced.

5.2 Performance Results

To evaluate the performance of our algorithm three test cases will be presented, one case of instantaneous mixtures, one of delayed mixtures and one of convolved mixtures. These results were collected using a 2 by 2 separation routine with inputs two 100 sec news broadcasts, consisting solely of speech, sampled at 11025 Hz. Given the interest of real-time applications a real time version of the algorithm was used to obtain these results. The data was presented only once to the algorithm and parameters were chosen so that real-time performance was possible from the test hardware[†]. After the data pass the estimated unmixing FIR matrix was multiplied with the corresponding mixing matrix to obtain a *performance matrix*. This matrix is an indication of how well the inputs were separated, and has to be close to the unit FIR matrix (a matrix where the diagonal elements are the delta function and the rest is zero) to denote success.

[†] Tests were performed on a machine with SPECint95: 4.2 and SPECfp95: 5

5.2.1 Instantaneous mixtures

In order to test that the algorithm does converge, the simplest problem was set up. In this example the sources were mixed instantaneously using a mixing matrix of:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \tag{47}$$

The elements of the product of the unmixing matrix with the mixing matrix are shown in the following figure:

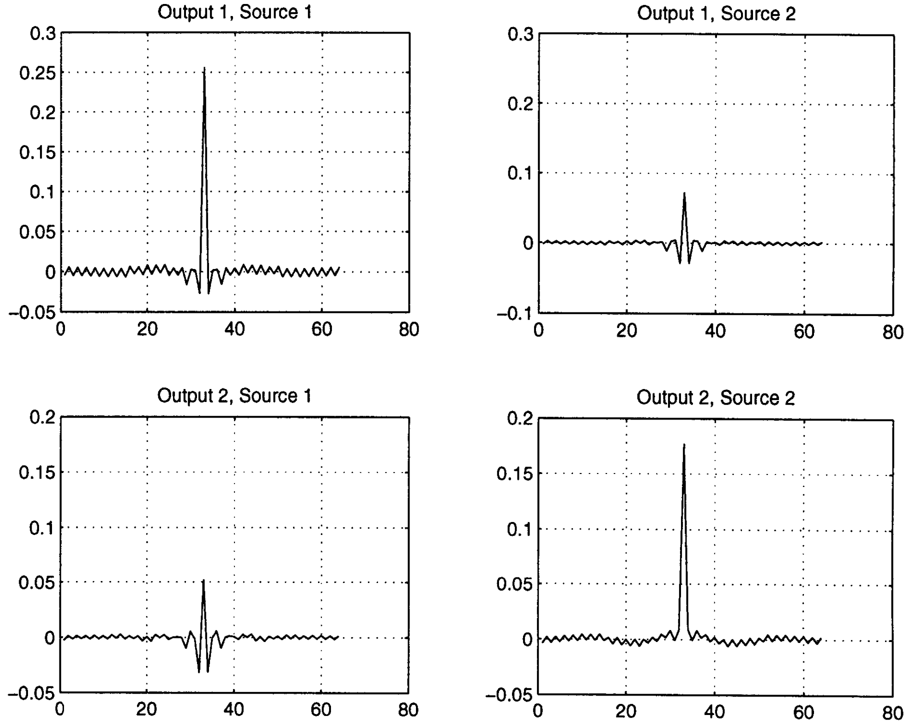

Figure 13 **Instantaneous unmixing results in the time domain**

In order to evaluate the performance of the algorithm, we also need to consider the signals that were used for training. As mentioned above the inputs were speech signals. Speech signals do not have significant high frequency content so training of the high frequency networks is usually bad. This is evident in the plots where the interfering signals are seen as highpass filters, while the cleaned signals are more impulse-like. This can be seen better in the following plots which are the frequency domain representations of the performance matrix. The dashed lines are the interfering signals and the solid lines are the desired signals:

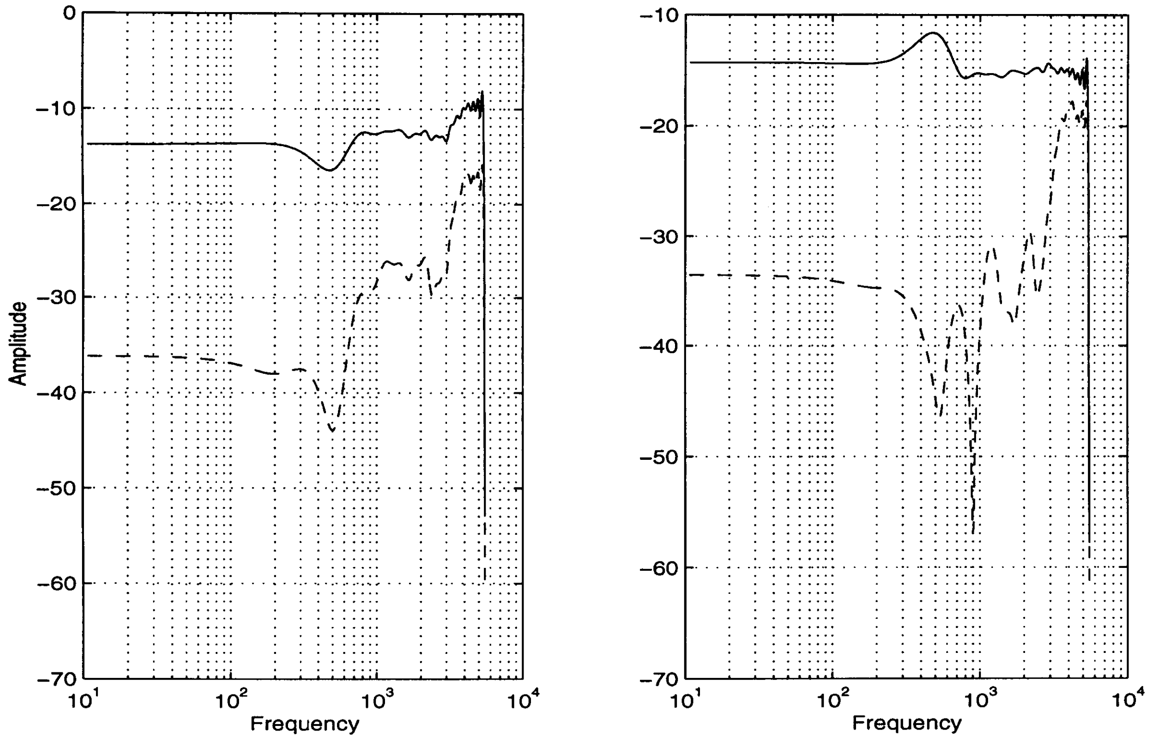

Figure 14 **Instantaneous mixing results in the frequency domain**

By looking at these plots it is evident that the interfering sources were suppressed on average by more than 30dB. Performance starts to degrade in frequency regions were speech is not as active (3kHz and up). Audible separation was almost perfect. Only the high frequency content of loud consonants from the interfering sources was faintly heard.

5.2.2 Delayed Mixtures

The next test used the same weights as the instantaneous mixture case but this time the cross filters included delays of 100 and 56 samples respectively. The mixing matrix was thus:

$$\mathbf{A} = \begin{bmatrix} 2 & \begin{bmatrix} 0^{(1)} \; 0^{(2)} \; \ldots \; 0^{(100)} \; 1^{(101)} \end{bmatrix} \\ \begin{bmatrix} 0^{(1)} \; 0^{(2)} \; \ldots \; 0^{(56)} \; 1^{(57)} \end{bmatrix} & 1 \end{bmatrix} \qquad (48)$$

where the exponents represent index number. The results in this case were:

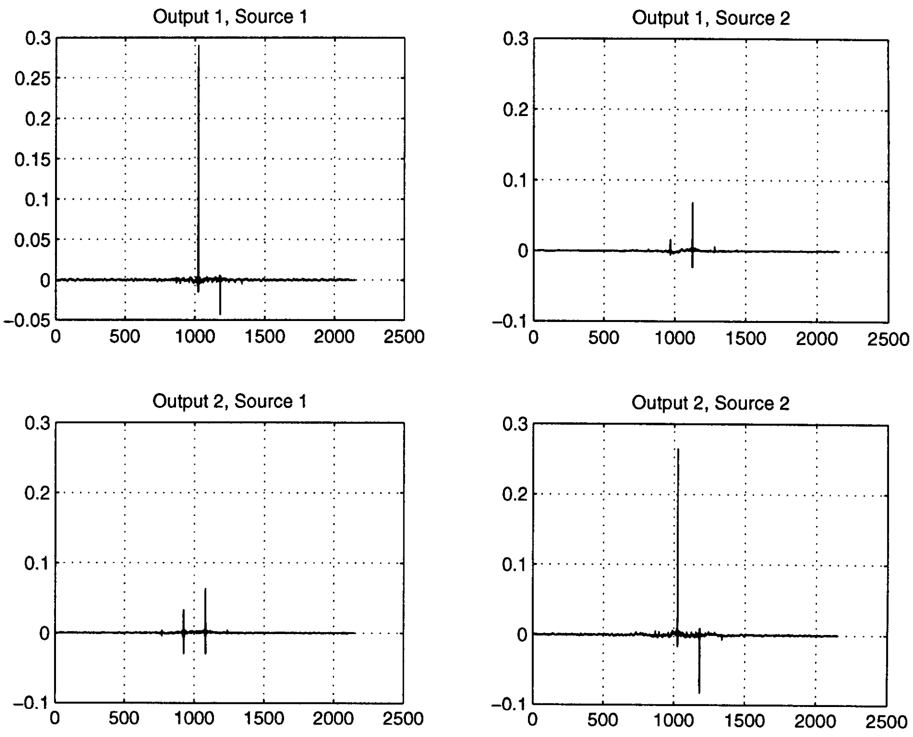

Figure 15 **Delayed mixing results in the time domain**

A couple of observations may to be made for these plots. It is clear that the algorithm did not spend any resources to eliminate the echoes caused by mixing in a single signal. Time domain algorithms would spend some their resources eliminating the echoes and in process of doing so they risk getting stuck in local minima, Since this algorithm is not attempting to lower the entropy of the signals but the cross-entropy it converges in only a few presentations. Audible separation was again almost perfect.

In order to get a better sense of the separation in this case we also present the frequency domain plots:

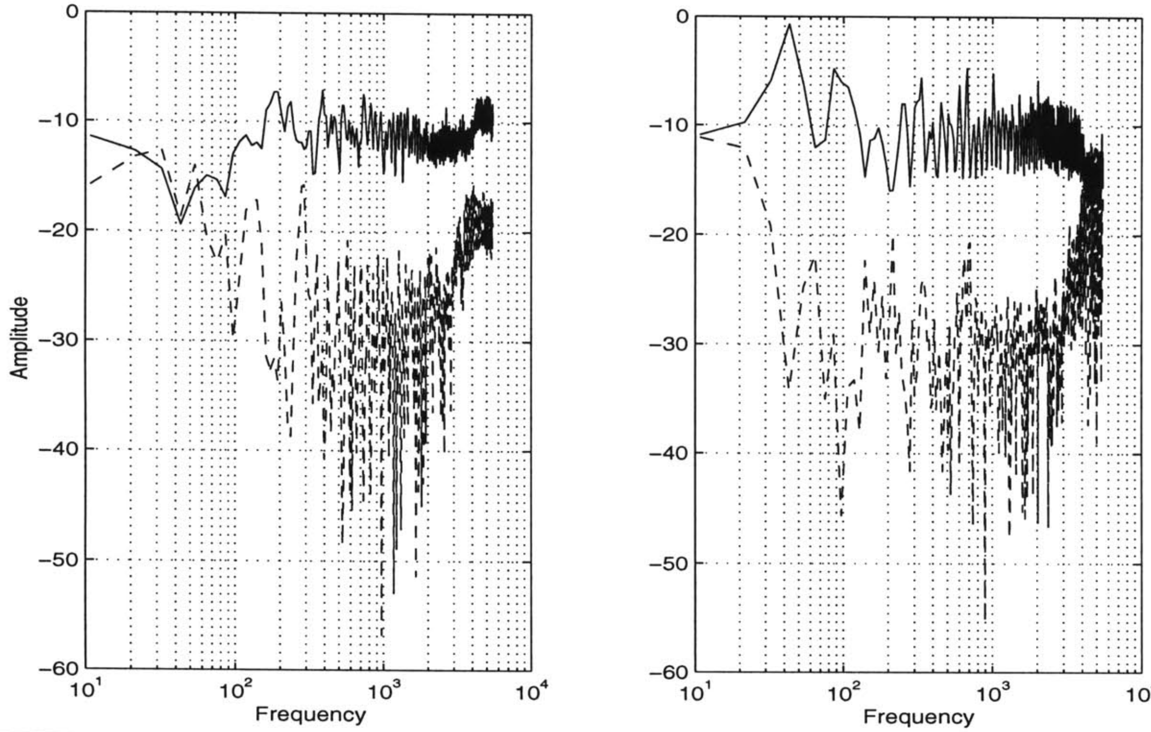

Figure 16 **Delayed mixing results in the frequency domain**

5.2.3 Convolved mixtures

After assuring that the algorithm works for simple cases a more difficult problem was set up. This time the mixing filters were composed of exponentially decaying Cauchy noise. Cauchy noise exhibits a resemblance to room responses since it has a low probability for high values and higher probability for lower values (see Figure 17). The high values are a good model of early reflections, while the lower values are modeling the ambience of the room.

The cross-filters of the mixing matrix were delayed by a random amount of samples to simulate propagation delays. The results are shown in Figure 18 and Figure 19.

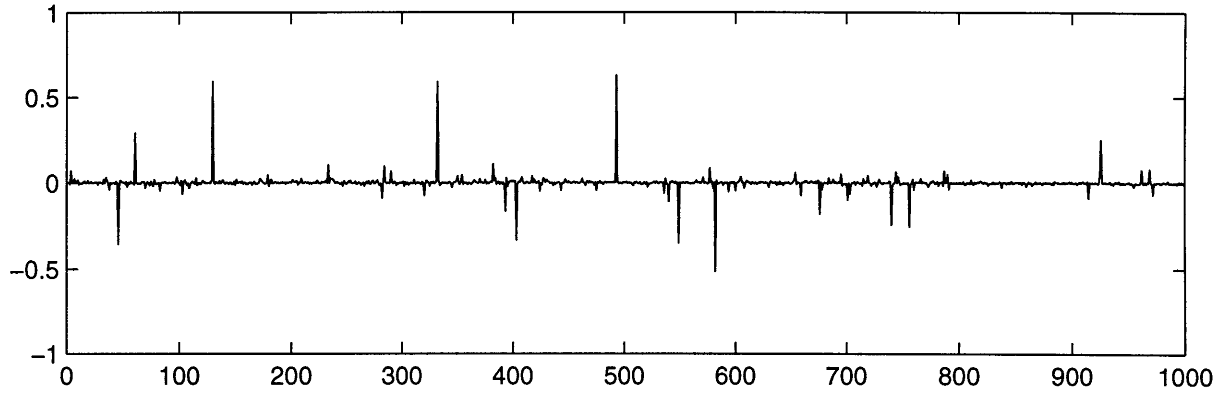

Figure 17 Cauchy noise

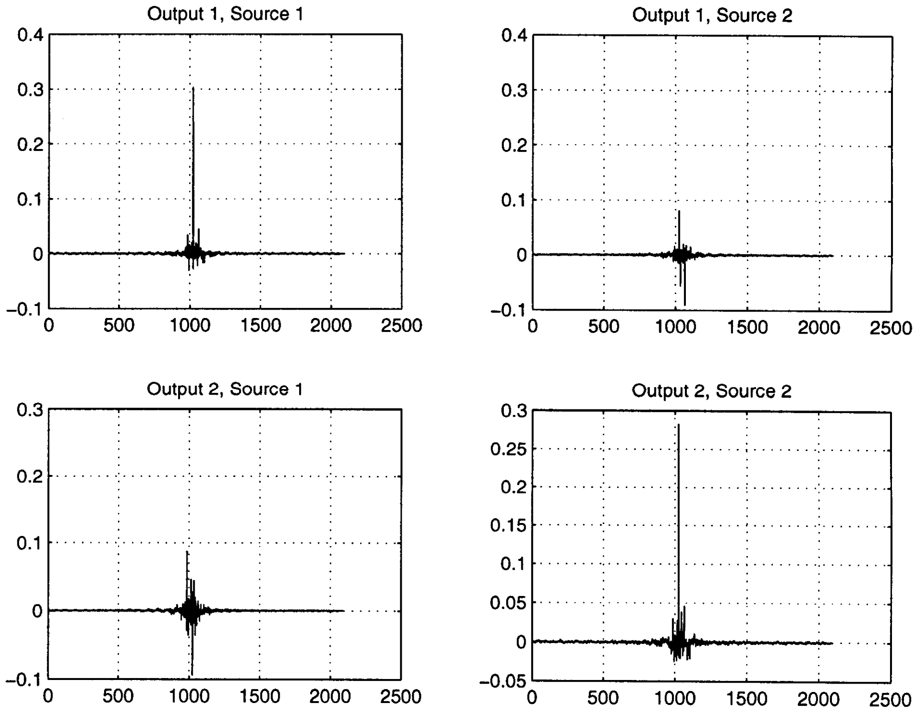

Figure 18 Convolved mixing results in the time domain

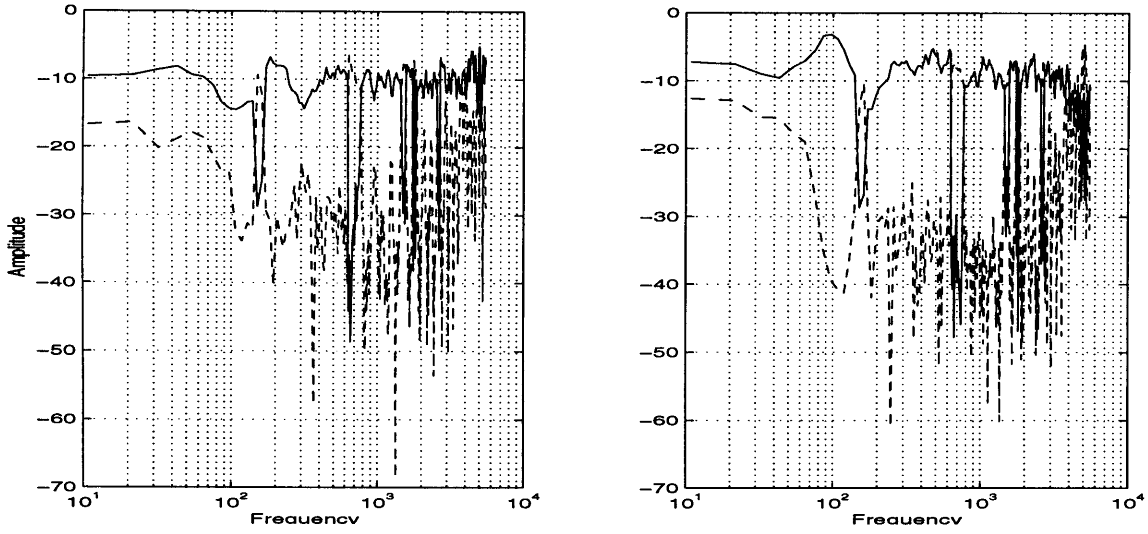

Figure 19 **Convolved mixing results in the frequency domain**

A very interesting observation comes from looking at the frequency responses. On average there was an attenuation by 20dB but for certain frequency bins we see the interfering sound being louder than the desired output. These are cases of frequency bins that converged to the wrong permutation. For these bins we have the interfering source being louder than the desired source. In order to avoid such problems more sophisticated adaptation schemes can be employed. In this case performance can be improved to obtain:

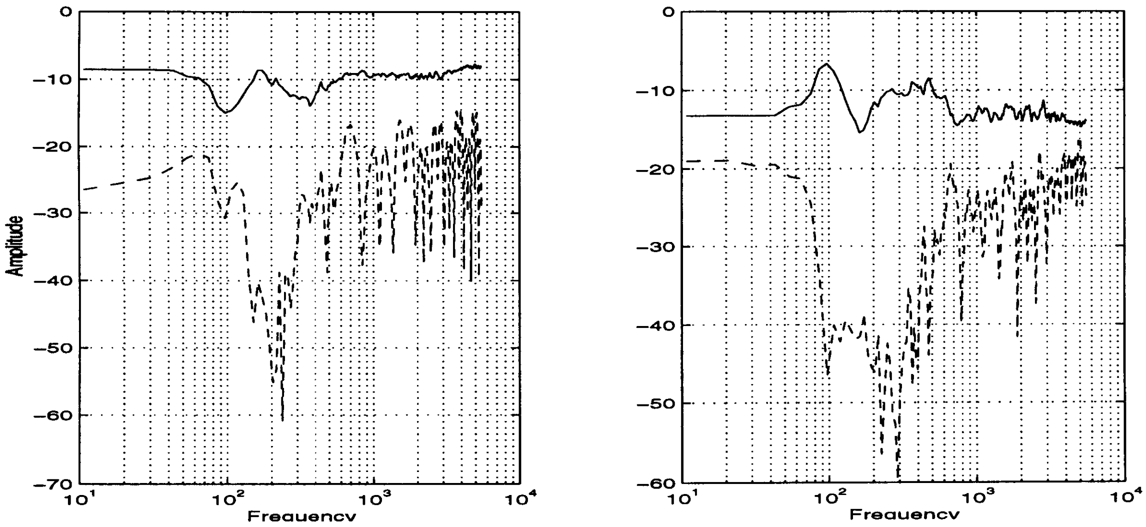

Figure 20 **Convolved mixing results in the frequency domain using parameter fine tuning**

As all systems based on neural networks, this one too requires careful selection of parameters. In order to obtain better results momentum was incorporated in the learning procedure, in addition to a more conservative learning rate with decay.

5.3 Future Work

5.3.1 Short term

So far the algorithm has two major problems. One being the problem of obtaining uniform permutation for the unmixing matrices over all frequency bins. An enhancement was implemented where the update matrix of each frequency bin would be scaled and added to the update of the neighboring bins. This was intended to help all frequency bins influence their neighbors and reach the same permutation. However this proved to generate very strong local minima since it introduced dependencies between the filter parameters. No convergence was achieved in this case even for simple cases.

Another problem is that of the scarceness of inputs. Recall that the algorithm receives data only after a frequency transformation is completed. This delays convergence by a factor of more than a couple of orders of magnitude. In order to fix this we can reduce the hop size of the STFT. An alternative method would be to implement a filterbank which will have an output for every sample. This will seriously increase computational requirements but it will be able to converge in very short times and track dynamically changing fields[†].

5.3.2 Long term

One of the major problem with the algorithms presented in this thesis is that they require *N* inputs to give *N* outputs. In some situations it is possible to setup a data collection scheme that suits these needs. However in there are many cases where we are presented with a single mix and asked to perform separation on this.

As mentioned in Chapter 2, there are some single input algorithms but their performance is not ideal in neither efficiency nor quality nor precision. It was shown however that these algorithms are closely related to the minimum mutual information cost function that some better performing techniques work with. So the question is whether it is possible to adapt the better formulated information theoretic approach to the single input separation schemes.

Preliminary work on this has spawned encouraging results. Toy cases were setup with scenes composed of elementary waveforms such as sines, triangles and square waves. After sinusoidal analysis the resulting tracks were examined and a search was conducted

[†] Convergence times in the test examples varied from 5 seconds to 20. Convergence times around 1 or 2 seconds are better for tracking changing scenes.

which sought for the grouping of the sinusoids that will offer the lowest possible entropy. The grouping configurations that were obtained that way successfully grouped sinusoids that belonged to the same waveforms together.

In addition to this there was some behavior which resembled human judgement. A harmonic tone was produced and periodically more and more harmonics were taken out. After a certain number of harmonics were removed the resulting sound was perceived as two tones, at that same threshold the grouping algorithm output two groups instead of one, both of which described the two perceived sounds. Such behavior is highly desirable since one cost function was used to describe Gestalt grouping rules but also offered deeper insight which would otherwise be unattainable.

At this stage this implementation is based on a simple search algorithm not very faster than exhaustive search. Plans are being made to redesign this algorithm as a self-organizing system based on information theory which is a much more plausible model than search algorithms.

5.4 Conclusions

The problem tackled in this thesis was that of source separation from convolved mixtures. It was shown that instantaneous unmixing algorithms are not capable of finding a correct solution and that more sophisticated algorithms are required. With the introduction of the FIR linear algebra it was possible to use the already existing derivations of instantaneous algorithms to form new approaches that can separate convolved mixtures. It wad also shown that the computational efficiency of the convolved mixture algorithms was very poor. In order to improve efficiency, a frequency domain formulation was proposed which offered superior performance gain and very good convergence features.

The same information theoretic features that were applied to solving this problem were also linked to popular approaches for auditory perception. Preliminary work has shown good results from applying the strong mathematical principles developed here to harmonic grouping approaches. Future work will include algorithms that combine desirable features from engineering oriented multi-input algorithms and systems developed as auditory models to obtain more robust and easy to use performance.

Appendix A. Information Theory Basics

A.1 Introduction

Information theory was formulated by Shannon while at Bell Labs, who was working on the problem of efficiently transmitting information over a noisy communication channel. His approach employed probability and ergodic theory to study the statistical characteristics of communication systems. Since then information theory has been used in a variety of disciples, well beyond telecommunications, ranging from physics to medicine to computer science. In this appendix I'll present parts of information theory that are relevant to this document. Interested readers are pointed out to Shannon and Weaver (1963) for a complete introduction and to Papoulis (1991) for a more rigorous treatment.

A.2 Definition of Entropy

Entropy is the fundamental measure of information theory. It is a very broad concept and it is used to measure the uncertainty of a random variable. It is defined as:

$$H(x) = \int_{-\infty}^{\infty} P(x) \cdot \log P(x) dx = \langle \log \frac{1}{P(x)} \rangle \tag{49}$$

Where x is a random variable and $P(\cdot)$ is its probability density function. The angled brackets $\langle \cdot \rangle$ denote expectation. Depending on the base of the logarithm that is used the units of entropy change. The common units are *nats* for base e and *bits* for base 2.

Since the material in this thesis is mainly on discrete mathematics we'll also consider the discrete definition of entropy:

$$H(x) = -\sum_{x \in \aleph} P(x) \cdot \log P(x) \tag{50}$$

where $\aleph$ is the set that x belongs to. Entropy is bounded from below at 0. An entropy of 0 denotes zero uncertainty which is the case for deterministic processes. From above, the limit is at log(a) for a random variable distributed from 0 to a. This is the case when we have a uniform distribution where uncertainty is maximal. As an example consider the coin toss case. For a fair coin the heads/tails probabilities are:

$$P(heads) = 0.5 \tag{51}$$

$$P(tails) = 0.5 \tag{52}$$

So the entropy is:

$$H(coin) = -[P(heads) \cdot \ln P(heads) + P(tails) \cdot \ln P(tails)] = \ln 2 \tag{53}$$

So we have maximum uncertainty.

If we had a maximally biased coin (towards heads), then:

$$P(heads) = 1 \tag{54}$$

$$P(tails) = 0 \tag{55}$$

and:

$$H(coin) = -[P(heads) \cdot \ln P(heads) + P(tails) \cdot \ln P(tails)] = 0^{\dagger} \tag{56}$$

and if the entropy is 0 we are always certain about the results.

In coding theory entropy has also been used as a measure of the length of the shortest possible description for a random variable sequence.

†. Elaluated at the limit where $\lim_{x \to 0^+} x\ln(x) = 0$

A . 3 Joint and Conditional Entropy

If we have a set of random variables we can also define their joint entropy and their conditional entropy. For the random variables x and y from the sets $\aleph$ and $\mathfrak{Z}$ respectively, the joint entropy is defined for the continuous case as:

$$H(x, y) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(x, y) \cdot \log P(x, y) dx dy = \langle \log \frac{1}{P(x, y)} \rangle \qquad (57)$$

and for the discrete case as:

$$H(x, y) = -\sum_{x \in \aleph} \sum_{y \in \mathfrak{Z}} P(x, y) \cdot \log P(x, y) = \langle \log \frac{1}{P(x, y)} \rangle \qquad (58)$$

for the discrete case. Joint entropy is a measure of overall uncertainty of a set of variables. Taking the entropy of a random vector as $H(\mathbf{x})$ we mean the joint entropy between all of the vector elements.

The conditional entropy of x and y is defined as:

$$H(y|x) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(x, y) \cdot \log P(y|x) dx dy \qquad (59)$$

for the continuous case and:

$$H(y|x) = -\sum_{x \in \aleph} \sum_{y \in \mathfrak{Z}} P(x, y) \cdot \log P(y|x) \qquad (60)$$

for the discrete case, and is a measure of uncertainty of y given certainty of x.

A . 4 Kullback-Leibler Entropy

Also known as the relative entropy, the Kullback-Leibler entropy is a measure of difference between two random variables. The definitions are:

$$K(f,g) = \int_{-\infty}^{\infty} f(x) \cdot \log\frac{f(x)}{g(x)}dx \tag{61}$$

for the continuous case and:

$$K(f,g) = \sum_{x} f(x) \cdot \log\frac{f(x)}{g(x)} \tag{62}$$

for the discrete case, where $f(\cdot)$ and $g(\cdot)$ are the probability densities of the variable that we are comparing. This measure is also called the Kullback-Leibler distance since it exhibits some distance characteristics, that of being positive and that of being equal to zero if $f(x) = g(x)$. It is not a real distance though since $K(f,g) \neq K(g,f)$.

A . 5 Mutual Information

One way to measure statistical independence between two random variables is to look at their mutual information content. This is defined as the Kullback-Leibler distance between the joint probability of these variables and the product of the individual probabilities. The mutual information between two random variables x and y from the sets $\aleph$ and $\Im$, is given by:

$$I(x,y) = K(P(x), P(x) \cdot P(y)) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(x,y) \cdot \log\frac{P(x,y)}{P(x) \cdot P(y)}dxdy \tag{63}$$

for the continuous case and:

$$I(x,y) = K(P(x), P(x) \cdot P(y)) = \sum_{x \in \aleph}\sum_{y \in \Im} P(x,y) \cdot \log\frac{P(x,y)}{P(x) \cdot P(y)} \tag{64}$$

for the discrete case, where $P(\cdot)$ is the probability density function. Mutual information gives us the amount of information that y contains about x.

From the definition we can derive some of the properties of mutual information:

$$I(x,y) = I(y,x) \tag{65}$$

$$I(x, x) = H(x) \tag{66}$$

$$I(x, y) = H(x) - H(x|y) = H(y) - H(y|x) = H(x) + H(y) - H(x, y) \tag{67}$$

Since mutual information is itself a Kullback-Leibler distance it is always positive and zero only if the two variables it measures are independent.

Appendix B. Derivation of Bell's Rule

B.1 Overview

Bell and Sejnowski (1996) were the first to propose a robust information theoretic learning rule for source separation. They argue that maximizing the information flow of the unmixing system it is possible to ensure statistical independence at the outputs. In order to do so they make use of:

$$I(input, output) = H(output) - H(input|output) \tag{68}$$

where I is the mutual information and H the entropy functions. Since the second term of the right hand side is not dependent on the unmixing structure, Bell argues that maximizing the output entropy will result in information maximization between the inputs and the outputs.

B.2 1 by 1 Case

Bell starts by making an output entropy maximization learning rule for a 1 input - 1 output problem. Obviously there is no separation happening here, only output entropy maximization. The sceleton of the derivation will be presented here and it will be generalized in the next section for N by N case.

Assume an input source *x*, and an output $y = g(w \cdot x)$, where *w* is an arbitrary weight variable, and $g(\cdot)$ is a non-linear monotic function. Our goal is to find the value of *w* which maximises the entropy of *y*.

The entropy of *y* is defined as:

$$H(y) = -\int_{-\infty}^{\infty} P(y) \cdot \log P(y) dy = \langle \log P(y) \rangle \qquad (69)$$

where $\langle \cdot \rangle$ denotes expectation and the *P*(*y*) is the probability density function (PDF) of *y*, which can be computed given the PDF of *x* from:

$$P(y) = \frac{P(x)}{\left|\frac{\partial y}{\partial x}\right|} \qquad (70)$$

Substituting Equation (70) in Equation (69) we obtain:

$$H(y) = \langle \log\left|\frac{\partial y}{\partial x}\right| \rangle - \langle \log P(x) \rangle \qquad (71)$$

Since we have no control over the second term in this equation, we can only optimize with respect to the first term. We can remove the expectation operator and define a stochastic gradient rule in which case we need to maximize $\log\left|\frac{\partial y}{\partial x}\right|$. This will be:

$$\Delta w \propto \frac{\partial H(y)}{\partial x} = \frac{\partial}{\partial x}\left(\log\left|\frac{\partial y}{\partial x}\right|\right) = \left(\frac{\partial y}{\partial x}\right)^{-1} \frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right) \qquad (72)$$

A good function to use as the sigmoid is the hyperbolic tangent for which we get:

$$\Delta w \propto \frac{1}{w} - 2xy \qquad (73)$$

B.3 N by N Case

For the N by N case we can use the same reasoning to obtain a learning rule. The inputs now are the vector **x** and the outputs are the vector **y**. Instead of a scalar weight we have to use a matrix **W**, so that $\mathbf{y} = g(\mathbf{W} \cdot \mathbf{x})$.

The main difference in this derivation is that the derivative of the transformation will now be the Jacobian of the transformation. The PDF of **y** will be:

$$P(\mathbf{y}) = \frac{P(\mathbf{x})}{|J|} \tag{74}$$

where:

$$J = det\left(\begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_1} \end{bmatrix}\right) \tag{75}$$

Similarly as in the previous section we maximize w.r.t. J and we derive the following rule:

$$\Delta\mathbf{W} \propto [\mathbf{W}^T]^{-1} - 2 \cdot \mathbf{y} \cdot \mathbf{x} \tag{76}$$

Bell has reported satisfying results in cases where N was up to 10.

Appendix C. Derivation of Amari's Rule

C.1 Overview

This appendix is split in two parts. The first part shows the derivation that Amari et al. (1996) used to reach a learning rule. The second part explains the natural gradient which Amari introduces in conjunction with his learning rule.

C.2 Learning Rule Derivation

The unmixing structure that was used here is the one in Figure 2 from Chapter 2. For this appendix we'll use x_i for the mixed inputs to the unmixing equation, s_i as the original sources, u_i as the unmixing outputs, **W** as our estimate of the unmixing matrix and **A** as the original mixing matrix.

The starting point in Amari's derivation was the Kullback-Leibler distance, The cost function that was used was the Kullback-Leibler distance between the joint entropy of the output and the product of the individual output entropies. If this distance is zero it means that $P(\mathbf{u}) = \prod_{i=1}^{N} P(u_i)$ which is the definition of statistical independence for the elements of u_i (where $P(x)$ is the probability density function (PDF) of x). This distance is described as:

$$K(\mathbf{W}) = \int P(\mathbf{u}) \cdot \log \frac{P(\mathbf{u})}{\prod_{i=1}^{N} P(u_i)} d\mathbf{u} \tag{77}$$

The integral and the PDFs can be substituted using the entropy definition:

$$K(\mathbf{W}) = \sum_{i=1}^{N} H(u_i) - H(\mathbf{u}) \tag{78}$$

Where $H(\mathbf{u})$ is the joint entropy of the outputs and $H(u_i)$ is the marginal entropy of the *i*th output. Using a truncated Gram-Charlier expansion the individual entropies in the right hand side of the above equation can be expressed as:

$$H(u_i) \approx \frac{1}{2}\log(2\pi e) - \frac{\kappa_i(3)^2}{2 \cdot 3!} - \frac{\kappa_i(4)^2}{2 \cdot 4!} + \frac{5}{8}\kappa_i(3)^2\kappa_i(4) + \frac{1}{16}\kappa_i(4)^3 \tag{79}$$

where $\kappa_i(a)$ is the *a*th order moment of the *i*th output. Using Equation (79) and $H(\mathbf{u}) = H(\mathbf{x}) + \log|det(\mathbf{W})|$ we have:

$$K(\mathbf{W}) \approx -H(\mathbf{x}) - \log|det(\mathbf{W})| + \frac{N}{2}\log(2\pi e)$$
$$-\sum_{i=1}^{N}\left[\frac{\kappa_i(3)^2}{2 \cdot 3!} - \frac{\kappa_i(4)^2}{2 \cdot 4!} + \frac{5}{8}\kappa_i(3)^2\kappa_i(4) + \frac{1}{16}\kappa_i(4)^3\right] \tag{80}$$

Equation (80) will serve as our cost function. In order to find its gradient we need $\frac{\partial K(\mathbf{W})}{\partial w_{rc}}$, where w_{rc} is the *c*th element in the *r*th row of **W**. Using the approximation above this will be:

$$\frac{\partial K(\mathbf{W})}{\partial w_{rc}} = -q_{rc} + f(\kappa_r((3), \kappa_r(4))) \cdot \langle u_r^2 \cdot x_c \rangle + g(\kappa_r((3), \kappa_r(4)) \cdot \langle u_r^3 \cdot x_c \rangle) \tag{81}$$

where $\langle \cdot \rangle$ denotes expectation, q_{rc} are elements from $\mathbf{Q} = [\mathbf{W}^T]^{-1}$ and $f(x, y) = -\frac{1}{2}x + \frac{15}{4}xy$, $g(x, y) = -\frac{1}{6}y + \frac{5}{2}x^2 + \frac{3}{2}y^2$.

Since this is an on-line algorithm, we can replace the expected values and the moments by their instantaneous values. By performing these substitutions and rewriting the gradient using linear algebra notation, we reach:

$$\nabla K(\mathbf{W}) = [\mathbf{W}^T]^{-1} - f(\mathbf{u}) \cdot \mathbf{u}^T \qquad (82)$$

where $f(x) = \frac{3}{4}x^{11} + \frac{25}{4}x^9 - \frac{14}{3}x^7 - \frac{47}{4}x^5 + \frac{29}{4}x^3$. It is easy to see that this learning rule is almost the same rule that Bell derived. The only difference is the activation function. It should be noted that by replacing Amari's activation function with the hyperbolic tangent we get more stable learning. This is because $f(\cdot)$ is not bounded and large learning rates result in numerical overflows.

C.3 Natural Gradient

An additional observation that Amari makes is that the space we are optimizing in, is a Riemannian space. The gradient descent in general is defined as the Eucledian gradient multiplied by the metric of the space we are in[†]:

$$\nabla_R f = G^{-1} \cdot \nabla_E f \qquad (83)$$

where ∇_R is the Riemann gradient, ∇_E is the Eucledian gradient and G is the metric of the space.

Amari (1997) shows that in this particular problem the space is indeed Riemannian and that the metric can be substituted with a with a right multiplication by $\mathbf{W}^T\mathbf{W}$ (Cardoso and Laheld 1996), which gives:

$$\begin{aligned} \nabla_R K(\mathbf{W}) &= G^{-1} \cdot \nabla_E K(\mathbf{W}) = \nabla_E K(\mathbf{W}) \cdot \mathbf{W}^T\mathbf{W} = \\ &= [\mathbf{W}^{-T} - f(\mathbf{u}) \cdot \mathbf{x}^T] \cdot \mathbf{W}^T\mathbf{W} = [\mathbf{I} - f(\mathbf{u}) \cdot \mathbf{u}^T] \cdot \mathbf{W} \end{aligned} \qquad (84)$$

By performing steepest descent using the natural gradient convergence is faster and more stable. In addition to good convergence behaviour, there is also increased efficiency since the learning rule does not have a matrix inversion anymore.

[†] In the Eucledian space the metric is a unit matrix so it is traditionaly ommited.

Appendix D. Derivation for Convolved Mixture Time Domain Algorithms

D.1 Overview

In this appendix we'll present the derivations of the learning rules for the feedforward and the feedback algorithms for separating convolved sources in the time domain. The derivations are as presented in Torkkola (1996a). For this appendix we will use the same notation that we established in the previous chapters. We define the inputs to our sensors as x_i, the original sources s_i, the unmixing filters as h_{ij} and the outputs of the algorithms as u_i. As a sigmoid we use the hyperbolic tangent and we also define $y_i = \tanh(u_i)$. The length of the filters is *N*. For simplcity we'll consider the 2 by 2 case, generalizations for more dimensions follow the same derivation.

D.2 Feedforward Architecture

The outputs of the algorithm in the feedforward architecture are defined as:

$$u(t)_1 = \sum_{k=0}^{N} h_{11}(k) \cdot x_1(t-k) + \sum_{k=0}^{N} h_{12}(k) \cdot x_2(t-k) \tag{85}$$

and

$$u(t)_2 = \sum_{k=0}^{N} h_{21}(k) \cdot x_1(t-k) + \sum_{k=0}^{N} h_{22}(k) \cdot x_2(t-k) \tag{86}$$

The approach we use is to maximize the determinant of the logarithm of the Jacobian of the network, which is defined as:

$$\ln|J| = \ln\left(\frac{\partial y_1}{\partial x_1} \cdot \frac{\partial y_2}{\partial x_2} - \frac{\partial y_1}{\partial x_2} \cdot \frac{\partial y_2}{\partial x_1} = y'_1 \cdot y'_2 \cdot D\right) = \ln y'_1 + \ln y'_2 + \ln D \tag{87}$$

where:

$$D = \frac{\partial u_1}{\partial x_1} \cdot \frac{\partial u_2}{\partial x_2} - \frac{\partial u_1}{\partial x_2} \cdot \frac{\partial u_2}{\partial x_1} = h_{11}(0) \cdot h_{22}(0) - h_{12}(0) \cdot h_{21}(0) \tag{88}$$

and

$$y'_i = \frac{\partial u_i}{\partial x_i} \tag{89}$$

To obtain the learning rule we compute the gradient of Equation (87). As an example we compute the direction with respect to $h_{11}(0)$:

$$\frac{\partial \ln|J|}{\partial h_{11}(0)} = \frac{1}{y'_1} \cdot \frac{\partial y'_1}{\partial h_{11}(0)} + \frac{1}{y'_2} \cdot \frac{\partial y'_2}{\partial h_{11}(0)} + \frac{1}{D} \cdot \frac{\partial D}{\partial h_{11}(0)} \tag{90}$$

for the hyperbolic function we have $y'_i = -2 \cdot y_i$. Using that the partial derivatives in Equation (90) are:

$$\frac{\partial y'_1}{\partial h_{11}(0)} = \frac{\partial y'_1}{\partial y_1} \cdot \frac{\partial y_1}{\partial u_1} \cdot \frac{\partial u_1}{\partial h_{11}(0)} = -2 \cdot y'_1 \cdot y_1 \cdot x_1 \tag{91}$$

$$\frac{\partial y'_2}{\partial h_{11}(0)} = \frac{\partial y'_2}{\partial y_2} \cdot \frac{\partial y_2}{\partial u_2} \cdot \frac{\partial u_2}{\partial h_{11}(0)} = 0 \tag{92}$$

$$\frac{\partial D}{\partial h_{11}(0)} = h_{11}(0) \tag{93}$$

By repeating the same procedure for the rest of the parameters we end up with:

$$\Delta h_{11}(0) \propto \frac{h_{22}(0)}{D} - 2 \cdot y_1(t) \cdot x_1(t) \,,\, \Delta h_{12}(0) \propto \frac{-h_{21}(0)}{D} - 2 \cdot y_1(t) \cdot x_2(t) \tag{94}$$

$$\Delta h_{21}(0) \propto \frac{-h_{22}(0)}{D} - 2 \cdot y_2(t) \cdot x_1(t) \,,\, \Delta h_{22}(0) \propto \frac{h_{22}(0)}{D} - 2 \cdot y_2(t) \cdot x_2(t) \tag{95}$$

for the leading weights, and:

$$\Delta h_{ij}(k) \propto -2 \cdot y_i(t) \cdot x_j(t-k) \tag{96}$$

for the remaining weights.

D.3 Feedback Architecture

For the feedback network we use the same methodology but this time the output equations are:

$$u_1(t) = \sum_{k=0}^{N} h_{11}(k) \cdot x_1(t-k) + \sum_{k=0}^{N} h_{12}(k) \cdot u_2(t-k) \tag{97}$$

$$u_2(t) = \sum_{k=0}^{N} h_{22}(k) \cdot x_2(t-k) + \sum_{k=0}^{N} h_{21}(k) \cdot u_1(t-k) \tag{98}$$

In this case we have $D = h_{11} \cdot h_{22}$ and we follow the same steps as before to obtain:

$$\Delta h_{ii}(0) \propto -2 \cdot y_i(t) \cdot x_i(t) + \frac{1}{h_{ii}} \tag{99}$$

$$\Delta h_{ii}(0) \propto -2 \cdot y_i(t) \cdot x_i(t-k) \tag{100}$$

and

$$\Delta h_{ij}(k) = -2 \cdot y_i(t) \cdot u_j(t-k) \tag{101}$$

For both of the above structure we can also use IIR filters instead of FIR filters, the only difference would be in the output equation definition, the remaining derivation steps are similar.

Bibliography

Referenced Bibliography

The following bibliography is referenced from this thesis.

Amari, S-I., A. Cichocki, and H.H. Yang. 1996. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems 8.* MIT Press, Cambridge, MA.

Amari, S-I. 1997. Natural Gradient Works Efficiently in Learning, submitted to Neural Computation.

Amari, S-I., S.C. Douglas, A. Cichocki, and H.H. Yang, 1997 `Multichannel Blind Deconvolution and Equalization Using the Natural Gradient. In *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications*, Paris, France, pp. 101-104.

Atick, J.J. and A.N. Redlich. 1990. Towards a theory of early visual processing. *In Neural Computation 2*. pp. 308-320. MIT Press, Cambridge, MA.

Attneave, F. 1954. Informational aspects of visual perception. *Psychological Review 61*, pp. 183-193.

Barlow, H.B. 1959. Sensory mechanisms, the reduction of redundancy, and intelligence. In *National Physical Laboratory Symposium No. 10*, The Mechanization of Thought Processes.

Barlow, H.B. 1961. Possible principles underlying the transformation of sensory messages, In *Sensory Communication*, W. Rosenblith, ed., pp. 217-234. MIT Press, Cambridge, MA.

Barlow, H.B. 1989. Unsupervised learning. In *Neural Computation 1*, pp. 295-311. MIT Press, Cambridge, MA.

Bell, A.J. and T.J. Sejnowski. 1995. An information maximization approach to blind separation and blind deconvolution. In *Neural Computation 7*. pp. 1129-1159. MIT Press, Cambridge, MA.

Bell, A.J. and T.J. Sejnowski. 1996. The independent components of natural scenes. *Vision Research.* Submitted.

Bodden, M. 1993. Modeling human sound-source localization and the cocktail-party effect. In *Acta Acustica 1*. pp. 43-55.

Brown, G.J. 1992. Computational auditory scene analysis: A representational approach. Ph.D. dissertation, Uniersity of Sheffield, Computer Science Dept., Sept, 1992.

Burel, G. 1992. Blind separation of sources: A nonlinear algorithm, In *Neural Networks*, vol. ***5***, pp. 937-947.

Cardoso, J-F. and A. Souloumiac. 1993. Blind beamforming for non-gaussian signals. In *IEE Proceedings F, vol* ***140****, no 6,* pp. 362-370.

Cardoso, J-F., A. Belouchrani and B.H. Laheld. 1994. A new composite criterion for adaptive and iterative blind source separation. In *Proceedings of International Conference on Acoustics and Speech Signal Processing* 1994.

Cardoso, J-F. and B.H. Laheld. 1996. Equivariant adaptive source separation. In *IEEE Transactions on Signal Processing*, vol. ***44***, no 12, pp. 3017-3030, Dec. 1996.

Cardoso, J-F. 1997. Infomax and maximum likelihood for source separation, To appear in *IEEE Letters on Signal Processing*, April, 1997.

Comon, P. 1989. Independent component analysis - a new concept? In *Signal Processing* ***36***, pp. 287-314.

Cooke, M.P. 1991. Modeling auditory processing and organization. Ph.D. thesis, University of Sheffield, Dept. of computer science.

Ellis, D.P.W. 1992. A perceptual representation of sound. Masters thesis, MIT EECS Department.

Haykin, S. 1996. Adaptive filter theory. Third edition, Prentice-Hall.

Herault, J., C. Jutten. 1991. Blind separation of sources, part I, An adaptive algorithm based on neuromimetic architecture. *Signal Processing* ***24***, pp. 1-10.

Hopfield, J.J. 1991. Olfactory computation and object perception. *Proceedings of the National Academy of Sciences* ***88***, 6462-6466.

Lambert, R. H., 1996. Multichannel blind deconvolution: FIR matrix algebra and separation of multipath mixtures. Ph.D. dissertation, University of Southern California, EE dept. May 1996.

Linsker, R. 1988. Self-Organization in a perceptual network. In *Computer* ***21*** (March), pp. 105-117.

McKay, D. 1996. Maximum Likelihood and Covariant Algorithms for Independent Component Analysis, *Draft paper available at:* `ftp://wol.ra.phy.cam.ac.uk/pub/mackay/ica.ps.gz`

Matsuoka, K., M. Ohya, and M. Kawamoto. 1995. A neural net for blind separation of non-stationary signals. In *Neural Networks,* vol. ***8***, pp. 411-419.

Molgedey L. and H.G. Schuster. 1994. Separation of a mixture of independent signals using time delayed correlations. In *Physical Review Letters* ***72***, 3634-3637.

Nakatami, T., H.G. Okuno, and T. Kawabata. 1994. Auditory stream segregation in auditory scene analysiswith a multi-agent system. In *AIII Conference Proceedings*, 1994.

Papoulis, A. 1991. "Probability, Random Variables and Stochastic Processes", McGraw-Hill series in Electrical Engineering.

Parsons, T.W. 1976. Separation of speech from interfering speech by means of harmonic selection. *Journal of the Acoustical Society of America,* vol. ***60***, pp. 911-918.

Platt, J. and F. Faggin. 1992. In *Advances in Neural Information Processing* ***4***, J. Moody, S. Hanson, R. Lippmann, eds., pp. 730-737, Morgan-Kaufmann.

Redlich, A.N. 1993. Redundancy reduction as a strategy for unsupervised learning. *Neural Computation* ***5***. pp. 289-304. MIT Press, Cambridge, MA.

Shannon, C. and W. Weaver. 1963. "The Mathematical Theory of Communication" University of Illinois Press.

Stockham, T.G., T.M. Cannon, and R.B. Ingerbretsen. 1975. Blind deconvolution through digital signal processing. In *Proc. IEEE,* vol. ***63***, pp 678-692.

Torkkola, K. 1996a. Blind separation of convolved sources based on information maximization. In *Neural Networks for Signal Processing VI.* Kyoto Japan, IEEE press.

Torkkola, K. 1996b. Blind separation of delayed sources based on information maximization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing.* Atlanta, GA.

Weintraub, M. 1985. A theory and computational model of auditory monaural sound separation. Ph.D. dissertation, Stanford University, EE Dept.

Additional Bibliography

The following bibliography is suggested reading. On relevant topics. Most of these papers can be found at: `http://sound.media.mit.edu/~paris/ica.html`

Amari, S-I., A. Cichocki and H.H Yang. 1995. Recurrent Neural Networks for Blind Separation of Sources, In *Proceedings 1995 International Symposium on Nonlinear Theory and Applications NOLTA'95*, Vol.*1*., pp.37-42. Tokyo, Japan.

Amari, S-I.. 1996. Neural Learning in Structured Parameter Spaces , In *Neural Information Processing Systems* 96.

Amari, S-I. 1996. Information Geometry of Neural Networks --- New Bayesian Duality Theory --- , In *ICONIP'96*

Amari, S-I. 1996. Gradient Learning in Structured Parameter Spaces: Adaptive Blind Separation of Signal Sources , In *WCNN'96*

Amari, S-I. and J-F. Cardoso. 1997. Blind Source Separation - Semiparametric Statistical Approach, *submitted to IEEE Transactions on Signal Processing.*

Amari, S-I. 1997. Superefficiency in Blind Source Separation , submitted to *IEEE Transactions on Signal Processing.*

Amari, S-I. and N. Murata. 1997. Statistical Analysis of Regularization Constant - From Bayes, MDL and NIC Points of View, In *International Work-Conf. on Artificial and Natural Neural Networks* 97.

Bell A.J. & Sejnowski T.J. 1995. Fast blind separation based on information theory, in *Proceedings of the International Symposium on Nonlinear Theory and Applications*, vol. *1*, pp. 43-47, Las Vegas, Dec. 1995

Bell, A. and T.J. Sejnowski. 1995. Fast Blind Separation based on Information Theory, In *Proceedings 1995 International Symposium on Nonlinear Theory and Applications NOLTA'95*, Vo*l*., pp. 43-47. Tokyo, Japan.

Belouchrani A. and J-F. Cardoso. 1994. Maximum likelihood source separation for discrete sources. In *Proceedings EUSIPCO*, pp 768-771, Edinburgh.

Belouchrani A. and J-F. Cardoso. 1995. Maximum Likelihood Source Separation by the Expectation-Maximization Technique: Deterministic and Stochastic Implementation, In *Proceedings 1995 International Symposium on Nonlinear Theory and Applications NOLTA'95*, Vol.*1*.,pp.49-53. Tokyo, Japan.

Bregman, A.S. 1990. Auditory Scene Analysis, MIT Press, Cambridge, MA.

Cardoso, J-F. 1989. Source separation using higher order moments In *Proceedings ICASSP,* pages 2109-2112.

Cardoso, J-F. 1990. Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem. In *Proceedings ICASSP,* pages 2655-2658.

Cardoso, J-F. 1992. Iterative techniques for blind source separation using only fourth order cumulants. In *Proceedings EUSIPCO,* pages 739-742.

Cardoso, J-F. and B. Laheld. 1992. Adaptive blind source separation for channel spatial equalization. In *Proceedings of COST 229 workshop on adaptive signal processing*, pages 19-26.

Cardoso, J-F. and A. Souloumiac. 1993. An efficient technique for blind separation of complex sources. In *Proceedings IEEE Signal Processing Workshop on Higher-Order Statistics*, Lake Tahoe, USA, pages 275-279.

Cardoso, J-F. 1994. On the performance of source separation algorithms. In *Proceedings EUSIPCO*, pp 776-779, Edinburgh.

Cardoso, J-F. 1995. The equivariant approach to source separation. In *Proceedings NOLTA*, pp 55-60, 1995.

Cardoso, J-F. 1995. A tetradic decomposition of 4th-order tensors: application to the source separation problem. In M. Moonen and B. de Moor, editors, *Algorithms, architectures and applications*, volume III of SVD and signal processing, pp 375-382.

Cardoso, J-F., S.Bose, and B. Friedlander. 1995. Output cumulant matching for source separation. *In Proceedings IEEE Signal Processing Workshop on Higher-Order Statistics*, Aiguablava, Spain, pp 44-48.

Cardoso, J-F. 1995. The Invariant Approach to Source Separation, In *Proceedings 1995 International Symposium on Nonlinear Theory and Applications NOLTA'95*, Vol.*1*., pp. 55-60. Tokyo, Japan.

Cardoso, J-F. 1996. Performance and implementation of invariant source separation algorithms In *Proceedings ISCAS'96.*

Cardoso J-F., S. Bose, and B. Friedlander. 1996. On optimal source separation based on second and fourth order cumulants. In *Proceedings IEEE Workshop on SSAP,* Corfou, Greece.

Cardoso, J-F. and P. Comon. 1996. Independent component analysis, a survey of some algebraic methods. In *Proceedings ISCAS'96*, vol.2, pp. 93-96.

Cichocki, A., W. Kasprzak and S-I. Amari. 1995. Multi-Layer Neural Networks with Local Adaptive Learning Rules for Blind Separation of Source Signals, In *Proceedings 1995 International Symposium on Nonlinear Theory and Applications NOLTA'95*, Vol.*1*., pp.61-65. Tokyo, Japan.

Cichocki A. and W. Kasprzak. 1996. Nonlinear Learning Algorithms for Blind Separation of Natural Images. In *Neural Network World*, vol.6, 1996, No.4, IDG Co., Prague, 515-523.

Cichocki A., W. Kasprzak and S-I. Amari. 1996. Neural Network Approach to Blind Separation And Enhancement of Images. In *EUSIPCO'96*, Trieste, Italy.

Cichocki A., S-I. Amari, M. Adachi and W. Kasprzak. 1996. Self-Adaptive Neural Networks for Blind Separation of Sources. In *1996 IEEE International Symposium on Circuits and Systems, ISCAS'96*, Vol. 2, IEEE, Piscataway, NJ, pp. 157-160.

Deco, G. and D. Obradovic. 1995. An Information Theoretic Approach to Neural Computing. Springer-Verlag.

Duda, R.O., R.F. Lyon, and M. Slaney. 1990. Correlograms and the separation of sounds. In *Proceedings Asilomar Conference on Signals, Systems and Computers 1990.*

Ellis, D.P.W. 1991. A wavelet-based sinusoid model of sound for auditory signal separation. In *Proceedings of International Computer Music Conference 1991*. pp. 86-89.

Ellis, D.P.W. 1996. Prediction driven computational auditory scene analysis. Ph.D. thesis. EECS Department.

Girolami, M, and C. Fyfe. 1996. Blind Separation of Sources Using Exploratory Projection Pursuit Networks. In *Speech and Signal Processing, International Conference on the Engineering Applications of Neural Networks*, ISBN 952-90-7517-0, London, pp. 249-252.

Girolami, M. and C. Fyfe. 1996. Higher Order Cumulant Maximisation Using Nonlinear Hebbian and Anti-Hebbian Learning for Adaptive Blind Separation of Source Signals. In *Proceedings IWSIP-96, IEEE/IEE International Workshop on Signal and Image Processing*, Advances in Computational Intelligence, Elsevier Science, pp141 - 144, Manchester.

Girolami, M. and C. Fyfe. 1997. Multivariate Density Factorisation for Independent Component Analysis : An Unsupervised Artificial Neural Network Approach. In *AISTATS-97, 3'rd International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, Florida.

Girolami, M. and C. Fyfe. 1996. Negentropy and Kurtosis as Projection Pursuit Indices Provide Generalised ICA Algorithms, *NIPS-96 Blind Signal Separation Workshop*, (Org A. Cichocki & A.Back), Aspen, Colorado.

Girolami, M. and C. Fyfe. 1997. A Temporal Model of Linear Anti-Hebbian Learning, In *Neural Processing Letters,* Vol 4, Issue 3, Jan 1997.

Hartman, W.A. 1988. Pitch perception and the segregation and integration of auditory entities, In *Auditory Function*, G.M. Edelman, W.E. Gail and W.M. Cowan (Eds.), New York, NY, John Wiley and Sons, Inc.

Huang, J. N. Ohnishi and N. Sugie. 1995. Sound Separation Based on Perceptual Grouping of Sound Segments, In *Proceedings 1995 International Symposium on Nonlinear Theory and Applications NOLTA'95*, Vol.*1*., pp.67-72. Tokyo, Japan.

Jutten, C. and J-F. Cardoso. 1995. Separation of Sources: Really Blind ? , In *Proceedings 1995 International Symposium on Nonlinear Theory and Applications NOLTA'95*, Vol.*1*.,pp. 79-84. Tokyo, Japan.

Karhunen, J., Wang, L., and Vigario, R. 1995. Nonlinear PCA Type Approaches for Source Separation and Independent Component Analysis, In *Proceedings of the 1995 IEEE International Conference on Neural Networks (ICNN'95)*, Perth, Australia, November 27 - December 1, 1995, pp. 995-1000.

Karhunen, J., Wang, L., and Joutsensalo, J. 1995. Neural Estimation of Basis Vectors in Independent Component Analysis, In *Proceedings of the International Conference on Artificial Neural Networks ICANN-95*, Paris, France, October 9-13, 1995, pp. 317-322.

Karhunen, J. 1996. Neural Approaches to Independent Component Analysis and Source Separation. To appear in *Proc. 4th European Symposium on Artificial Neural Networks (ESANN'96)*, April 24 - 26, 1996, Bruges, Belgium.

Kasprzak W. and A. Cichocki. 1996. Hidden Image Separation From Incomplete Image Mixtures by Independent Component Analysis. *ICPR'96*, Vienna.

Laheld, B. and Cardoso, J-F. 1994. Adaptive source separation with uniform performance. In *Proceedings EUSIPCO*, pages 183-186, Edinburgh.

De Lathauwer, L., P. Comon, B. De Moor and J. Vandewalle. 1995. Higher-Order Power Method - Application in Independent Component Analysis , In *Proceedings 1995 International Symposium on Nonlinear Theory and Applications NOLTA'95*, Vol.*1*.,pp. 91-96. Tokyo, Japan.

Lee, T.W., A. Bell and R. Lambert. 1996. Blind separation of delayed and convolved sources, accepted for publication in *Advances in Neural Information Processing Systems*, MIT Press, Cambridge MA

Lee, T.W., A. Bell and R. Orglmeister. 1997. Blind Source Separation of Real World Signals, to appear in *IEEE International Conference Neural Networks*, Houston, 1997.

Matsuoka, K. and M. Kawamoto. 1995. Blind Signal Separation Based on a Mutual Information Criterion, In *Proceedings 1995 International Symposium on Nonlinear Theory and Applications NOLTA'95*, Vol.*1*.,pp. 85-91. Tokyo, Japan.

Mellinger, D.K. 1991. Event formation and separation in musical sound. Ph.D. thesis, CCRMA, Stanford University.

Oja, E., Karhunen, J., Wang, L., and Vigario, R. 1995. Principal and independent components in neural networks - recent developments. *Proc. VII Italian Workshop on Neural Nets WIRN'95*, May 18 - 20, 1995, Vietri sul Mare, Italy..

Oja, E. 1995. The nonlinear PCA learning rule and signal separation - mathematical analysis. Helsinki University of Technology, *Laboratory of Computer and Information Science*, Report A26.

Oja, E. and Taipale, O.1995. Applications of learning and intelligent systems- the Finnish technology programme. In *Proceedings of the International Conference on Artificial Neural Networks ICANN-95, Industrial Conference*, Oct. 9 - 13, 1995, Paris, France.

Oja, E. 1995. PCA, ICA, and nonlinear Hebbian learning. In *Proceedings of the International Conference on Artificial Neural Networks ICANN-95*, Oct. 9 - 13, 1995, Paris, France, pp. 89 - 94.

Oja, E. and Karhunen, J. 1995. Signal separation by nonlinear Hebbian learning. In M. Palaniswami, Y. Attikiouzel, R. Marks II, D. Fogel, and T. Fukuda (Eds.), Computational Intelligence - a Dynamic System Perspective. New York: IEEE Press, pp. 83 - 97.

Pearlmutter, B. and Lucas C. Parra. 1996. A context-sensitive generalization of ICA. In *International Conference on Neural Information Processing*. September 1996, Hong Kong.

Vercoe, B. and D. Cumming. 1988. Connection machine tracking of polyphonic audio. In *Proceedings of International Computer Music Conference 1988*. pp. 211-218.

Yang H.H. and S-I. Amari. 1996. Adaptive On-Line Learning Algorithms for Blind Separation --- Maximum Entropy and Minimum Mutual Information , accepted for publication in *Neural Computation.*

Zhu, J., X-R. Cao, and R-W Liu, Blind Source Separation Based on Output Independence - Theory and Implementation , In *Proceedings 1995 International Symposium on Nonlinear Theory and Applications NOLTA'95*, Vol.*1*.,pp. 97-102. Tokyo, Japan.

2755-95