# Fast Convolutive Blind Speech Separation via Subband Adaptation

*François Duplessis-Beaulieu*

Department of Electrical & Computer Engineering
McGill University
Montréal, Canada

October 2002

---

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering.

# Abstract

*Blind source separation* (BSS) attempts to recover a set of statistically independent sources from a set of mixtures knowing only the structure of the mixing network, and the hypothesized probability distribution function of the sources. The case where the sources are immobile persons speaking in a reverberant room is of particular interest, because it represents a first step toward unlocking the so-called "cocktail party problem". Due to the reverberations, BSS in the time domain is usually expensive in terms of computations, but the number of computations can be significantly decreased if separation is carried out in subbands.

An implementation of a subband-based BSS system using DFT filter banks is described, and an adaptive algorithm tailored for subband separation is developed. Aliasing present in the filter bank (due to the non-ideal frequency response of the filters) is reduced by using an oversampled scheme. Experiments, conducted with two-input two-output BSS systems, using both subband and fullband adaptation, indicate that separation and distortion rates are similar for both systems. However, the proposed 32-subband system is approximately 10 times computationally faster than the fullband system.

## Résumé

La *séparation aveugle de sources* (SAS) consiste à récupérer un ensemble de sources statistiquement indépendantes d'un ensemble de mixtures en connaissant uniquement la structure du réseau de mélange et la fonction de distribution hypothétique des sources. Le cas où les sources sont des gens immobiles qui parlent dans une pièce réverbérante s'avère intéressant, car cela représente un premier pas vers la résolution du problème de *cocktail party*. À cause des réverbérations, la SAS dans le domaine du temps est habituellement coûteuse en termes de calculs, mais le nombre de ceux-ci peut être significativement réduit si la séparation est effectuée en sous-bandes.

Une implémentation d'un système de SAS en sous-bandes en utilisant des bancs de filtres de type DFT est décrite et un algorithme adaptatif pour la séparation en sous-bandes est développé. Le recouvrement spectral présent dans le banc de filtres (causé par la réponse fréquencielle non idéale des filtres) est réduit en utilisant un échantillonnage non critique. Des expériences, effectuées avec des systèmes de SAS à deux entrées et à deux sorties, en utilisant l'adaptation en sous-bandes et en pleine bande, indiquent que les taux de séparation et de distorsion sont similaires pour les deux systèmes. Par contre, le système à 32 sous-bandes se révèle être approximativement 10 fois plus rapide en temps de calculs que le système en pleine bande.

## Acknowledgments

The completion of this thesis would not have been possible without the valuable advice and direction of my supervisor, Prof. Benoît Champagne. Furthermore, I would like to thank him and *le Fonds québécois de la recherche sur la nature et les technologies* (previously known as FCAR) for providing financial support to complete this research.

I am grateful to my fellow graduate students, Benoît, Mark, Paxton, Xiaojian, Dorothy, Aziz and others, in the Telecommunications and Signal Processing Laboratory. Their companionship was very much appreciated, and they all contributed to create a pleasant work environment.

Finally, I would like to thank my family for their love and support. They have always encouraged me to continue my studies, and I am forever indebted to them. To my mother, France, my father, Laurent, and my sister, Isabelle, thank you.

# Contents

**1 Introduction** **1**
1.1 Overview of the Blind Source Separation Problem . . . . . 1
1.2 Applications of BSS . . . . . 3
1.3 Historical Perspective . . . . . 5
1.4 Thesis Contribution . . . . . 6
1.5 Thesis Organization . . . . . 7

**2 Blind Separation of Sources** **9**
2.1 Problem Statement . . . . . 9
2.1.1 Instantaneous Mixing . . . . . 10
2.1.2 Convolutive Mixing . . . . . 11
2.2 Separation Criteria . . . . . 12
2.2.1 Maximum Likelihood Estimation . . . . . 13
2.2.2 Information Theoretic Approach . . . . . 16
2.3 Adaptive Algorithms for BSS . . . . . 19
2.3.1 Instantaneous Mixtures . . . . . 19
2.3.2 Convolutive Mixtures . . . . . 21
2.3.3 Activation Functions . . . . . 26
2.4 Chapter Summary . . . . . 29

**3 DFT Filter Banks for Subband Adaptation** **30**
3.1 Blind Source Separation in Subbands . . . . . 30
3.2 DFT Filter Banks . . . . . 33
3.3 Decimation and Interpolation . . . . . 37
3.3.1 Decimation . . . . . 37

3.3.2 Interpolation . . . 40
3.4 Prototype Filter Design . . . 41
3.4.1 Cancellation of Phase Distortion . . . 43
3.4.2 Interpolation of Quadrature Mirror Filters . . . 44
3.5 Implementation Using the Weighted Overlap-Add Method . . . 47
3.6 Chapter Summary . . . 49

**4 Blind Separation Using Subband Adaptation 50**
4.1 Aliasing in the Subband-Based BSS System . . . 50
4.1.1 Derivation of an Expression for Aliasing . . . 51
4.1.2 Possible Solutions to Eliminate Aliasing . . . 53
4.2 Adaptive Algorithm for Subband BSS . . . 54
4.2.1 Probability Distribution Function of Bandpass Speech . . . 54
4.2.2 Adaptation Equation . . . 57
4.2.3 Source Permutations . . . 60
4.3 Computational Complexity . . . 60
4.4 Chapter Summary . . . 63

**5 Experimental Results 64**
5.1 Methodology . . . 64
5.1.1 Performance Measures . . . 64
5.1.2 Data Generation . . . 67
5.1.3 Simulation Programming . . . 69
5.2 Results and Discussion . . . 70
5.2.1 Performance of Fullband and Subband BSS Systems . . . 70
5.2.2 Distortions in Subband-Based Systems . . . 72
5.2.3 Subjective Testing . . . 73
5.3 Chapter Summary . . . 74

**6 Conclusion 75**
6.1 Summary of Our Work . . . 75
6.2 Future Work . . . 77
6.2.1 Source Permutation Problem . . . 77
6.2.2 Automatic Step Size Adjustment . . . 78

6.2.3 Moving Speakers . . . . . 78

**References** **79**

# List of Figures


1.1 BSS in a room with two speakers. . . . . 2
1.2 Inner working of BSS. . . . . 3
1.3 Four separated speech sources. . . . . 4

2.1 Feedforward convolutive mixing model with two sources. . . . . 12
2.2 Feedforward blind separation system proposed by Bell and Sejnowski (1995). 16
2.3 Feedforward convolutive de-mixing network with two sources. . . . . 23
2.4 Feedback convolutive de-mixing network with two sources. . . . . 25
2.5 Gaussian, super-Gaussian and sub-Gaussian pdf's. . . . . 27

3.1 A subband-based BSS system ($N' = N - 1$ and $K' = K - 1$). . . . . 31
3.2 Non-ideal filters used for subband analysis and synthesis in a two-subband system. . . . . 32
3.3 Uniform DFT filter banks (for analysis and synthesis). . . . . 34
3.4 Complex modulation and decimation in an analysis DFT filter bank. . . . 35
3.5 Subband channels for $K = 8$. . . . . 36
3.6 A decimator. . . . . 37
3.7 Derivation of an efficient structure for decimation using an FIR filter. . . . 39
3.8 An interpolator. . . . . 40
3.9 Derivation of an efficient structure for interpolation using an FIR filter. . . 42
3.10 DFT interpolation of a QMF. . . . . 46
3.11 Comparison between prototype filters generated using traditional and DFT interpolation. . . . . 47
3.12 Prototype filters for 16 and 32-subband BSS systems. . . . . 48

4.1 Feedback convolutive de-mixing networks used in a two-subband BSS system. 51

4.2 Histogram illustrating the experimentally observed phase distribution of bandpass speech samples. . . . . 55
4.3 Experimentally observed magnitude distribution of bandpass speech samples. 56

5.1 Mixing and de-mixing systems. . . . . 65
5.2 Speaker positions in the recording room. . . . . 68
5.3 Distortion in the subband systems. . . . . 72

# List of Tables

4.1 Output generation and weight update equations for subband BSS. . . . . 60
4.2 Some symbols and their description. . . . . 61
4.3 Number of real multiplications needed for subband BSS without a WOA realization. . . . . 62
4.4 Number of real multiplications needed for subband BSS with a WOA realization. . . . . 62
4.5 Comparison between the number of multiplications required for fullband BSS and subband BSS (16 and 32-subband implementations). . . . . 63
4.6 Computational gains for a 16 and a 32-subband systems. . . . . 63

5.1 Equipment used for data recording. . . . . 69
5.2 Performance of BSS systems. . . . . 70
5.3 Computational speed and time gains of two-input two-output BSS systems. 71
5.4 Number of listeners (out of 4) who found that the processed file was the most separated. . . . . 73

# Chapter 1

# Introduction

In this chapter, the problem of blind source separation is first introduced in general terms. Possible real-world applications that result from solving this problem are then discussed. A brief historical survey follows, and original work presented in this thesis is outlined. Finally, a synopsis of this thesis along with a list of mathematical notations end the chapter.

## 1.1 Overview of the Blind Source Separation Problem

In this thesis, we are concerned with the *blind source separation* (BSS) problem. In this context, a "source" is a signal produced by a physical phenomenon, such as speech uttered in a conversation between two persons, the electrical activity coming from a person's brain, or the music played by mixing each instrument in a concert. "Source separation" refers to the task where one is trying to recover the original sources when several mixtures of these sources are observable. "Blind source separation" means that separation is attempted without knowing the physical realizations of the original sources, and how they were mixed together in the first place.

To clarify these ideas, let us consider the situation depicted in Fig. 1.1, which illustrates two persons speaking in a room, with respective speech signal $s_1(t)$ and $s_2(t)$. After propagation within the room, these signals are captured by two microphones, which pick up two combinations — or two mixtures — of both speakers, denoted $x_1(t)$ and $x_2(t)$. For some reasons, it would be very desirable to have a device that automatically isolates one speaker from the other, simply by processing the speech mixtures collected by the microphones. The process by which we obtain the contribution of each speaker, denoted $y_1(t)$ and $y_2(t)$,

from several mixtures is a particular example of what BSS aim to do. We are primarily interested in speech separation in this thesis, but most of the ideas exposed here can easily be extended to other signals and situations as well.

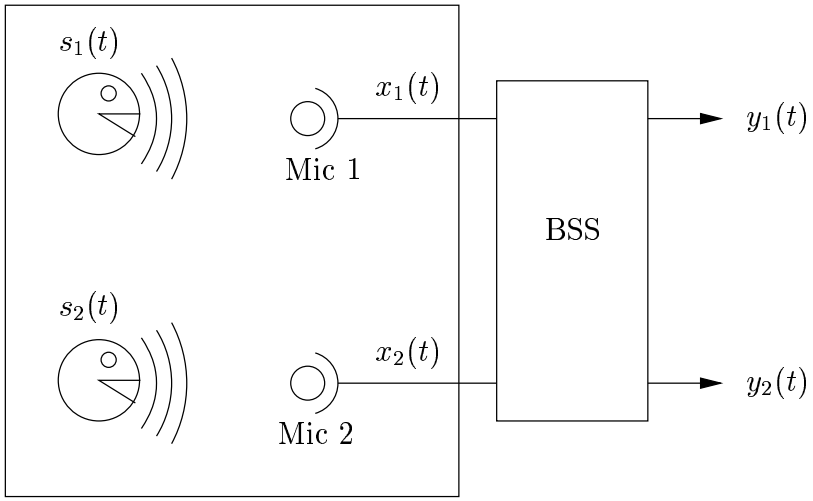



**Fig. 1.1** BSS in a room with two speakers.

Succinctly, BSS attempts to recover a set of sources from a set of mixtures. To separate the sources, there is one simple, but essential, condition that the sources must adhere to. BSS utilizes the *statistical independence* of the sources to recover them, that is, each source must not be statistically related to the others. We assume that this condition holds when people are speaking in the scenario depicted above, even if this might not be entirely true. Classically, BSS employs the scheme illustrated in Fig. 1.2. First, a measure of independence is defined. According to this measurement, parameters of a de-mixing network are adapted so that the outputs become as independent as possible. Since the sources are independent, they can ideally be recovered simply by restoring statistical independence at the end of the de-mixing network.

BSS is a particularly powerful technique because it necessitates very few assumptions. In theory, only statistical independence of the sources and a model of the mixing phenomenon are needed. In addition, a maximum of one Gaussian distributed source must be assumed. But in practice, knowing the number of sources and their statistical properties greatly helps to simplify derivations and computations.

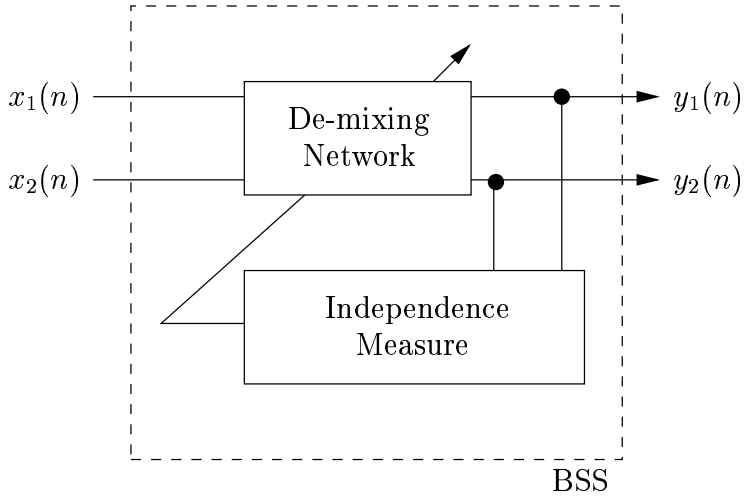



**Fig. 1.2** Inner working of BSS.

## 1.2 Applications of BSS

In this section, a few applications that make use of BSS are described. Even if this thesis focuses on speech applications, it is worth mentioning that one of the most successful applications of BSS has been reported in the biomedical signal processing field. In fact, BSS has proven to be an excellent analysis tool for electroencephalographic (EEG) data. EEG signals are collected by several electrodes placed on the scalp that measure electrical brain activity. Data can be modelled as several superposed noise sources (eye blinks, muscle noise, cardiac contamination, line noise) and brain activity (alpha and theta bursts, for example) [1], [2]. Identifying and removing noise sources from brain activity facilitates the interpretation of EEG recordings by medical specialists. Similarly, functional magnetic resonance images (fMRI) can be cleaned up with BSS [1], [2].

Contrary to EEG data, the mixing model in speech applications is far more complex. Due to the reverberant nature of a room, where sound waves are reflected off various surfaces, like walls and furniture, multiple scaled and delayed versions of each source are involved. Separation is thus more difficult. Isolating speech signals coming from a room is a classic signal processing problem known as the "cocktail party problem" [3]. The cocktail party problem refers to the difficulty an automated device, even a very sophisticated one, can extract a specific voice in an environment similar to a cocktail party, where several conversations, noise and music are heard at the same time. As an example of BSS appli-

cation, Fig. 1.3(a) illustrates the waveforms of four mixed speech sources recorded in an ideal and quiet environment; Fig. 1.3(b) shows the separated sources after BSS processing, as described in Chap. 2.

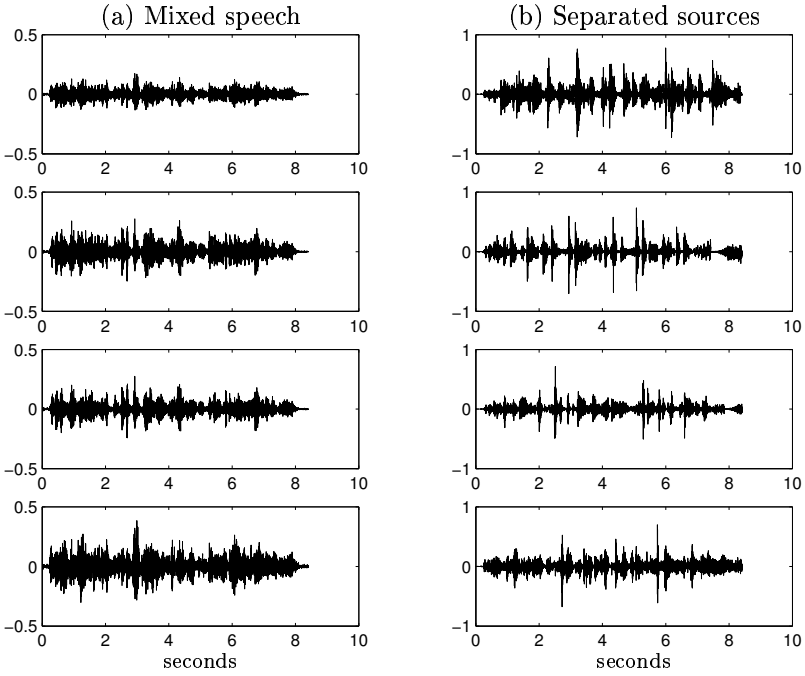



**Fig. 1.3** Four separated speech sources.

Using BSS to solve the cocktail party problem would be useful for many audio applications. In teleconferencing applications, it is very desirable to separate and transmit the contribution of each local speakers [4]. BSS can also be used in this context to cancel the acoustical echoes, that is to eliminate the effect of the loudspeaker and microphone couplings [5]. Moreover, automatic speech recognition systems have always performed poorly in a environment where one's voice is corrupted by other audio sources. Using BSS to enhance a specific voice improves the recognition rate tremendously [6]. Speech enhancement is also very desirable for cellular phones, since speech coders do not perform well in the presence of noise components. However, one quickly realizes that the number of noise sources can be very big, and BSS cannot be applied straightforwardly [7]. Hearing aids are another application that can benefit from BSS [7]. Some people suffering from a severe hearing impairment cannot distinguish one specific voice from the others, especially

in a room where many loud conversations occur simultaneously. Implementing BSS in an hearing device would allow these people to follow a specific conversation more easily. Note that there remain many challenges to be solved before BSS is implemented in a successful commercial product. Some of these challenges will be described later in this thesis.

BSS can also be applied to telecommunication problems. For instance, if we consider CDMA (Code Division Multiple Access), a data transmission method based on spread spectrum techniques used by second (IS-95) and third generation (CDMA 2000, UTRA FDD) mobile phones, BSS can be employed to separate convolved mixtures for symbol detection [8]. Indeed, BSS in a mobile system can help in extracting the data stream belonging to a particular phone from those intended to other phones.

## 1.3 Historical Perspective

We give in this section a short historical review of BSS. A more complete historical survey can be found in [9]. Around 1986, Jutten and Hérault were the firsts to look into the blind source separation problem [10]. They proposed a working separating network, but their adaptation rule was derived using an ad hoc procedure. Since separation relied on the independence of the sources, Jutten and Hérault coined the term *independent component analysis* (ICA) to describe their approach, because it conveys ideas similar to principal component analysis (PCA). However, the goal of PCA is to transform correlated components into uncorrelated ones, whereas ICA is concerned with statistically independent components. Using the new framework established by Jutten and Hérault, Comon published in 1994 a detailed mathematical study of the BSS problem [11]. Comon proposed cost functions based on higher order statistics to adapt the separating network.

In 1995, Bell and Sejnowski merged two lines of research together: the use of higher order statistics for blind separation, and information theoretic unsupervised learning rules used in neural networks. As a result, Bell and Sejnowski came up with the information-maximization (or "infomax") principle [12]. Briefly, the infomax principle states that, under certain conditions, independent outputs can be obtained by maximizing the joint entropy. BSS was now cast in an information theoretic framework, which led to elegant cost functions and adaptation rules. An important milestone in the short history of BSS was reached. Amari *et al.* [13] found in 1996 a better adaptation scheme by applying the "natural" gradient to the cost function proposed by Bell and Sejnowski. Adaptation speed

was significantly improved. About at the same time, Torkkola used the infomax principle and the natural gradient to derive a de-mixing network capable of separating speech in a reverberant environment [14].

Meanwhile, Cardoso and Laheld independently proposed in 1996 a blind separation algorithm based on maximum likelihood estimation [15]. The "new" algorithm closely resembles the adaptation rule of Bell and Sejnowski combined with Amari's natural gradient. Cardoso and Laheld also analyzed the stability of their adaptation rule. Cardoso then showed in 1997 that Bell and Sejnowski's information theoretic approach was equivalent to maximum likelihood estimation [16].

In 1999, Lee combined the architecture of Bell and Sejnowski's de-mixing network, the natural gradient of Amari, and the stability analysis of Cardoso and Laheld to derive the extended infomax algorithm [1]. Unlike the algorithms proposed by Bell and Sejnowski, the extended infomax algorithm is capable of separating a wide variety of sources, regardless of their statistics.

More recently, research efforts have focused on practical considerations and on real-world applications of BSS. For example, time-varying environments [17], [18] and blind processing with a different number of microphones than sources [19], [20] have been considered.

## 1.4 Thesis Contribution

If BSS is undertaken in a room, as illustrated in Fig. 1.1, the de-mixing network will require many parameters to take into account the many reflections in a reverberant environment. Adapting all these parameters is very expensive in terms of computational resources, and real-time separation may become impossible.

The goal of this thesis is to reduce the number of computations necessary to separate sources in a reverberant environment. For reasons detailed later in this thesis, the number of computations can be significantly reduced if the parameters are adapted in subbands. Subband adaptation differs from conventional (fullband) adaptation as follows. If subband adaptation is implemented, the signal is divided into several adjacent frequency intervals, also called subbands. Each interval has its own de-mixing network that is adapted independently. Once the signals in each frequency interval have been processed, they are combined together so that the recovered sources are obtained. The computational efficiency of this approach results from the possibility of reducing the sampling rate within each subband.

In this thesis, subband decomposition is carried out via a DFT filter bank. Adaptation rules used in the literature are modified for subband signals. The resulting BSS system is evaluated according to three criteria:

- Distortion of the resulting signal,
- Signal separation level achieved,
- Computational complexity.

Compared to fullband BSS systems, the subband systems developed in this thesis exhibit about the same distortion and separation rates. However, a significant improvement in terms of computational speed can be noted. For instance, the proposed 32-subband BSS system is more than 10 times faster than the fullband system for a two-input two-output scenario.

To the best of our knowledge, there does not exist in the literature a complete study of a subband-based BSS implementation. This thesis attempts to remedy the lack of information in that domain.

## 1.5 Thesis Organization

This thesis is organized as follows.

Chapter 2 contains a detailed literature review of BSS for real sources. Two different approaches to BSS are described: maximum likelihood estimation and the infomax principle. Adaptation equations for the feedforward and the feedback de-mixing networks are derived. This chapter ends with a discussion on the extended infomax algorithm.

Chapter 3 introduces the concept of subband adaptation for BSS. In particular, the DFT filter banks used for subband analysis and synthesis are described in details.

Chapter 4 describes how the BSS framework presented in Chap. 2 can be recast for subband adaptation. Adaptation equations previously derived are modified for subband operation, and a mathematical analysis of the distortion introduced by the filter banks is given.

Chapter 5 discusses various testing strategies to assess the performance of BSS systems. Fullband and subband BSS systems are compared according to several criteria, such as distortion rate, separation quality, and computational complexity.

Finally, Chapter 6 summarizes the results presented in this thesis, and some ideas for future research are proposed.

The following notations are used throughout the thesis. The superscripts $T$ and $H$ respectively stand for the transpose and the Hermitian transpose of a vector or a matrix. All vectors considered in this thesis are column vectors, and are denoted by lower case bold letters, e.g. $\boldsymbol{x}$. We reserve the use of upper case bold letters for matrices, e.g. $\boldsymbol{A}$. In addition, $\Re\{z\}$ and $\Im\{z\}$ denote the real and imaginary part of $z$, respectively. The expression $E[x]$ represents the expectation of the random variable $x$. Lastly, we use the symbols $\mathbb{R}$ and $\mathbb{Z}$ to denote the set of all real and integer numbers, respectively.

# Chapter 2

# Blind Separation of Sources

In this chapter, a detailed literature review on blind source separation is given. We mainly focus on speech sources. Two mixing models are considered: a simple mixing model that does not take into account propagation delays and sound wave reflections on furniture and walls (instantaneous mixing), and a more complex model that incorporates such effects (convolutive mixing). In this chapter, we use the former model to introduce new concepts in a simple manner, whereas the latter model is used to develop the concepts in a more general and realistic framework. This chapter is divided into four sections. It begins with a mathematical formulation of the BSS problem. Two popular blind separation criteria are then presented. According to these criteria, adaptive algorithms capable of separating sources from mixtures are derived. Lastly, the concepts presented in this chapter are summarized.

## 2.1 Problem Statement

In order to derive a blind separation algorithm, the mixing process must be modelled appropriately. We will consider two models: the first (and the simplest) illustrates an instantaneous mixing situation. The second model is more elaborate and depicts a convolutive mixing scenario. Let us consider $N$ sources $s_i(n), i = 1, \ldots, N$, modelled as independent zero-mean random processes with at most one of them Gaussian distributed. As pointed out later in this chapter, the restriction on the number of Gaussian sources stems from the fact that the separation criterion attempts to exploit non-Gaussianity of the sources.

In this thesis, we are interested in the particular case where the number of sources $N$

is the same as the number of mixtures (or sensors) $M$, i.e. $N = M$. Usually, in a real world application, the number of sensors remains fixed, whereas the number of sources varies. Two alternative situations can occur: there are less sources than sensors ($N < M$), or more sources than sensors ($N > M$). If $N < M$, some sensors contain redundant information. It is possible to exploit this additional information to improve separation quality [20]. Traditionally, dimensionality is reduced by projecting the data on a subspace. On the other hand, if $M < N$, there is a lack of information, and BSS cannot be expressed in terms of a precise mathematical solution. Nevertheless, BSS is still possible in this case. For instance, a maximum a posteriori (MAP) estimator has recently been derived by exploiting sparsity of the sources. However, it was reported that speech mixtures do not normally satisfy the sparsity condition in the time domain [19].

### 2.1.1 Instantaneous Mixing

The instantaneous mixing model is described in this section. Let us consider the situation where the sources are picked up by an array of sensors $\boldsymbol{x}$ as modelled by

$$\boldsymbol{x} = \boldsymbol{A}_* \boldsymbol{s}, \tag{2.1}$$

where $\boldsymbol{A}_*$ is an unknown $N \times N$ mixing matrix, $\boldsymbol{s} = [s_1, \ldots, s_N]^T$ is a $N$-vector of source components $s_i$, and $\boldsymbol{x} = [x_1, \ldots, x_N]^T$ is a $N$-vector of mixture components $x_i$. In this chapter, all quantities of interest are assumed to be real valued, i.e. $s_i \in \mathbb{R}$, $x_i \in \mathbb{R}$, etc. Since the matrix multiplication combines the sources with their present values only, Eq. 2.1 can be used to model instantaneous mixing. Physically, instantaneous mixing could represent immobile speakers in a noise-free non-reverberant room where the speed of sound is infinite. Of course, such a mixing model does not represent a real room environment, but it is nonetheless convenient to consider a simple model first.

The goal of BSS is to find a linear transform, represented by a $N \times N$ transformation matrix $\boldsymbol{W}$, such that

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} = \boldsymbol{W}\boldsymbol{A}_*\boldsymbol{s} \tag{2.2}$$

is a good estimate of the sources $\boldsymbol{s}$. The matrix $\boldsymbol{A}_*$ must be non-singular so that it can be inverted to recover $\boldsymbol{s}$ from $\boldsymbol{x}$. However, without any further information, it is impossible

to find a systematic way of perfectly inverting the system, i.e.

$$\boldsymbol{W}\boldsymbol{A}_* = \boldsymbol{I} \Rightarrow \boldsymbol{W} = \boldsymbol{A}_*^{-1}, \tag{2.3}$$

where $\boldsymbol{I}$ is the $N \times N$ identity matrix. Since the observations are obtained by a product of two unknowns (namely, $\boldsymbol{A}_*$ and $\boldsymbol{s}$), the original energy level of each source cannot be restored. In other words, we cannot distinguish the energy of the sources from the energy introduced by the mixing matrix. Hence, the amplitude of the recovered sources will be scaled by an arbitrary factor. In addition, the signals in the recovered vector $\boldsymbol{y}$ may be permuted with respect to those in $\boldsymbol{s}$. When the mixing and the de-mixing matrices are multiplied together, the result can be decomposed as follows:

$$\boldsymbol{C} = \boldsymbol{P}\boldsymbol{D} = \boldsymbol{W}\boldsymbol{A}_*, \tag{2.4}$$

where $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{D}$ is a diagonal scaling matrix [21]. A perfect separation occurs when $\boldsymbol{C}$ contains only one non-zero element in each row and in each column. We refer to such a matrix as a non-mixing matrix.

### 2.1.2 Convolutive Mixing

To implement BSS in a real acoustic environment, one must consider the reverberant nature of a room. Instantaneous BSS cannot be used in this context, since sensors would pick up multiple delayed and scaled copies of the sources. Thus, we need a more elaborate model if we want to be able to separate speech signals recorded in a real room.

Many acoustic environments can be modelled as a feedforward mixing network, as illustrated in Fig. 2.1. In this figure, each branch of the network contains a filter. Mathematically, in discrete-time, the convolutive mixing model can be expressed as

$$x_i(n) = \sum_{j=1}^{N} \boldsymbol{a}_{ij}^T \boldsymbol{s}_j(n) \quad \text{for } 1 \leq i \leq N, \tag{2.5}$$

where $\boldsymbol{a}_{ij} = [a_{ij0}, \ldots, a_{ij(L-1)}]^T$ is a $L$-vector representing the impulse response from source $j$ to sensor $i$, and $\boldsymbol{s}_j(n) = [s_j(n), \ldots, s_j(n-L+1)]^T$ is a vector that comprises the $L$ past values taken by source $j$. In theory, the impulse response of a room can be infinite, but it can be well approximated using only a finite number of taps if $L$ is large

enough.

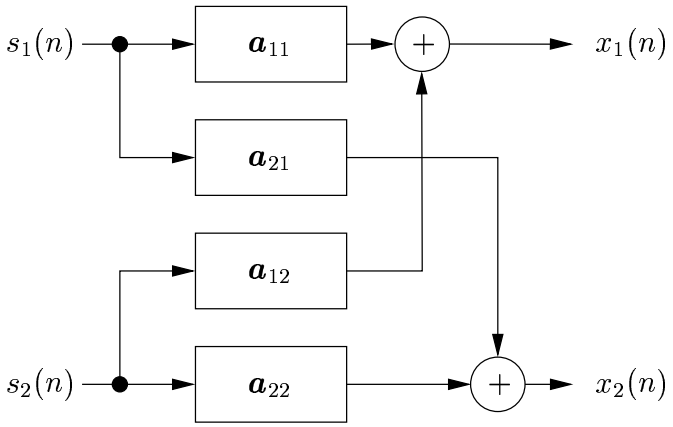



**Fig. 2.1** Feedforward convolutive mixing model with two sources.

The goal of convolutive BSS is to find a non-instantaneous linear transformation $\mathcal{T}_i$ that operates on $\boldsymbol{x}_1(n), \ldots, \boldsymbol{x}_N(n)$, where $\boldsymbol{x}_i(n) = [x_i(n), \ldots, x_i(n-L+1)]^T$, such that

$$y_i(n) = \mathcal{T}_i(\boldsymbol{x}_1(n), \ldots, \boldsymbol{x}_N(n)) \text{ for } 1 \leq i \leq N \tag{2.6}$$

is a good approximation of $s_i(n)$. Note that, depending on the separation scheme which is used, $y_i(n)$ may be a delayed version of $s_i(n)$. The transformation $\mathcal{T}_i$ can be a linear feedforward or feedback network. Similarly to the instantaneous case, the recovered sources may be permuted with respect to the original sources. Moreover, each recovered source can be corrupted by a gain and a filtering ambiguity. It is important to note that BSS itself does not dereverberate the signal. In general, non-causal filters are needed to invert the system in Fig. 2.1. Whenever the echo is stronger then the direct path, the system may become non-minimum phase, and, for the inverse to remain stable, one must use non-causal filters [22].

## 2.2 Separation Criteria

In this section, two separation criteria are described. Each of these criteria leads to a cost function which measures statistical independence between the output components of

a blind separation system. Parameters of the separation system are then adapted so that the cost function is maximized.

Referring to the instantaneous model of Eq. 2.1, provided that $\boldsymbol{s}$ contains independent components, and that at most one of them is Gaussian distributed, Darmois's theorem (stated in [11]) ensures us that if the output components $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{s}$ are independent, then $\boldsymbol{C}$ is a non-mixing matrix [23]. In other words, if independence is restored via a linear transformation, then $\boldsymbol{y}$ must be a scaled and permuted version of $\boldsymbol{s}$. This theorem is of utmost importance, because it means that BSS is possible by recovering statistical independence. Independent component analysis (ICA) thus provides useful analysis tools for BSS. ICA consists in finding a linear transformation such that components of a vector become as independent as possible, in the sense that a function measuring independence is maximized.

### 2.2.1 Maximum Likelihood Estimation

In this section, we develop a cost function for BSS using maximum likelihood (ML) estimation. A cost function suitable for BSS, denoted $\phi(\cdot)$, is a real-valued function which exhibits the following behaviour: when maximized, source separation is achieved. Hence, we have $\phi(\boldsymbol{C}\boldsymbol{s}) \leq \phi(\boldsymbol{s})$ with equality only if $\boldsymbol{C}\boldsymbol{s}$ is a scaled and permuted version of $\boldsymbol{s}$ [23]. This section begins with a short recapitulation of ML estimation. We then show how ML estimation can be used for BSS.

The objective of ML estimation is to find an unknown, but in this case non-random, parameter $\theta$ (possibly multi-dimensional) such that the normalized log-likelihood function is maximum. Let us suppose that we observe $T$ independent realizations $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_T$ of a random vector $\boldsymbol{r}$. The normalized log-likelihood function can be expressed as

$$L_T(\theta) = \frac{1}{T} \sum_{t=1}^{T} \log f(\boldsymbol{r}_t; \theta), \tag{2.7}$$

where $f(\cdot\,; \theta)$ denotes the probability density function (pdf) of $\boldsymbol{r}$ parametrized by $\theta$. The function $L_T(\cdot)$ indicates how "likely" the observations $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_T$ were produced from $f(\cdot\,; \theta)$. As the number of observations approaches infinity, then according to the law of large numbers, $L_T(\theta)$ converges in probability to its mean. If we denote $f_*(\cdot)$ as the true pdf of $\boldsymbol{r}$,

we can write

$$L_T(\theta) \overset{T\to\infty}{\longrightarrow} L(\theta) = \int f_*(\boldsymbol{r}) \log f(\boldsymbol{r};\theta) d\boldsymbol{r}, \tag{2.8}$$

which is the likelihood equation.

In the context of instantaneous BSS, ML estimation can be used in the following way (generalization to convolutive BSS is very straightforward as it will be shown later). Firstly, let us suppose that $q(\cdot)$ denotes the *hypothesized* pdf of the sources, and define $\tilde{\boldsymbol{s}} = [\tilde{s}_1, \ldots, \tilde{s}_N]^T$ as the hypothesized source vector. Since the sources are independent, the pdf of $\tilde{\boldsymbol{s}}$ can be expressed as follows:

$$q(\tilde{\boldsymbol{s}}) = \prod_{i=1}^{N} q_i(\tilde{s}_i). \tag{2.9}$$

In this context, the true pdf of $\boldsymbol{r}$, $f_*(\cdot)$, becomes the true pdf of $\boldsymbol{x}$. The parameter to be estimated $\theta$ is the unknown mixing matrix $\boldsymbol{A}$. Since the input components are related to the source components by the relation $\boldsymbol{x} = \boldsymbol{A}_*\boldsymbol{s}$, the parametric pdf of $\boldsymbol{x}$ depends on two factors: the unknown mixing matrix $\boldsymbol{A}$ and the hypothesized pdf of the sources $q(\cdot)$. Thus, the parametric pdf of $\boldsymbol{x}$ is denoted by $f(\cdot\,;\boldsymbol{A}, q)$. As a result, when applied to the BSS problem, Eq. 2.8, the likelihood equation, becomes [16]

$$L(\boldsymbol{A}) = \int f_*(\boldsymbol{x}) \log f(\boldsymbol{x};\boldsymbol{A}, q) d\boldsymbol{x}. \tag{2.10}$$

At this point, we need to define two quantities in order to develop Eq. 2.10 further: the joint entropy and the Kullback-Leibler divergence.

**The joint entropy** of a pdf $f$ of a random vector $\boldsymbol{x}$, denoted $H(f)$, is defined as

$$H(f) = -\int f(\boldsymbol{x}) \log f(\boldsymbol{x}) d\boldsymbol{x} = -E[\log f(\boldsymbol{x})]. \tag{2.11}$$

**The Kullback-Leibler divergence** between pdf's $f$ and $g$, denoted $K(f\|g)$, is defined as

$$K(f\|g) = \int f(\boldsymbol{x}) \log \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} d\boldsymbol{x}. \tag{2.12}$$

A convenient abuse of notation is to use $K(\boldsymbol{x}\|\boldsymbol{y})$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are random vectors, to denote the Kullback-Leibler divergence between the pdf of $\boldsymbol{x}$ and $\boldsymbol{y}$. Likewise, we often

use $H(\boldsymbol{x})$ to denote the entropy of the pdf of $\boldsymbol{x}$. Intuitively, the joint entropy quantifies the "randomness" of a random vector, and the Kullback-Leibler divergence measures the "closeness" of two pdf's. By letting

$$f(\boldsymbol{x};\boldsymbol{A},q) = \frac{f(\boldsymbol{x};\boldsymbol{A},q)}{f_*(\boldsymbol{x})} f_*(\boldsymbol{x}) \tag{2.13}$$

in Eq. 2.10, we obtain

$$\begin{aligned} L(A) &= \int f_*(\boldsymbol{x}) \log \frac{f(\boldsymbol{x};\boldsymbol{A},q)}{f_*(\boldsymbol{x})} d\boldsymbol{x} + \int f_*(\boldsymbol{x}) \log f_*(\boldsymbol{x}) d\boldsymbol{x} \\ &= -K(\boldsymbol{A}_*\boldsymbol{s} \| \boldsymbol{A}\tilde{\boldsymbol{s}}) - H(\boldsymbol{A}_*\boldsymbol{s}). \end{aligned} \tag{2.14}$$

The second term of Eq. 2.14 is a constant (it does not depend on $\boldsymbol{A}$) and may be dropped. Therefore, the cost function associated with ML estimation is

$$\begin{aligned} \phi_{ML}(\boldsymbol{y}) &= -K(\boldsymbol{A}_*\boldsymbol{s} \| \boldsymbol{A}\tilde{\boldsymbol{s}}) \\ &= -K(\boldsymbol{A}^{-1}\boldsymbol{x} \| \tilde{\boldsymbol{s}}) \\ &= -K(\boldsymbol{y} \| \tilde{\boldsymbol{s}}), \end{aligned} \tag{2.15}$$

where the second equality stems from the fact that the Kullback-Leibler divergence remains unchanged when an invertible transformation is applied to the sample space [16]. Upon maximization of the cost function, the distance[1] between the pdf of the output components of the separation system and the hypothesized pdf of the sources is minimized.

As mentioned in Sec. 2.1, we limit the number of Gaussian distributed sources to one. Equation 2.15 justifies this condition. Since a linear combination of two Gaussian sources results in a Gaussian output, the Kullback-Leibler divergence between the hypothesized pdf of the source and that of the output would be minimized even if the sources are not separated. Hence, the cost function cannot separate a mixture of Gaussian sources.

In the next section, we will derive another cost function using a completely different approach. The latter relies on information theoretic quantities, and is very popular in the literature.

[1] The Kullback-Leibler divergence is not really a distance, since it is not symmetric.

### 2.2.2 Information Theoretic Approach

In 1995, Bell and Sejnowski proposed a new approach to BSS. They suggested to use information theoretic measurements to guide the separation. Their new approach, which has been named "information maximization" principle (or "infomax" principle), has been very well-received by the BSS research community. But, as shown in this section, in the end, infomax boils down to nothing more than ML estimation.

The idea behind the infomax principle is simple: if mutual information shared by $N$ components is zero, then the components are statistically independent. Let us consider the output of a simple feedforward blind separation system, as illustrated in Fig. 2.2. As reported by Bell and Sejnowski, under certain conditions, maximizing the joint entropy of the output components can approximately minimize the mutual information [12]. However, maximization of the entropy of $\boldsymbol{y}$ is problematic, because it may diverge to infinity [16]. Thus, a nonlinear function $p : \mathbb{R}^N \rightarrow (a, b)^N$ is introduced, where $(a, b)$ here denotes the interval of the real line between points $a$ and $b$. This nonlinear function maps the $N$-dimensional real plane to a $N$-dimensional finite plane, and is an invertible component-wise monotonously increasing function. It is convenient to define $p_i(y_i), 1 \leq i \leq N$, as the $i$-th component of $p(\boldsymbol{y})$.

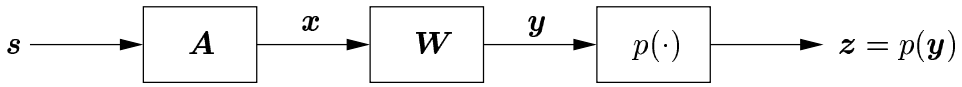



**Fig. 2.2** Feedforward blind separation system proposed by Bell and Sejnowski (1995).

Hence, maximizing the joint entropy of $\boldsymbol{z} = p(\boldsymbol{W}\boldsymbol{x})$ with respect to $\boldsymbol{W}$ is the course of action the infomax principle proposes. In order to gain more insights about the infomax principle, let us rewrite the joint entropy in terms of the sum of the marginal entropies and the mutual information $I(\cdot)$. The mutual information is defined as the Kullback-Leibler divergence between the pdf of $\boldsymbol{z}$ and the marginal pdf of the components of $\boldsymbol{z}$, i.e.

$$I(\boldsymbol{z}) = K(\boldsymbol{z} \| z_1 \dots z_N). \tag{2.16}$$

Hence, we can write

$$H(\boldsymbol{z}) = \sum_{i=1}^{N} H(z_i) - I(\boldsymbol{z}), \tag{2.17}$$

where $H(z_i) = -\int f(z_i) \log f(z_i) dz_i$ denotes the marginal entropy of $z_i$. Since the random variable $z_i = p_i(y_i)$ is a function of $y_i$, its pdf can be written in terms of the pdf of $y_i$. Given that $p_i(\cdot)$ is a one-to-one function, we have [24]

$$f_{z_i}(z_i) = \frac{f_{y_i}(y_i)}{|p_i'(y_i)|}. \tag{2.18}$$

Substituting the above expression in Eq. 2.17 yields

$$H(\boldsymbol{z}) = -I(\boldsymbol{z}) - \sum_{i=1}^{N} E\left[\log \frac{f_{y_i}(y_i)}{|p_i'(y_i)|}\right]. \tag{2.19}$$

In Eq. 2.19, if the derivative of the function $p_i(\cdot)$ is equal to the pdf of the recovered source $y_i$ (in other words, if $p_i(\cdot)$ is the cumulative pdf function (cdf) of $y_i$), i.e.

$$f_{y_i}(y_i) = |p_i'(y_i)|, \tag{2.20}$$

then the sum becomes zero, and the joint entropy simplifies to

$$H(\boldsymbol{z}) = -I(\boldsymbol{z}). \tag{2.21}$$

Therefore, provided that the function $p_i(\cdot)$ corresponds to the pdf of the sources, maximizing the joint entropy is equivalent to minimizing the mutual information among the output components. Hence, independent components can be obtained by maximizing the joint entropy. The cost function associated with the infomax principle is thus

$$\phi_{IN}(\boldsymbol{y}) = H(\boldsymbol{z}) = H(p(\boldsymbol{y})). \tag{2.22}$$

However, finding a function that matches exactly the cdf of the sources is not a trivial task. Some approximations have to be considered. In practice, we have to keep in mind that the second term of Eq. 2.19 will interfere and a sub-optimal minimization of the mutual information will generally occur. The choice of a pdf tailored to speech signals will be

discussed in Sec. 2.3.3.

As noted in [16], the infomax principle and ML estimation lead to the same cost function. The link between ML estimation and the infomax principle can be established by first assuming that the cdf of the recovered sources, $p_i(\cdot)$, corresponds to the cdf of the hypothesized sources (see Eq. 2.9):

$$p_i(\tilde{s}_i) = \int_{-\infty}^{\tilde{s}_i} q_i(r)dr. \tag{2.23}$$

As a consequence, the random variable $p_i(\tilde{s}_i)$ is distributed uniformly on $(0,1)$, which implies that $p(\tilde{\boldsymbol{s}})$ is a uniform random vector on $(0,1)^N$ [16]. By denoting $\boldsymbol{u} = p(\tilde{\boldsymbol{s}})$ and since $f_{\boldsymbol{u}}(\boldsymbol{u}) = \prod_{i=1}^{N} 1 = 1$ for $\boldsymbol{u} \in (0,1)^N$, we can write

$$\begin{aligned}
\phi_{IN}(\boldsymbol{y}) &= H(p(\boldsymbol{y})) = H(\boldsymbol{z}) \\
&= -\int f_{\boldsymbol{z}}(\boldsymbol{z}) \log f_{\boldsymbol{z}}(\boldsymbol{z}) d\boldsymbol{z} \\
&= -\int f_{\boldsymbol{z}}(\boldsymbol{z}) \log \frac{f_{\boldsymbol{z}}(\boldsymbol{z})}{f_{\boldsymbol{u}}(\boldsymbol{u})} d\boldsymbol{z} \\
&= -K(p(\boldsymbol{y}) \| \boldsymbol{u}) \\
&= -K(\boldsymbol{y} \| \tilde{\boldsymbol{s}}),
\end{aligned} \tag{2.24}$$

where the last equality follows from the invariance of the Kullback-Leibler divergence when an invertible transformation is applied to the sample space. As we can see from Eqs. 2.24 and 2.15, $\phi_{IN}(\boldsymbol{y}) = \phi_{ML}(\boldsymbol{y})$. Therefore, under above assumptions, both approaches yield exactly the same cost function. Nevertheless, interesting insights can be obtained by developing the cost function in an information theoretic framework. Basically, according to Eq. 2.19, the cost function measures two mismatches [23]:

1. The mismatch caused by the statistical dependence of $y_1, \ldots, y_N$;

2. The mismatch between the pdf of $y_1, \ldots, y_N$ and the hypothesized pdf of the sources.

Numerous other approaches to BSS have been proposed. Techniques based on negentropy maximization (Kullback-Leibler divergence between $f_{\boldsymbol{u}}(\boldsymbol{u})$ and a Gaussian pdf), cumulants (higher-order moments), and nonlinear principal component analysis have been

used. However, many of these different approaches can be unified in an information-theoretic framework [25] (and thus in a ML estimation framework).

## 2.3 Adaptive Algorithms for BSS

We now derive adaptive algorithms for BSS. These algorithms can easily be implemented on a general-purpose digital processor. In the previous section, the cost function has been developed for instantaneous BSS, but, as shown in this section, it can still be used to derive adaptive algorithms for convolutive BSS.

### 2.3.1 Instantaneous Mixtures

In this section, the source separation cost function is used to develop two adaptive algorithms for instantaneous mixtures. These two algorithms differ by the type of gradient that is used to maximize the cost function. The first algorithm is derived using the absolute gradient, while the second utilizes the natural gradient. These two types of gradient are explained in this section.

In both cases, adaptation of the de-mixing matrix $\boldsymbol{W}$ in Eq. 2.2 is carried out using the following scheme:

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + \mu \Delta \boldsymbol{W}(n), \tag{2.25}$$

where $\mu$ is an adaptive step size, $\Delta \boldsymbol{W}$ is a gradient derived from the cost function, and $n$ is a time index. The step size is often set using a trial and error procedure, and a compromise between the rate of convergence and the separation quality has to be made. A small step size makes the system converge slower, but the separation quality is better once a steady state is reached [26].

#### Absolute Gradient

The absolute gradient consists of a plain maximization of the cost function. Maximization becomes easier to carry out if the cost function is expressed as follows

$$\begin{aligned} \phi(\boldsymbol{y}) &= K(\boldsymbol{y} \| \tilde{\boldsymbol{s}}) = H(\boldsymbol{z}) \\ &= -E[\log f_{\boldsymbol{z}}(\boldsymbol{z})]. \end{aligned} \tag{2.26}$$

Using the Jacobian of the transformation from $\boldsymbol{x}$ to $\boldsymbol{z}$ (see Fig. 2.2), defined by the determinant of the matrix of partial derivatives

$$J(\boldsymbol{x},\boldsymbol{z}) = \det \begin{bmatrix} \partial z_1/\partial x_1 & \cdots & \partial z_1/\partial x_n \\ \vdots & \ddots & \vdots \\ \partial z_n/\partial x_1 & \cdots & \partial z_n/\partial x_n \end{bmatrix} = \det(\boldsymbol{W}) \prod_{i=1}^{N} p_i'(y_i), \tag{2.27}$$

in Eq. 2.26 yields

$$\begin{aligned} \phi(\boldsymbol{y}) &= -E\left[\log \frac{f_*(\boldsymbol{x})}{|J(\boldsymbol{x},\boldsymbol{z})|}\right] \\ &= H(\boldsymbol{x}) + E[\log|J(\boldsymbol{x},\boldsymbol{z})|]. \end{aligned} \tag{2.28}$$

If we substitute Eq. 2.27 in Eq. 2.28, the joint entropy can be re-written as follows

$$\phi(\boldsymbol{y}) = H(\boldsymbol{z}) = H(\boldsymbol{x}) + \log|\det \boldsymbol{W}| + \sum_{i=1}^{N} E[\log|p_i'(y_i)|]. \tag{2.29}$$

In order to obtain the so-called absolute gradient, we have to differentiate the above equation with respect to $\boldsymbol{W}$, and approximate the expectation by its instantaneous value. It can be shown that [27]

$$\frac{\partial H(\boldsymbol{z})}{\partial \boldsymbol{W}} = \Delta \boldsymbol{W} = (\boldsymbol{W}^{-1})^T - \Phi(\boldsymbol{y})\boldsymbol{x}^T, \tag{2.30}$$

where

$$\Phi(\boldsymbol{y}) = \left[-\frac{p_1''(y_1)}{p_1'(y_1)}, \ldots, -\frac{p_N''(y_N)}{p_N'(y_N)}\right]^T. \tag{2.31}$$

The function $\Phi(\cdot)$ is sometimes referred as the "activation function"[2]. The absolute gradient is not an efficient algorithm in the sense that its convergence speed can be very slow. Indeed, its performance is not uniform, and depends on the mixing matrix $\boldsymbol{A}_*$ [23]. For instance, if the condition number of $\boldsymbol{A}_*$ is large, inverting the mixing system can be a very difficult task, and takes more time to converge than otherwise. Furthermore, Eq. 2.30 requires a matrix inversion at each iteration, which is an expensive computation, and therefore not a desirable feature.

[2]This terminology comes from the field of neural networks.

As reported by Chicharo and Xi [28], the matrix inversion can easily be avoided. The authors observed that performance of the algorithm does not change significantly if the term $(\boldsymbol{W}^{-1})^T$ in Eq. 2.30 is dropped. Therefore, a simpler learning rule which requires less computational resources can be formulated as follows:

$$\Delta \boldsymbol{W} = -\Phi(\boldsymbol{y})\boldsymbol{x}^T. \tag{2.32}$$

However, this algorithm can still converge very slowly if $\boldsymbol{A}_*$ is not well conditioned.

### Natural Gradient

Amari *et al.* introduced the notion of natural gradient, and proposed in [13] to scale Eq. 2.30, the absolute gradient, by $\boldsymbol{W}^T\boldsymbol{W}$:

$$\begin{aligned} \Delta \boldsymbol{W} &= [(\boldsymbol{W}^{-1})^T - \Phi(\boldsymbol{y})\boldsymbol{x}^T]\boldsymbol{W}^T\boldsymbol{W} \\ &= [\boldsymbol{I} - \Phi(\boldsymbol{y})\boldsymbol{y}^T]\boldsymbol{W}. \end{aligned} \tag{2.33}$$

This scaling not only improves convergence speed considerably, but it also decreases the amount of computations needed since a matrix inversion is no longer necessary (compare Eq. 2.33 with Eq. 2.30). Moreover, contrary to the absolute gradient, performance of the natural gradient does not depend on the mixing matrix $\boldsymbol{A}_*$. Convergence speed only depends on the characteristics of the sources, even if the mixing matrix is close to being singular. In the literature, this feature is referred to as the equivariant property. A complete justification of the origin and properties of the natural gradient can be found by studying differential geometry, as explained in [29]. In a few words, the natural gradient exploits the structure of the parameter space (composed of all non-singular $N \times N$ matrices). The parameter space can be represented as a Riemannian space, and the natural gradient represents the steepest descent (or ascent) in that space [21].

Independently, Cardoso and Laheld introduced in [15] the notion of relative gradient. In the context of BSS, the relative gradient coincides with the natural gradient.

## 2.3.2 Convolutive Mixtures

A more elaborate separating network than the one considered in Fig. 2.2 is required to process convolutive mixtures. Two de-mixing networks are presented in this section, based,

respectively, on a feedforward and a feedback architecture. Adaptation equations are derived for both networks.

### Feedforward Architecture

Figure 2.3 illustrates a feedforward separating network. Each box represents a finite impulse response (FIR) filter of length $L$. The nonlinear functions, $p_i(\cdot)$, are not illustrated in this figure as in Fig. 2.2, since the cost function implicitly incorporates such functions as shown later. Referring to Fig. 2.3, for mathematical convenience, we will not explicitly use $\boldsymbol{w}_{ij}$ to derive an adaptation equation. Instead, we define $\boldsymbol{W}_k$ as a $N \times N$ matrix whose component $ij$ corresponds to the $k$-th entry of vector $\boldsymbol{w}_{ij}$, i.e.

$$[\boldsymbol{W}_k]_{ij} = [\boldsymbol{w}_{ij}]_k, \tag{2.34}$$

where the brackets here indicates a particular element of a matrix or a vector. Using this notation, the network is characterized by the input-output relation

$$\boldsymbol{y}(n) = \sum_{k=0}^{L-1} \boldsymbol{W}_k \boldsymbol{x}(n-k), \tag{2.35}$$

where $\boldsymbol{y}(n) = [y_1(n), \ldots, y_N(n)]^T$, and $\boldsymbol{x}(n) = [x_1(n), \ldots, x_N(n)]^T$. The length of the filters $\boldsymbol{w}_{ij}$ is chosen to match the reverberation time of the room. A room with a strong reverberation (echoes that last for a long time) requires more taps.

An adaptive algorithm for convolutive BSS is derived as follows. Regardless of the de-mixing network which is used (instantaneous or convolutive), according to Eq. 2.28, the cost function can be expressed as

$$\phi(\boldsymbol{y}(n)) = H(\boldsymbol{x}(n)) + E[\log|J|], \tag{2.36}$$

where $J$ is the Jacobian of the de-mixing network. The first term in the above equation can be dropped since it is a constant (thus, not useful for maximization purposes), and the expectation in the second term can be omitted since we are interested in a stochastic gradient. Therefore, the cost function becomes

$$\phi(\boldsymbol{y}(n)) = \log|J|. \tag{2.37}$$

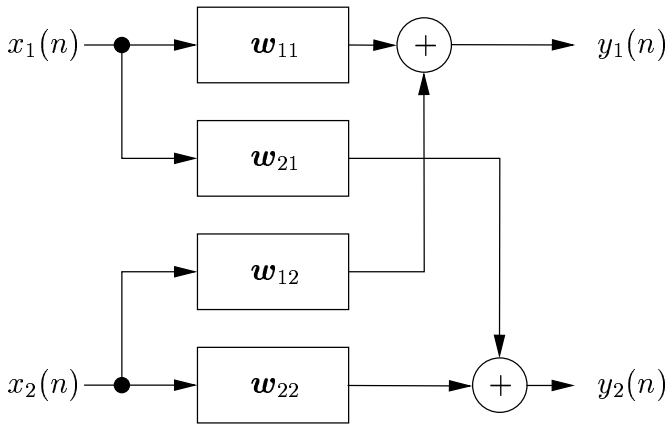



**Fig. 2.3** Feedforward convolutive de-mixing network with two sources.

Hence, in order to find an adaptation equation for convolutive BSS, we have to find the Jacobian of the de-mixing network, and maximize the logarithm of the absolute value of that expression.

For the feedforward network, the Jacobian is

$$J = \det(\boldsymbol{W}_0) \prod_{i=1}^{N} p_i'(y_i(n)). \tag{2.38}$$

Taking the logarithm of the absolute value of this expression yields

$$\log|J| = \log|\det \boldsymbol{W}_0| + \sum_{i=1}^{N} \log|p_i'(y_i(n))|. \tag{2.39}$$

If we substitute $y_i(n)$ in the above equation according to Eq. 2.35, an adaptive algorithm can be found by maximizing Eq. 2.39 with respect to $\boldsymbol{W}_k$. Two cases have to be considered, namely $k = 0$ and $k \geq 1$ [20]. It can be shown that

$$\frac{\partial \log|J|}{\partial \boldsymbol{W}_k} = \begin{cases} (\boldsymbol{W}_0^{-1})^T - \Phi[\boldsymbol{y}(n)]\boldsymbol{x}^T(n), & k = 0 \\ -\Phi[\boldsymbol{y}(n)]\boldsymbol{x}^T(n-k), & k \geq 1. \end{cases} \tag{2.40}$$

As expected, we obtain the same result as with instantaneous mixtures when $k = 0$ (see Eq. 2.30). Therefore, for $k = 0$, we can use the natural gradient to avoid a matrix inversion.

For $k \geq 1$, the gradient is already computationally simple, so we can use the absolute gradient. In the end, we obtain the following adaptation equations

$$\boldsymbol{W}_k(n+1) = \boldsymbol{W}_k(n) + \mu \Delta \boldsymbol{W}_k(n) \tag{2.41}$$

with

$$\Delta \boldsymbol{W}_k(n) = \begin{cases} [\boldsymbol{I} - \Phi[\boldsymbol{y}(n)]\boldsymbol{x}^T(n)\boldsymbol{W}_0^T(n)]\boldsymbol{W}_0(n), & k = 0 \\ -\Phi[\boldsymbol{y}(n)]\boldsymbol{x}^T(n-k), & k \geq 1. \end{cases} \tag{2.42}$$

Adaptation of the feedforward network can be carried out in the frequency domain, resulting in a very efficient implementation [22], [3]. However, the network has the drawback of whitening the signal due to the filters $\boldsymbol{w}_{ii}$, which tend to remove temporal redundancies as well as spatial ones. The whitening problem can be particularly annoying for speech signals, and it can be avoided by using a feedback network instead [14], as described in the next section.

### Feedback Architecture

Convolutive mixtures can also be processed by a feedback network as illustrated in Fig. 2.4. It is not necessary to include the self-connecting loops in Fig. 2.4 to separate the sources, so we always set $\boldsymbol{w}_{ii} = \boldsymbol{0}$.

We will proceed exactly as in the previous section to obtain the adaptation equation for the feedback network. The input-output relation is now

$$\boldsymbol{y}(n) = \boldsymbol{x}(n) + \sum_{k=0}^{L-1} \boldsymbol{W}_k \boldsymbol{y}(n-k), \tag{2.43}$$

where all the diagonal elements of $\boldsymbol{W}_k$ are zero. The Jacobian of the network becomes [30]

$$J = \prod_{i=1}^{N} p_i'(y_i), \tag{2.44}$$

which, according to Eq. 2.37, implies the following cost function

$$\log|J| = \sum_{i=1}^{N} \log|p_i'(y_i)|. \tag{2.45}$$

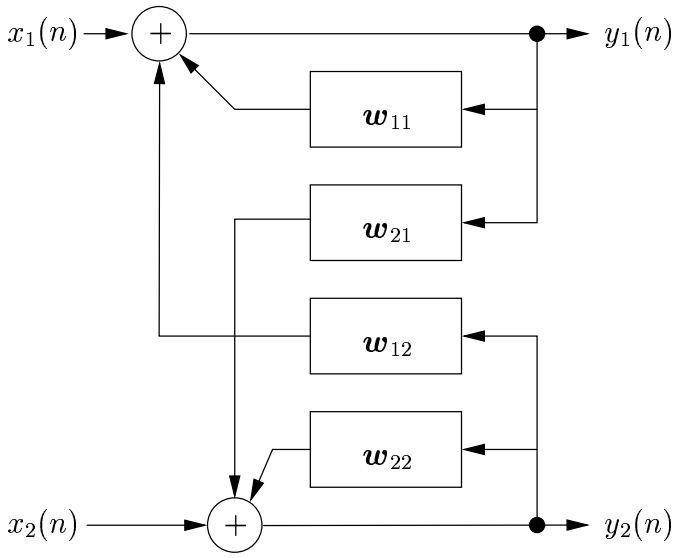



**Fig. 2.4** Feedback convolutive de-mixing network with two sources.

Maximizing the cost function yields [31]

$$\frac{\partial \log |J|}{\partial \boldsymbol{W}_k} = \Delta \boldsymbol{W}_k = \Phi[\boldsymbol{y}(n)]\boldsymbol{y}^T(n-k). \tag{2.46}$$

The previous equation, combined with Eq. 2.41, can be used to update the coefficients of the feedback network.

Contrary to the feedforward network, whitening of the signals is no longer an issue. Because each filtered component is added to a branch coming from a different input, the feedback network will not remove temporal redundancies [14]. The feedback network has also the advantage of requiring less parameters to adapt since the self-connecting loops are not necessary.

Note that if the mixing system has a non-minimum phase characteristic, which can occur if an echo is stronger than the direct path [22], then non-causal filters must be implemented to inverse the system [32]. The frequency domain implementation of the feedforward network easily allows such an non-causal extension [22], [3]. As shown in [33], it is also straightforward to implement non-causal filters in the feedback network, and modify the adaptive algorithm accordingly.

### 2.3.3 Activation Functions

The adaptation equations derived for instantaneous and convolutive BSS necessitate evaluation of the activation function $\Phi_i(\cdot)$, defined earlier as

$$\Phi_i(y_i) = -\frac{p_i''(y_i)}{p_i'(y_i)}, \tag{2.47}$$

where $p_i(\cdot)$ should be equal to the cdf of source $i$. In fact, when $p_i(\cdot)$ corresponds exactly to the cdf, it has been shown in [34] that the error variance of the separated signals is minimized. Bell and Sejnowski proposed in [12] two activation functions that were widely used in the literature, namely

$$p_i(y_i) = \frac{1}{1+e^{-y_i}} \Rightarrow \Phi_i(y_i) = 1 - \frac{2}{1+e^{-y_i}} \tag{2.48}$$

and

$$p_i(y_i) = \tanh(y_i) \Rightarrow \Phi_i(y_i) = 2\tanh(y_i). \tag{2.49}$$

These functions are only capable of separating super-Gaussian sources (or, in other words, sources with positive kurtosis), like speech [1]. But a better activation function tailored to speech signals has been proposed [34]. It is well known that speech can be approximated by a Laplacian pdf, which is given by

$$p_i'(y_i) = \frac{\alpha}{2} e^{-\alpha|y_i|}, \tag{2.50}$$

where $\alpha$ is a positive constant. Therefore, the following activation function is appropriate for speech sources:

$$\Phi_i(y_i) = \alpha\,\mathrm{sign}(y_i), \tag{2.51}$$

In this case, the step size is defined as follows

$$\mu = \frac{\mu'}{\alpha}, \tag{2.52}$$

where $\mu'$ is a small positive constant. Using this new definition is convenient, because we avoid having to choose a value for $\alpha$. For speech sources, Eq. 2.51 outperforms many activation functions proposed earlier in the literature [34]. Even if Eq. 2.51 was specifically

derived for speech sources, it can still separate, in practice, other audio sources such as music.

If there exists no knowledge about the pdf of the sources, then a fixed activation function may fail to separate some sources. There is a certain limit a mismatch between $p_i(\cdot)$ and the cdf of a source that the separating system can tolerate [23]. We already have several activation functions capable of separating super-Gaussian sources (see Eqs. 2.48, 2.49, and 2.51). Lee and Sejnowski suggested to process a mixture of sub-Gaussian and super-Gaussian sources using the extended infomax algorithm [1]. The extended infomax algorithm makes use of two activation functions: one for super-Gaussian sources and one for sub-Gaussian sources (or sources with positive and negative kurtosis, respectively, as illustrated in Fig. 2.5). A switching criterion is then applied to choose the most suitable function at a given time.

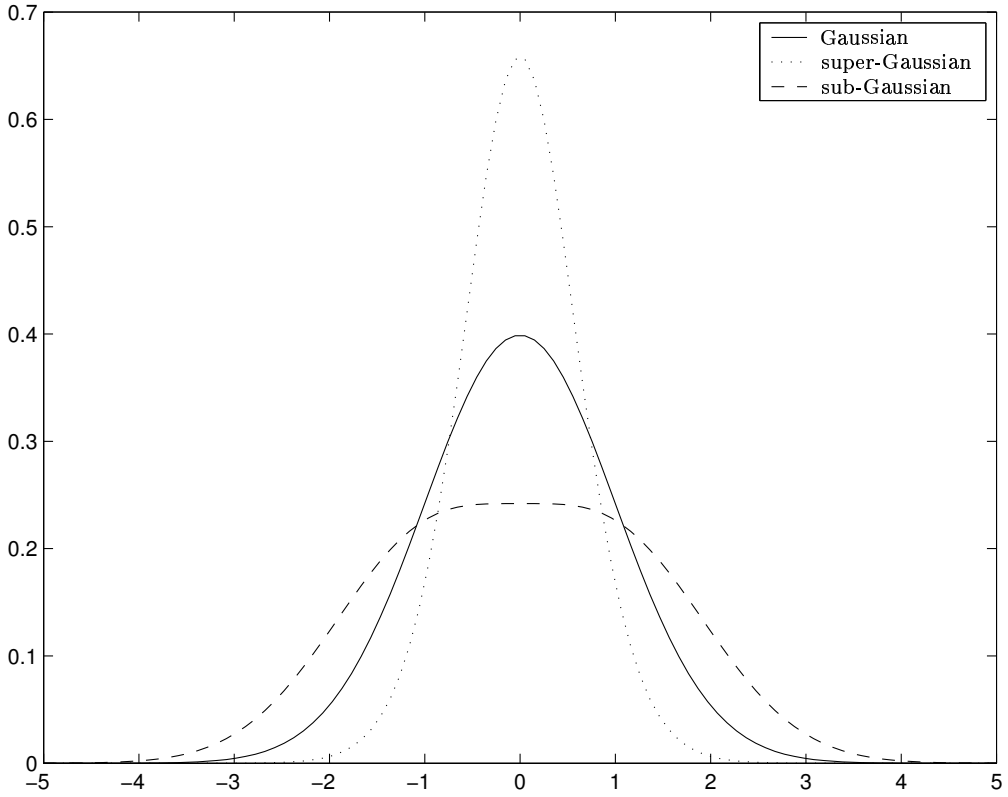



**Fig. 2.5** Gaussian, super-Gaussian and sub-Gaussian pdf's.

The sub-Gaussian source is modelled by the sum of two Gaussian distributed sources

with mean $\pm\mu$ and variance $\sigma^2$

$$p_i'(y_i) = \frac{1}{2\sqrt{2\pi}\sigma}\left(e^{-\frac{(y_i-\mu)^2}{2\sigma^2}} + e^{-\frac{(y_i+\mu)^2}{2\sigma^2}}\right). \tag{2.53}$$

Obviously, the pdf becomes bimodal if $\mu$ is large enough. It can be shown that the kurtosis always remains negative (or zero if $\mu = 0$) [1]. The super-Gaussian source is modelled by the following pdf

$$p_i'(y_i) = \nu f_G(y_i)\,\mathrm{sech}^2(y_i), \tag{2.54}$$

where $\nu \approx 0.6057$ is a constant so that $\int_{-\infty}^{\infty} p_i'(y_i)dy_i = 1$, and $f_G(y_i)$ is the pdf of a Gaussian random variable with mean 0 and variance 1, i.e.

$$f_G(y_i) = \frac{e^{-y_i^2/2}}{\sqrt{2\pi}}. \tag{2.55}$$

The proposed super-Gaussian source has an even more pronounced peakedness than a Laplacian pdf. Setting $\mu = 1$ and $\sigma^2 = 1$ for the sub-Gaussian source yields the following activation functions [1]:

$$\Phi_i(y_i) = \begin{cases} y_i - \tanh(y_i) & \text{for sub-Gaussian sources} \\ y_i + \tanh(y_i) & \text{for super-Gaussian sources} \end{cases} \tag{2.56}$$

Both functions differ only by a change of sign before the hyperbolic tangent. Switching between the two signs can be done by evaluating the kurtosis within a given time frame. If it is positive, a "+" sign is used, and vice-versa. But an alternate approach that works better in practice utilizes a stability analysis of the separating process. A sufficient stability condition for instantaneous BSS can be stated as follows [35]: $\kappa_i > 0$ for $i = 1, \ldots, N$ with

$$\kappa_i = E[\Phi_i'(y_i)]E[y_i^2] - E[\Phi_i(y_i)y_i], \tag{2.57}$$

where

$$\Phi_i(y_i) = y_i + k_i \tanh(y_i), \quad k_i = \pm 1. \tag{2.58}$$

Substituting $\Phi_i(y_i)$ in Eq. 2.57 yields

$$\kappa_i = k_i(E[\operatorname{sech}^2(y_i)]E[y_i^2] - E[y_i \tanh(y_i)]). \tag{2.59}$$

Stability is guaranteed if $\kappa_i > 0$ for all sources. Therefore, the sign of $k_i$ must be the same as $E[\operatorname{sech}^2(y_i)]E[y_i^2] - E[y_i \tanh(y_i)]$. This amounts choosing $k_i$ such that

$$k_i = \operatorname{sign}(E[\operatorname{sech}^2(y_i)]E[y_i^2] - E[y_i \tanh(y_i)]). \tag{2.60}$$

The extended infomax algorithm is mainly used for biomedical signal processing, because it is difficult to anticipate the pdf of all the sources.

## 2.4 Chapter Summary

The basic concepts of BSS were introduced in this chapter. Two different mixing models were given. De-mixing networks were then discussed in details, and adaptation equations were derived. Since the adaptation equations are mathematically expressed in terms of the pdf of the sources, functions that model such pdf were enumerated. Finally, an algorithm that does not require knowledge of the statistical pdf of the sources was presented.

# Chapter 3

# DFT Filter Banks for Subband Adaptation

In this chapter, the concept of subband adaptation is introduced. Firstly, an overview of BSS in subbands is presented, and a discussion about DFT filter banks follows. Interpolation and decimation schemes for DFT filter banks are then described. A discussion about prototype filter design is also given. Finally, an efficient structure for the DFT filter bank, the weighted overlap-add realization, is briefly mentioned, and a summary ends the chapter.

## 3.1 Blind Source Separation in Subbands

As outlined in Chap. 2, convolutive BSS necessitates learning of potentially long filters to inverse a mixing system. These long filters are expensive to adapt due to the many coefficients that have to be considered, which could make real-time applications impossible. Using subband adaptation, we can adapt several shorter filters instead of a long one. Adapting short filters is more efficient from a computational complexity point of view [36]. However, subband analysis and synthesis introduce distortions in the signal. The challenge is to limit these distortions to an acceptable level while preserving an efficient structure for subband-based BSS systems.

Figure 3.1 depicts a BSS system operating in subbands. The subband-based system works as follows. Initially, each microphone signal is separated into $K$ subbands, each of them corresponding to a certain frequency range of equal width. Since the bandwidth of

the signal in each subband is now reduced, the sampling rate can be lowered by a factor $M \leq K$. Sampling rate reduction is performed using a digital decimator, represented by the symbol $\downarrow M$ in Fig. 3.1. The mixtures are then independently separated in each subband. After the separation, the processed signals are digitally interpolated so that their sampling rate is restored at its original value. The signals coming from the different subbands that belong to the same source are then added together to reconstruct the fullband signal. The amount of computations is lowered because adaptation occurs when the sampling rate is reduced. Hence, subband-based BSS systems have to adapt less parameters, since the number of taps of each de-mixing filter decreases due to the bandwidth reduction. As shown in Chap. 4, even if each subband channel now requires its own BSS system, it is still more efficient than fullband adaptation, where only one BSS system is needed, but with longer de-mixing filters.

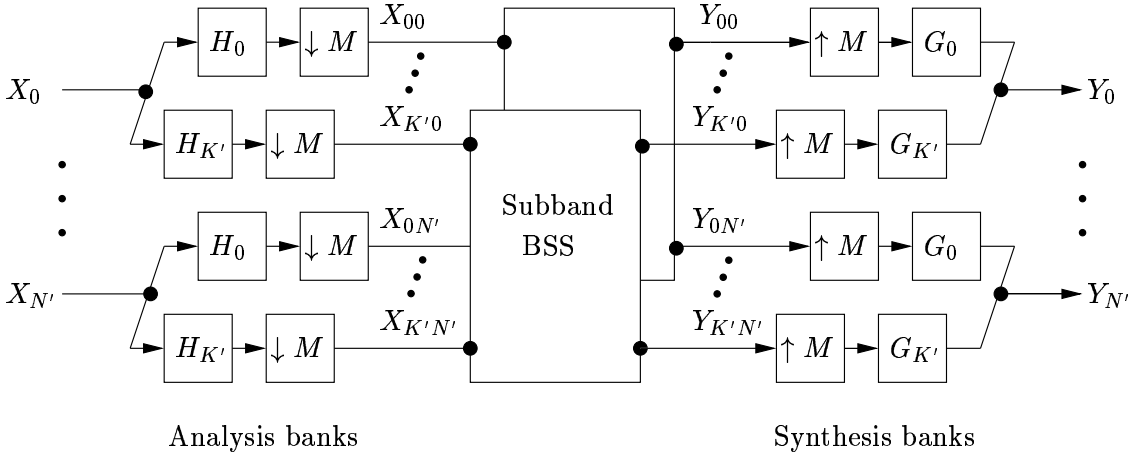



**Fig. 3.1** A subband-based BSS system ($N' = N - 1$ and $K' = K - 1$).

Subband analysis and synthesis are implemented using two filter banks, referred to as the analysis and synthesis banks, respectively. Filter banks are basically an array of filters, each of them extracting from the signal all components of an appropriate frequency range. In practical situations, filter banks consist of non-ideal filters, such as the ones illustrated in Fig. 3.2 for a two-subband system. Note that these filters overlap in frequency, and distort the signal. Due to the non-ideal characteristics of the filters, distortions are introduced by the filter banks. Specifically, three types of distortions can be distinguished for a system consisting of a direct cascade of an analysis and a synthesis bank, i.e. without intermediate

processing [37]:

**Aliasing distortion,** which results from the presence of aliased components in the system output (due to the non-observance of the Nyquist condition when decimation occurs);

**Amplitude distortion,** which is caused by the non-flat magnitude response of the system;

**Phase distortion,** which is caused by the non-linear phase response of the system.

A filter bank which is not affected by any of these three distortions is said to be characterized by the perfect reconstruction (PR) property. With PR filter banks, it is possible to obtain a perfect replica of the input at the output of a subband-based system. A perfect replica is obtained when the signal differs only by a gain and a delay, i.e.

$$Y_i(z) = cz^{-d}X_i(z), \quad c \neq 0. \tag{3.1}$$

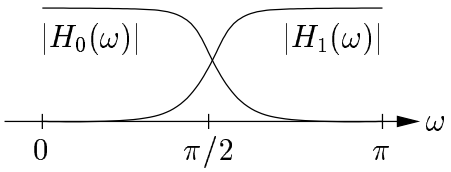



**Fig. 3.2** Non-ideal filters used for subband analysis and synthesis in a two-subband system.

Most of the theory related to PR filter banks has been developed for unprocessed signals in subbands[1], and cannot be readily applied to the problem under consideration in this thesis. A near-PR filter bank allows much more relaxed conditions than those required by a PR filter bank. A near-PR filter bank can exhibit certain amount of distortions, provided they are below an acceptable level. Since the design of a near-PR filter bank is often simpler, and, from a practical point of view, both PR and near-PR filter banks can yield approximately the same result in audio applications, where processing of subband signals is involved, we will aim for the near-PR property in this thesis.

If $M = K$, the filter bank is referred to as a critically sampled filter bank. Usually, in a critically sampled filter bank, after filtering, the signal is not perfectly bandlimited,

[1]In this context, this means that BSS would not be attempted, and, in each subband, the input would simply be connected to the output signal.

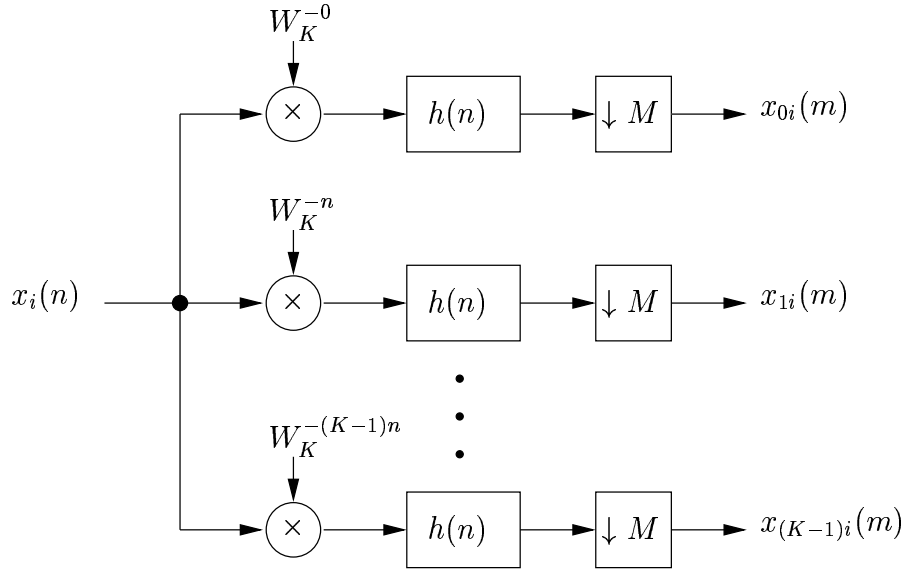
and there remain high frequency components that get transformed into aliased components after sampling rate reduction. Techniques, based on PR theory, exist to cancel these aliased components during synthesis, but are mostly limited to the case where the analysis and synthesis banks are directly cascaded together, which is not the case in our work. Regardless whether the analysis and synthesis banks are cascaded together, setting $M < K$ reduces the energy of aliased components, since the sampling rate reduction is less severe. When $M < K$, the filter bank is referred to as an oversampled filter bank. In this thesis, we will use an oversampled scheme, since distortion would be present even for PR systems due to subband processing.

## 3.2 DFT Filter Banks

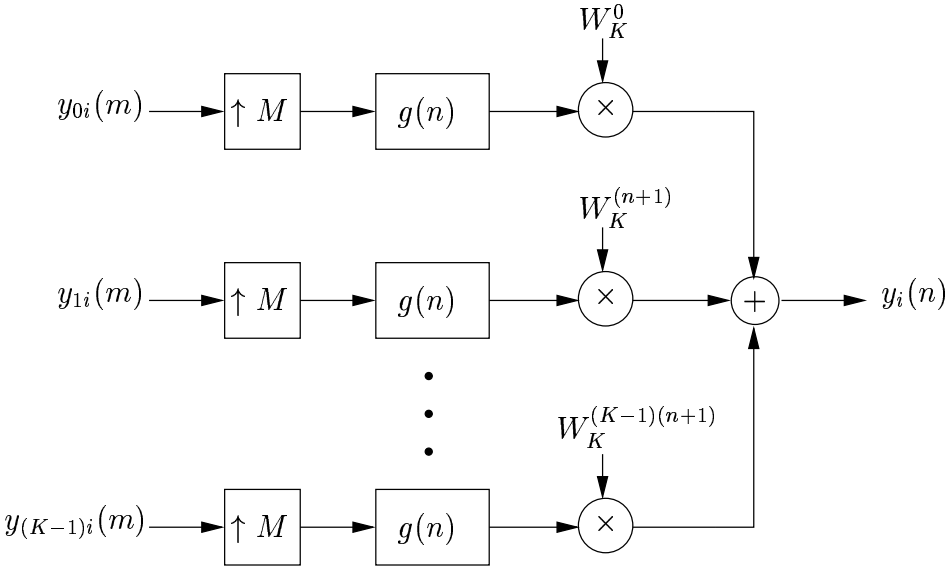
Many filter bank structures suitable for subband processing have been reported in the literature, see e.g. [38], [37]. In this thesis, we will use DFT filter banks for subband analysis and synthesis, mainly due to their versatility, efficiency and simplicity. A DFT filter bank makes use of complex modulations. Other filter banks could have been used. For instance, some filter banks are based on a tree-like structure, and others use cosine modulation. Cosine modulated filter banks (CMFB) [39] could have been used in this thesis instead of DFT filter banks. CMFB produce real modulated signals instead of complex ones, which is definitely more convenient, because complex operations could be avoided. However, due to the modulation scheme employed, CMFB are more difficult to implement, especially if we want to satisfy the PR (or near-PR) property [40].

Two DFT filter banks are needed for subband processing: one for subband analysis and one for synthesis. Both filter banks use a low-pass prototype filter either to limit the bandwidth of the modulated signal before decimation, or to remove periodic repetitions of the spectrum after interpolation. A $K$-subband analysis and synthesis DFT filter bank pair is illustrated in Fig. 3.3. Note that, due to sampling rate reduction, there are two time indexes, namely $n$ and $m$. Subband processing is performed at low sampling rate, $m$. DFT filter banks are simple, and operate as follows.

Firstly, consider the analysis bank. The spectrum of the signal in subband channel $k$ is shifted by $(2\pi k)/K$ to the left using a complex modulation, i.e. multiplication by $W_K^{-nk}$, $W_K = e^{j2\pi/K}$. After modulation, the signal is filtered with an FIR low-pass prototype filter $h(n)$ of length $D$ and cutoff frequency $\pi/K$. It is then decimated by a factor $M$. The result




(a) Analysis bank.



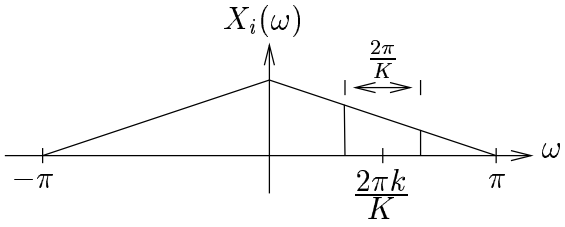

(b) Synthesis bank.

**Fig. 3.3** Uniform DFT filter banks (for analysis and synthesis).

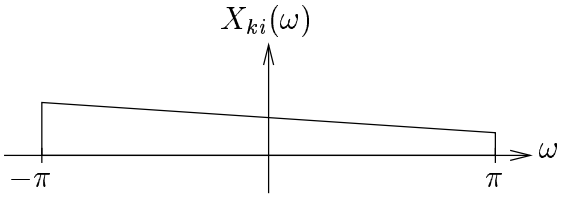
of such operations is illustrated in Fig. 3.4, where $X_i(\omega)$ and $X_{ki}(\omega)$ respectively denote the discrete-time Fourier transform (DTFT) of $x_i(n)$ and $x_{ki}(m)$. Using an ideal low-pass filter with cutoff frequency $\pi/K$, i.e.

$$H(\omega) = \begin{cases} 1, & -\pi/K < \omega < \pi/K \\ 0, & \text{otherwise}, \end{cases} \tag{3.2}$$

would result in a filter bank free from distortions. But since an ideal filter is not realizable in practice (that is, it cannot be stable and causal), designing a prototype filter is not trivial, and will be fully considered in Sec. 3.4.




(a) Fullband signal.




(b) Subband signal.

**Fig. 3.4** Complex modulation and decimation in an analysis DFT filter bank.

In the synthesis bank, the signal in each subband is upsampled by $M$, and filtered by $g(n)$, the synthesis prototype low-pass filter. Like the filter $h(n)$, $g(n)$ is also characterized by a cutoff frequency of $\pi/K$. At the end, each signal is demodulated, and added together so that a fullband signal is obtained. Unlike traditional designs, demodulation is implemented by multiplying the signal by $W_K^{k(n+1)}$, as suggested in [41]. Classically, demodulation is carried out using $W_K^{nk}$, which corresponds to the complex conjugate of the expression used

for modulation [38], but it has been shown that the expression proposed in [41] eliminates phase distortion more naturally. Finally, note that an equivalent interpretation of the DFT filter bank in terms of complex bandpass filters can also be formulated [38].

Modulation and demodulation closely resemble the definition of the discrete Fourier transform (DFT), hence the name DFT filter banks. It should be pointed out that modulation is of an even type, i.e. the frequency mid-point of each band is located at

$$\omega_k = \frac{2\pi k}{K}, \quad k = 0, 1, \ldots, K-1, \tag{3.3}$$

as illustrated in Fig. 3.5 for $K = 8$. This implies that, for real signals like speech, subband 0 differs from the other subbands in the sense that it contains real-valued samples, and its effective bandwidth is halved. Assuming $K$ even, a similar observation applies to the band centered at $\omega_{K/2} = \pi$. Moreover, even if we speak of a $K$-subband DFT filter bank, in reality, for real sources, only the first $1 + K/2$ subbands are relevant. In fact, information contained in the second half is redundant, since the spectrum of a real signal $y_i(n)$ for $-\pi < \omega < 0$ can easily be retrieved from the spectrum limited to $0 < \omega < \pi$ with the relation [42]

$$Y_i(\omega) = Y_i^*(-\omega), \tag{3.4}$$

where $Y_i(\omega)$ is the DTFT of $y_i(n)$. If we eliminate the second half of the subbandss, the imaginary part will not cancel anymore when the signals are added together at the end. However, basic derivations show that simply throwing away the imaginary part at the end results in the same real signal as if the second half of the subbands was considered.

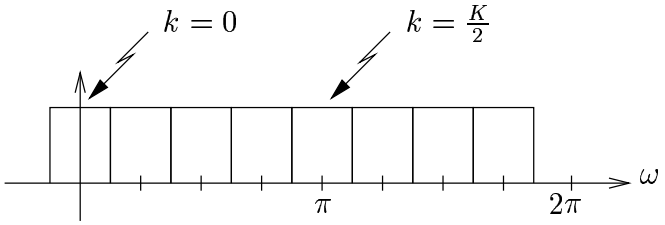



**Fig. 3.5** Subband channels for $K = 8$.

Finally, as mentioned in Sec. 3.1, the best way to handle distortions introduced by the filter banks is to use oversampling ($M < K$). The decimation factor $M$ should be set to a level compatible with audio applications, while attempting to preserve computational

efficiency of the subband approach.

## 3.3 Decimation and Interpolation

Since decimators and interpolators are two important components of DFT filter banks, it is worthwhile to review how they operate in the z-domain. These results will enable us to study the origins of various distortions introduced by the filter banks. Efficient implementations of the decimator and interpolator are also discussed. Expressions derived in this section consist of classical results in the field of multirate digital processing, and can be found in [43].

### 3.3.1 Decimation

The decimation of a signal $x(n)$ by a factor $M$ can be realized by the system illustrated in Fig. 3.6. The systems consists of a low-pass filter $h(n)$, whose purpose will be explained shortly, followed by a sampling rate compressor by an integer factor $M$. The sampling rate compressor retains only one out of $M$ samples of its input signal, $v(n)$. The resulting decimated output is denoted $y(m)$.

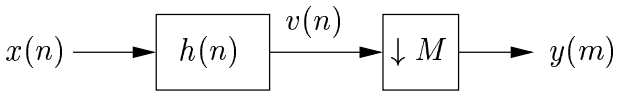



**Fig. 3.6** A decimator.

The operation of the sampling rate compressor may be expressed in mathematical terms as

$$y(m) = v(mM), \quad m \in \mathbb{Z}. \tag{3.5}$$

Based on this time-domain relationship, it can be shown that the z-transform of $y(m)$, $Y(z)$, is [43]

$$Y(z) = \frac{1}{M} \sum_{l=0}^{M-1} V(W_M^{-l} z^{1/M}), \tag{3.6}$$

where $V(z)$ denotes the z-transform of $v(n)$. According to Fig. 3.6, $V(z)$ is simply a filtered

version of $X(z)$, and can be written as

$$V(z) = H(z)X(z), \tag{3.7}$$

where $H(z)$ is the z-transform of $h(n)$. Therefore, if we substitute Eq. 3.7 in Eq. 3.6, $Y(z)$ becomes

$$Y(z) = \frac{1}{M}\sum_{l=0}^{M-1} H(W_M^{-l}z^{1/M})X(W_M^{-l}z^{1/M}). \tag{3.8}$$

It is easier to explain the role of filter $H(z)$ by reformulating Eq. 3.8 in the frequency domain. To this end, we evaluate the z-transform on the unit circle, i.e. $z = e^{j\omega}$, thus obtaining the DTFT of $y(m)$:

$$Y(\omega) = \frac{1}{M}H\left(\frac{\omega}{M}\right)X\left(\frac{\omega}{M}\right) + \frac{1}{M}\sum_{l=1}^{M-1} H\left(\frac{\omega - 2\pi l}{M}\right)X\left(\frac{\omega - 2\pi l}{M}\right), \tag{3.9}$$

where the term $l = 0$ has been isolated from the rest of the sum. The term corresponding to $l = 0$ is an expanded version of the original spectrum by $M$, whereas those corresponding to $l \neq 0$ represent the expanded spectrum shifted by multiples of $2\pi$. These terms are necessary to make $Y(\omega)$ $2\pi$-periodic. If we focus on the band $-\pi < \omega < \pi$, the terms corresponding to $l \neq 0$ are responsible for aliasing, and to avoid aliasing of these terms, the filter $H(\omega)$ must not permit frequency components above $\pi/M$, i.e.

$$H(\omega) = \begin{cases} 1, & -\pi/M < \omega < \pi/M \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}$$

The decimator illustrated in Fig. 3.6 is not very efficient. Indeed, filtering all the samples is a waste of resources, since for every block of $M$ samples, $M-1$ are discarded. Since it is already assumed that the prototype filter is FIR, it can be implemented using a direct form structure as illustrated in Fig. 3.7(a). A more efficient structure can be derived if one realizes that each gain of the filter can be commuted with the sampling rate compressor. The result of this operation is illustrated in Fig. 3.7(b). The filter now operates at frequency $F/M$ instead of $F$, which reduces computations by a factor of $M$.

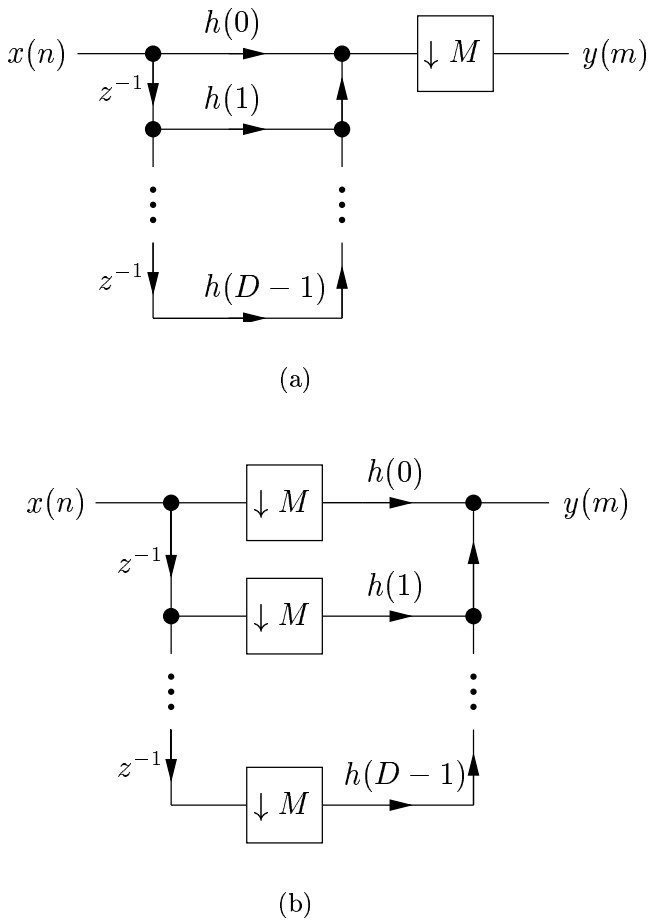



**Fig. 3.7** Derivation of an efficient structure for decimation using an FIR filter.

### 3.3.2 Interpolation

Let us now consider the interpolation of a signal $x(m)$ by a factor $M$, as depicted in Fig. 3.8. Interpolation is the dual of decimation; instead of decreasing the sampling rate by $M$, we are interested in increasing it by $M$. Interpolating a signal can be realized in two steps. First, $M-1$ zeros are inserted between each sample using a sampling rate expander. Then, the signal is "smoothed" using a low-pass filter. The resulting interpolated output is denoted $y(n)$.

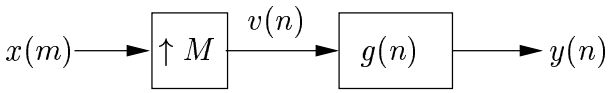



**Fig. 3.8** An interpolator.

Referring to Fig. 3.8, we have

$$v(n) = \begin{cases} x(n/M), & \text{for } n = \ldots, -2M, -M, 0, M, 2M, \ldots \\ 0, & \text{otherwise.} \end{cases} \tag{3.11}$$

From the above time-domain relation, the z-transform of $v(n)$, $V(z)$, can be obtained as [43]

$$V(z) = X(z^M). \tag{3.12}$$

The DTFT of $v(n)$ can be obtained by setting $z = e^{j\omega}$, i.e.

$$V(\omega) = X(M\omega). \tag{3.13}$$

Thus, the entire spectrum of $x(m)$, over the interval $-\pi < \omega < \pi$, is compressed to $-\pi/M < \omega < \pi/M$. Moreover, images of the compressed spectrum, centered at $\pm 2\pi/M, \pm 4\pi/M, \ldots$, are also created. The purpose of the low-pass filter is precisely to remove these unwanted images. Therefore, the filter $G(\omega)$ must remove frequency components above $\pi/M$. Note that, contrary to decimation, the low-pass filter must have a gain of $M$ in its passband if we want to preserve the amplitude of the signal, i.e. $y(n) = x(n/M)$. Indeed, let us assume that $G(\omega)$ is an ideal low-pass filter with a cutoff frequency $\pi/M$ and a gain of $G$. Selecting

$n = 0$ for mathematical convenience, we can write

$$\begin{aligned} y(0) &= \int_{-\pi}^{\pi} H(\omega) X(M\omega) d\omega \\ &= G \int_{-\pi}^{\pi} X(\omega') d\omega' / M \\ &= \frac{G}{M} x(0). \end{aligned} \tag{3.14}$$

Therefore, a passband gain of $M$ is necessary so that both the original and the interpolated signals have the same amplitude.

As it was the case with the decimator, it is possible to find a more efficient structure for interpolation than the one illustrated in Fig. 3.8. In this figure, filtering is done right after the zeros have been inserted. This is not an efficient scheme, since many zeros get filtered, which generates unnecessary computations[2]. Let us assume that the low-pass filter length, $D$, is a multiple of $M$. At each time instant, we do not need to worry about the weights of the filter that are multiplied by a zero. So, at time $m = 0$, let us define a new filter, $g_0(m)$, that only comprises the relevant weights at that precise instant. If we proceed similarly at $m = \pm 1, \pm 2, \ldots$, we can note that the filters repeat themselves after every block of $M$ consecutive samples. Hence, we only need to consider a set of $M$ filters, such that

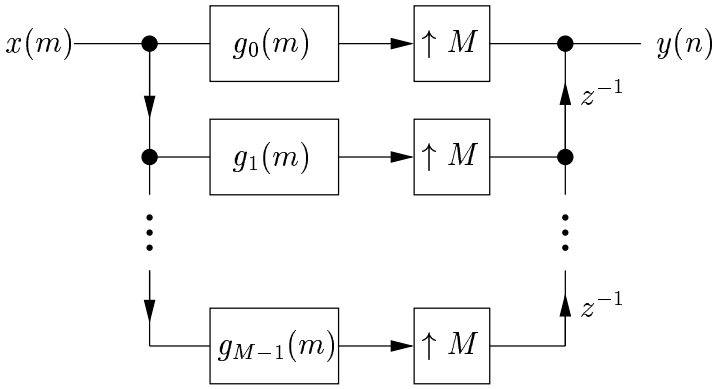
$$g_i(m) = g(mM + i), \quad \text{for } i = 0, 1, \ldots, M-1. \tag{3.15}$$

These filters, referred to as polyphase filters [43], can be used in a polyphase network to derive an efficient structure for interpolation, as illustrated in Fig. 3.9(a). In practice, the structure in Fig. 3.9(a) can be realized by the equivalent network of Fig. 3.9(b), which makes use of a circular commutator.

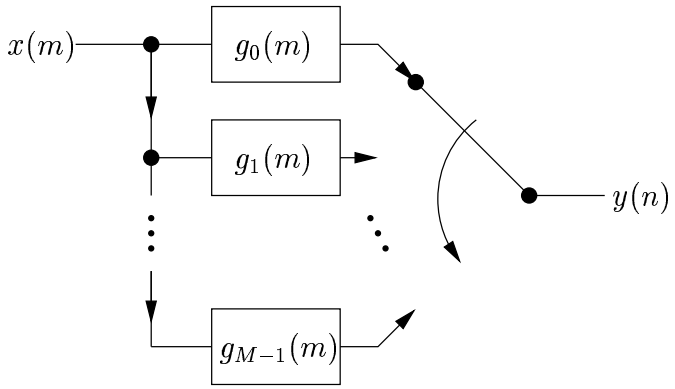
## 3.4 Prototype Filter Design

In this section, we discuss the process of designing a prototype filter suitable for an oversampled DFT filter bank. Two main points are asserted in this section. Firstly, it is shown that if the synthesis prototype filter is a time flipped version of the analysis prototype, then

[2]Note that by transposing the structure in Fig. 3.7(b), we can obtain an efficient structure for interpolation, but we propose, in this section, an alternative approach based on polyphase FIR structures.




(a) Polyphase structure for an interpolator.




(b) Commutator structure for a polyphase interpolator.

**Fig. 3.9** Derivation of an efficient structure for interpolation using an FIR filter.

phase distortion can be eliminated by demodulating the signal with a particular function. Secondly, a simple way to generate analysis prototype filters by interpolating tabulated Quadrature Mirror Filters is discussed.

### 3.4.1 Cancellation of Phase Distortion

Referring to Fig. 3.3, if we take into account that, in the analysis filter bank, decimation is performed on a modulated signal, then Eq. 3.8 becomes

$$X_{ki}(z) = \frac{1}{M} \sum_{l=0}^{M-1} H(z^{1/M} W_M^{-l}) X_i(z^{1/M} W_M^{-l} W_K^k). \tag{3.16}$$

Similarly, for the synthesis filter bank and using Eq. 3.12, the output of the system after interpolation can be written as

$$Y_i(z) = \sum_{k=0}^{K-1} Y_{ki}(z^M W_K^{-kM}) G(z W_K^{-k}) W_K^k. \tag{3.17}$$

Here, we focus on the transmission properties of the cascade of an analysis and a synthesis filter bank. Accordingly, we let $X_{ki}(z) = Y_{ki}(z)$; in other words, we do not attempt to separate the mixtures. We can find the transfer function of the overall system by substituting Eq. 3.16 in Eq. 3.17. Thus, we obtain

$$Y_i(z) = \sum_{l=0}^{M-1} T_l(z) X_i(z W_M^{-l}), \tag{3.18}$$

where

$$T_l(z) = \frac{1}{M} \sum_{k=0}^{K-1} W_K^k H(z W_K^{-k} W_M^{-l}) G(z W_K^{-k}). \tag{3.19}$$

For $l = 1, 2, \ldots, M-1$, $T_l(z)$ is approximately zero, provided that aliasing can be neglected. Since the low-pass filters $H(z)$ and $G(z)$ have the same cutoff frequency, $\pi/K$, the passband of $H(zW_K^{-k}W_M^{-l})$ falls in the stopband of $G(zW_K^{-k})$, and vice-versa [41]. Thus, the transfer function of the system can be approximated as follows

$$Y_i(z) \approx T_0(z) X_i(z), \tag{3.20}$$

where, from Eq. 3.19,

$$T_0(z) = \frac{1}{M}\sum_{k=0}^{K-1} W_K^k H(zW_K^{-k})G(zW_K^{-k}). \tag{3.21}$$

Now that we have obtained the transfer function of the overall system, it is shown, as reported in [41], that phase distortion can be eliminated by letting the synthesis filter $g(n)$ be a time flipped version of $h(n)$, i.e.

$$g(n) = h(D-n-1), \quad n = 0, 1, \ldots, D-1, \tag{3.22}$$

and by demodulating the signal by $W_K^{k(n+1)}$. Note that if $h(n)$ has linear phase, then $g(n) = h(n)$ since $h(n)$ will be symmetric with respect to the mid-point of its impulse response. In the z-domain, $g(n)$ can be expressed as

$$G(z) = z^{-(D-1)}H(z^{-1}). \tag{3.23}$$

Using Eq. 3.23, the transfer function $T_0(z)$ in Eq. 3.21 becomes

$$T_0(z) = z^{-(D-1)}\frac{1}{M}\sum_{k=0}^{K-1} W_K^{kD} H(zW_K^{-k})H(z^{-1}W_K^k). \tag{3.24}$$

If we express the transfer function in the frequency domain, we obtain

$$T(e^{j\omega}) = e^{-j(D-1)\omega}\frac{1}{M}\sum_{k=0}^{K-1} |H(e^{j\omega}W_K^{-k})|^2, \tag{3.25}$$

which, obviously, has a linear phase characteristic. Thus, phase distortion is eliminated.

### 3.4.2 Interpolation of Quadrature Mirror Filters

Before considering interpolation of Quadrature Mirror Filters (QMF), let us attempt to cancel amplitude distortion. Designing a filter that cancels amplitude distortion is a difficult task. In this thesis, we use a simple design method based on the interpolation of QMF, which produces filters with an acceptable amount of distortions [41].

To cancel amplitude distortion, the magnitude response of the approximate transfer

function given in Eq. 3.25 must be flat, i.e.

$$\sum_{k=0}^{K-1} |H(e^{j\omega} W_K^{-k})|^2 = 1, \quad \text{for all } \omega. \tag{3.26}$$

We are now facing an optimization problem, which is generally solved using computer-aided methods. At the same time, we would like to have the frequency response of the ideal low-pass filter in Eq. 3.2, and the flatness condition described in Eq. 3.26. However, with FIR filters, these two requirements can only be approximated. In mathematical terms, we want to minimize the error function given by [41]

$$E = \sigma E_s + E_r, \tag{3.27}$$

where $\sigma$ is a positive weighting factor, and

$$E_s = \int_{\omega_s}^{\pi} |H(e^{j\omega})|^2 d\omega \tag{3.28}$$

$$E_r = \int_0^{2\pi} \left( \sum_{k=0}^{K-1} |H(e^{j\omega} W_K^{-k})|^2 - 1 \right) d\omega. \tag{3.29}$$

The first term of Eq. 3.27 refers to the stopband energy ($\omega_s$ being the stopband frequency), and should ideally be zero to avoid aliasing as much as possible. The second term of Eq. 3.27 corresponds to Eq. 3.26, which expresses the desired flatness of the magnitude response so that magnitude distortion is avoided.

Unfortunately, such optimization is not trivial to carry out. So instead of optimizing Eq. 3.27 to generate an appropriate prototype filter, we choose a simpler approach. This approach is described here from an engineering perspective. As reported in [41], performing a $K/2$-point interpolation on a QMF yields a good prototype filter. QMF are well-studied in the literature, and their coefficients have been tabulated in many places, such as in [38]. Interpolation can be achieved using the traditional technique previously described in Sec. 3.3 (insertion of zeros followed by low-pass filtering), but for the design of prototype filters, better characteristics are obtained using DFT interpolation [44], as illustrated in Fig. 3.10. DFT interpolation works as follows. First, the QMF impulse response (of length $D_0$) is transformed into the frequency domain using a $D_0$-point DFT. The resulting

sequence is then split into two sequences of equal length, and $D - D_0$ zeros are inserted between the two sequences, forming a new sequence of length $D = D_0(K/2)$. A $D$-point inverse DFT is finally applied to the new sequence, thereby obtaining the impulse response of the prototype filter, $h(n)$. Filters obtained by this method have linear phase, and should therefore introduce a delay of $(D-1)/2$ in the system.

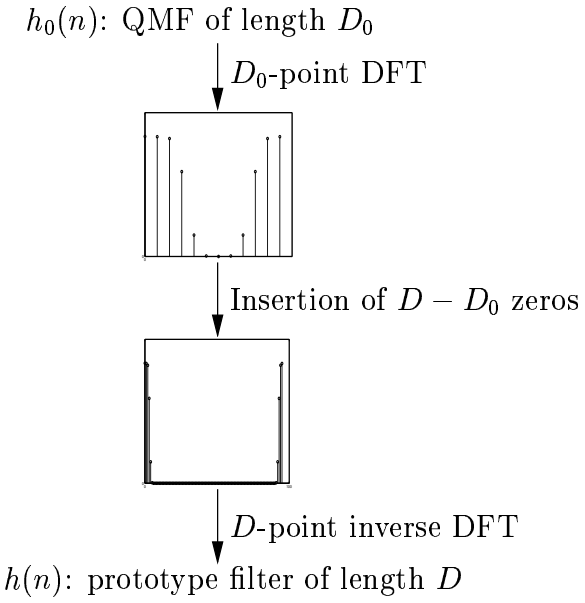



**Fig. 3.10** DFT interpolation of a QMF.

The interpolation factor, $I = D/D_0$, is related to the number of subband channels by the following relation

$$I = \frac{D}{D_0} = \frac{K}{2}, \tag{3.30}$$

since the cutoff frequency of the interpolated prototype is scaled by $1/I$. Moreover, to implement the weighted overlap-add realization as described in Sec. 3.5, $D$ must be a multiple of $K$.

Figure 3.11 compares magnitude responses of a QMF 12 A interpolated by a factor 8 (thus having a normalized cutoff frequency of $0.5/16 = 0.03125$) using both interpolation methods. DFT interpolation generates filters that behave slightly better, especially in the beginning of the stopband. In the case illustrated in Fig. 3.11, there is a difference of 5 dB in the first sidelobes, and about 1 or 2 dB in the rest of the stopband. For subband

BSS, we have designed two prototype filters using DFT interpolation. Their frequency characteristics (magnitude and phase responses) are illustrated in Fig. 3.12. These filters (of length 96 and 192) result from a 8 and 16-point interpolation of a QMF 12 A, and are therefore suitable for a 16 and 32-subband systems, respectively.

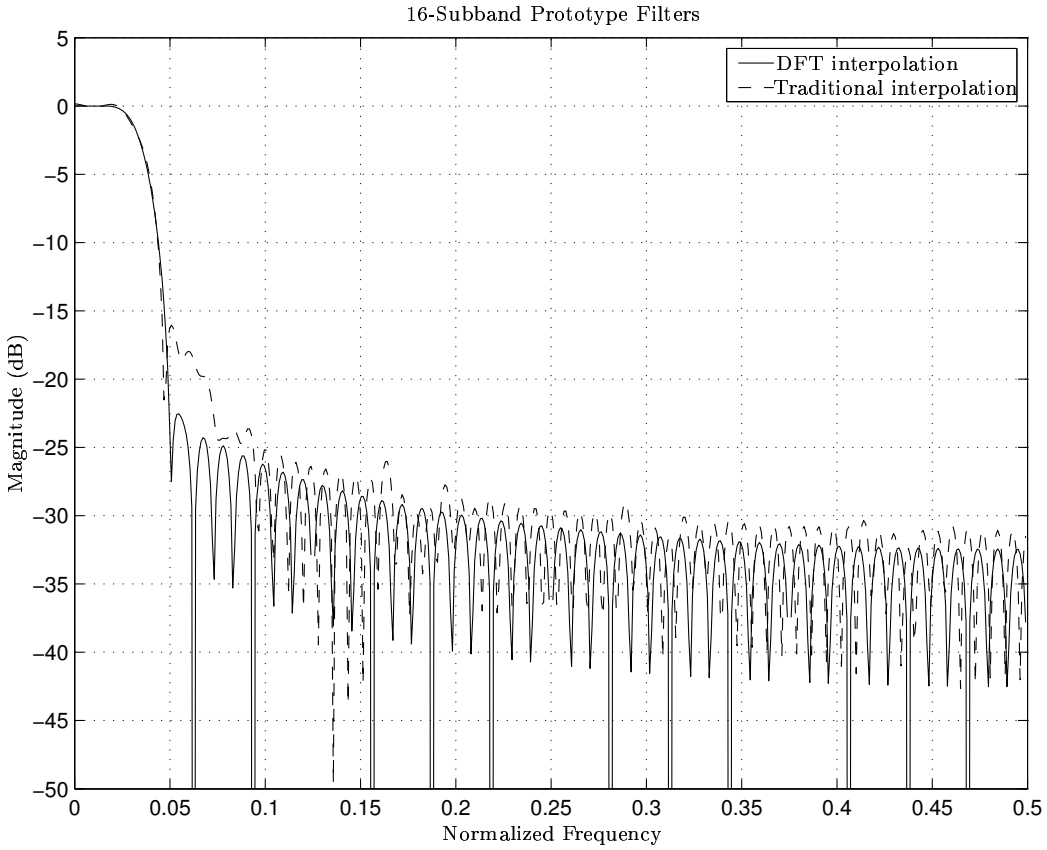



**Fig. 3.11** Comparison between prototype filters generated using traditional and DFT interpolation.

## 3.5 Implementation Using the Weighted Overlap-Add Method

If one implements the DFT filter bank straightforwardly as illustrated in Fig. 3.3, the overhead generated by subband analysis and synthesis can become very expensive, especially if the prototype filter is long. There are different possible solutions to obtain an efficient implementation. For instance, the overhead can significantly be reduced by using a polyphase

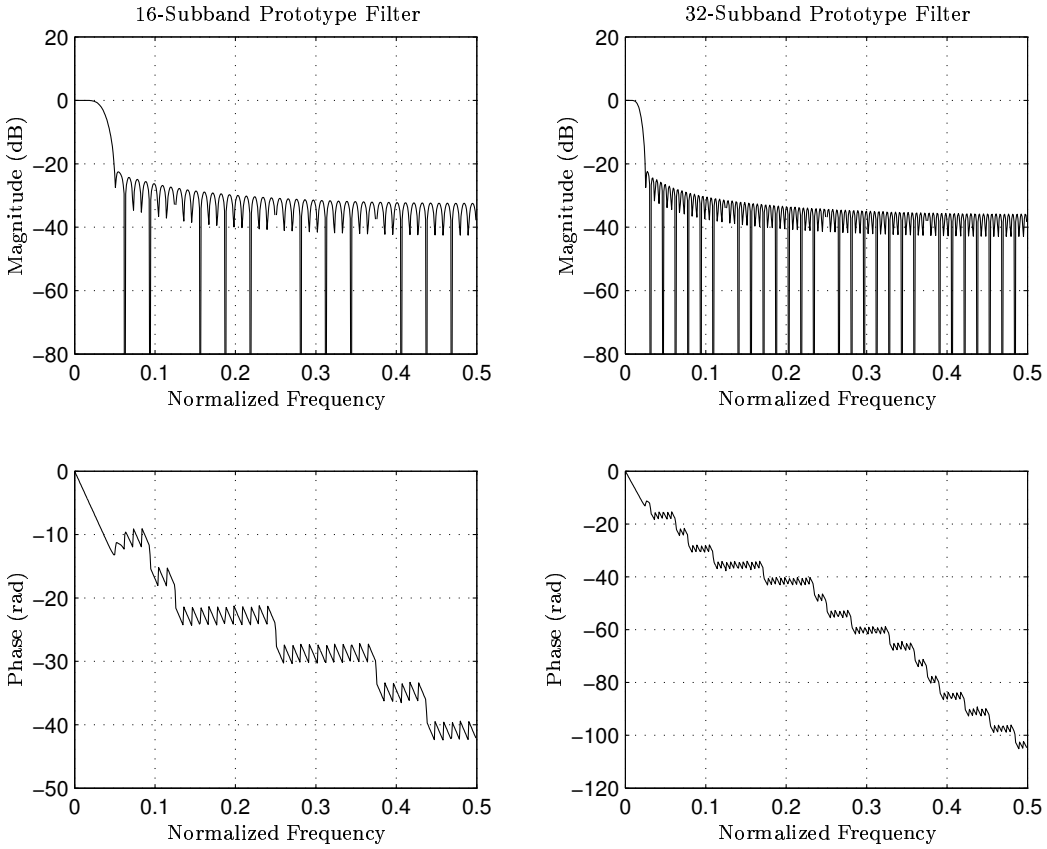



**Fig. 3.12** Prototype filters for 16 and 32-subband BSS systems.

structure or a weighted overlap-add (WOA) realization [38].

The polyphase structure is an efficient realization of a DFT filter bank. It is based on the polyphase decimator and interpolator structures, described in Sec. 3.3, and can be used with IIR or FIR prototype filters. However, the polyphase structure imposes a strict relation between $K$ and $M$. In fact, $K$ must be a multiple of $M$, which severely limits the number of possibilities for an oversampled scheme. Therefore, the polyphase structure is not suitable for this thesis.

As opposed to the polyphase structure, the WOA realization imposes no restriction on $K$ and $M$, but can generally be implemented using FIR filters only. The WOA realization is a more appropriate method for this thesis, because we want to be able to fine-tune $M$ as necessary to reduce distortions. The WOA method interprets the DFT filter bank as a block transform, and uses the fast Fourier transform (FFT) to optimize the computations. Quantitative examples of complexity reduction with the WOA implementation can be found in Sec. 4.3.

The WOA method is fully described in [38], and we will not reproduce the algorithm in this thesis. However, since we use a non-standard function for demodulation, we refer the reader to [41] for a description of WOA that uses this function.

## 3.6 Chapter Summary

We have introduced in this chapter the concept of subband adaptation, which is attractive because it may reduce the amount of computations needed for an adaptive algorithm. We have proposed to use a pair of DFT filter banks to carry out subband analysis and synthesis. All the components of a DFT filter bank were described: modulators, demodulators, decimators and interpolators. A method to generate prototype filters for decimation and interpolation was proposed. Two analysis prototype filters were obtained using DFT interpolation of a QMF 12 A, which results in a filter of length 96 for the 16-subband system, and a filter of length 192 for the 32-subband system. The corresponding synthesis prototype filters were obtained by time-reversing the impulse response of the analysis filters. Finally, we have mentioned that the weighted overlap-add realization is a very efficient structure for a DFT filter bank.

# Chapter 4

# Blind Separation Using Subband Adaptation

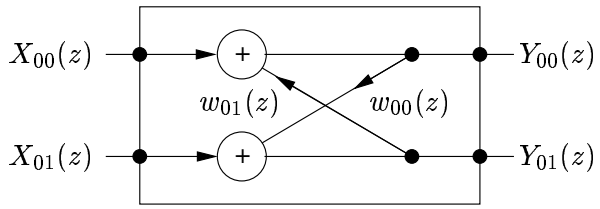
Most of the mechanisms involved in subband adaptation have been explained in Chap. 3. In this chapter, there remains to explain how source separation can be performed in subbands, or to reveal the content of the "Subband BSS" box in Fig. 3.1. But before deriving an adaptation equation, we study the problem of aliasing for a two-subband system in details. The mathematical results hereby obtained justify the need for an oversampled scheme. We then derive an adaptation equation suitable for subband separation, and compare the computational complexity of subband and fullband adaptation. Lastly, the results obtained in this chapter are summarized.

## 4.1 Aliasing in the Subband-Based BSS System

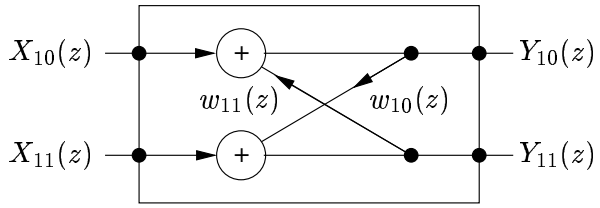
As discussed in Sec. 3.1, subband analysis and synthesis introduce distortions in the reconstructed signals. In this section, we show that aliasing cannot be easily cancelled for a critically sampled BSS system. Simple listening tests confirm that aliasing is a real problem under this condition. As mentioned in Chap. 3, a solution to this problem is to use an oversampled filter bank. The goal of this section is to find an analytic expression for aliasing in the z-domain, and to enumerate possible solutions to cancel aliasing.

### 4.1.1 Derivation of an Expression for Aliasing

Assuming we have a feedback network in each subband channel that performs convolutive BSS, we show that it is not trivial to cancel aliasing for a critically sampled system. To simplify the mathematical analysis, we assume a two-subband system with two microphones, but the results could be generalized for a $K$-subband multi-microphone system. Let us consider the subband-based BSS system with $N = 2$ illustrated in Fig. 3.1. Figure 4.1 illustrates the feedback de-mixing network used in each subband. We define $X_{ij}(z)$ (or $Y_{ij}(z)$) as the z-transform of the $j$-th mixture (or recovered source) in the $i$-th subband, $X_i(z)$ (or $Y_i(z)$) as the the z-transform of the $i$-th fullband mixture (or recovered source), and $H_i(z)$ (or $G_i(z)$) as the z-transform of the analysis (or synthesis) filter of the $i$-th subband. To find an expression for aliasing in the z-domain, we proceed in three steps. We will analyze and write an equation for subband analysis, subband synthesis, and convolutive BSS. By combining these equations together, an expression for aliasing will be found.




(a) Subband 0.




(b) Subband 1.

**Fig. 4.1** Feedback convolutive de-mixing networks used in a two-subband BSS system.

To simplify notation, let us define these three matrices:

$$\boldsymbol{X}(z) = \begin{pmatrix} X_{00}(z) & X_{01}(z) \\ X_{10}(z) & X_{11}(z) \end{pmatrix}, \quad \boldsymbol{H}_m(z) = \begin{pmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{pmatrix}, \quad \boldsymbol{X}_m(z) = \begin{pmatrix} X_0(z) & X_0(-z) \\ X_1(z) & X_1(-z) \end{pmatrix}. \tag{4.1}$$

According to Eq. 3.8, subband analysis corresponds to the following mathematical expression

$$\boldsymbol{X}(z) = \frac{1}{2}\boldsymbol{H}_m(z^{1/2})\boldsymbol{X}_m^T(z^{1/2}). \tag{4.2}$$

For subband synthesis, let us consider the following definitions

$$\tilde{\boldsymbol{Y}}(z) = \begin{pmatrix} Y_0(z) \\ Y_1(z) \end{pmatrix}, \qquad \boldsymbol{Y}(z) = \begin{pmatrix} Y_{00}(z) & Y_{01}(z) \\ Y_{10}(z) & Y_{11}(z) \end{pmatrix}, \qquad \boldsymbol{G}(z) = \begin{pmatrix} G_0(z) \\ G_1(z) \end{pmatrix}. \tag{4.3}$$

Then, according to Eq. 3.12, subband synthesis can be expressed as

$$\tilde{\boldsymbol{Y}}(z) = \boldsymbol{Y}^T(z^2)\boldsymbol{G}(z). \tag{4.4}$$

Convolutive BSS, which uses a feedback network, is not as straightforward to express in matrix form as subband analysis and synthesis. Let us first define

$$A_0(z) = 1 - w_{00}(z)w_{01}(z), \tag{4.5}$$
$$A_1(z) = 1 - w_{10}(z)w_{11}(z). \tag{4.6}$$

Then, according to Fig. 4.1, we have

$$Y_{00}(z) = \frac{1}{A_0(z)}[X_{00}(z) + w_{01}(z)X_{01}(z)], \tag{4.7}$$
$$Y_{01}(z) = \frac{1}{A_0(z)}[X_{01}(z) + w_{00}(z)X_{00}(z)], \tag{4.8}$$
$$Y_{10}(z) = \frac{1}{A_1(z)}[X_{10}(z) + w_{11}(z)X_{11}(z)], \tag{4.9}$$
$$Y_{11}(z) = \frac{1}{A_1(z)}[X_{11}(z) + w_{10}(z)X_{10}(z)], \tag{4.10}$$

which can be written in matrix form by using these definitions

$$\boldsymbol{W}(z) = \begin{pmatrix} w_{00}(z) & w_{01}(z) \\ w_{10}(z) & w_{11}(z) \end{pmatrix}, \qquad \boldsymbol{A}(z) = \begin{pmatrix} \frac{1}{A_0(z)} & 0 \\ 0 & \frac{1}{A_1(z)} \end{pmatrix}. \tag{4.11}$$

Hence, in matrix form, the separating network can be formulated as follows

$$\boldsymbol{Y}(z) = \boldsymbol{A}(z)\big(\boldsymbol{X}(z) + [\boldsymbol{W}(z) \odot \boldsymbol{X}(z)]\boldsymbol{J}\big), \tag{4.12}$$

where $\boldsymbol{J} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ is a permutation matrix, and $\odot$ denotes element-wise multiplication. A mathematical description of aliasing can be obtained by combining Eqs. 4.2, 4.4 and 4.12:

$$\tilde{\boldsymbol{Y}}(z) = \frac{1}{2}\Big(\boldsymbol{X}_m(z)\boldsymbol{H}_m^T(z) + \boldsymbol{J}\big(\boldsymbol{W}^T(z^2) \odot [\boldsymbol{X}_m(z)\boldsymbol{H}_m^T(z)]\big)\Big)\boldsymbol{A}(z^2)\boldsymbol{G}(z). \tag{4.13}$$

### 4.1.2 Possible Solutions to Eliminate Aliasing

In order to see the aliased components more clearly, we can extract $Y_0(z)$ from $\tilde{\boldsymbol{Y}}(z)$ in Eq. 4.13. Again, to simplify notation, let the analysis high-pass filter of subband 1 be a modulated version of the low-pass filter of subband 0, i.e.

$$H_0(z) = H(z), \tag{4.14}$$

$$H_1(z) = H(-z). \tag{4.15}$$

Expanding Eq. 4.13 for $Y_0(z)$ yields

$$\begin{aligned} Y_0(z) = {} & \frac{1}{2}\left(\frac{1}{A_0(z^2)}H(z)G_0(z) + \frac{1}{A_1(z^2)}H(-z)G_1(z)\right)X_0(z) \\ & + \frac{1}{2}\left(\frac{1}{A_0(z^2)}H(-z)G_0(z) + \frac{1}{A_1(z^2)}H(z)G_1(z)\right)X_0(-z) \\ & + \frac{1}{2}\left(\frac{w_{01}(z^2)}{A_0(z^2)}H(z)G_0(z) + \frac{w_{11}(z^2)}{A_1(z^2)}H(-z)G_1(z)\right)X_1(z) \\ & + \frac{1}{2}\left(\frac{w_{01}(z^2)}{A_0(z^2)}H(-z)G_0(z) + \frac{w_{11}(z^2)}{A_1(z^2)}H(z)G_1(z)\right)X_1(-z). \end{aligned} \tag{4.16}$$

The terms $X_0(-z)$ and $X_1(-z)$ represent aliased terms, and must be cancelled. To get rid of these terms, we must have

$$\frac{1}{A_0(z^2)}H(-z)G_0(z) + \frac{1}{A_1(z^2)}H(z)G_1(z) = 0 \tag{4.17}$$

$$\frac{w_{01}(z^2)}{A_0(z^2)}H(-z)G_0(z) + \frac{w_{11}(z^2)}{A_1(z^2)}H(z)G_1(z) = 0. \tag{4.18}$$

Equations 4.17 and 4.18 represent an homogeneous system of linear equations which has only the zero solution, i.e.

$$G_0(z) = 0 \tag{4.19}$$

$$G_1(z) = 0. \tag{4.20}$$

Of course, these are not useful filters in practice, since they eliminate everything, including the non-aliased components.

In short, with a critically sampled filter bank, it is not trivial to cancel aliasing. We have attempted such cancellation by deriving an appropriate mathematical expression for the synthesis filters, and failed. Hence, to eliminate aliasing, we have to look for another solution. The oversampled filter bank, which prevents aliasing in the first place, is a good alternative. Of course, we have to make a compromise, a small $M$ would completely eliminate aliasing, but the system would no longer be very efficient. A good tradeoff, as shown in Chap. 5, appears to be $M = 12$ for a 16-subband system, and $M = 24$ for a 32-subband system.

## 4.2 Adaptive Algorithm for Subband BSS

In this section, an algorithm for subband BSS is developed. A model for bandpass speech is given first, and an adaptation equation is then derived. A short introduction to the source permutation problem is also given.

### 4.2.1 Probability Distribution Function of Bandpass Speech

As reported in Chap. 2, the Laplacian pdf provides a good model for the distribution of speech sources. A gradient for convolutive source separation was developed using the

Laplacian distribution. In this section, we propose a pdf suitable for bandpass speech, based on empirical considerations. This pdf will be used to derive an adaptive algorithm for subband BSS.

Due to the modulation scheme employed, the phase of bandpass speech appears to be uniformly distributed, as illustrated by the phase histogram in Fig. 4.2. Figure 4.2 was obtained by computing the histogram of a three-minute conversation between two persons recorded in a hall, and sampled at 16 kHz. Samples were taken from subband 2 of a 16-subband filter bank, but very similar results were observed in other subbands. Hence, phase can be considered "random" in this context, and the proposed pdf should only depend on the magnitude of the complex random variable.

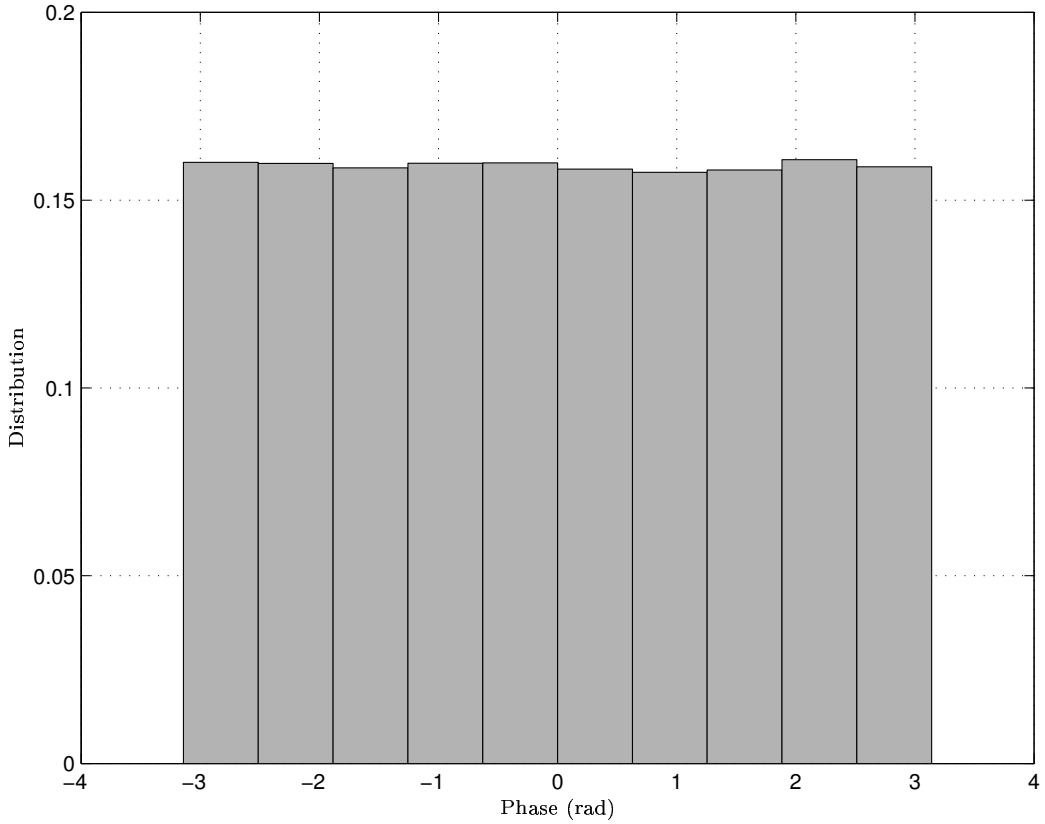



**Fig. 4.2** Histogram illustrating the experimentally observed phase distribution of bandpass speech samples.

The magnitude distribution of bandpass speech, obtained under the same experimental conditions as above, is illustrated in Fig. 4.3, and clearly exhibits a non-uniform distri-

bution. Points on Fig. 4.3 were generated by computing the histogram of the magnitude of bandpass speech samples using very narrow bins. Proceeding as in [45], we also show in Fig. 4.3 a Laplacian and a Gaussian pdf having the same variance as the magnitude distribution of bandpass speech. Due to the many silent parts in the recording used to generate the distribution, both the Laplacian and the Gaussian pdf's do not seem to match the experimental curve well. A better match would have been obtained if all the silent parts in the experimental data were removed. Nevertheless, for the purpose of BSS, since an exact match is not needed, a Laplacian pdf can be a good model for the magnitude of bandpass speech.

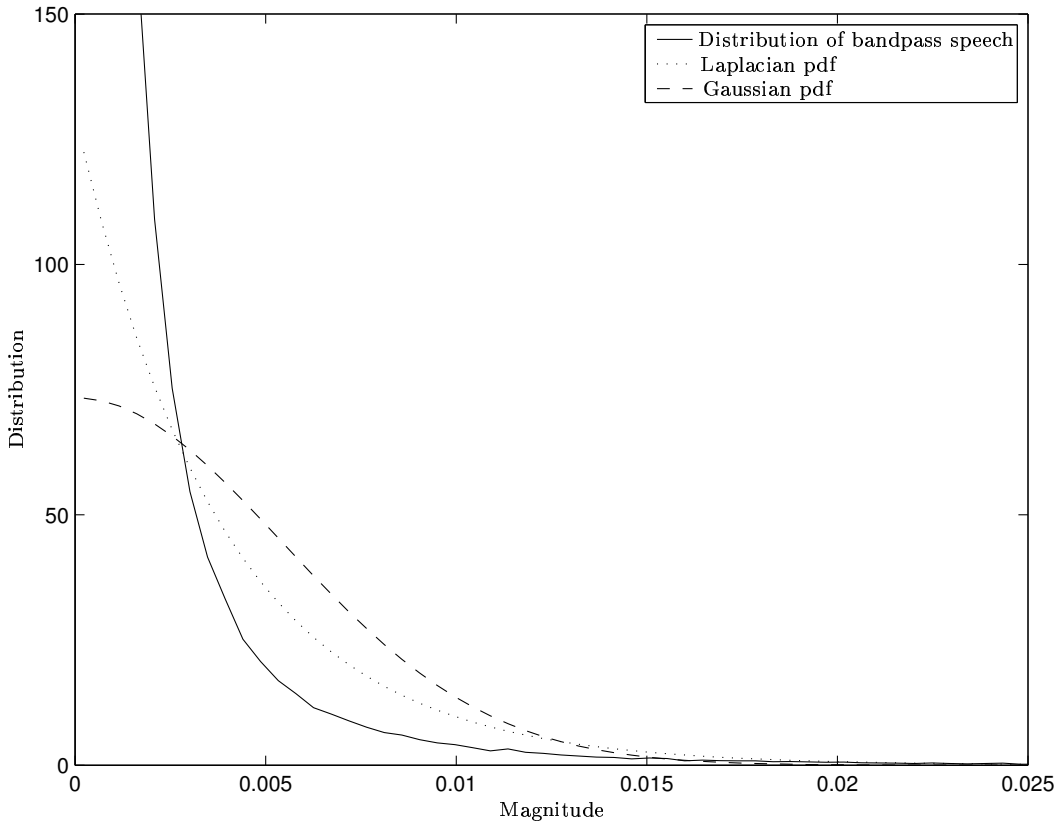



**Fig. 4.3** Experimentally observed magnitude distribution of bandpass speech samples.

Based on the above considerations, we propose to use the following pdf to approximate

the distribution of bandpass speech

$$p_i'\left(y_{ki}^R(m), y_{ki}^I(m)\right) = \alpha \cdot \exp\left(-\alpha\sqrt{\left(y_{ki}^R(m)\right)^2 + \left(y_{ki}^I(m)\right)^2}\right), \tag{4.21}$$

where $\alpha$ is a positive constant, $y_{ki}^R(m) = \Re\{y_{ki}(m)\}$, and $y_{ki}^I(m) = \Im\{y_{ki}(m)\}$. Results presented in Chap. 5 show that the Laplacian model works well for separating speech in subbands.

### 4.2.2 Adaptation Equation

Adaptation equations and activation functions for source separation developed in Chap. 2 are not appropriate for a subband-based separation system, because, among other things, the signals to separate are now complex (except in subband 0 and $K/2$). Complex modulation shifts the signal in frequency, and thus transforms a real signal into a complex one. Literature on complex convolutive BSS is very sparse. For instance, in [46], Smaragdis proposed an adaptive scheme to separate complex signals in the frequency domain, but the results cannot be easily transposed to subband adaptation.

To derive a separation algorithm, there are two factors that must be taken into consideration. A complex pdf that models the sources closely enough has to be provided (a pdf was proposed in Sec. 4.2.1), and a complex gradient operator must be used to find an adaptive algorithm. Fortunately, the cost function obtained in Chap. 2 (Eq. 2.24) can still be applied, since it only involves the pdf of the sources, whose range is always real-valued, regardless whether the sources are real or complex.

Let us use a feedback de-mixing network to separate the sources in each subband, as illustrated in Fig. 2.4 (as before, the self-connecting loops are ignored). We denote each filter by $\boldsymbol{w}_{kij}$ instead of $\boldsymbol{w}_{ij}$, because $k$ is needed to indicate to which subband the filter refers to. To simplify the application of the complex gradient operator, notation used in this section differs from Chap. 2. Let $\boldsymbol{y}_{ki}(m) = [y_{ki}(m), y_{ki}(m-1), \ldots, y_{ki}(m-L+1)]^T$,

and

$$x_{ki}^R(m) = \Re\{x_{ki}(m)\}, \quad y_{ki}^R(m) = \Re\{y_{ki}(m)\}, \quad \boldsymbol{y}_{ki}^R(m) = \Re\{\boldsymbol{y}_{ki}(m)\}, \quad \boldsymbol{a}_{kij} = \Re\{\boldsymbol{w}_{kij}\}, \tag{4.22}$$

$$x_{ki}^I(m) = \Im\{x_{ki}(m)\}, \quad y_{ki}^I(m) = \Im\{y_{ki}(m)\}, \quad \boldsymbol{y}_{ki}^I(m) = \Im\{\boldsymbol{y}_{ki}(m)\}, \quad \boldsymbol{b}_{kij} = \Im\{\boldsymbol{w}_{kij}\}. \tag{4.23}$$

Using this notation, the output of the feedback de-mixing network becomes

$$\begin{aligned} y_{ki}(m) &= x_{ki}(m) + \sum_{j\neq i} \boldsymbol{w}_{kij}^H \boldsymbol{y}_{kj}(m) \\ &= x_{ki}^R(m) + j x_{ki}^I(m) + \sum_{j\neq i} (\boldsymbol{a}_{kij}^T - j\boldsymbol{b}_{kij}^T)(\boldsymbol{y}_{kj}^R(m) + j\boldsymbol{y}_{kj}^I(m)), \end{aligned} \tag{4.24}$$

which can be separated into real and imaginary components as follows

$$y_{ki}^R(m) = x_{ki}^R(m) + \sum_{j\neq i} (\boldsymbol{a}_{kij}^T \boldsymbol{y}_{kj}^R(m) + \boldsymbol{b}_{kij}^T \boldsymbol{y}_{kj}^I(m)), \tag{4.25}$$

$$y_{ki}^I(m) = x_{ki}^I(m) + \sum_{j\neq i} (\boldsymbol{a}_{kij}^T \boldsymbol{y}_{kj}^I(m) - \boldsymbol{b}_{kij}^T \boldsymbol{y}_{kj}^R(m)). \tag{4.26}$$

According to Eq. 2.45, the cost function is based on the Jacobian of the network. For a feedback network, the Jacobian is given by Eq. 2.44. Substituting Eq. 4.21, the pdf of bandpass speech, in Eq. 2.45 yields the following cost function

$$\begin{aligned} \log|J| &= \sum_{i=1}^{N} \log p_i'\left(y_{ki}^R(m), y_{ki}^I(m)\right) \\ &= -\alpha \sum_{i=1}^{N} \sqrt{\left(y_{ki}^R(m)\right)^2 + \left(y_{ki}^I(m)\right)^2} + N\alpha. \end{aligned} \tag{4.27}$$

Note that the cost function is a real function, but the filters to adapt are complex. Maximization of the cost function can be done using the complex gradient operator as defined in [47]. By definition, the gradient is equal to

$$\nabla \log|J| = \frac{1}{2}\left(\frac{\partial \log|J|}{\partial \boldsymbol{a}_{kij}} + j\frac{\partial \log|J|}{\partial \boldsymbol{b}_{kij}}\right). \tag{4.28}$$

To evaluate the gradient, we proceed in two steps, calculating the real and imaginary part separately. Thus, the real part of the gradient corresponds to

$$\frac{\partial \log |J|}{\partial \boldsymbol{a}_{kij}} = -\alpha \frac{y_{ki}^R(m)\boldsymbol{y}_{kj}^R(m) + y_{ki}^I(m)\boldsymbol{y}_{kj}^I(m)}{\sqrt{\left(y_{ki}^R(m)\right)^2 + \left(y_{ki}^I(m)\right)^2}}, \tag{4.29}$$

and the imaginary part amounts to

$$\frac{\partial \log |J|}{\partial \boldsymbol{b}_{kij}} = -\alpha \frac{y_{ki}^R(m)\boldsymbol{y}_{kj}^I(m) + y_{ki}^I(m)\boldsymbol{y}_{kj}^R(m)}{\sqrt{\left(y_{ki}^R(m)\right)^2 + \left(y_{ki}^I(m)\right)^2}}. \tag{4.30}$$

An expression for the gradient of the cost function can be obtained by combining Eqs. 4.28, 4.29 and 4.30:

$$\begin{aligned} \nabla \log |J| &= -\frac{\alpha}{2} \frac{y_{ki}^R(m) - j y_{ki}^I(m)}{\|y_{ki}(m)\|} \boldsymbol{y}_{kj}(m) \\ &= -\frac{\alpha}{2} \frac{y_{ki}^*(m)}{\|y_{ki}(m)\|} \boldsymbol{y}_{kj}(m). \end{aligned} \tag{4.31}$$

Hence, using a feedback network, complex coefficients of the separating filters can be updated with the following scheme

$$\boldsymbol{w}_{kij}(m+1) = \boldsymbol{w}_{kij}(m) - \mu \frac{y_{ki}^*(m)}{\|y_{ki}(m)\|} \boldsymbol{y}_{kj}(m), \tag{4.32}$$

where $\mu$ is the step size. The step size is defined as

$$\mu = \mu' \frac{\alpha}{2}, \tag{4.33}$$

where $\mu'$ a small real positive constant. An optimal value for $\mu$ depends on several factors, such as $K$, the number of subbands, and $L$, the de-mixing filter length. Some examples are given in Chap. 5. Note that since the sampling rate has been reduced by $M$, the number of taps we would use for fullband adaptation, $L$, can be cut down by $M$ for subband adaptation.

In short, to implement BSS based on a feedback de-mixing network in subbands, we need two main equations, namely the output generation and weight update equations,

which are given in Tab. 4.1.

| | |
|---|---|
| Output generation | $y_{ki}(m) = x_{ki}(m) + \sum_{j \neq i} \boldsymbol{w}_{kij}^H \boldsymbol{y}_{kj}(m)$ |
| Weight update | $\boldsymbol{w}_{kij}(m+1) = \boldsymbol{w}_{kij}(m) - \mu \frac{y_{ki}^*(m)}{\|y_{ki}(m)\|} \boldsymbol{y}_{kj}(m)$ |

**Table 4.1** Output generation and weight update equations for subband BSS.

### 4.2.3 Source Permutations

As observed in Chap. 2, at the output of a BSS system, the recovered sources can be permuted with respect to the original sources. In fact, the system cannot retrieve information about the original location of the sources, since both the sources themselves and the mixing system remain unknown when separation is attempted. Therefore, when subband adaptation is used, there is no guarantee that, for each subband channel, the sources follow the same permutation. When the fullband signal is reconstructed, frequency bands belonging to different sources could be added together, and the resulting signal would not appear to be separated. In addition, the subband signal could be corrupted by a gain ambiguity. However, several experiments, conducted with two sources recorded in a real-room environment, let us believe that the problem of different source permutations across each subband occurs rarely.

In all the experiments conducted so far with a subband-based BSS system, the same permutation in all subbands has always occurred. We were able to note this fact by listening to each subband individually. Moreover, if one subband with a reasonable amount of energy is deliberately permuted with respect to the others, listening tests confirm that the resulting signal does not seem separated.

A thorough study of source permutations in subbands is beyond the scope of this thesis. As mentioned in Chap. 6, the conclusion, it will be classified as possible future work.

## 4.3 Computational Complexity

We compare in this section the computational complexity of fullband and subband BSS. Computational complexity is measured in terms of the number of real multiplications required to process an array of $N$ samples. The number of multiplications will be expressed

in terms of several parameters that characterize the system. These parameters are recalled in Tab. 4.2.

| Symbol | Description |
|---|---|
| $N$ | Number of sources |
| $L$ | Number of taps used for fullband adaptation |
| $K$ | Number of subbands |
| $M$ | Decimation and interpolation factor |
| $D$ | Prototype filter length |

**Table 4.2** Some symbols and their description.

For the fullband system, we consider a feedback de-mixing network. The number of real multiplications that this network requires is $2N(N-1)L$. For the subband system, we have to take into account the overhead generated by the filter banks. Systems with and without a WOA realization are considered. Table 4.3 gathers the number of computations for the various components of a system without a WOA realization. Based on the results reported in [41], the number of real multiplications for a system using a WOA realization can be found in Tab. 4.4. Using the results presented in Tabs. 4.3 and 4.4, a computational gain can be defined by considering the amount of computations needed for fullband adaptation versus subband adaptation. If we do not take into account the overhead due to the filter banks, the ratio is

$$\gamma_0 = \frac{2}{3} \cdot \frac{M^2}{K}. \tag{4.34}$$

If we were using a critically sampled scheme ($M = K$), then $\gamma_0$ would be directly proportional to the number of subbands, $K$. But with an oversampled scheme, the choice of $M$ is a compromise between computational efficiency and aliasing reduction. Furthermore, according to Eq. 4.34, to increase the gain, one simply has to use more subbands for subband adaptation. However, if we take into account the overhead introduced by the filter banks, as the number of subbands increases, there is a point where the filter banks become more expensive than BSS. Thus, if we take into consideration the overhead, the following

computational gains can be found

$$\gamma_{WOA} = \frac{M^2}{K} \cdot \frac{2(N-1)L}{2(D/K + \log_2 K) + 3(N-1)L} \tag{4.35}$$

$$\gamma = \frac{M^2}{K} \cdot \frac{2(N-1)L}{2M(M+D) + 3(N-1)L}, \tag{4.36}$$

where $\gamma_{WOA}$ and $\gamma$ denote the computational gains of the subband systems with and without the WOA realization, respectively.

| | Real multiplications |
|---|---|
| Modulation and demodulation | $NK$ |
| Decimation and interpolation | $(2NKD)/M$ |
| Output generation | $[2N(N-1)LK]/M^2$ |
| Weight update | $[N(N-1)LK]/M^2$ |
| Total | $2NK(1+D/M)$ $+3[N(N-1)LK]/M^2$ |

**Table 4.3** Number of real multiplications needed for subband BSS without a WOA realization.

| | Real multiplications |
|---|---|
| WOA | $2N(D+K\log_2 K)/M$ |
| Output generation | $[2N(N-1)LK]/M^2$ |
| Weight update | $[N(N-1)LK]/M^2$ |
| Total | $2N(D+K\log_2 K)/M$ $+3[N(N-1)LK]/M^2$ |

**Table 4.4** Number of real multiplications needed for subband BSS with a WOA realization.

We now present numerical examples of the computational complexity. For $N = 2$ and $L = 1152$, we have evaluated the number of multiplications required by a fullband system, a 16-subband system ($K = 16$, $M = 12$ and $D = 96$), and a 32-subband system ($K = 32$, $M = 24$ and $D = 192$), with and without a WOA realization. The results are presented in Tab. 4.5. In addition, the computational gains are given in Tab. 4.6. Due to the long filters used for subband analysis and synthesis, we note that, according to the results of

Tab. 4.6, subband adaptation necessitates a huge overhead if the WOA realization is not implemented.

| | Real multiplications |
|---|---|
| Fullband BSS | 4608 |
| Subband BSS, with WOA, 16-subband | 822 |
| Subband BSS, with WOA, 32-subband | 443 |
| Subband BSS, without WOA, 16-subband | 1344 |
| Subband BSS, without WOA, 32-subband | 1536 |

**Table 4.5** Comparison between the number of multiplications required for fullband BSS and subband BSS (16 and 32-subband implementations).

| | 16-subband | 32-subband |
|---|---|---|
| $\gamma_0$ | 6.00 | 12.00 |
| $\gamma_{WOA}$ | 5.97 | 11.92 |
| $\gamma$ | 3.42 | 3.00 |

**Table 4.6** Computational gains for a 16 and a 32-subband systems.

## 4.4 Chapter Summary

We have described in this chapter a subband-based BSS system using a pair of DFT filter banks. A 16 and a 32-subband systems were proposed. The 16-subband system uses a decimation factor of 12 to reduce aliasing to an acceptable level, and good results were obtained with a factor of 24 for the 32-subband system. These results are justified experimentally in Chap. 5. Subband separation is performed with a feedback de-mixing network, and, since the signals are complex, we had to use a complex gradient operator to derive the adaptation equation. Due to the long prototype filters, using a WOA realization is recommended for an efficient implementation. We have noted that subband systems are more efficient than fullband systems in terms of computational cost.

# Chapter 5

# Experimental Results

This chapter is concerned with the performance analysis of the proposed BSS systems. The assessment methodology is described first, and different performance results are then given and discussed. Finally, a summary of this chapter is presented.

## 5.1 Methodology

This section begins with a survey of the techniques proposed in the literature to assess the performance of BSS systems. Limitations of these techniques are pointed out, and an evaluation procedure that overcomes some of these limitations is described. We then explain how data for performance evaluation was generated. Finally, we make some comments about the programming aspect of the simulations conducted in this chapter.

### 5.1.1 Performance Measures

Performance evaluation of BSS systems is far from being standardized, even if there is some effort made toward that direction [48]. Many researchers working on BSS have their own approach to quality assessment. As a result, it has become difficult to compare the BSS algorithms proposed in the literature, because of the different measures and data sets which are used. We propose in this section to use the method suggested in [48]. As described in this section, this method provides performance rates based on separation and distortion scores using real recordings.

There are two main categories of data that can be used for performance evaluation:

synthetic and real data. Synthetic data are produced by artificial mixing, using, for example, the impulse responses of a room at various positions. Considering Fig. 5.1, if artificial mixing is used, the sources $s_i$, the mixtures $x_i$, and the recovered sources $y_i$ are available. In this case, performance measurement is straightforward, because we can directly compare the original sources $s_i$ with the recovered sources $y_i$, and compute a distance between the two signals. Another way to measure performance would be to convolve the impulse responses of the modelled room with the de-mixing filters, and measure how similar the responses are to a pure delay system. However, convolving the impulse responses is not necessarily a good indication of separation quality. For example, if the sources have very low energy in a given frequency range, the system may fail to identify separating filters for this range, but separation would still be successful [48].

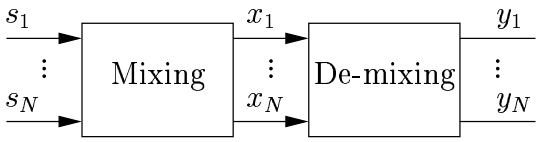



**Fig. 5.1** Mixing and de-mixing systems.

It is easy to measure performance with synthetic data, but there are always some characteristics of a real environment that cannot be faithfully reproduced in a controlled environment. Using real recordings provides more realistic conditions, but the sources are no longer available, which makes performance assessment more difficult. Among the performance assessment tools proposed in the literature, waveform or spectrogram plots of the separated speech signals are frequently used. However, such plots do not reveal much information about separation quality, and, in many cases, cannot even confirm that the system is really separating the sources properly. Scores given by automatic speech recognizers are also a popular way to measure performance of BSS systems. But in fact, recognition scores do not provide an accurate performance measure, because they depend on many factors other than separation quality. For instance, even if speech is perfectly separated, room reverberation could distort the signal so that recognition scores are very poor.

Schobben *et al.* proposed in [48] an evaluation method that combines the best of both worlds: the realism of real room recordings, and the possibility of getting performance

measures as with artificial mixing. The idea is to record the sources in a real-world environment, while letting only one of them active at a time. The mixtures are then obtained by adding all contributions together, i.e.

$$x_i(n) = \sum_{j=1}^{N} x_{i,s_j}(n), \quad i = 1, 2, \ldots N, \tag{5.1}$$

where $x_{i,s_j}(n)$ denotes the contribution of speaker $j$ to microphone $i$. Since sound waves combine themselves in an additive manner, Eq. 5.1 seems plausible from a physical point a view. Similarly, $y_{i,s_j}(n)$ denotes the recovered source $i$ when speaker $j$ is active. Note that all the persons participating in the recording must be present in the room, even if they are recorded in turn, because the mere presence of a person in a room can change its impulse response. This evaluation method is limited to the case where speakers are immobile. In practice, it would be difficult to recreate the exact same movement more than once, unless very thorough precautions are taken.

When the sources are recorded separately, separation quality can be given by the following power ratio in decibels

$$S_i = 10 \log \left( \frac{E\left[y_{i,s_i}^2(n)\right]}{E\left[\sum_{j \neq i} y_{i,s_j}^2(n)\right]} \right). \tag{5.2}$$

Hence, as the power of source $i$ becomes stronger compared to the power coming from the other sources, separation quality is better, and $S_i$ increases. In addition to separation quality, there is a second parameter that can be computed for performance assessment: distortion of the signal. Indeed, BSS is not very useful if a signal is well separated (all alien sources are inaudible), but is so distorted that it cannot be understood. Distortion can be measured using

$$D_i = 10 \log \left( \frac{E[(x_{i,s_i}(n) - \lambda_i y_i(n-d))^2]}{E[x_{i,s_i}^2(n)]} \right), \tag{5.3}$$

where $\lambda_i = E[x_{i,s_i}^2(n)]/E[y_i^2(n)]$, and $d$ is a delay introduced by the BSS system (in [48], no delay was considered). Since the prototype filters have linear phase, the delay is given by

$$d = 2\left(\frac{D-1}{2}\right) - M = D - 1 - M, \tag{5.4}$$

which implies that $d = 95 - 12 = 83$ for a 16-subband system, and $d = 191 - 24 = 167$ for a 32-subband system. According to Eq. 5.3, distortion is minimal if the recovered source $y_i(n-d)$ is equal to $x_{i,s_i}(n)$, up to a gain factor $\lambda_i$. As a consequence, any dereverberation attempt made by the BSS system will be considered as distortion, since the reference signal is $x_{i,s_i}(n)$, and not $s_i(n)$ (the sources are not directly available). Note that these definitions can be used for subband and fullband systems, since they only refer to input and output signals.

The evaluation method that we have described above is not perfect. Measuring quality of speech in decibels can be misleading, because it does not take into account the psychoacoustic properties of the human ear. For instance, two speech signals having the same separation rate in decibels can be perceived very differently in terms of separation quality by a human listener. In Sec. 5.2.3, we will compare the separation rates given by Eq. 5.2 to scores obtained by subjective testing. Differences will be noticed.

### 5.1.2 Data Generation

To test the BSS systems with the method described in the previous section, we have made three data sets. Each set of data comprises four audio recordings that are obtained in a hard walled office room, which is not acoustically isolated. Two microphones are used to record two live speakers who are counting aloud from one to ten in English and French. As mentioned in Sec. 5.1.1, we record each source separately. Hence, each person speaks in turn, and, at the end, we obtain four audio files, which represent:

- Contribution of Speaker 1 to Microphone 1,
- Contribution of Speaker 2 to Microphone 1,
- Contribution of Speaker 1 to Microphone 2,
- Contribution of Speaker 2 to Microphone 2.

The three data sets differ by the position of the speakers, as shown in Fig. 5.2. In Position $a$, both speakers are standing back-to-back in the middle of the room, facing each microphone directly. The direct path is thus strong, cross-talk is weak, and recorded speech should be easy to separate. In Position $b$, both speakers are side-by-side near the right wall, directly facing the microphones. This position should still offer a strong direct path, but

cross-talk should be stronger than in Position $a$. In Position $c$, the two speakers are facing each other at each end of the room. Cross-talk is very strong in this position, and the direct path is not as powerful as in the other positions. Speech recorded in this position should be difficult to separate. In all positions, in addition of speech, a certain amount of noise is also picked up by the microphones. We have identified three major sources of noise:

- Noise produced by the computer fan present in the room,
- Noise generated internally by the recording equipment,
- Noise coming from the street near the room (car, people).

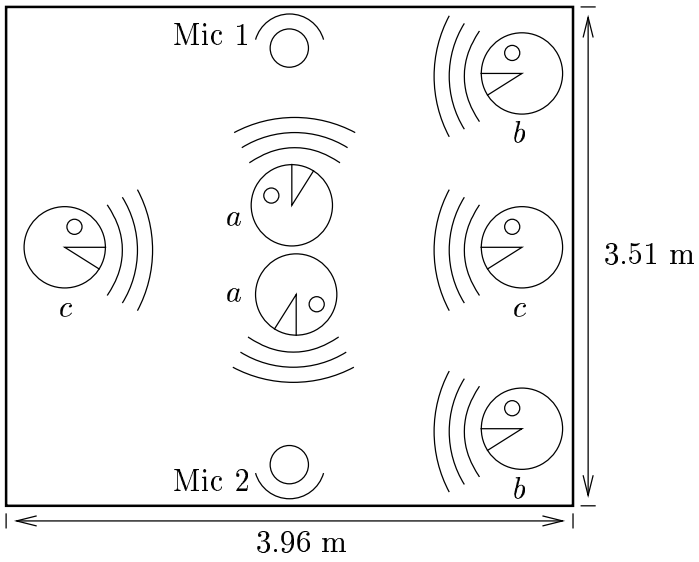



**Fig. 5.2** Speaker positions in the recording room.

Table 5.1 lists the equipment used to make the recordings. Note that the microphones are omnidirectional, and incorporate a high-pass filter that limits frequency components below 60 Hz, which helps to reduce noise generated by the power lines. The analog signal is converted to a digital signal using a sampling frequency of 11.025 kHz, and a 16-bit uniform quantizer.

| Equipment | Manufacturer | Model |
|---|---|---|
| Microphones | Audio-Technica | AT8531 |
| Microphone mixer | Tascam | MX-80 |
| A/D converter | M-Audio | Delta 1010 |

**Table 5.1** Equipment used for data recording.

### 5.1.3 Simulation Programming

For comparison purposes, we have implemented a fullband and a subband system. The fullband system uses a feedback de-mixing network. Coefficients of the network are adapted according to Eq. 2.41 in conjunction with the gradient derived in Eq. 2.46. Output samples are generated with Eq. 2.43. As for the subband system, it is implemented using feedback de-mixing networks. Filter weights are adapted with Eq. 4.32, and output samples are processed according to Eqs. 4.25 and 4.26.

The C programming language was used to implement the BSS systems, because of speed issues. However, the prototype filters were designed using the *MATLAB* technical computing environment. Since the C programming language (ISO C89) does not provide any facility to easily represent complex numbers, an external library, the *GNU Scientific Library* [49] was used for this purpose. Sound file manipulations (loading and saving) were carried out using the *Audio File Library* [50], which provides a uniform interface to access the most popular audio file formats. All the samples are converted internally to 64-bit floating-point representations, and are scaled to $(-1.0, 1.0)$ before processing.

Moreover, the software has the option of doing several iterations on the same set of files. Of course, between each iteration, coefficients of the separating network are not reset to zero; adaptation continues with the current weight values. As described in Sec. 5.1.1, to assess performance of the system, the software expects the sources to be recorded separately. Contributions to each microphone are added internally by the software, and the network is adapted with respect to the resulting signal. At the end, both the separation and distortion rates are computed and displayed to the user.

To implement the WOA realization, the FFT routines provided by the *FFTW* library [51] are used. These FFT's are optimized for real data, and will only compute the first $1 + K/2$ samples of the transformation. The other half is never used, since it does not contain pertinent information.

## 5.2 Results and Discussion

In this section, we assess the performance of the BSS systems in regard to several aspects. Performance rates and computational speed of the fullband and subband systems are given, and briefly discussed. We then analyze the distortion in subband systems using different decimation factors. Finally, results of subjective testing are presented, and compared to those previously obtained.

### 5.2.1 Performance of Fullband and Subband BSS Systems

We have tested the performance of the fullband and subband systems by measuring the separation and distortion rates as defined in Eqs. 5.2 and 5.3, respectively. Results are gathered in Tab. 5.2, and were obtained for speakers in Position $a$, $b$ and $c$ (see Fig. 5.2). Two numbers, separated by a '/', are given for each entry. These numbers correspond to the first and second source. The performance rates reported in Tab. 5.2 were computed after 8 iterations (i.e. each set of files was processed 8 times). The step size $\mu$, the number of filter taps in the feedback de-mixing network, and the decimation factor $M$ that were used to separate the sources are also given in the table.

| | | | Subband BSS | |
|---|---|---|---|---|
| | Position | Fullband BSS | 16-subband | 32-subband |
| Separation (dB) | $a$ | 9.12/7.29 | 9.16/7.52 | 9.48/7.83 |
| | $b$ | 8.05/4.85 | 7.99/4.64 | 8.50/4.74 |
| | $c$ | 3.33/2.27 | 7.11/6.00 | 6.21/6.10 |
| Distortion (dB) | $a$ | -8.34/-6.80 | -8.50/-6.92 | -8.56/-7.00 |
| | $b$ | -7.78/-5.05 | -7.63/-4.91 | -7.94/-4.96 |
| | $c$ | -3.65/-3.32 | -6.24/-5.78 | -5.55/-4.94 |
| $\mu$ | | $1 \times 10^{-4}$ | $5 \times 10^{-3}$ | $5 \times 10^{-2}$ |
| taps | | 1152 | 96 | 48 |
| $M$ | | 1 | 12 | 24 |

**Table 5.2** Performance of BSS systems.

We have tried to choose the best step size for each system, but more fine-tuning is possible. The performance of each system appears to be similar, except for Position $c$. In this case, distortion and separation rates of the fullband system are much worse than those of the subband systems. But if we choose $\mu = 1 \times 10^{-5}$ for the fullband system in

Position $c$, we obtain separation rates of 6.08/4.27 dB, instead of 3.33/2.27 dB, which are closer to the rates of the subband systems. Informal listening tests described in Sec. 5.2.3 confirm that the sources in Tab. 5.2 are separated to a certain extent.

Furthermore, we may note that the position of the speakers relative to the microphones has a direct influence on the performance rates. Better separation and lower distortion are obtained when the direct path is strong, and cross-talk is weak. Recall that the strongest direct path is obtained in Position $a$, whereas the strongest cross-talk can be found in Position $c$.

Computational complexity is given in Tab. 5.3. These results represent the average processing time of one iteration of 10-second speech files, and were obtained with the same parameters used in Tab. 5.2. A *Pentium 3* processor clocked at 933 MHz was used to carry out the computations. Gains in terms of processing times with respect to the fullband system are also given. Theoretical gains are reproduced here from Tab. 4.6 for comparison purposes.

| | | Time (s) | Real gain | Theoretical gain |
|---|---|---|---|---|
| | Fullband BSS | 60.1 | – | – |
| 16-subband | With WOA | 10.4 | 5.79 | 5.97 |
| | Without WOA | 16.1 | 3.73 | 3.42 |
| 32-subband | With WOA | 5.4 | 11.05 | 11.92 |
| | Without WOA | 17.6 | 3.41 | 3.00 |

**Table 5.3** Computational speed and time gains of two-input two-output BSS systems.

A strong correlation between the theoretical and real gains can be noted. Of course, an exact match was not expected, since the theoretical gains are obtained by counting (approximately) the number of multiplications. There are many factors other than the number of multiplications that influence the computational speed of a program running on a digital processor. For example, pipelines, caches, and operations like indexing, looping and branching, affect the computational speed of a program. The 32-subband WOA realization was the fastest system, being 11 times faster than the corresponding fullband system, and 1.9 times faster than the corresponding 16-subband system. Hence, the goal of reducing the number of computations is reached, without a significant impact on the separation and distortion rates.

### 5.2.2 Distortions in Subband-Based Systems

We are now interested in the distortion, as defined in Eq. 5.3, produced by subband systems. Figure 5.3 illustrates the distortion for 16 and 32-subband systems as a function of $M$, the decimation factor, when speakers are in Position $a$ . Results were obtained using $\mu = 0$, that is, no separation is attempted.

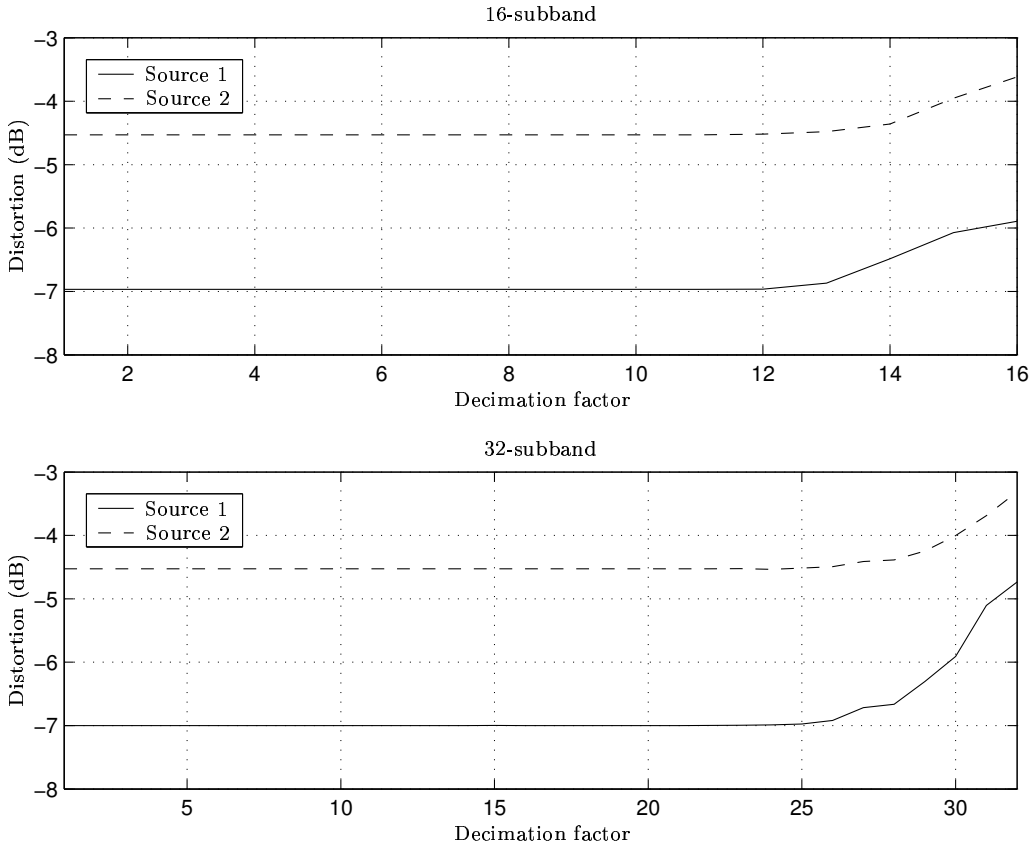



**Fig. 5.3** Distortion in the subband systems.

In both systems the distortion increases as the decimation factor increases. In fact, as $M$ approaches $K$, the distortion increases steeply, but below a certain point, it remains steady. By setting $M = 12$ for a 16-subband system, and $M = 24$ for a 32-subband system, the steep distortion increase toward the end is avoided. Choosing a smaller $M$ would not reduce distortion further, but can severely slow down the computational speed. Informal listening tests performed in Sec. 5.2.3 confirm that, with such $M$'s, distortions remain below

an acceptable level.

### 5.2.3 Subjective Testing

Informal subjective tests involving human listeners were conducted to ensure that the results given by Eqs. 5.2 and 5.3 are meaningful. These tests were divided in two parts, and attempt to validate the results presented in Tab. 5.2 and Fig. 5.3.

The first part of the test was carried out as follows. For each separation quality score in Tab. 5.2, subjects listened to the mixtures and to the recovered speech files. They were then asked to select which one appeared to be the most separated. Four persons participated to the test, and results are given in Tab. 5.4. In this table, we have indicated the number of listeners who chose the output of the BSS system as the most separated signal. Again, two numbers, separated by a '/', are given for each entry, corresponding to the first and second source. The percentages in the last row of Tab. 5.4 are obtained by adding the pairs of results for Positions $a$, $b$ and $c$, and by dividing the sum by 24.

| | | Subband BSS | |
|---|---|---|---|
| Position | Fullband BSS | 16-subband | 32-subband |
| $a$ | 4/2 | 4/4 | 4/3 |
| $b$ | 3/3 | 4/2 | 1/3 |
| $c$ | 2/3 | 4/2 | 4/4 |
| Total | 71 % | 83 % | 79 % |

**Table 5.4** Number of listeners (out of 4) who found that the processed file was the most separated.

In general, people thought that the most separated signal was indeed the signal collected at the output of the BSS systems. However, the number of listeners is somewhat low in two cases. In Position $c$, with the fullband system, a score of 2/3 was obtained. The corresponding separation rates are 3.33/2.27 dB, as indicated in Tab. 5.2, which are the lowest rates that were obtained. Hence, in this case, both the separation rates and subjective tests indicate that separation was not very successful. Another low score, 1/3, was obtained in Position $b$ with the 32-subband system. However, the corresponding separation rates, 8.50/4.74 dB, is in the same range as the rates obtained in Position $b$ by the other BSS systems. Thus, measuring separation quality in decibels has its limitations. Good

separation rates can sometimes be misleading, since listeners may feel that, after all, the resulting signal is not well separated.

In the second part of the test, we asked the subjects to compare two speech signals in terms of distortion, and select the less distorted one. Speech files were processed by the 16 and 32-subband systems using $M = 1, 12, 16$, and $M = 1, 24, 32$, respectively ($\mu$ was set to zero, as in Fig. 5.3). For each system, listeners evaluated each of the three possible pairs. When the critically sampled signal was compared to the oversampled signals, 96 % of the choices were in favour of the oversampled ones. No clear preference was noted when the two oversampled signals (e.g. $M = 1$ and $M = 12$ in the 16-subband system) were compared. These results are consistent with those illustrated in Fig. 5.3.

## 5.3 Chapter Summary

In this chapter, we have described a method to assess the performance of BSS systems in terms of separation and distortion. Data generation and how the BSS systems were programmed have been described. Performance rates have been obtained, and indicated that a certain amount of separation was achieved. We have noted that computational speed was better if subband adaptation was used in conjunction with a WOA realization. The choice of an oversampled scheme to reduce distortion was also justified experimentally. Lastly, the results were compared to the scores obtained by informal listening tests, and the two were generally consistent.

# Chapter 6

# Conclusion

This thesis was concerned with the BSS problem applied to speech sources in a reverberant environment. Computational efficiency of BSS systems can be problematic, so the goal of this thesis was to propose a computationally fast BSS algorithm based on subband adaptation. Subband analysis and synthesis were performed via DFT filter banks. Moreover, equations suitable for subband separation were derived. In the end, performance in terms of separation and distortion rates of the subband system was similar to the fullband system. However, the 32-subband system was about 10 times faster for a two-input two-output scenario.

In this chapter, a summary of the work is provided. Ideas for future work are then proposed.

## 6.1 Summary of Our Work

A literature review of BSS was presented in Chap. 2. Two mixing scenarios were considered: instantaneous and convolutive mixing. Instantaneous mixing was used to introduce the concepts of BSS. In this context, a cost function for blind separation, based on ML estimation, was given. The cost function attempts to measure statistical independence at the end of a separating network. Assuming that the sources are independent, separation can be achieved by adapting the parameters of a de-mixing network such that independence, measured by the cost function, is restored. A second cost function was obtained using an information theoretic approach, referred to as the infomax principle. The infomax principle was mentioned because of its historical importance, and we showed that in

the end both cost functions were equivalent. Adaptation equations for separating reverberant sources were developed by maximizing the cost function. Two de-mixing networks were considered: a feedforward and a feedback network. Due to its inherent structure, the feedforward network whitens the sources, which is a major drawback for speech signal; this problem does not occur with the feedback network. In any case, to compute the adaptation equation, it is necessary to evaluate an activation function. Activation functions depend on the pdf of the sources, and for speech sources, using the Laplacian pdf was suggested.

We introduced in Chap. 3 the idea of subband adaptation to reduce computational costs. The necessary tools for subband adaptation were described. Briefly, a pair of DFT filter banks was used for subband analysis and synthesis. A DFT filter bank is made of three components: a complex modulator to shift the signal in frequency, a low-pass prototype filter to limit the bandwidth of the signal, and a decimator to decrease the sampling rate or an interpolator to increase the sampling rate (depending whether the filter bank is used for subband analysis or synthesis). There is one set of these components for each subband, and they all operate on a different frequency range (complex modulation is different for each subband). The effect of decimation and interpolation in the z-domain was investigated next. These results were used in Chap. 4 to derive an expression for aliased components. Design of the prototype filters was then discussed. Appropriate filters were obtained by interpolating tabulated QMF. Two prototype filters were designed, one suitable for a 16-subband system, and one suitable for a 32-subband system. Finally, the WOA realization was mentioned, which is a very efficient method to implement a DFT filter bank.

In Chap. 4, we merged the results of Chaps. 2 and 3 together, and developed a BSS system that operates in subbands using feedback de-mixing networks in each subband. Firstly, under the assumption of a critically sampled scheme, a mathematical expression for aliased terms in the subband system was obtained. We noted that aliasing could not be cancelled easily. Therefore, to reduce aliasing, we proposed to use an oversampled scheme. For a 16-subband system, a decimation factor of 12 yielded inaudible distortions. Similarly, a decimation factor of 24 appeared to be a good compromise for a 32-subband system. An adaptive algorithm was then derived for subband systems. The same cost function presented in Chap. 2 was used, but since the parameters of the de-mixing networks were now complex (due to the modulation), we had to employ a complex gradient operator to maximize the cost function. In order to obtain an activation function for the adaptive algorithm, distribution of bandpass speech was analyzed empirically. Simple experiments

showed that phase was distributed uniformly, and that magnitude could be modelled as a Laplacian random variable. Lastly, the computational complexity of subband systems was established. We found approximately that a theoretical gain of $2M^2/3K$ could be achieved for a two-input two-output system. This gain was calculated by counting the number of multiplications required by the fullband system over those of the subband system.

Performance of the subband systems was measured in Chap. 5. Firstly, different strategies for performance assessment were enumerated. We selected an evaluation method that provided the realism of real recordings while permitting objective separation and distortion measures. These measures were possible to obtain by recording the sources separately. Indeed, by having access to the contribution of each source to each microphone, it was straightforward to define separation and distortion rates. These rates were computed empirically for the fullband, the 16-subband, and the 32-subband systems using three different speaker positions. For each position, rates were almost identical among all the BSS systems. But it was noted that the performance rates depended on the positions of the speakers in the room. For instance, a strong direct path and weak cross-talk yielded the best performance rates. Informal listening tests were conducted, and confirmed that there existed a strong correlation between the separation rates and the perceived degree of separation. But due to the psychoacoustic properties of the human ear, performance rates could sometimes be misleading, i.e. a high separation rate does not automatically imply that the sources appear well separated to a human listener. Computational complexity was also measured, and it turned out that theoretical gains reported in Chap. 4 were very close to the empirical gains.

## 6.2 Future Work

We describe in this section future directions for work on subband-based BSS systems. Interesting issues include the source permutation problem, automatic step size adjustment, and moving speakers.

### 6.2.1 Source Permutation Problem

The source permutation problem was briefly mentioned in Sec. 4.2.3. Since source separation is performed independently in each subband, there is no guarantee that the sources will be recovered in the same order. Furthermore, a gain ambiguity can also be present.

The order in which the sources are recovered is certainly not a random phenomenon, and needs to be studied in the context of subband adaptation. In the experiments conducted so far, different source permutations across each subband never occurred, but we provided no explanation for this fact. However, in the context of frequency domain adaptation, the source permutation problem has been documented in the literature, e.g. see [32]. A good starting point would be to familiarize ourselves with this literature, and note to which extent this information is pertinent for subband adaptation.

### 6.2.2 Automatic Step Size Adjustment

In this thesis, the step size was chosen using an ad hoc method. Performance rates reported in Tab. 5.2 can vary dramatically if the step size is changed. Numerous methods have been proposed to adjust the step size in a optimal manner, such that convergence speed is high, and steady-state error is low [52]. It should be worth investigating the use of self-adjusting step sizes for BSS, as in [53]. Moreover, we should exploit the fact that each subband is processed independently by using a different step size in each subband.

### 6.2.3 Moving Speakers

The BSS systems designed in this thesis assumed immobile sources. In other words, the mixing network remained static. In a real world application, persons usually move while they are speaking. However, the mere act of turning one's head can completely change the impulse response, and the mixing network to be inverted. Modelling a dynamic mixing network should be considered in the design of BSS systems, as in [17], [54]. A new testing methodology would have to be proposed, since the one used in this thesis (based on [48]) is limited to non-moving sources.

# References


[1] T.-W. Lee, M. Girolami, and T. J. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed sub-Gaussian and super-Gaussian sources," *Neural Computation*, vol. 11, no. 2, pp. 417–441, 1999.

[2] A. J. Bell, "Information theory, independent-component analysis, and applications," in *Unsupervised Adaptive Filtering* (S. Haykin, ed.), vol. 1, ch. 6, pp. 237–264, Wiley-Interscience Publication, 2000.

[3] T.-W. Lee, R. Orglmeister, A. Ziehe, and T. Sejnowski, "Combining time-delayed decorrelation and ICA: Towards solving the cocktail party problem," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, (Seattle, USA), pp. 1249 – 1252, May 1998.

[4] D. Schobben and P. Sommen, "A new algorithm for joint blind signal separation and acoustic echo canceling," in *Proc. of the Fifth International Symposium on Signal Processing and its Applications*, vol. 2, (Brisbane, Australia), pp. 889 – 892, Aug. 1999.

[5] T. Okuno and M. O. Tokhi, "Stereophonic acoustic echo cancellation using blind source separation as post-processing," in *Proc. of the International Workshop on Acoustic Echo and Noise Control*, (Darmstadt, Germany), pp. 75 – 78, Sept. 2001.

[6] S. Choi, H. Hong, H. Glotin, and F. Berthommier, "Multichannel signal separation for cocktail party speech recognition: A dynamic recurrent network," in *Proc. of the 6th International Conference on Spoken Language Processing*, (Beijing, China), Oct. 2000.

[7] K. Torkkola, "Blind separation of delayed and convolved sources," in *Unsupervised Adaptive Filtering* (S. Haykin, ed.), vol. 1, ch. 8, pp. 321–375, Wiley-Interscience Publication, 2000.

[8] R. Cristescu, T. Ristaniemi, J. Joutsensalo, and J. Karhunen, "Blind separation of convolved mixtures for CDMA systems," in *Proc. of the X European Signal Processing Conference*, (Tampere, Finland), pp. 619 – 622, Sept. 2000.

[9] T. Lee, *Independent Component Analysis: Theory and Applications*. Kluwer Academic Publisher, 1998.

[10] J. Herault and C. Jutten, "Space or time adaptive signal processing by neural network models," in *Neural Networks for Computing: AIP Conference Proceedings*, vol. 151, (Snowbird, USA), pp. 206 – 211, 1986.

[11] P. Comon, "Independent component analysis, a new concept ?," *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[12] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[13] S. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems* (D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, eds.), vol. 8, pp. 757–763, The MIT Press, 1996.

[14] K. Torkkola, "Blind separation of convolved sources based on information maximization," in *Proc. IEEE Workshop on Neural Networks for Signal Processing*, (Kyoto, Japan), pp. 423 – 432, Sept. 1996.

[15] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE Trans. on Signal Processing*, vol. 44, no. 12, pp. 3017–3030, 1996.

[16] J.-F. Cardoso, "Infomax and maximum likelihood for blind source separation," *IEEE Signal Processing Letters*, vol. 4, pp. 112–114, Apr. 1997.

[17] A. Koutras, E. Dermatas, and G. Kokkinakis, "Blind speech separation of moving speakers in real reverberant environments," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, (Istanbul, Turkey), pp. 1133 – 1136, June 2000.

[18] L. Vielva, D. Erdogmus, C. Pantaleon, I. Santamaria, J. Pereda, and J. Principe, "Underdetermined blind source separation in a time-varying environment," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, (Orlando, USA), pp. 3049 – 3052, May 2002.

[19] L. Vielva, D. Erdogmus, and J. Principe, "Underdetermined blind source separation using a probabilistic source sparsity model," in *Proc. of 3rd International Conference on Independent Component Analysis and Blind Signal Separation*, (San Diego, USA), pp. 675 – 679, Dec. 2001.

[20] A. Koutras, E. Dermatas, and G. Kokkinakis, "Improving simultaneous speech recognition in real room environments using overdetermined blind source separation," in *Proc. Eurospeech*, vol. 2, (Aalborg, Denmark), pp. 1009–1012, Sept. 2001.

[21] S. Amari and A. Cichocki, "Adaptive blind signal processing – neural network approaches," *Proc. of the IEEE*, vol. 86, pp. 2026–2048, Oct. 1998.

[22] T.-W. Lee, A. J. Bell, and R. H. Lambert, "Blind separation of delayed and convolved sources," in *Advances in Neural Information Processing Systems* (M. C. Mozer, M. I. Jordan, and T. Petsche, eds.), vol. 9, pp. 758 – 764, The MIT Press, 1997.

[23] J.-F. Cardoso, "Blind signal separation: Statistical principles," *Proc. of the IEEE*, vol. 86, pp. 2009–2025, Oct. 1998.

[24] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 3rd ed., 1991.

[25] T. Lee, M. Girolami, A. Bell, and T. Sejnowski, "A unifying information-theoretic framework for independent component analysis," *International Journal on Mathematical and Computer Modeling*, vol. 31, pp. 1–21, Mar. 2000.

[26] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 3rd ed., 1996.

[27] J. Benesty, "An introduction to blind source separation of speech signals," in *Acoustic Signal Processing for Telecommunication* (S. L. Gay and J. Benesty, eds.), ch. 15, pp. 321–329, Kluwer Academic Publisher, 2000.

[28] J. Xi and J. F. Chicharo, "A simplified infomax approach for blind signal separation," in *Proc. Fifth International Symposium on Signal Processing and its Application*, vol. 1, (Brisbane, Australia), pp. 43 – 46, Aug. 1999.

[29] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[30] K. Torkkola, "Blind separation of delayed sources based on information maximization," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, (Atlanta, USA), pp. 3509 – 3512, May 1996.

[31] S. Choi and A. Cichocki, "Adaptive blind separation of speech signals: Cocktail party problem," in *Proc. of the International Conference on Speech Processing*, (Seoul, Korea), pp. 617 – 622, Aug. 1997.

[32] K. Torkkola, "Blind separation for audio signals – are we there yet?," in *Proc. of the 1st Workshop on Independent Component Analysis and Blind Signal Separation*, (Aussois, France), pp. 239 – 244, Jan. 1999.

[33] Y. Guo, F. Sattar, and C. Koh, "Blind separation of temporomandibular joint sound signals," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, (Phoenix, USA), pp. 1069 – 1072, Mar. 1999.

[34] N. Charkani and Y. Deville, "Optimization of the asymptotic performance of time domain convolutive source separation algorithms," in *Proc. of the European Symposium on Artificial Neural Networks*, (Bruges, Belgium), pp. 273 – 278, Apr. 1997.

[35] J.-F. Cardoso, "Entropic contrasts for source separation: Geometry and stability," in *Unsupervised Adaptive Filtering* (S. Haykin, ed.), vol. 1, ch. 4, pp. 139–189, Wiley-Interscience Publication, 2000.

[36] A. Gilloire and M. Vetterli, "Adaptive filtering in sub-bands," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, (New York, USA), pp. 1572 – 1575, Apr. 1988.

[37] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice-Hall, 1993.

[38] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Prentice-Hall, 1983.

[39] R. D. Koilpillai and P. P. Vaidyanathan, "Cosine-modulated fir filter banks satisfying perfect reconstruction," *IEEE Trans. on Signal Processing*, vol. 40, pp. 770 – 783, Apr. 1992.

[40] H. Bolcskei and F. Hlawatsch, "Oversampled cosine modulated filter banks with perfect reconstruction," *IEEE Trans. on Circuits and Systems – II: Analog and Digital Signal Processing*, vol. 45, pp. 1057 – 1071, Aug. 1998.

[41] Q.-G. Liu, B. Champagne, and D. K. Ho, "Simple design of oversampled uniform DFT filter banks with applications to subband acoustic echo cancellation," *Signal Processing*, vol. 80, pp. 831–847, June 2000.

[42] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice-Hall, 1996.

[43] R. E. Crochiere and L. R. Rabiner, "Interpolation and decimation of digital signals — a tutorial review," *Proc. of the IEEE*, vol. 69, pp. 300 – 331, Mar. 1981.

[44] X. Lu and D.-S. Yu, "Acoustic echo cancellation by using subband structure." McGill University, 2000.

[45] M. D. Paez and T. H. Glisson, "Minimum mean-squared-error quantization in speech PCM and DPCM systems," *IEEE Trans. on Communications*, vol. COM-20, pp. 225–230, Apr. 1972.

[46] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," in *Proc.of the International Workshop on Independence & Artificial Neural Networks*, (Tenerife, Spain), Feb. 1998.

[47] D. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proc.*, vol. 130, pp. 11–16, Feb. 1983.

[48] D. Schobben, K. Torkkola, and P. Smaragdis, "Evaluation of blind signal separation methods," in *Proc. of the 1st Workshop on Independent Component Analysis and Blind Signal Separation*, (Aussois, France), pp. 261–266, Jan. 1999.

[49] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi, *GNU Scientific Library Reference Manual*. Network Theory Ltd, 2001.

[50] P. Creek and D. Moccia, *Digital Media Programming Guide*, ch. 7. Silicon Graphics, Inc., 1996.

[51] M. Frigo and S. G. Johnson, "FFTW: An adaptive software architecture for the FFT," in *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, (Seattle, USA), pp. 1381–1384, May 1998.

[52] S. Koike, "A class of adaptive step-size control algorithms for adaptive filters," *IEEE Trans. on Signal Processing*, vol. 50, pp. 1315 – 1326, June 2002.

[53] T. P. von Hoff, A. G. Lindgren, and A. N. Kaelin, "Step-size control in blind source separation," in *Proc. of the 2nd International Workshop on Independent Component Analysis and Blind Signal Separation*, (Helsinki, Finland), pp. 509 – 514, June 2000.

[54] A. Koutras, E. Dermatas, and G. Kokkinakis, "Blind speech separation of moving speakers using hybrid neural networks," in *Proc. Eurospeech*, vol. 2, (Aalborg, Denmark), pp. 997 – 1000, Sept. 2001.