UNDERDETERMINED BLIND SOURCE SEPARATION FOR SPEECH SIGNALS

Nikolaos Gkalelis

Supervised by
Dipl. -Ing. (FH) Robert Aichner
Prof. Dr. -Ing. Walter Kellermann

Submitted as partial fulfillment of the requirements for
the degree of Master of Computational Engineering
at the Friedrich-Alexander-University-Erlangen-Nuremberg

3 November 2003 – 3 May 2004
Erlangen,Germany

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, 3.5.2004

Nikolaos Gkalelis
Mittlere Schulstr. 4
91054 Erlangen
Deutschland

Declaration

I declare that the work is entirely my own, and was produced with no assistance from third parties. I certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references direct and indirect are indicated as such and have been cited accordingly.

Erlangen, 3rd May 2004

Nikolaos Gkalelis
Mittlere Schulstr. 4
91054 Erlangen
Germany

Acknowledgments

I would like to thank Prof. W. Kellermann for giving me the opportunity to pursue my MSc thesis in his excellent research group and also for the warm encouragement through all the hard stages of the thesis. Moreover, I am grateful to Robert Aichner for his professional supervision and mentorship, which expanded significantly my knowledge in digital signal processing, and improved my soft skills.

I would like to express my gratitude to Herbert Buchner for the fruitfull discussions which enhanced considerably the overall project. Moreover, for his care at the finalization of the thesis. I am indebted to Felix, for the same reason and for the overall help through the project.

Thanks are due to all the colleagues in the computer lab, Stefan, Jens and other, for all the pleasant and humorous atmosphere and for many interesting scientific and non-scientific discussions.

Most of all, my warmest thanks to my parents and my sister for the great support all the years of my studies. For the same reason my best thanks to Momke.

Zusammenfassung

Blinde Quellentrennung addressiert das Problem aus Q linearen Mischsignalen P Quellen zu separieren. Für den quadratischen Fall, d.h. $P = Q$ wurden in den letzten Jahren viele Algorithmen entwickelt, welche in realistischen Szenarien gute Trennungsergebnisse liefern. In jüngster Zeit wurde verstärkt das schwierigere Problem der unterbestimmten Quellentrennung untersucht, d.h. die Anzahl der Quellen P übertrifft die Anzahl der Sensoren Q. In dieser Arbeit werden vielversprechende Ansätze zur unterbestimmten blinden Quellentrennung untersucht. Ausserdem wird ein umfassender Überblick über sogenannte "Deflation"-Ansätze gegeben, welche aus den Mischsignalen die getrennten Quellensignale nacheinander extrahieren. Es wird eine Vorgehensweise präsentiert, welche erlaubt vorhandene Algorithmen für "i.i.d."-Signale zu modifizieren und auf Audiosignale anzuwenden. Anschliessend werden Zusammenhänge zu konventionellen "Generalized Sidelobe Canceller"-Strukturen aufgezeigt. Experimentelle Ergebnisse in halligen Umgebungen zeigen die Anwendbarkeit der vorgeschlagenen Algorithmen. Am Schluß werden mögliche Erweiterungen präsentiert, welche zukünftige Forschungsthemen darstellen können.

Abstract

Blind source separation (BSS) addresses the problem to separate Q sources from P linear mixtures. In the quadratic case, that is $Q = P$, many algorithms exhibiting good performance in real-world scenarios have been proposed. Recently many researchers have been investigating the more difficult scenario that the mixing system is underdetermined, that is, the number of sources outnumbers the number of mixtures. In this thesis promising approaches for underdetermined blind source separation of speech mixtures are investigated. Moreover a comprehensive overview of deflation approaches which extract the desired source signals from the mixtures sequentially is given. It is shown how deflation approaches usually designed for i.i.d. signals can be modified and applied to audio signals. Furthermore links between the proposed deflation approach and conventional Generalized Sidelobe Canceller (GSC) structures are shown. Experimental results in reverberant environments show the applicability of the presented algorithms for speech signals. In the end possible extensions are presented which might be further topics of research.

Contents

List of Figures

Chapter 1

Introduction

In speech signal processing, blind source separation (BSS) addresses the problem to separate Q speech sources from P microphone signals, without any a priori knowledge about the source signals, mixing conditions or the sensor array configuration. The lack of prior knowledge is compensetated by a statistically strong but often physically plausible assumption of spatial independence between the source signals, i.e. source separation exploits primarily *spatial diversity* [11].

The underlying mixing system is convolutive, consisting of the room impulse responses, which can be conveniently modeled as linear time-invariant (LTI) filters (Fig. 1.1).

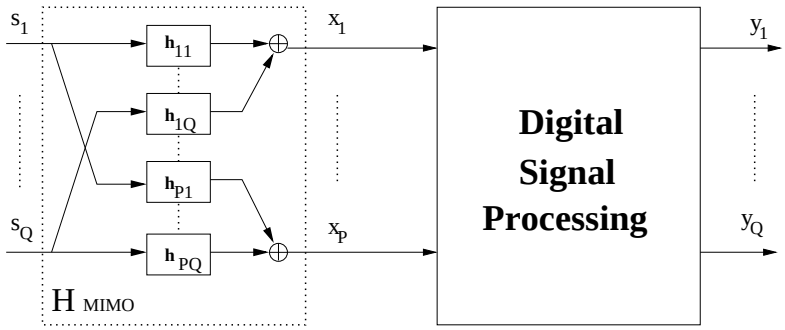

Figure 1.1: Block diagram of BSS.

Standard BSS methods assume the quadratic case, i.e., the number of sources is equal to the number of sensors, and therefore perfect reconstruction of the signals is possible [BSS]. Several algorithms have been proposed which demonstrate good performance even for convolutive mixtures. In more realistic scenarios, e.g. cocktail party problem, the speakers outnumber the microphones and consequently the mixing system is underdetermined. Hence, conventional BSS methods

are not applicable.

The last years, many researchers have been dealing with the underdetermined BSS problem. Mainly, they try to exploit the so-called sparseness of speech sources. Sparseness means that a representation of the signals exists where they have disjoint support, i.e. they do not overlap. Jilmaz and Rickard [38] demonstrated by experiments that speech signals in instantaneous mixtures are sparse enough in the time-frequency domain. Moreover, based on the direction of arrival (doa) of the sources, they constructed time-frequency masks to separate them, thus exploiting the *spectral diversity* of speech signals. While in the free-field case the method gives good results, when exposed to convolutive mixtures the performance is not satisfying. Recently, Araki et.al. [4] combined time-frequency masking with independent component analysis (ICA), succeeding better results even for reverberant enclosures.

Another possible way to solve the underdetermined BSS problem may be to separate the sources sequentially after deflating, i.e. removing, their contribution from the mixtures. Deflationary techniques consist of a blind signal extraction method, where only one signal is extracted, and a mean square error (MSE) criterion to estimate and then remove the contribution of the separated source from the mixture. Several deflationary techniques were initially designed for temporally i.i.d. signals and gradually evolved for BSS of signals with various distributions. Such an example is the blind deconvolution method proposed by Shalvi and Weinstein, which gradually extended to one of blind source factor separation (BSFS) of speech signals. Moreover, it is interesting to note that deflationary techniques appear similarities with Generalized Sidelobe Cancellers (GSC), which may be exploited for the construction of efficient algorithms.

This thesis aims to investigate promising techniques for underdetermined BSS of speech signals. In particular, the objective is to examine one method based on sparseness and one deflationary. For the latter the similarity with the GSC should be explored, from which efficient algorithms may arise. Before proceeding to the core of the dissertation it is valuable to outline its structure.

Organization of the thesis

We distinguish six chapters. The current, chapter 1, introduces the fundamental concepts, (and) thus preparing the reader for the core of the thesis.

In Chapter 2 we gradually introduce the underdetermined BSS problem and then discuss how sparseness of speech signals can be exploited to give the necessary additional information to solve this problem. Then we examine a relevant method which combines time-frequency masking and conventional BSS.

Chapter 3 is the longest and most important of the thesis. There we examine the evolution of a single channel blind deconvolution method for i.i.d. signals

to one of BSFS of speech signals. Along with this comprehensive overview, the necessary modifications imposed on the original algorithm in order to arrive to its final stage are clearly shown. This treatment may motivate for further improvement of the method or application of this strategy to similar blind deconvolution methods in the field.

In Chapter 4 the proposed BSFS method is combined with a MSE criterion to form a deflationary BSS algorithm. It is shown that the resulting algorithm has strong similarities with GSC structures, which may be exploited to produce efficient algorithms.

Chapter 5 includes the experimental results from both methods. From this experiments the applicability of the algorithms for speech signals were verified and interesting conclusions arised.

Finally, in Chapter 6 we give a short summary and conclusions upon the discussed topics, and propose possible extensions of the algorithms for future consideration.

For the convenience of the reader, at the beginning of each chapter we give a brief picture of its contents. The same is attempted for each section depending on its size. At the end of each chapter again a brief summary is presented along with pointing out the more important previously discussed topics.

Chapter 2

Underdetermined BSS utilizing the Sparseness of Speech Signals

This chapter deals with the underdetermined BSS of speech signals and more specifically with these methods that utilize the sparseness of speech signals.

The performance of these methods is affected crucially from the propagation conditions which may be anechoic or echoic. These properties as well as most of the properties of the problem are captured in the mixing matrix. The invertibility of the mixing matrix determines the solution of the problem. In the underdetermined case traditional inversion of this matrix is not possible. A way to overcome this problem is to utilize the sparseness of the speech signals. In the first section of this chapter, the BSS problem is formulated mathematically for both the anechoic or echoic conditions, and the implications of the mixing matrix are discussed. Then we consentrate on the underdetermined BSS problem and we indicate how it can be solved by utilizing the sparseness of speech signals.

In the next section we narrow the discussion to analyze a specific category of underdetermined BSS methods, the so-called time-frequency masking methods. These methods assume that speech signals are sparse in the time-frequency domain and that the mixing is anechoic. The first assumption implies that each time-frequency coefficient of the microphone signals belongs to only one source. The second assumption allows to label uniquely each time-frequency bin with the source it belongs to, utilizing the one-to-one relationship between a source signal and its propagation path. Therefore the label of each bin is the attenuation and the delay parameter of the propagation path. Even when this idealized model is employed, there are two essential factors that still affect the performance of these methods. These factors are the window selection for the ST-DFT and the spacing distance between adjacent microphones. The first parameter determines the faithfulness of the spectrum estimate, and the second parameter relates the temporal and spatial sampling frequency and determines the existence or not of aliasing.

In reality speech signals partially overlap even in the frequency domain. Moreover, when the propagation is echoic a source propagates to a microphone through more than one paths. Therefore labeling the bins with the attenuation and delay parameters of the propagation is not effective anymore. Consequently, in realistic scenarios ideal masks can not be constructed. Then, measuring the performance of a mask becomes an essential matter. All the above issues are discussed in detail in the second section too.

Time-frequency masking methods appear strong similarities with source localization techniques. Therefore improvement of time-frequency masking methods may come by combining them with beamforming approaches. In the third section we describe such an algorithm which combines time-frequency masking and ICA. First a pair of sources is extracted and then ICA is applied to separate the sources from the pair. The labeling parameter in the time-frequency domain is the direction of arrival (DOA) of the sources.

In the last section we give some conclusions on the topics discussed in this chapter and discuss possible improvements of the time-frequency masking methods.

2.1 Basic Concepts

This section after reviewing briefly the BSS problem, it concentrates gradually to the underdetermined case, and discusses how the sparseness of speech signals can be utilized to solve the particular problem.

In the first subsection the mathematical formulation of BSS is given for both reverberant and free-field mixing conditions. This formulation is represented as a set (or sets) of linear equations. The mixing matrix is fundamental on the attributes of the system of equations. Therefore in the next subsection we discuss the properties and implications of the mixing matrix to the BSS problem.

Finally in the last subsection we concentrate in the underdetermined BSS and discuss how sparseness of speech signals can be utilized in a suitable domain to separate them, for both anechoic and echoic mixtures. For echoic mixtures, we discuss the additional requirements imposed to the mapping function which transforms the signals from the time-domain to the domain where they are maximally sparse.

2.1.1 Reverberant vs. free-field mixing

The propagation of a speech signal $s_q(t)$ in *air* is with finite speed and involves reverberation. Therefore the recorded signal $x_p(t)$ in a microphone consists of a direct (delayed) copy of the sound source and its multi-path copies, modified by

the environment. If the medium of propagation is assumed time invariant then it can be modeled with a LTI filter h_{pq} enclosing the respective impulse responses of the environment. Consequently, the microphone signal will be given by the convolution of the speech signal with the filter ([30], [1, p. 13])

$$x_p(t) = \int_\tau h_{pq}(\tau) \cdot s_q(t-\tau)\,. \tag{2.1}$$

BSS considers Q sources and P sensor signals. Mixing of sound sources in the air is linear and hence we can write

$$x_p(t) = \sum_{q=1}^{Q} \int_\tau h_{pq}(\tau) \cdot s_q(t-\tau)\,, \quad p = 1, \ldots, P\,. \tag{2.2}$$

In the *free-field*, a sound signal propagating from the q-th source to the p-th microphone is attenuated by a gain factor a_{pq} and delayed by a time τ_{pq}, i.e. the propagation does not involve reverberation. Therefore the filters h_{pq} are reduced to

$$h_{pq}(t) = a_{pq} \cdot \delta(t-\tau_{pq})\,, \quad p = 1, \ldots, P\,, \tag{2.3}$$

where $\delta(t)$ is the Dirac delta function. Thus, the microphone signals become

$$x_p(t) = \sum_{q=1}^{Q} a_{pq} \cdot s_q(t-\tau_{pq})\,, \quad p = 1, \ldots, P\,. \tag{2.4}$$

In the discrete time domain (2.2), (2.4) can be written respectively as

$$x_p(n) = \sum_{q=1}^{Q} \sum_{\kappa} h_{pq_\kappa} \cdot s_q(n-\kappa_{pq})\,, \quad p = 1, \ldots, P\,, \tag{2.5}$$

$$x_p(n) = \sum_{q=1}^{Q} a_{pq} \cdot s_q(n-\kappa_{pq})\,, \quad p = 1, \ldots, P\,, \tag{2.6}$$

or in vector form

$$\begin{aligned} \mathbf{x}(n) &= \mathbf{H}_n * \mathbf{s}(n) \\ &= \sum_{\kappa} \mathbf{H}_\kappa \cdot \mathbf{s}(n-\kappa)\,, \end{aligned} \tag{2.7}$$

$$\mathbf{x}(n) = \mathbf{A} \cdot \mathbf{s}(n)\,, \tag{2.8}$$

where $\mathbf{s}(n)$ is a $Q \times 1$ column vector collecting the source signals, $\mathbf{x}(n)$ is similarly the P-channel microphone signal, $\mathbf{H}_n$ is the $P \times Q$ mixing matrix

containing the mixing coefficients at time n with $\{\mathbf{H}_\kappa\}$ being the respective $P \times Q$ matrix impulse response sequence of the propagation medium, $\mathbf{A}$ is the mixing matrix in the free-field case (e.g., see [19, p. 882], [11]), and κ_{pq} is the delay in number of samples. These structures are depicted above

$$\mathbf{s}(n) = [s_1(n), \ldots, s_Q(n)]^T , \tag{2.9}$$

$$\mathbf{x}(n) = [x_1(n), \ldots, x_Q(n)]^T , \tag{2.10}$$

$$\mathbf{H}_\kappa = \begin{bmatrix} h_{11,\kappa} & \cdots & h_{1Q,\kappa} \\ \vdots & \ddots & \vdots \\ h_{P1,\kappa} & \cdots & h_{PQ,\kappa} \end{bmatrix} , \tag{2.11}$$

$$\mathbf{A} = \begin{bmatrix} a_{11} \cdot \delta(n - \kappa_{11}) & \cdots & a_{1Q} \cdot \delta(n - \kappa_{1Q}) \\ \vdots & \ddots & \vdots \\ a_{P1} \cdot \delta(n - \kappa_{P1}) & \cdots & a_{PQ} \cdot \delta(n - \kappa_{PQ}) \end{bmatrix} . \tag{2.12}$$

where $\delta(n)$ in the discrete time domain represents the unit impulse function. In Subsect. 3.2.1 we show an equivalent multichannel block formulation of (2.7).

2.1.2 Implications of the mixing matrix into the BSS problem

BSS deals with the problem of separating Q unknown sources by observing P microphone signals. The mixing matrix $\mathbf{H}_n$ is the parameter of interest and determines the method to solve the problem. Moreover, the following two assumptions for $\mathbf{H}_n$ are necessary,

- The mixing system $\mathbf{H}_n$ is stable in the sense that $\sum_\kappa ||\mathbf{H}_\kappa|| < \infty$ (e.g. [19]). This is true for realistic scenarios because the propagation of a speech source to a microphone involves finite number of paths.

- It is important to note that $\mathbf{H}_n$ represents a linear transformation. Therefore (2.7) is solvable under the assumptions that $\mathbf{H}_n$ is full rank.

The method of estimating the sources depends on the relation between P and Q, i.e. the dimension of $\mathbf{H}_n$. When $Q = P$, i.e. the number of sources is equal to the number of sensors, the linear system of equations in (2.7) is called *determined* and has a unique solution given by the inversion of $\mathbf{H}_n$. In the *overdetermined* case ($P > Q$) an exact solution of (2.7) may or may not exist. The left generalized inverse of $\mathbf{H}_n$ identifies the exact $\mathbf{s}(n)$ if exists, or approximates it with the optimal estimate in the least squares (LS) sense (e.g. see Subsect. 3.2.2) .

In the *underdetermined* case ($P < Q$) there are infinitely possible vectors $\mathbf{s}(n)$ which satisfy (2.7). The actual $\mathbf{s}(n)$ is the one with the minimum L_2 norm. There are mainly two ways to acquire the minimum norm solution. In the first,

the right generalized inverse of $\mathbf{H}_n$ is estimated and then applied to the microphone signal $\mathbf{x}(n)$. This solution is the minimum norm solution. Another class of algorithms utilize the sparseness of speech signals to design better inversion strategies ([37, 21]) and identify the minimum norm solution. In the following we consentrate in sparseness methods.

2.1.3 Exploitation of sparseness for underdetermined BSS

A signal is sparse if only few of its samples are significantly different from zero. Sparseness is ussualy modeled by the Laplacian probability density function (pdf). Recalling that the pdf of speech signals in the time domain is close to a Laplacian, we can assume that they are sparse. Moreover, speech signals have harmonic structure and therefore it is expected that they will be even more sparse in the frequency domain.

Conventional BSS methods are based in the assumption that the source signals are spatially independent. Several methods for the underdetermined case make the additional assumption that the source signals are *disjoint*, i.e. they do not overlap in some domain. This assumption is based on the fact that speech signals are sparse and therefore they rarely overlap. The more the assumption of disjointness is fullfilled the better the separation performance of these methods is.

Experimental work revealed that sound sources in the time domain are not sufficiently sparse [8]. In contrary, subjective tests have shown that speech signals are *approximately* disjoint [38, 2, 3, 7] in the frequency domain, when the frequency resolution is between ten and twenty Herz ($10 - 20$ Hz).

In the following we see how the *spectral diversity* of speech signals can be exploited.

Sparseness in anechoic mixtures

Starting from the simpler case, i.e., free-field we write the mixing matrix in terms of its columns and then we expand (2.8)

$$\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_Q] \,, \tag{2.13}$$

$$\mathbf{x}(n) = \sum_{q=1}^{Q} \mathbf{a}_q \cdot s_q(n) \,. \tag{2.14}$$

This formulation shows explicitly that the measurement vector $\mathbf{x}(n)$ is a linear combination of the columns of the mixing matrix $\mathbf{a}_q$, scaled by the respective source sample $s_q(n)$ at time instant n. If the sources are disjoint then at every time instant n the measurement vector will be equal to one of the columns of $\mathbf{A}$,

say $\mathbf{a}_{q_1}$, scaled by the non-zero source $s_{q_1}(n)$. Speech signals are more disjoint in the frequency domain, hence we transform (2.14)

$$T\{\mathbf{x}(n)\} = T\{\sum_{q=1}^{Q} \mathbf{a}_q \cdot s_q(n)\} \tag{2.15}$$

$$= \sum_{q=1}^{Q} T\{\mathbf{a}_q\} \cdot T\{s_q(n)\}\,, \tag{2.16}$$

where $T\{\cdot\}$ represents a suitable invertible linear mapping, e.g. short-time discrete fourier transform (ST-DFT), discrete cosine transform (DCT), wavelet, etc.

Note that in the anechoic case each source $s_q(n)$ comes from only one path to each microphone. Therefore the corresponding $\mathbf{a}_q$ is a scalar vector representing a single bearing.

Sparseness in echoic mixtures

Proceeding as in the anechoic case, we express (2.7) in terms of the columns of the mixing matrix $\mathbf{H}_n$

$$\mathbf{H}_n = [\mathbf{h}_{1,n}, \ldots, \mathbf{h}_Q, n]\,, \tag{2.17}$$

$$\mathbf{x}(n) = \sum_{q=1}^{Q} \mathbf{h}_{q,n} * s_q(n)\,, \tag{2.18}$$

$$= \sum_{q=1}^{Q} \sum_{\kappa} \mathbf{h}_{q,\kappa} \cdot s_q(n-\kappa)\,. \tag{2.19}$$

Now $\mathbf{h}_{q,n}$ is a multichannel filter carrying information for several steering vectors, due to multipath propagation. Therefore in the time domain the sparseness assumption should be a lot stronger because we do not have just multiplication but convolution.

Moreoever $T\{\cdot\}$ should be a suitable transform that not only transform the signals in a domain where they are sparse but also change the convolution operator to instantaneous multiplication. For instance such a transform is the ST-DFT which consentrates the energy of the signal to a few components and convolution becomes multiplication.

We should also note that the representations of the column filters of H_κ will not be just scalar vectors but they will have a specific signature in that domain. This will smear the signals but fortunately it will not affect the degree of sparseness in the representation domain. Instead the main problem is that the columns of the

mixing matrix will not just carry a clear bearing but they will contain information of several steering vectors and therefore it is difficult to identify them.

Several methods exploit the sparseness of speech signals to separate them. For instance some methods are: clustering techniques to identify the columns of the mixing matrix and then demix the sources [8], subspace methods [39], time-frequency masking methods , other heuristic approaches [37], or combination of the above. In the rest of the chapter we will discuss time-frequency approaches which are probably the most popular one for underdetermined BSS either alone, or combined with other methods.

2.2 Fundamentals of Time-Frequency Masking

In the previous section we formulated the problem of BSS for anechoic and echoic conditions. Then we concentrated in the underdetermined BSS case and we discussed in general how the sparseness of speech signals can be utilized to separate them.

In this section we focus on a specific category of underdetermined BSS methods, which utilize the sparseness of speech signals in the time-frequency domain. In these methods the sensor signals are transformed in the time-frequency domain and then, with an appropriate mask, the components of each signal are collected separately, thus performing time-frequency masking.

With rigorous mathematical treatment it is shown how the sparseness of speech signals in the ST-DFT domain can be utilized for BSS, and how the propagation path parameters (attenuation and delay) can be derived and used to label the time-frequency grid points. Moreover, issues such as time delay estimation restrictions, and window effects of the ST-DFT, which affect the performance of these methods are examined in detail.

We investigate the application of these methods to three different models, ideal , echoic and anechoic. For all the the models we assume the following

- The speakers are spatially distant.
- The source signals are in the far-field of the microphone array.
- There are two omnidirectional microphones for signal acquisition.

For the ideal model we additionally assume that speech signals (a) do not overlap in the frequency domain, and (b) the propagation is in the free-field. In the anechoic model the free-field propagation assumption still holds, but now the signals partially overlap. Finally, the realistic scenario of propagation in reverberant environments is investigated, where both of the ideal case assumptions are violated.

2.2.1 Ideal model

The short-time (or time-dependent) discrete Fourier transform (ST-DFT) operation is essential in time-frequency masking methods ([38, p. 3], [1, pp. 13-15]). The ST-DFT of a sequence $x_p(n)$ is given from

$$\begin{aligned} X_p(k,m) &= F^w\{x_p(n)\} \\ &= \sum_{n=0}^{L_w-1} x_p(n+m\cdot L_w)\cdot w(n)\cdot W_w^{k\cdot n}\,, \end{aligned} \tag{2.20}$$

where $F^w\{\cdot\}$ is the ST-DFT operator, $X_p(k,m)$ is a two-dimensional function representing the signal in the time-dependent Fourier domain, k is the independent variable of the DFT domain, m is the block index representing time, $w(n)$ is a window sequence of length L_w, and W_w is the rotation factor

$$W_w = e^{-j\cdot\frac{2\cdot\pi}{L_w}}\,. \tag{2.21}$$

Time-frequency masking is based on the concept of W-disjoint orthogonality (W-DO) proposed by [38]: Two functions $s_{q_1}(k,m),\ s_{q_2}(k,m)$ are W-DO if for a given window function $w(n)$ the supports of their ST-DFT are disjoint.

The W-DO assumption can be stated concisely as

$$s_{q_1}(k,m)\cdot s_{q_2}(k,m) = 0 \quad \forall\ k,\ m\,, \tag{2.22}$$

which is satisfied if either of $s_{q_1}(k,m)$ or $s_{q_2}(k,m)$ is equal to zero. Note that W-DO is particularly convenient for the analysis of short-time stationary signals, like speech, as it is related with the ST-DFT.

For the ideal model we additionally assume the following

- The speech signals are W-DO.
- The propagation is in the free-field, i.e. the speech signals arrive to the microphones through only *one* path.

Further, the above properties of the Fourier transform are necessary

$$F^w\left\{\sum_{q=1}^{Q} s_q(n)\right\} = \sum_{q=1}^{Q} S_q(k,m)\,, \tag{2.23}$$

$$F^w\{a_{pq}\cdot s_q(n)\} = a_{pq}\cdot S_q(k,m)\,, \tag{2.24}$$

$$F^w\{s_q(n-\kappa_{pq})\} = S_q(k,m)\cdot W_w^{k\cdot\kappa_{pq}}\,. \tag{2.25}$$

By transforming the microphone signals in the ST-DFT and applying (2.23), (2.24), (2.25) we take

$$X_p(k,m) = \sum_{q=1}^{Q} a_{pq} \cdot S_q(k,m) \cdot W_w^{k \cdot \kappa_{pq}}, \quad p = 1, 2. \tag{2.26}$$

In matrix notation (2.26) becomes

$$\begin{bmatrix} X_1(k,m) \\ X_2(k,m) \end{bmatrix} = \begin{bmatrix} a_{11} \cdot W_w^{k \cdot \kappa_{11}} & \cdots & a_{1Q} \cdot W_w^{k \cdot \kappa_{1Q}} \\ a_{21} \cdot W_w^{k \cdot \kappa_{21}} & \cdots & a_{2Q} \cdot W_w^{k \cdot \kappa_{2Q}} \end{bmatrix} \cdot \begin{bmatrix} S_1(k,m) \\ \vdots \\ S_Q(k,m) \end{bmatrix}, \tag{2.27}$$

or equivalently

$$\begin{bmatrix} X_1(k,m) \\ X_2(k,m) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 1 \\ a_1 \cdot W_w^{k \cdot \kappa_1} & \cdots & a_Q \cdot W_w^{k \cdot \kappa_Q} \end{bmatrix} \cdot \begin{bmatrix} S_1(k,m) \\ \vdots \\ S_Q(k,m) \end{bmatrix}, \tag{2.28}$$

where $a_q = \frac{a_{2q}}{a_{1q}}$ is the ratio between the attenuation factors imposed in the q-th source when propagates to the sensors and $\kappa_q = \kappa_{2q} - \kappa_{1q}$ is the difference between the corresponding delays.

Due to the W-DO assumption every point (k,m) in the time-frequency grid will belong only to one of the sources $S_q(k,m)$. Therefore (2.28) is reduced to

$$\begin{bmatrix} X_1(k,m) \\ X_2(k,m) \end{bmatrix} = \begin{bmatrix} 1 \\ a_q \cdot W_w^{k \cdot \kappa_q} \end{bmatrix} \cdot S_q(k,m). \tag{2.29}$$

Hence, if we know the points (k,m) associated with each source, a binary mask can be constructed for the extraction of the particular source

$$M_q^b(k,m) = \begin{cases} 1, & S_q(k,m) \neq 0 \\ 0, & \text{otherwise} \end{cases}, \quad q = 1, \ldots, Q. \tag{2.30}$$

Then applying these masks to one of the microphone signals we extract the sources in the time-frequency domain

$$S_q(k,m) = X_p(k,m) \cdot M_q^b(k,m). \quad q = 1, \ldots, Q, \tag{2.31}$$

If at least one of the window samples $w(n)$ is non-zero, then the windowed signals, i.e. $s_q(n + m \cdot L_w) \cdot w(m) \quad q = 1, \ldots, Q$, can be reconstructed by the DFT synthesis equation and then by inverting the window function the source signals can be retrieved (e.g. [32, pp. 693-774]), thus succeeding BSS.

Labeling

In order to construct the masks some means of *labeling* are necessary. In (2.28) we observe that each time-frequency bin is parameterized by the attenuation factor a_q and the time-delay κ_q imposed by the *propagation medium* to the sources. One of these values is enough for labeling the grid points but the estimation of both leads to more robust labeling. In order to gain some insight, we show above a direct way to compute these parameters (e.g. [38, p. 6]).

Considering (2.28), a_q for each grid point is

$$\left|\frac{X_2(k,m)}{X_1(k,m)}\right| = a_q(k,m)\,, \tag{2.32}$$

and for $\tau_q(k,m)$

$$\begin{aligned} -\mathrm{Im}\left(\log\left(\frac{X_2(k,m)}{X_1(k,m)}\right)\right) &= \Delta\phi(k,m) \\ &= \omega_k \cdot \kappa_q(k,m) \\ &= \frac{2\cdot\pi\cdot k}{L_w}\cdot\kappa_q(k,m)\,, \end{aligned} \tag{2.33}$$

rearranging (2.33)

$$\tau_q(k,m) = \frac{L_w}{2\cdot\pi\cdot k}\cdot\mathrm{Im}\left(\log\left(\frac{X_1(k,m)}{X_2(k,m)}\right)\right)\,, \tag{2.34}$$

where $\Delta\phi(k,m)$ is the phase difference between the signals arriving in the two microphones in radians, ω_k is the frequency of the k-th frequency component ($k = 0,\ldots,L_w-1$) in radians per sample, and the indices (k,m) were added where necessary in order to show that the particular quantity is computed for each grid point.

(2.32) (2.34) can be written compactly as a two-dimensional parameter for each grid point

$$(a_q,\kappa_q)(k,m) = \left(\ \left|\frac{X_2(k,m)}{X_1(k,m)}\right|\ ,\ \frac{L_w}{2\cdot\pi\cdot k}\cdot\mathrm{Im}\left(\log\left(\frac{X_1(k,m)}{X_2(k,m)}\right)\right)\ \right)\,. \tag{2.35}$$

The ideal model assumed free field propagation. Thus each source $s_q(n)$ is associated with only one pair (a_q,κ_q) characterizing the propagation medium between the source and the two sensors. Moreover, the sources are assumed spatially distant. Therefore there are Q distinct parameter pairs $(a_1,\kappa_1),\ldots,(a_Q,\kappa_Q)$. This implies that a histogram of the parameter pairs $(a_q,\kappa_q)(k,m)$ computed for all grid points will reveal Q distinct values corresponding to the Q sources. Then each grid point can be labeled accordingly, and the mask for each source can be constructed as in (2.30).

Restrictions when $\kappa_q(k,m)$ is used as the separation criterion

The estimation of the time delay $\kappa_q(k,m)$ from the phase difference $\Delta\phi(k,m)$ as in (2.33) imposes some geometrical restrictions on the position of the sources and the sensors (e.g. see [30, pp. 1-14], [35, pp.2-11], [38, pp. 6-7] and for a more detailed analysis [20, pp. 84-91]) . We discuss them in the following.

The sound sources are assumed to be in the *far-field* of the microphone array, i.e. the distance of a source from the array is much greater than the distance between the microphones. Then the spherical wavefronts emanating from the sources can be approximated as plane wavefronts. Therefore the sound waves reaching the microphones will be parallel to each other and perpendicular to the direction of propagation, as shown in Fig. 2.1.

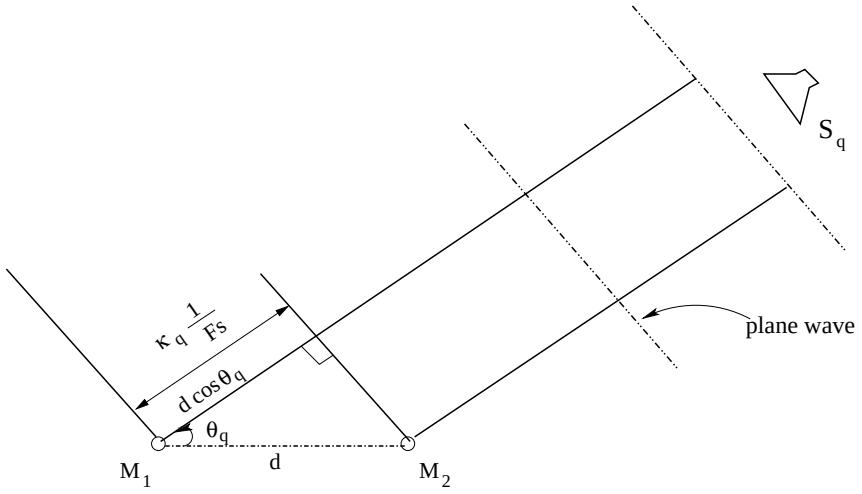

Figure 2.1: Geometrical setup under the assumption of far-field source.

We can see that source $s_q(n)$ arrives to microphone M_1 with delay κ_q relative to its arrival to M_2 because it has to travel an extra distance $d \cdot \cos\theta$. Therefore

$$\kappa_q = \frac{d \cdot \cos\theta_q}{v_s}\,, \tag{2.36}$$

where θ_q is the direction of arrival (DOA) of $s_q(n)$ and $v_s \approx 355\frac{m}{sec}$ is the velocity of sound.

The phase deference $\Delta\phi(k,m)$ between the microphone signals $X_1(k,m)$ and $X_2(k,m)$ is only detectable if it is in the range $[-\pi,\pi]$. Therefore from (2.33) we can write

$$|\omega_k \cdot \kappa_q| \leq \pi\,, \tag{2.37}$$

where $|\omega_k| \leq \pi$.(2.37) should be satisfied also for $|\omega_{k,max}| = \pi$, hence

$$|\kappa_q| \leq 1\,\text{sample delay}\,. \tag{2.38}$$

This means that as long as the delay between the two microphone readings is less than a sample, the estimated phase will be accurate. One sample delay is $\frac{1}{F_s}$ seconds, where F_s is the sampling rate in samples per second. Replacing κ_q of (2.36) into the inequality (2.38) we take

$$\frac{d \cdot |\cos\theta_q|}{v_s} \leq \frac{1}{F_s}\,, \tag{2.39}$$

$$d \leq \frac{v_s}{F_s} \cdot \frac{1}{|\cos\theta_q|}\,. \tag{2.40}$$

It is $|\cos\theta_q| \leq 1$. As we do not have any control on the DOA of the source we assume the worst case, i.e. $\theta_q = 0^o$. Moreover, $|\omega_k| \leq \pi$ meaning that $|\omega_{k,max}| = \pi$ and thus (2.37) becomes

$$d \leq \frac{v_s}{F_s}\,. \tag{2.41}$$

d represents the spatial sampling interval of the wavefield, like F_s for the temporal sampling. To avoid spatial aliasing d should be

$$d \leq \frac{\lambda_{\min}}{2} = \frac{v_s}{2 \cdot F_{\max}}\,, \tag{2.42}$$

where $\lambda_{\min}$ is the wavelength corresponding to the maximum frequency component $F_{\max}$ of the bandlimited source signal. Similarly, for preventing from temporal aliasing

$$F_s \geq 2 \cdot F_{\max}\,, \tag{2.43}$$

(2.42), (2.43),(2.41) show that both sampling rates dependent on the frequency content of the signal and they are interrelated. Indeed in (2.41) d is inversely proportional to F_s meaning that there is a trade-off between the two sampling rates. For instance, if the sensor signal is bandlimited with $F_{\max} = 4\text{KHz}$ and we sample at the Nyquist rate $F_{\text{N}} = 8\text{KHz}$ then from (2.41) the microphone distance should be $d \leq \frac{355}{8000} = 0.0444\,\text{m}$ in order to avoid spatial aliasing. Note again that this consideration becomes important only when κ_q is estimated from the phase difference $\Delta\phi(k,m)$. For instance, algorithms computing the time-delays of signals using cross-correlations are not restricted in this manner.

In order to show another problem of methods utilizing κ_q as the separation parameter, we split the propagation plane, defined by the source position and the microphone array line, into two half-planes as in Fig. 2.2.

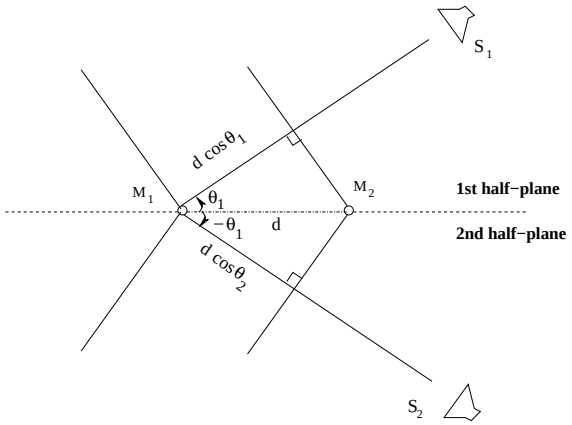

Figure 2.2: Illustration of the front-back ambiguity problem.

The angles are measured with respect to the microphone array line. Angles at the first half-plane are measured counter-clockwise and belong to $[0, \pi]$, and in the second half-plane clockwise and belong to $[0, -\pi]$. If two sources are coming from angles θ_1 and $\theta_2 = -\theta_1$, as in Fig. 2.2, then they are not separable in respect to κ_q (we have assumed omnidirectional sensors). This is the so-called front-back ambiguity of the array. Therefore in addition to the spatial diversity assumption we require for the DOA θ_{q_1}, θ_{q_2} of any two sources $s_{q_1}(n)$, $s_{q_2}(n)$ to be $\theta_{q_2} \neq -\theta_{q_1}$.

Window selection for spectrum estimation

In time-frequency masking, first the frequency domain representation of each sensor signal is *estimated* and then the delay and attenuation parameters, κ_q, a_q, are computed in order to separate the signals. Therefore the vulnerable point of the method, when an ideal model is assumed, is the spectrum estimation. The performance of the separation depends entirely on the accuracy of the spectrum estimation method. DFT spectrum estimation techniques depend on several factors like spectral sampling, blocking artifacts, windowing effects and other. Here we consentrate on window selection.

At each block time instant m a segment of the source signal $s_q(n)$ is viewed through a multiplication with a window function $w(n)$ of length L_w, and then it is transformed in the frequency domain. Multiplication in the time-domain corresponds to convolution in the frequency domain, i.e.,[1]

$$S_q(e^{j\cdot\omega}) = \frac{1}{2\cdot\pi} \cdot S_{q,\mathrm{true}}(e^{j\cdot\omega}) * W(e^{j\cdot\omega})\,. \tag{2.44}$$

[1]Here the discrete-time Fourier transform (DTFT) is used. The conclusions hold equivalently for the DFT, considering that DFT can be retrieved from appropriately sampling the DTFT.

where $S_{q,\text{true}}(e^{j\cdot\omega})$ is the true spectrum of the source signal. The convolution of $W(e^{j\cdot\omega})$ with $S_{q,\text{true}}(e^{j\cdot\omega})$ will tend to smooth sharp peaks and discontinuities in $S_{q,\text{true}}(e^{j\cdot\omega})$.

Ideally we want $W(e^{j\cdot\omega})$ to be equal to an impulse response. In reality the window spectrum consists of a main lobe and side lobs. Therefore it is desirable to have as narrow as possible main lobe, for better frequency resolution, and as low as possible side lobes' amplitude in order to avoid spectral smearing. The main lobe mostly depends on the length L_w of the window function and becomes narrower as L_w increases. The side lobes depend on the shape (amount of tapering) of the window function $w(n)$. In particular, it is desirable to taper the window to decrease smearing, and to use as long window as feasible to increase the frequency resolution.

On the other hand speech signals are short-time stationary. Therefore in the ST-DFT the primary purpose of the window is to limit the extend of the sequence to be transformed so that the spectral characteristics are reasonable stationary over the duration of the window. The more rapidly the signal characteristics change over time, the shorter the window should be. Therefore the choice of window length becomes a trade-off between frequency resolution (long window to narrow the main lobe) and time resolution (short window to fulfill the assumption of stationarity and capture the short-time properties of the signal).

In reality speech signals are not entirely W-DO in the frequency domain. The degree of overlap depends on the frequency resolution and it is not necessarily less if the resolution is large. Therefore the length of the window function determines not only the resolution of the spectrum but also the degree of W-DO. We discuss this matter in the next subsection.

2.2.2 Realistic mixing models

Speech signals in microphone mixtures are only approximately W-DO. Moreover, when the environment is echoic, reverberation complicates the estimation of the source signals from the corresponding path characteristics (attenuation and delay). We discuss these matters in detail in the following.

Anechoic model

The ideal model of the previous subsection assumed that speech signals in the time-frequency domain do not overlap. In reality this is partially true, i.e. in instantaneous, anechoic and echoic mixtures the signals are approximately W-DO. Frequency domain speech signals are sparse in that a small percentage of the time-frequency components capture a large percentage of the overall energy. Therefore like in (2.22) for W-DO, the *approximate* W-DO can be stated concicely

as

$$s_{q_1}(k,m) \cdot s_{q_2}(k,m) \approx 0 \quad \forall\, k,\, m\,. \tag{2.45}$$

This means that either $s_{q_1}(k,m)$ or $s_{q_2}(k,m)$ are very close to zero at each grid point (k,m).

The fact that the signals are approximately disjoint differentiates the analysis of the anechoic model with the analysis given for the ideal model in the following points

- The binary mask proposed in (2.30) is not the ideal one anymore.
- For the choice of the window function of the ST-DFT, the additional requirement of increasing the degree of W-DO should be taken into account.

That a binary mask can not be ideal, is obvious if we consider that few of the DFT coefficients of a speech signal have large magnitude and most of them are close but not exactly zero. Therefore a binary mask will allow some interference and cause some distortion on the objective signal. Consequently an ideal mask should appropriately weight the ST-DFT coefficients, depending on the portion of the objective signal in the specific time-frequency bin, i.e., $0 \leq M(k,m) \leq 1$. Unfortunately such a mask can not be constructed because we can never know the exact portion of the signal in a bin. Instead only binary masks (or approximately binary masks in the sense that zero and one are replaced with scalars close to zero and one respectively) can be constructed.

Now consider the family of the binary masks defined from the equation below

$$M_q^{\beta}(k,m) = \begin{cases} 1, & 20 \cdot \log\left(\frac{|S_q(k,m)|}{|C_q(k,m)|}\right) \geq \beta \\ 0, & \text{otherwise} \end{cases}, \quad q = 1, \ldots, Q\,, \tag{2.46}$$

where $C_q(k,m)$ is the summation of the sources interfering with the objective source $S_q(k,m)$ at bin (k,m)

$$C_{q_1}(k,m) = \sum_{\substack{q_1=1 \\ q \neq q_1}}^{Q} S_q(k,m)\,. \tag{2.47}$$

These kind of masks exist in reality because speech signals are sparse and it may be possible to identify them with algorithms like the one described in Subsect. 2.2.1. Then β would be the number of dB that the objective signal dominates the interfering ones in the bin of the worst case.

For the evaluation of the time-frequency masks described until now, the following performance criteria can be used [38]

- The preserved-signal ratio (PSR) of the mask, PSR_M, is defined as

$$PSR_M := \frac{\|M(k,m) \cdot S_q(k,m)\|^2}{\|S_q(k,m)\|^2} \tag{2.48}$$

  This measurement shows how well the mask preserves the source of interest $s_q(n)$, or equivalently how much the mask *distorts* this source. As $0 \leq M(k,m) \leq 1$, it is $\text{PSR}_M \leq 1$, with $\text{PSR}_M = 1$ for the ideal mask.

- The signal-to-interference ratio (SIR) of the mask, SIR_M, is defined as

$$SIR_M := \frac{M(k,m) \cdot S_q(k,m)}{M(k,m) \cdot C_q(k,m)} \tag{2.49}$$

  which is the signal-to-interference ratio in each time-frequency bin after using the mask to demix. Note that $\text{SIR}_M \in [0, \infty]$ with $\text{SIR}_M = \infty$ when we do not have interference.

The above criteria can be combined in order to form a measure of the degree of the W-DO of the signals in the mixture, in respect to a particular mask. The proposed criterion, WDO_M, is a normalized difference between the signal energy maintained in masking and the interference energy maintained in masking

$$\begin{aligned} WDO_M : &= \frac{\|M(k,m) \cdot S_q(k,m)\|^2 - \|M(k,m) \cdot C_q(k,m)\|^2}{\|S_q(k,m)\|^2} && (2.50)\\ &= PSR_M - \frac{PSR_M}{SIR_M} && (2.51)\\ &= PSR_M \left(1 - \frac{1}{SIR_M}\right) . && (2.52) \end{aligned}$$

Because $\text{SIR}_M \in [0, \infty]$ it is $\left(1 - \frac{1}{SIR_M}\right) \in [-\infty, 0]$ and therefore $\text{WDO}_M \in [-\infty, 1]$. In the case where the signals are W-DO then $\text{PSR}_M = 1$ and $\text{SIR}_M = \infty$ and thus $\text{WDO}_M = 1$. On the other hand $\text{WDO}_M = 1$ only when both $\text{PSR}_M = 1$ and $\text{SIR}_M = \infty$. Concluding the above expressions are equivalent. Summarizing the properties of WDO_M

$$\begin{aligned} WDO_M &\leq 1 , && (2.53)\\ WDO_M = 1 &\Leftrightarrow \begin{cases} PSR_M = 1 , \\ SIR_M = \infty . \end{cases} , \\ &\Leftrightarrow s_{q_1}(k,m) \cdot s_{q_2}(k,m) = 0 \quad \forall\, k, m . && (2.54) \end{aligned}$$

In [38] several subjective tests were performed on instantaneous mixtures of up to ten sources and confirmed the validity of the WDO_M criterion. In brief the

correspondence of the WDO_M value and the subjective grading of the separation is summarized below

$$\begin{aligned} WDO_M \in [0.8, 1] &\Rightarrow \text{perfect results}, \\ WDO_M \in [0.6, 0.6] &\Rightarrow \text{minor artifacts or interference}, \\ WDO_M < 0.6 &\Rightarrow \text{unsatisfying results}. \end{aligned}$$

The fact that speech signals are approximately W-DO puts an additional requirement to the ST-DFT, i.e. the ST-DFT should be optimized to produce not only a faithful spectrum estimate, which was discussed in Subsect. 2.2.1 but also increase the W-DO of the speech signals in the mixture. Consequently, most of the delay and attenuation parameters would be estimated well and the signals will be as disjoint as possible, thus succeeding to build the best possible separation mask. As were discussed, the parameters that can be tuned in the ST-DFT, is the length and the shape of the window function.

In [3, 2] the sparseness and disjointness of speech signals were investigated in respect to the frequency resolution. As a measure of sparseness the cumulative distribution of the power spectrum, P, were used, in order to investigate the power concentration of speech signals in specific harmonics.

The frequency resolution, f_{res}, in the DFT domain is given from

$$f_{res} = \frac{F_s}{L_w}. \tag{2.55}$$

For instance, the f_{res} of a signal sampled in the Nyquist frequency of $8\,KHz$ and transformed with the ST-DFT using a window of length 512 samples, is 15.625 Hz.

In [3, 2] P of a male and a female speech were computed for different f_{res}. The Hamming window was used for the ST-DFT. It was shown that

- As f_{res} increased from 80 to 10 Hz, the value of P decreased, meaning that the power of the signal concentrated on specific components. The P changed slightly when f_{res} was between 10 to 20 Hz.
- When f_{res} increased more, below 10 Hz, P were increased. This is due the non-stationarity of speech as were already discussed.
- The female speech was more sensitive to f_{res}, but both the speech signals showed similar behavior.

Concluding the optimum f_{res} for maximally sparse signals was between 10 to 20 Hz.

Moreover in [3, 2] the degree of overlap were investigated in respect to the frequency resolution. The evaluation parameter were the ratio of the amplitudes a_q. The experimental results showed again that the optimum f_{res} were again in the area of 10 to 20 Hz. [2]

The degree of W-DO in respect to the shape and length of the window function of the ST-DFT were investigated also from [7, p. 212], [38, p. 5]. In both works the optimum window function was the Hamming window, with other popular windows giving similar performance except from the rectangular. The optimum frequency resolution was again between 10 to 20 Hz.

Echoic model

Until now we assumed that a source signal is comming to a microphone from only one path. Realistic scenarios usually include *reverberation*, i.e. a source signal reaches a microphone from several paths. The degree of W-DO of the signals is not serioulsly affected as soon as the harmonic structure of a source signal is not changing significantly when propagating in different paths.

In contrary the problem now is the identification of the source signal components from the path parameters, i.e. the propagation delay and attenuation, (k_q, a_q) . In echoic environments several such parameters, k_q, a_q, refer to each source. Moreover, the parameters referring to one source may overlap or be very close with the parameters referring to another source. Consequently, several components of one source will be identified as belonging to another source. The 'spread' of these parameters increases in parallel with the reverberation.

It is apparent that the type of the sound acquition equipment now becomes important. For instance, directional sensors improve the performance of the method, as the effect of reverberation is reduced. Further the choice of the type of the sound acquition equipment (type of microphones, spacing between the sensors, etc.) usually determines which separation criterion to use, i.e. κ_q , a_q, or both (e.g. see [3, pp. 151-152]).

2.3 A method combining Time-Frequency Masking and ICA

The previous discussion revealed the main problems of time-frequency masking methods, namely that the speech signals are not perfectly W-DO in the time-frequency domain, and that in reverberation environments each source refers to

[2] These results certify also the strong relationship between sparseness and disjointness.

several paths which may be very similar with portion of paths referring to another source. Therefore simple clustering methods based on the characteristics of the paths (κ_q, a_q) will not give satisfying separation results. Consequently, in anechoic cases , but especially in reverberant environments more sophisticated methods are necessary.

Time-frequency masking methods which use κ_q, and a_q as separation parameters are similar to source localization techniques. Therefore their combination with beamforming methods may give improved results. In this frame [4] combined time-frequency masking with independent component analysis (ICA) [3] to separate three sources from two mixtures (3×2 case).

First the DOA is estimated and used as a criterion to label the time-frequency bins. Then in the ST-DFT domain one source is extracted from the sensor signals using time-frequency masking. Therefore the BSS problem is reduced to the quadratic case (2×2) and the two remaining sources can be separated by a conventional BSS algorithm. The same procedure is repeated two times to acquire all three sources, as shown in Fig. 2.3.

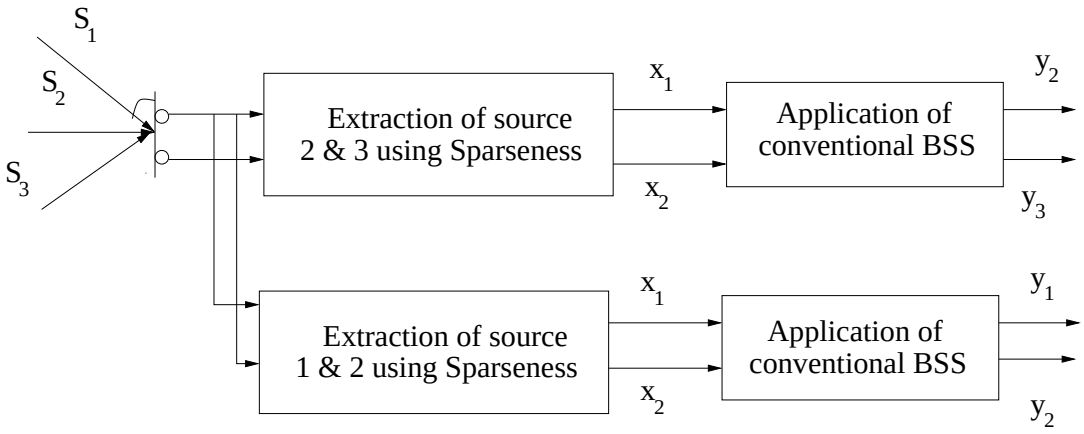

Figure 2.3: Time-frequency masking combined with a conventional BSS.

The algorithm is summarized in detail in the following

- Transform the two microphone mixtures in the ST-DFT domain using (2.20). We depict the equation again for convenience

$$X_p(k,m) = \sum_{n=0}^{L_w-1} x_p(n + m \cdot L_w) \cdot w(n) \cdot e^{\frac{-j \cdot 2 \cdot \cdot \pi \cdot k \cdot n}{L_w}} . \qquad (2.56)$$

[3] For more information about ICA look [18].

- Calculate the phase difference for each time-frequency bin as in (2.33).

$$\Delta\phi(k,m) = -\mathrm{Im}\left(\log\left(\frac{X_2(k,m)}{X_1(k,m)}\right)\right) . \tag{2.57}$$

- Compute the DOA for each bin

$$\theta(k,m) = \cos^{-1}\left(\frac{\phi(k,m)\cdot c}{f_{res}\cdot k\cdot d}\right) . \tag{2.58}$$

  Note that the criterion of the DOA depends on $\Delta\phi$ and is equivalent with the criterion of the time delay, κ_q, as discussed before.

- Plot a histogram for each frequency index k, i.e. built a histogram with the time-frequency components (k,m), which result from keeping constant the frequency index k and varying the time (block) index m. Do this for every k, i.e. $k = 0,\ldots,L_w-1$. For each frequency index k the histogram must reveal three peaks $\Theta_1^k, \Theta_2^k, \Theta_3^k$, which refer to the DOA of the three sources, $s_1(n), s_2(n), s_3(n)$, accordingly. The value of the DOA should not differ significantly along the histograms. Then their average will give a good estimate of the true DOA of the sources, i.e. $\Theta_q = \sum_{k=0}^{L_w-1}\Theta_q^k, \quad q = 1,2,3$.

- Based on the peaks Θ_q detected in the histogram construct two binary time-frequency masks

$$\mathrm{M}_q = \begin{cases} 1, & \theta_{min} \le \theta(k,m) \le \theta_{max} \\ 0, & \text{otherwise} \end{cases} , \quad q = 1,3 , \tag{2.59}$$

  where

$$q = 1 \;\Rightarrow\; \begin{cases} \theta_{min} &=& \Theta_1 - \Delta \\ \theta_{max} &=& 180^o \end{cases} , \tag{2.60}$$

$$q = 3 \;\Rightarrow\; \begin{cases} \theta_{min} &=& 0^o \\ \theta_{max} &=& \Theta_1 + \Delta \end{cases} , \tag{2.61}$$

  and Δ is a deign parameter (threshold) to decide the trade-off between separation performance and distortion. Hence, M_1, i.e. $q = 1$, eliminates source $s_1(n)$ and allow $s_2(n), s_3(n)$ in the mixture and M_3 eliminates $s_3(n)$ and allow the other two sources.

- Apply one of the masks to eliminate one source from the mixture and then use a conventional BSS method to separate the other two sources.

- Repeat again the last step using the other mask in order to allow in the mixture the previously eliminated source. Then the quadratic BSS method will retreive a better estimate of this source also.

The advantage of the combined method over the pure time-frequency masking methods is that time-frequency components of one source whose DOA indicates that they belong to a neighboring source will not be lost, as in conventional time-frequency masking methods. Therefore the quadratic BSS method will improve the separation and reduce the distortion (or else musical noise) of the source estimates.

Experimental result showed that this method give good results for the anechoic case, but the performance degrades with reverberation. The reason is that time-frequency masking methods assume the anechoic model, and as discussed before the parameters of time delay and attenuation of a path do not form anymore a faithful separation criterion. Therefore even if we extract pairs of signals a large part of components of the objective sources is eliminated from the mixture with the third source. We should also note that no improvement comes in the degree of W-DO of the signals with this method over the conventional time-frequency one.

2.4 Conclusions

The objective of this chapter was to analyze the methods which utilize the sparseness of speech signals to solve the underdetermined BSS problem, overweighting the time-frequency masking methods. The basic assumption of these methods were that

- The speech signals are W-DO in the time-frequency domain.
- The propagation of the speech signals is anechoic.

Under these assumptions there is a binary time-frequency mask that perfectly extract a source from the mixtures. For the idealized model the correct identification of this mask depends on

- The window function (shape and length) of the ST-DFT, which determines the faithfulness of the spectrum estimate.
- The spacing between adjacent microphones, which determines the trade-off between temporal and spatial sampling and therefore decides the existence of aliasing in time or space.

In realistic mixtures, speech signals are approximately W-DO. Therefore there is not exist a binary mask to extract entirely one source and cancel all the other. In this case, the optimal mask will be the one that collects the most of the energy of the objective signal and cancels most of the interfering energy. Under this frame a performance criterion of the mask, WDO_M, was proposed. This criterion measures the degree of the W-DO of the speech signals in the mixtures, as 'viewed' by the mask.

Moreover, when the environment is echoic, a signal arrives to a microphone through several paths. Therefore the precise identification of the optimal mask of a source, even in the sense described above, is not possible. Consequently, several coefficients that should be judged as belonging to one source, they are collected from other sources.

Time-frequency methods are similar to source localization techniques. Therefore the above problem is soften, if time-frequency methods are combined with beamforming approaches. This is shown in Sect. 2.3, where time-frequency masks are constructed, which extract pairs of sources. The labeling criterion is the DOA of the time-frequency components. Then ICA is used to separate the sources from the pairs.

From the above analysis we distinguish three main points to focus for the enhancement of the time-frequency masking methods

- Improve the precision of the spectrum identification method , such that most of the delay (or equivalently DOA) and attenuation parameters for labeling the source signals are estimated correctly.

- For echoic environments the labeling parameters of the sources overlap. Find techniques which can collect the source components even if their labeling parameters overlap.

- Investigate other domains where the degree of W-DO of the source signals in the mixtures is larger.

Therefore possible ways of enhancement of the methods are

- Improve the spectrum estimation method, e.g. investigate other windows for the ST-DFT method (e.g. cosine taper), evaluate more precise spectrum estimation techniques (e.g. spectrum averaging [39]).

- Employ source localization and beamforming techniques, e.g. for better DOA estimation [35, 6].

- Investigate more sophisticated clustering techniques, like neural networks, nearest neighbor and other.

- Investigate other transform methods. Promicing methods are expected to be the one that give higher degree of energy compaction in the transformation domain, compared to the ST-DFT. For instance, the discrete cosine transform (DCT) may give more sparse signals in the DCT domain [32, pp. 596-599].

- Time-frequency masking exploits spatial and spectral diversity of speech signals. In addition exploit the short time temporal structure of the speech signals, for both labeling and separating the signals.

Chapter 3

A class of Blind Deconvolution methods extended for Blind Source Factor Separation of Speech Signals

Single-channel blind deconvolution[1] (BD) is the DSP operation that reconstructs the original source signal that has been convolved with an unknown filter, representing e.g. the acoustical room impulse response. By definition the convoluting system is LTI and usually the input signal is assumed temporally independent and identically distributed (i.i.d.). In the case of a multichannel signal, e.g. several speakers or several telecomunications signals, the operation is called *multichannel blind deconvolution*.

There are three basic differences between blind deconvolution and BSS :

- In blind deconvolution there may be only one source, i.e. the underlying system may be single-input single-output (SISO), while in BSS there are always multiple sources and the mixing system may be multiple input and multiple output (MIMO).

- The interest in blind deconvolution is to deconvolve the source from the channel. In contrary, the task of BSS is to separate the sources, i.e. the outputs of the demixer may be filtered versions of the original sources.

- In blind deconvolution the input is assumed to be an i.i.d. signal, while in BSS it may possess various distributions and therefore may be temporally correlated, e.g., speech.

[1] Often called unsupervised deconvolution in adaptive filtering, or blind equalization in communications. For the latter the deconvoluting system is called equalizer.

Nevertheless, many techniques of convolutive BSS have been developed by extending methods originally designed for blind deconvolution. A usual practice is to use a Source Factor Separation (SFS) technique, where one source is separated from the mixtures, and combine it with a deflationary approach, where the sources are extracted one by one after deflating, i.e. removing, them from the mixed signals. The final step is to modify the algorithm for the separation of temporally correlated sources with various distributions, not only temporally i.i.d.

This chapter investigates an approach based on the single-channel blind deconvolution method proposed by Shalvi and Weinstein [34]. Initially Martone [27, 28] extended the method to MIMO blind deconvolution. Then Inouye and Tanebe [19] proposed a deflationary approach for global convergence and for independent but not necessarily identically distributed signals, therefore improving Martone's method. Finally Kawamoto and Inouye [24] further modified the method to allow for various signals, i.e. i.i.d, SOS or HOS colored. Their deflationary BSS method inherited the desirable attributes of the one proposed from Shalvi and Weinstein, namely global convergence at a very fast rate and separation even in the case of non-minimum phase systems.

The study of the development of the method is considered essential for three reasons:

- A thorough understanding of the underlying principles is necessary.
- The extension of the particular method to similar problems is more easily handled.
- The general practice of extending a method of blind deconvolution to a deflationary BSS may be applied to similar algorithms in the field. This may be useful in practical applications where only one desired signal is needed as, e.g., in hearing aids.

In the first section the blind deconvolution algorithm is derived from its fundamental principle and in the second section it is extended to source factor separation of various speech signals. The derivation concludes in the next chapter with the deflationary BSS approach.

The description is in the discrete time domain which makes the ideas and mathematical derivations easier to grasp. The unified treatment of this approach is the contribution of this chapter.

3.1 An exponential Blind Deconvolution method

In the following figure an unknown signal s_q is convolved with the unknown system $\mathbf{h}_{pq}$, and the estimated filter $\mathbf{w}_{rq}$ is used to recover them from the output x_p.

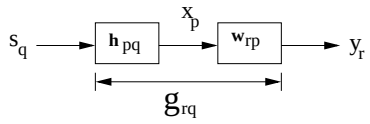

Figure 3.1: Singe channel blind deconvolution

For perfect deconvolution of the original source, $\mathbf{w}_{rq}$ ideally forces the cascade filter $\mathbf{g}_{rq}$ to be equal with a filter $\boldsymbol{\delta}_{rq}$, whose κ-th element is

$$\delta_{rq,\kappa} = \alpha_{rq} \cdot \delta(\kappa - k_{rq}) \,, \tag{3.1}$$

where

$\delta(\cdot)$	is the unit impulse,
α_{rq}	is a complex number denoting a scale change and phase shift,
k_{rq}	is an integer denoting a time shift.

If the filter taps $g_{rq,\kappa}$ were *known* a solution to the problem could be achieved by the following two steps *iterative* procedure[2] [34]:

$$\boxed{\begin{array}{rcll} g'_{rq,\kappa} & = & g^{u}_{rq,\kappa} \cdot (g^{*}_{rq,\kappa})^{v} & \kappa = -\infty, \ldots, \infty \,, \\ g''_{rq,\kappa} & = & \dfrac{1}{\left\|\mathbf{g}'_{rq}\right\|} \cdot g'_{rq,\kappa} & \kappa = -\infty, \ldots, \infty \,, \end{array}} \tag{3.2}$$

where

$\prime, \prime\prime$	denote the first and the second step of an iteration respectively ,
u, v	are non negative integers such that $u + v \geq 2$ [3],
$\left\|\mathbf{g}'_{rq}\right\|$	is the norm of the vector $\mathbf{g}'_{rq}$ spanned by the taps of the filter $\mathbf{g}'_{rq}$, ($\left\|\mathbf{g}'_{rq}\right\| = \sqrt{\mathbf{g}'^{H}_{rq}\mathbf{g}'_{rq}} = \sqrt{\sum_{\kappa} g'^{2}_{rq,\kappa}}$, where $\mathbf{g}'_{rq} = \left[g'_{rq,-\infty}, \ldots, g'_{rq,\infty}\right]^{T}$, and $g'_{rq,\kappa}$ is given from (3.2)).

Under the condition that the so called leading tap is larger than all the others, all but this tap will converge to zero at an *exponential rate*. A simple conceptual explanation[4] is:

– At the second step, which is a normalization, the magnitude of each tap becomes less than one.

[2]Where the g''_{rq} at the end of an iteration will become the g_{rq} of the next iteration.
[3]This follows logically because otherwise the taps will not be exponentiated.
[4]The convergence analysis and a relevant example can be found in the appendix.

– The subsequent exponentiations of the first step force, at an exponential rate, all but the leading tap to zero magnitude.

– After few iterations all the energy of the filter, which is normalized to one, is gathered at the leading tap.

Therefore by this iteration scheme the filter $\mathbf{g}_{rq}$ will converge always to the desired solution[5] of Equation 3.1 , where the time shift k_{rq} matches the position of the leading tap[6].

Unfortunately this idea is not directly applicable because in practical situation the filter $\mathbf{h}_{pq}$ is unknown. Instead $\mathbf{g}_{rq}$ should be computed implicitly from $\mathbf{w}_{rq}$, (i.e. exponentiated and normalized), at each iteration cycle. This is the subject of the following subsections.

Initially we formulate the problem in matrix notation. Then the MSE criterion is used to project the algorithm from "*g-*" to "*w-domain*", where altering $\mathbf{w}_{rq}$ causes the desirable effect to $\mathbf{g}_{rq}$. The derived algorithm is still not applicable to real-world scenarios because the involved systems are not known. SOS and HOS are employed to solve the problem of system identification and gradually turn the algorithm to a semi- and finally entirely blind one.

3.1.1 Problem formulation

For ilustration purposes it is convenient to express linear convolutions as matrix multiplications. Without loss of generality we make the following assumptions:

- Both $\mathbf{h}_{pq}$ and $\mathbf{w}_{rp}$ are causal FIR filters.
- As we are dealing with speech signals in this thesis, we assume that all the signals and systems are real valued.

Suppose the lengths of $\mathbf{h}_{pq}$, $\mathbf{w}_{rp}$ are N, L respectively, where $N > L$. Therefore the length of the cascade filter $\mathbf{g}_{rq}$ will be $K = N + L - 1$. The filter taps of the filters can be concatenated to vectors:

$$\mathbf{h}_{pq} = [h_{pq,0}, h_{pq,1}, \ldots, h_{pq,N-1}]^T , \quad (3.3)$$

$$\mathbf{w}_{rp} = [w_{rp,0}, w_{rp,1}, \ldots, w_{rp,L-1}]^T , \quad (3.4)$$

$$\mathbf{g}_{rq} = [g_{rq,0}, g_{rq,1}, \ldots, g_{rq,K-1}]^T . \quad (3.5)$$

[5]Except from pathological cases, for example if initially there are two leading taps.

[6]Note that the leading tap *position is preserved.* through the iterations, a fact that may be exploited in possible algorithms during the initialization of the fi lter [34, p. 508].

Similarly L samples $x_p(n)$ and K samples $s_q(n)$ form:

$$\mathbf{s}_q(n) = [s_q(n), s_q(n-1), \ldots, s_q(n-K+1)]^T , \tag{3.6}$$

$$\mathbf{x}_q(n) = [x_q(n), x_q(n-1), \ldots, x_q(n-L+1)]^T . \tag{3.7}$$

Now starting from $\mathbf{g}_{rq}$, convolution sums are expressed as matrix multiplications. Therefore

$$g_{rq,n} = \sum_{\kappa=0}^{L-1} h_{pq,n-\kappa} \cdot w_{rp,\kappa}, \qquad n = 0, \ldots, K-1 , \tag{3.8}$$

is written as

$$\begin{bmatrix} g_{rq,0} \\ g_{rq,1} \\ \vdots \\ g_{rq,N-1} \\ \vdots \\ g_{rq,K-1} \end{bmatrix} = \begin{bmatrix} h_{pq,0} & 0 & \cdots & 0 \\ h_{pq,1} & h_{pq,0} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ h_{pq,L-1} & \ddots & \ddots & h_{pq,0} \\ \vdots & \ddots & \ddots & \vdots \\ h_{pq,N-1} & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{pq,N-1} \end{bmatrix} \cdot \begin{bmatrix} w_{rp,0} \\ w_{rp,1} \\ \vdots \\ w_{rp,L-1} \end{bmatrix} . \tag{3.9}$$

The n^{th} sample of x_p is given from

$$x_p(n) = \sum_{\kappa=0}^{N-1} h_{pq,\kappa} \cdot s_q(n-\kappa), \tag{3.10}$$

or in matrix notation

$$x_p(n) = \begin{bmatrix} h_{pq,0} & h_{pq,1} & \cdots & h_{pq,N-1} \end{bmatrix} \cdot \begin{bmatrix} s_q(n) \\ s_q(n-1) \\ \vdots \\ s_q(n-N+1) \end{bmatrix} , \tag{3.11}$$

and for L samples

$$\begin{bmatrix} x_p(n) \\ x_p(n-1) \\ \vdots \\ x_p(n-L+1) \end{bmatrix} = \begin{bmatrix} h_{pq,0} & h_{pq,1} & \cdots & h_{pq,N-1} & 0 & \cdots & 0 \\ 0 & h_{pq,0} & h_{pq,1} & \cdots & h_{pq,N-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & h_{pq,0} & h_{pq,1} & \cdots & h_{pq,N-1} \end{bmatrix} \cdot \begin{bmatrix} s_q(n) \\ s_q(n-1) \\ \vdots \\ s_q(n-N+1) \\ \vdots \\ s_q(n-K+1)) \end{bmatrix} . \tag{3.12}$$

All the above equations can be written in compact form

$$\begin{aligned} \mathbf{g}_{rq} &= \mathbf{H}_{pq}^T \cdot \mathbf{w}_{rp}\,, && (3.13)\\ x_p(n) &= \mathbf{s}_q(n)^T \cdot \mathbf{g}_{rq}\,, && (3.14)\\ \mathbf{x}_p(n) &= \mathbf{H}_{pq} \cdot \mathbf{s}_q(n)\,. && (3.15) \end{aligned}$$

The $L \times (N + L - 1)$ matrix $\mathbf{H}_{pq}$ is the so called *filtering or Sylvester matrix* arising in several DSP problems. The Sylvester structure of this matrix is also used in the context of BSS (see, e.g., [9, 10]) Note that its rows consist of the taps of $\mathbf{h}_{pq}$ plus $L - 1$ zeros.

Likewise, the convolution sum of the output

$$\begin{aligned} y_r(n) &= \sum_{\kappa=0}^{L-1} w_{rp,\kappa} \cdot x_p(n-\kappa)\\ &= \sum_{\kappa=0}^{K-1} g_{rq,\kappa} \cdot s_q(n-\kappa)\,, && (3.16) \end{aligned}$$

is expressed as

$$\begin{aligned} y_r(n) &= \mathbf{x}_p(n)^T \cdot \mathbf{w}_{rq}\\ &= \mathbf{s}_q(n)^T \cdot \mathbf{g}_{rq}\,. && (3.17) \end{aligned}$$

3.1.2 From *g-domain* to *w-domain* with a MSE criterion

The iterative procedure described in the beginning of the chapter is not directly applicable because $\mathbf{g}_{rq}$ is unknown. Instead, $\mathbf{g}'_{rq}$ and $\mathbf{g}''_{rq}$ should be computed *implicitly* .

The *first iteration step* in (3.2) can be formulated as a Minimum Squared Error (MSE) estimation problem[7] – upgrade $\mathbf{g}_{rq}$ such that its distance from $\mathbf{g}'_{rq}$ is minimized

$$\min_{\hat{\mathbf{g}}'_{rq}} \left\| \hat{\mathbf{g}}'_{rq} - \mathbf{g}'_{rq} \right\|^2 \,, \qquad (3.18)$$

where $\mathbf{g}'_{rq}$ is the filter containing the taps of $\mathbf{g}_{rq}$ exponentiated to the power of $u + v$

$$\mathbf{g}'_{rq} = \left[g_{rq,0}^{u+v}, g_{rq,1}^{u+v}, \ldots, g_{rq,K-1}^{u+v} \right]^T \,, \qquad (3.19)$$

[7] On the contrary, Minimum Mean Square Error (MMSE) estimation is not applicable here because the filters are assumed to be deterministic.

and $\hat{\mathbf{g}}'_{rq}$ is the estimator of $\mathbf{g}'_{rq}$. The estimator will be computed by adjusting $\mathbf{w}_{rp}$ at each iteration step. Let $\mathbf{w}'_{rp}$ correspond to the first iteration step. Then from (3.13) we obtain

$$\hat{\mathbf{g}}'_{rq} = \mathbf{H}^T_{pq} \cdot \hat{\mathbf{w}}'_{rp}\,, \tag{3.20}$$

where $\hat{\mathbf{w}}'_{rp}$ is the estimator of $\mathbf{w}'_{rp}$ in vector form. Replacing $\hat{\mathbf{g}}'_{rq}$ and evaluating

$$\min_{\hat{\mathbf{w}}'_{rp}} \left\|\mathbf{H}^T_{pq} \cdot \hat{\mathbf{w}}'_{rp} - \mathbf{g}'_{rq}\right\|^2 \quad \Rightarrow \quad \min_{\hat{\mathbf{w}}'_{rp}}((\mathbf{H}^T_{pq} \cdot \hat{\mathbf{w}}'_{rp} - \mathbf{g}'_{rq})^T \cdot (\mathbf{H}^T_{pq} \cdot \hat{\mathbf{w}}'_{rp} - \mathbf{g}'_{rq}))\,, \tag{3.21}$$

by taking the gradient w.r.t. $\hat{\mathbf{w}}'_{rp}$, equating it to zero and assuming that the transpose of the filtering matrix $\mathbf{H}^T_{pq}$ is of full rank, the MSE estimator is acquired[8]

$$\nabla_{\hat{\mathbf{w}}'_{rp}} \left\|\mathbf{H}^T_{pq} \cdot \hat{\mathbf{w}}'_{rp} - \mathbf{g}'_{rq}\right\|^2 \stackrel{!}{=} 0 \quad \Rightarrow \quad \hat{\mathbf{w}}'_{rp_{MSE}} = (\mathbf{H}_{pq} \cdot \mathbf{H}^T_{pq})^{-1} \cdot \mathbf{H}_{pq} \cdot \mathbf{g}'_{rq}\,. \tag{3.22}$$

Assuming that $\hat{\mathbf{w}}'_{rp_{MSE}}$ is the true estimate of $\mathbf{w}'_{rp}$ the adjusted filter is

$$\mathbf{w}'_{rp} = (\mathbf{H}_{pq} \cdot \mathbf{H}^T_{pq})^{-1} \cdot \mathbf{H}_{pq} \cdot \mathbf{g}'_{rq}\,. \tag{3.23}$$

Let $\mathbf{w}''_{rp}$ correspond to the *second iteration step*. From Equation 3.20 the norm of $\mathbf{g}'_{rq}$ is

$$\begin{aligned} \left\|\mathbf{g}'_{rq}\right\| &= \left\|\mathbf{H}^T_{pq} \cdot \mathbf{w}'_{rp}\right\| \\ &= \mathbf{w}'^T_{rp} \cdot \mathbf{H}_{pq} \cdot \mathbf{H}^T_{pq} \cdot \mathbf{w}'_{rp}\,, \end{aligned} \tag{3.24}$$

and thus normalizing $\mathbf{g}'_{rq}$ is equivalent to

$$\mathbf{w}''_{rp} = \frac{\mathbf{w}'_{rp}}{\sqrt{\mathbf{w}'^T_{rp} \cdot \mathbf{H}_{pq} \cdot \mathbf{H}^T_{pq} \cdot \mathbf{w}'_{rp}}}\,. \tag{3.25}$$

[8]The $K \times L$ matrix $\mathbf{H}^T_{pq}$ is of full rank i.e. of full column rank because $L < K$ or more precisely $\operatorname{rank}\left\{\mathbf{H}^T_{pq}\right\} = L$. Then the $L \times L$ matrix $\mathbf{H}_{pq} \cdot \mathbf{H}^T_{pq}$ is also of full rank, and therefore invertible. This can be generalized for any, possibly non-minimum phase, LTI system $\mathbf{h}_{pq}$. If the respective $K \times L$ $\mathbf{H}^T_{pq}$ matrix, with up to infi nite number of rows K, is of full column rank, then the $L \times L$ matrix $\mathbf{H}_{pq} \cdot \mathbf{H}^T_{pq}$ is invertible and (3.22) is valid. Moreover Shalvi and Weinstein showed that the solution of (3.22) converges to the optimum solution in the mean square sense (Wiener solution), subject to the fi nite-length restriction of the equalizer $\mathbf{w}_{rp}$, such that the truncation effect is negligible. They stated also the necessary conditions such that $\mathbf{H}^T_{pq}$ is of full column rank, i.e., $\mathbf{h}_{pq}$ is stable in the sense $\sum_\kappa |h_{pq,\kappa}| < 0$, and its inverse, $h^{-1}_{pq,\kappa}$, $\kappa = -\infty, \ldots, \infty$, exists. The detailed analysis is given in [34, pp. 507-508].

In contrary, in predictive deconvolution, which is simple linear prediction employed for deconvolution, the minimum phase assumption for the convolutive system $\mathbf{h}_{pq}$ is fundamental [13, pp. 241-243,p. 685], [36, p. 423,p. 431]. This point reveals one of the advantages of the proposed method over similar ones in the fi eld.

Note that multiplying (3.25) from left with $\mathbf{H}_{pq}^T$ will give $\mathbf{g}_{rq}'' = \frac{\mathbf{g}_{rq}'}{\|\mathbf{g}_{rq}'\|} \Rightarrow \|\mathbf{g}_{rq}''\| = 1$.

Once again, the target of the first step is to change filter $\mathbf{g}_{rq}$ to a filter $\mathbf{g}_{rq}'$ with taps the taps of $\mathbf{g}_{rq}$ exponentiated to the power of $u+v$. $\mathbf{g}_{rq}$ is not accessible but it can be expressed in terms of $\mathbf{w}_{rp}$. Therefore change $\mathbf{w}_{rp}$ to $\mathbf{w}_{rp}'$ such that $\mathbf{g}_{rq}$ to change to $\mathbf{g}_{rq}'$. The desired $\mathbf{w}_{rp}'$ is given in Equation 3.23 by solving an MSE problem.– The second step is a simple normalization, and following the same logic, scale $\mathbf{w}_{rp}'$ to $\mathbf{w}_{rp}''$ such that $\mathbf{g}_{rq}'$ to change to $\mathbf{g}_{rq}''$ with energy one.

$$\mathbf{g}_{rq} = \mathbf{H}_{pq}^T \cdot \mathbf{w}_{rp} \quad \rightarrow \quad \mathbf{g}_{rq}' = \mathbf{H}_{pq}^T \cdot \mathbf{w}_{rp}' \,, \tag{3.26}$$

$$\mathbf{g}_{rq}'' = \mathbf{H}_{pq}^T \cdot \mathbf{w}_{rp}'' \quad \text{where} \quad \|\mathbf{g}_{rq}''\| = 1 \,. \tag{3.27}$$

Therefore the implicit algorithm of Equation 3.2, say in g-domain, changed to an explicit one, say in w-domain. This algorithm computes an MSE estimate of g_{rq}' and performs its normalization in the w-domain, at every iteration cycle. It is given in the box below.

$$\boxed{\begin{aligned} \mathbf{w}_{rp}' &= (\mathbf{H}_{pq} \cdot \mathbf{H}_{pq}^T)^{-1} \cdot \mathbf{H}_{pq} \cdot \mathbf{g}_{rq}' \,, \\ \mathbf{w}_{rp}'' &= \frac{\mathbf{w}_{rp}'}{\sqrt{\mathbf{w}_{rp}'^T \cdot \mathbf{H}_{pq} \cdot \mathbf{H}_{pq}^T \cdot \mathbf{w}_{rp}'}} \,. \end{aligned}} \tag{3.28}$$

The algorithm is still not applicable because the systems $\mathbf{H}_{pq} \cdot \mathbf{H}_{pq}^T$ and $\mathbf{H}_{pq} \cdot \mathbf{g}_{rq}'$ are not known. *The problem of blind deconvolution has been transformed to one of blind system identification.*

3.1.3 Semi-blind identification of $\mathbf{H}_{pq} \cdot \mathbf{H}_{pq}^T$ and $\mathbf{H}_{pq} \cdot \mathbf{g}_{rq}'$

The matrix products in (3.28) are identified from second and higher order cumulants. For simplicity we make the following assumptions

- all random processes are zero-mean and
- $u+v$ in (3.2) equals two.

The first assumption allows to express second and third order cumulants with the respective expectations (for the properties of cumulants used along the document refer to appendix B. For more information about cumulants consult [31]). $\mathbf{H}_{pq} \cdot \mathbf{g}_{rq}'$ is estimated from the $u+v+1$ *order* cumulants between the microphone samples x_p and $u+v$ output samples y_r of the deconvolution system. Therefore both

assumptions together allow to use third order expectations for the identification of $\mathbf{H}_{pq} \cdot \mathbf{g}'_{rq}$, i.e.

$$\mathbf{E}\left\{x_p(n-\kappa_i) \cdot y_r(n) \cdot y_r(n)\right\} \tag{3.29}$$

It is helpful to rewrite the filtering matrix (3.12) in terms of its rows

$$\mathbf{H}_{pq} = \left[\ (\mathbf{H}_{pq}^0)^T, (\mathbf{H}_{pq}^1)^T, \ldots, (\mathbf{H}_{pq}^{L-1})^T\ \right]^T\ , \tag{3.30}$$

where

$$\mathbf{H}_{pq}^{\kappa_i} = \left[\ \mathbf{0}_{\kappa_i}^T, \mathbf{h}_{pq}^T, \mathbf{0}_{L-1-\kappa_i}^T\ \right], \qquad \kappa_i = 0, 1, \ldots, L-1\,, \tag{3.31}$$

is the κ_i-th row of the filtering matrix, with dimension $1 \times K$, and $\mathbf{0}_\kappa$ is a vector of κ zeros. Now the third order expectation of (3.29) can be written in terms of inner products

$$\begin{aligned} & \mathbf{E}\left\{x_p(n-\kappa_i) \cdot y_r(n) \cdot y_r(n)\right\} \\ = \ & \mathbf{E}\left\{\mathbf{H}_{pq}^{\kappa_i} \cdot \mathbf{s}_q(n) \cdot \mathbf{g}_{rq}^T \cdot \mathbf{s}_q(n) \cdot \mathbf{s}_q(n)^T \cdot \mathbf{g}_{rq}\right\} \\ = \ & \sum_{\kappa=\kappa_i}^{\kappa_i+N-1} h_{pq,\kappa-\kappa_i}\left\{s_q(n-\kappa) \cdot \mathbf{g}_{rq}^T \cdot \mathbf{s}_q(n) \cdot \mathbf{s}_q(n)^T \cdot \mathbf{g}_{rq}\right\}\ , \end{aligned} \tag{3.32}$$

where in the second step matrix multiplication were replaced by the corresponding convolution sum and the linearity property of expectations were applied. Doing the same for the other inner products, the terms $g_{rq,\ldots}$ and $h_{pq,\ldots}$ can be extracted from the expectation operator leading to

$$\sum_{\kappa=\kappa_i}^{\kappa_i+N-1} h_{pq,\kappa-\kappa_i} \sum_{\kappa_1=0}^{K-1} g_{rq,\kappa-\kappa_1} \sum_{\kappa_2=0}^{K-1} g_{rq,\kappa-\kappa_2} \mathbf{E}\left\{s_q(n-\kappa) \cdot s_q(n-\kappa_1) \cdot s_q(n-\kappa_2)\right\}\ , \tag{3.33}$$

where $\kappa_i = 0, 1, \ldots, L-1$.

Because s_q is an i.i.d. process the expectation in (3.33) becomes

$$\mathbf{E}\left\{s_q(n-\kappa_i) \cdot s_q(n-\kappa_1) \cdot s_q(n-\kappa_2)\right\} = \begin{cases} \gamma_q^3, & \kappa_i = \kappa_1 = \kappa_2 \\ 0, & \text{else}\ , \end{cases} \tag{3.34}$$

where γ_q^3 is the third order (central) moment of s_q.

Therefore all the cross-products in (3.34) will vanish, and because $N < K$, only the expectations for lags $\kappa = \kappa_1 = \kappa_2 = \kappa_i$ will remain, scaled by the filter tap products $h_{pq,\kappa_i} \cdot g_{rq,\kappa_i}^2$

$$\mathbf{E}\left\{x_p(n-\kappa_i) \cdot y_r(n) \cdot y_r(n)\right\} = \sum_{\kappa=\kappa_i}^{\kappa_i+N-1} h_{pq,\kappa-\kappa_i} \cdot g_{rq,\kappa-\kappa_i}^2 \cdot \gamma_q^3\,, \tag{3.35}$$

$\kappa_i = 0, 1, \ldots, L-1$ and in matrix notation

$$\mathbf{E}\left\{x_p(n-\kappa_i)\cdot y_r(n)\cdot y_r(n)\right\} = \mathbf{H}_{pq}^{\kappa_i}\cdot \mathbf{g}_{rq}^T \cdot \mathbf{g}_{rq}\cdot \gamma_q^3\,, \qquad \kappa_i = 0, 1, \ldots, L-1\,. \tag{3.36}$$

It should be noted that L unwanted taps $g_{rq,\kappa}^2$ are discarded by the $L-1$ zeros of $\mathbf{H}_{pq}^{\kappa_i}$ each time, such that the above two equations are equivalent.

Then concatenating the L expectations of (3.36) with lags $\kappa_i = 0, 1, \cdots, L-1$, to a vector leads to

$$\begin{bmatrix} \mathbf{E}\left\{x_p(n)\cdot y_r(n)\cdot y_r(n)\right\} \\ \mathbf{E}\left\{x_p(n-1)\cdot y_r(n)\cdot y_r(n)\right\} \\ \vdots \\ \mathbf{E}\left\{x_p(n-L+1)\cdot y_r(n)\cdot y_r(n)\right\} \end{bmatrix} = \mathbf{H}_{pq}\cdot \mathbf{g}_{rq}^T\cdot \mathbf{g}_{rq}\cdot\gamma_q^3\,, \tag{3.37}$$

or equivalently

$$\mathbf{E}\left\{\mathbf{x}_p(n)\cdot y_r(n)\cdot y_r(n)\right\} = \mathbf{H}_{pq}\cdot\Sigma_{qq}^3\cdot\mathbf{g}'_{rq}\,, \tag{3.38}$$

where $\mathbf{g}'_{rq}$ given in (3.19), for $u+v=2$ is

$$\mathbf{g}'_{rq} = \left[g_{rq,0}^2, g_{rq,1}^2\ldots, g_{rq,L-1}^2\right]^T\,, \tag{3.39}$$

and Σ_{qq}^3 is a $K\times K$ diagonal matrix defined as

$$\Sigma_{qq}^3 = \text{diag}\left\{\gamma_q^3,\ldots,\gamma_q^3\right\}\,. \tag{3.40}$$

(In the appendix the product $\mathbf{H}_{pq}^{\kappa_i}\cdot\mathbf{s}_q(n)\cdot\mathbf{g}_{rq}^T\cdot\mathbf{s}_q(n)\cdot\mathbf{s}_q(n)^T\cdot\mathbf{g}_{rq}$ is evaluated and the algorithm is derived in a more intuitive way, which gives some insightful conclusions).

For the *identification of* $\mathbf{H}_{pq}\cdot\mathbf{H}_{pq}^T$ second order expectations of microphone samples are concatenated to form a vector outer product. Expectations preserve linear transformation, therefore

$$\begin{aligned}\mathbf{E}\left\{\mathbf{x}_p\cdot\mathbf{x}_p^T\right\} &= \mathbf{E}\left\{\mathbf{H}_{pq}\cdot\mathbf{s}_q\cdot\mathbf{s}_q^T\cdot\mathbf{H}_{pq}^T\right\} \\ &= \mathbf{H}_{pq}\cdot\Sigma_{qq}\cdot\mathbf{H}_{pq}^T\,,\end{aligned} \tag{3.41}$$

where Σ_{qq} is the $K\times K$ correlation matrix of s_q, which is diagonal because s_q is an i.i.d. signal. If σ_q^2 is the standard deviation of s_q, Σ_{qq} can be expressed as

$$\Sigma_{qq} = \mathbf{E}\left\{\mathbf{s}_q\cdot\mathbf{s}_q^T\right\} = \text{diag}\left\{\sigma_q^2,\ldots,\sigma_q^2\right\}\,, \tag{3.42}$$

The same strategy may be followed to arrive to (3.39) for any order of cumulants. In summary the algorithm for i.i.d. signals is

$$\boxed{\begin{aligned} \mathbf{H}_{pq} \cdot \Sigma_{qq} \cdot \mathbf{H}_{pq}^T &= \text{cum}\left\{\mathbf{x}_p(n), \mathbf{x}_p(n)^T\right\}, \\ \mathbf{H}_{pq} \cdot \Sigma_{qq}^{u+v+1} \cdot \mathbf{g}'_{rq} &= \text{cum}\left\{\mathbf{x}_p(n), \underbrace{y_r(n), y_r(n), \ldots, y_r(n)}_{u+v}\right\}, \end{aligned}} \tag{3.43}$$

where Σ_{qq}^{u+v+1} is the generalization of Σ_{qq}^3 in (3.40)

$$\Sigma_{qq}^{u+v+1} = \text{diag}\left\{\gamma_q^{u+v+1}, \cdots, \gamma_q^{u+v+1}\right\}. \tag{3.44}$$

Therefore if some statistics of the input signal are known, i.e. $\gamma_q^{u+v+1}, \sigma_q^2$, the blind system identification and therefore the blind deconvolution problem can be solved from (3.43) and then (3.28) respectively. This algorithm was proposed by Shalvi and Weinstein [34] and later it was extended to an entirely blind algorithm from Inouye and Tanebe [19] as we describe in the next subsection.

3.1.4 Blind identification of $\mathbf{H}_{pq} \cdot \Sigma_{qq} \cdot \mathbf{H}_{pq}^T$ and $\mathbf{H}_{pq} \cdot \Sigma_{qq} \cdot \mathbf{f}'_{rq}$

The algorithm of the previous section is semi-blind because Σ_{qq} and Σ_{qq}^{u+v+1} are unknown. It can become blind by incorporating these matrices to the MSE problem (3.21).

(3.43) may be rewritten as follows

$$\begin{aligned} \mathbf{H}_{pq} \cdot \Sigma_{qq}^{u+v+1} \cdot \mathbf{g}'_{rq} &= \mathbf{H}_{pq} \cdot \Sigma_{qq} \cdot \Sigma_{qq}^{-1} \cdot \Sigma_{qq}^{u+v+1} \cdot \mathbf{g}'_{rq} \\ &= \mathbf{H}_{pq} \cdot \Sigma_{qq} \cdot \mathbf{f}'_{rq}, \end{aligned} \tag{3.45}$$

where $\Sigma_{qq} \cdot \Sigma_{qq}^{-1}$ is equal to the $K \times K$ identity matrix and $\mathbf{f}'_{rq}$ is a $K \times 1$ vector defined as

$$\begin{aligned} \mathbf{f}'_{rq} &= \Sigma_{qq}^{-1} \cdot \Sigma_{qq}^{u+v+1} \cdot \mathbf{g}'_{rq} \\ &= \text{diag}\left\{\frac{\gamma_q^{u+v+1}}{\sigma_q^2}, \ldots, \frac{\gamma_q^{u+v+1}}{\sigma_q^2}\right\} \cdot \left[g_{rq,0}^{u+v}, \ldots, g_{rq,K-1}^{u+v}\right]^T \\ &= \left[\frac{\gamma_q^{u+v+1}}{\sigma_q^2} \cdot g_{rq,0}^{u+v}, \ldots, \frac{\gamma_q^{u+v+1}}{\sigma_q^2} \cdot g_{rq,K-1}^{u+v}\right]^T. \end{aligned} \tag{3.46}$$

In (3.46), Σ_{qq}^{-1}, Σ_{qq}^{u+v+1} and $\mathbf{g}'_{rq}$ were replaced by using (3.31), (3.41) and (3.19) respectively.

It can be seen that $\mathbf{f}'_{rq}$ is a statistically scaled (i.e., $\frac{\gamma_q^{u+v+1}}{\sigma_q^2}$) version of filter $\mathbf{g}'_{rq}$. Inouye and Tanebe [19] showed that the statistically scaled taps $f'_{rq,\kappa}$ also

converge to the desired solution of (3.1) through a similar iterative procedure as (3.2), if the leading tap condition is satisfied and the source signal s_q is stationary[9]. The iterative procedure proposed in [19] is given as

$$\boxed{\begin{aligned} f'_{rq,\kappa} &= \frac{\gamma_q^{u+v+1}}{\sigma_q^2} \cdot g_{rq,\kappa}^{u+v} \qquad \kappa = 0, \cdots, K-1\,, \\ f''_{rq,\kappa} &= \frac{f'_{rq,\kappa}}{\sqrt{\mathbf{f}'^T_{rq} \cdot \Sigma_{qq} \cdot \mathbf{f}'_{rq}}} \qquad \kappa = 0, \cdots, K-1\,. \end{aligned}} \tag{3.47}$$

Now the MSE problem of Subsect. 3.1.2 is modified accordingly to approximate $\mathbf{f}'_{rq}$ instead of $\mathbf{g}'_{rq}$.That is update the cascade filter g_{rq} such that the resulting filter to be equal with f'_{rq}. Proceeding in parallel with the derivations in Subsect. 3.1.2 we arrive to a realizable algorithm with the following MSE criterion

$$\min_{\hat{\mathbf{g}}'_{rq}} \left\| \hat{\mathbf{g}}'_{rq} - \mathbf{f}'_{rq} \right\|^2 \tag{3.48}$$

$$= \min_{\hat{\mathbf{w}}'_{rq}} ((\mathbf{H}^T_{pq} \cdot \hat{\mathbf{w}}'_{rp} - \mathbf{f}'_{rq})^T \cdot (\mathbf{H}^T_{pq} \cdot \hat{\mathbf{w}}'_{rp} - \mathbf{f}'_{rq}))$$

$$= \min_{\hat{\mathbf{w}}'_{rq}} ((\mathbf{H}^T_{pq} \cdot \hat{\mathbf{w}}'_{rp} - \mathbf{f}'_{rq})^T \cdot \Sigma_{qq} \cdot (\mathbf{H}^T_{pq} \cdot \hat{\mathbf{w}}'_{rq} - \mathbf{f}'_{rq}))\,, \tag{3.49}$$

leads to the deconvolving filter

$$\mathbf{w}'_{rp} = (\mathbf{H}_{pq} \cdot \Sigma_{qq} \cdot \mathbf{H}^T_{pq})^{-1} \cdot \mathbf{H}_{pq} \cdot \Sigma_{qq} \cdot \mathbf{f}'_{rq}\,. \tag{3.50}$$

where in (3.49) the multiplication with the positive definite correlation matrix Σ_{qq} will not change the direction of estimation. Moreover, it is assumed that it takes into account weighting errors arising during the estimation process.

By adjusting $\mathbf{w}_{rp}$ to $\mathbf{w}'_{rp}$ of (3.50) causes the cascade filter $\mathbf{g}_{rq}$ to be equal with $\mathbf{f}'_{rq}$ of (3.46). Similarly we should adjust $\mathbf{w}'_{rp}$ to $\mathbf{w}''_{rp}$ such that to normalize $\mathbf{f}'_{rq}$, i.e. the cascade filter $\mathbf{g}_{rq}$ to become equal with $\mathbf{f}''_{rq}$. Then from (3.13)

$$\mathbf{f}'_{rq} = \mathbf{H}^T_{pq} \cdot \mathbf{w}'_{rp}\,, \tag{3.51}$$

$$\mathbf{f}''_{rq} = \mathbf{H}^T_{pq} \cdot \mathbf{w}''_{rp}\,. \tag{3.52}$$

By inserting these two equations into the normalization step of (3.47) we obtain

$$\mathbf{H}^T_{pq} \cdot \mathbf{w}''_{rp} = \frac{\mathbf{H}^T_{pq} \cdot \mathbf{w}'_{rp}}{\sqrt{\mathbf{w}'^T_{rp} \cdot \mathbf{H}_{pq} \cdot \Sigma_{qq} \cdot \mathbf{H}^T_{pq} \cdot \mathbf{w}'_{rp}}}\,, \tag{3.53}$$

[9]If the signal s_q is stationary then the ratio $\frac{\gamma_q^{u+v+1}}{\sigma_q^2}$ will be constant along the iterations and the leading tap will be preserved. In contrary, for time-varying sources the statistical scaling of the taps will fluctuate with time and so the leading tap. Consequently, the system will never converge. For the case of short-time stationary signals the method will converge as soon as the ratio is preserved. For more details about this matter see the convergence analysis section in appendix C.1

and assuming that a left inverse of $\mathbf{H}_{pq}^T$ exists we arrive to

$$\mathbf{w}''_{rp} = \frac{\mathbf{w}'_{rp}}{\sqrt{\mathbf{w}'^T_{rp} \cdot \mathbf{H}_{pq} \cdot \Sigma_{qq} \cdot \mathbf{H}_{pq}^T \cdot \mathbf{w}'_{rp}}} \,. \tag{3.54}$$

Let $\mathbf{R}_{pp}$ be the $L \times L$ matrix containing the cross-cumulants of the microphone samples $x_p(n)$, and $\mathbf{d}_{rp}$ the vector containing cross-cumulants between the microphone samples and the output $y_r(n)$ as

$$\mathbf{R}_{pp} = \text{cum}\left\{\mathbf{x}_p(n), \mathbf{x}_p(n)^T\right\} \,, \tag{3.55}$$

$$\mathbf{d}_{rq} = \text{cum}\left\{\mathbf{x}_p(n), \underbrace{y_r(n), \ldots, y_r(n)}_{u+v}\right\} \,. \tag{3.56}$$

Then from (3.43)

$$\mathbf{R}_{pp} = \mathbf{H}_{pq} \cdot \Sigma_{qq} \cdot \mathbf{H}_{pq}^T \,, \tag{3.57}$$

$$\mathbf{d}_{rq} = \mathbf{H}_{pq} \cdot \Sigma_{qq}^{u+v+1} \cdot \mathbf{g}'_{rq} \,. \tag{3.58}$$

Now by replacing (3.57), (3.58) into (3.50), (3.54) the blind deconvolution algorithm for i.i.d. signals is acquired

$$\boxed{\begin{aligned} \mathbf{w}'_{rp} &= \mathbf{R}_{pp}^{-1} \cdot \mathbf{d}_{rp} \,, \\ \mathbf{w}''_{rp} &= \frac{\mathbf{w}'_{rp}}{\sqrt{\mathbf{w}'^T_{rp} \cdot \mathbf{R}_{pp} \cdot \mathbf{w}'_{rp}}} \,. \end{aligned}} \tag{3.59}$$

The algorithm described here for the single channel case was originally proposed in [19] for multichannel blind deconvolution and then extended in [24] for BSS of speech signals. In the subsequent sections we follow this development.

3.2 Extension to Blind Source Factor Separation of Speech Signals

This section deals with the application of the leading tap blind deconvolution method to BSS of signals with various distributions, not only temporally independent but also SOS or HOS colored, e.g., like speech. Several methods for BSS utilize FIR structures to demix the sources as in Fig. 3.2

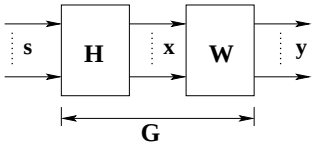

Figure 3.2: BSS using FIR structures.

where $\mathbf{s}$, $\mathbf{x}$ and $\mathbf{y}$ are Q, P and Q channel signals, and each system matrix $\mathbf{H}, \mathbf{W}$ consists of submatrices corresponding to the channels. These submatrices are Sylvester matrices containing the FIR filters. The mathematical relation between them is

$$\mathbf{x}(n) = \mathbf{H} \cdot \mathbf{s}(n)\,, \tag{3.60}$$

$$\mathbf{y}(n) = \mathbf{W} \cdot \mathbf{x}(n)\,. \tag{3.61}$$

It is useful to show their structure here.

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \dots & \mathbf{H}_{1Q} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \dots & \mathbf{H}_{2Q} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{H}_{P1} & \mathbf{H}_{P2} & \dots & \mathbf{H}_{PQ} \end{bmatrix}, \tag{3.62}$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{21} & \dots & \mathbf{W}_{Q1} \\ \mathbf{W}_{12} & \mathbf{W}_{22} & \dots & \mathbf{W}_{Q2} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{W}_{1P} & \mathbf{W}_{2P} & \dots & \mathbf{W}_{QP} \end{bmatrix}, \tag{3.63}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{21} & \dots & \mathbf{G}_{1Q} \\ \mathbf{G}_{12} & \mathbf{G}_{22} & \dots & \mathbf{G}_{2Q} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{G}_{1Q} & \mathbf{G}_{2Q} & \dots & \mathbf{G}_{QQ} \end{bmatrix} \tag{3.64}$$

where $\mathbf{G}$ is the cascade system matrix given from $\mathbf{G} = \mathbf{H}^T \cdot \mathbf{W} \cdot \mathbf{P}$, and $\mathbf{P}$ is a permutation matrix (the permutation matrix is not considered further here, for details see [19, p. 883]). Moreover, $\mathbf{W}$ and $\mathbf{G}$ can be expressed in terms of their columns $\mathbf{w_r}$, $\mathbf{g_r}$ respectively as in [19, p. 883] and defined in the next subsection

$$\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_r, \dots, \mathbf{w}_P]\,, \qquad \mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_r, \dots, \mathbf{g}_Q]\,. \tag{3.65}$$

There are mainly two ways to relate the BSS framework shown in Fig 3.2 to the blind deconvolution problem described in the previous sections. The first is to derive algorithms based on the combination of a blind source factor separation (BSFS) method with a deflationary approach, where the signals are extracted one by one after removing their contribution from the mixtures. The second is to conceive it related to a multichannel blind deconvolution problem. The problem

with this approach is that global convergence is not assured and therefore it will not be examined here. Nevertheless, we discuss the idea in the appendix in order to motivate for further research in the topic.

BSFS is the method which attempts to extract only one factor[10], i.e. source, out of the mixture. Hence, the multiple-input multiple-output (MIMO) FIR filter $\mathbf{W}$ used for BSS becomes a multiple-input single-output (MISO), and thus also the cascade filter $\mathbf{G}$ (left block diagram of Fig. 3.3). This means that they are reduced to one of their columns, i.e., $\mathbf{w}_r$ and $\mathbf{g}_r$ respectively. As it will be shown in the next subsection and depicted in Figure 3.3, by mathematical formulations it is possible to decouple the sources to refer to only one cascade filter $\mathbf{g}_{rq}$. Then, if the filters $\mathbf{g}_{rq}$ are viewed as taps of the filter $\mathbf{g}_r$, the leading tap blind deconvolution method, as derived in Sect. 3.1, can be analogously *projected* to a *leading filter* method where all the *filters* are cancelled except from one. This idea is formulated mathematically in subsections 3.2.2 and 3.2.3 for i.i.d. and speech signals respectively. The same method can be repeated Q times in combination with a deflationary approach to extract all the sources. The utilization of the leading tap blind deconvolution method to solve the BSFS problem is the subject of the next subsections. Its combination with a deflationary approach to solve the entire BSS problem is considered in the next chapter.

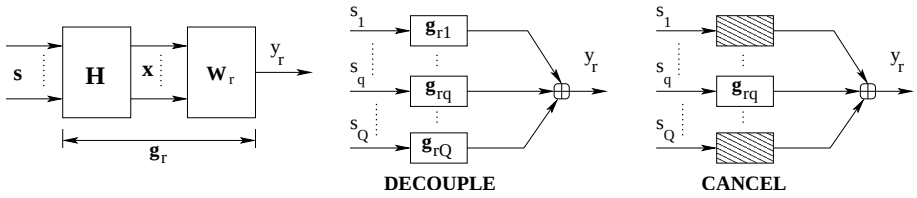

Figure 3.3: BSFS structure decoupled to Q FIR filters.

There are two more problems that should be solved before the method is capable of separating speech signals. The identification of $\mathbf{H}_{pq} \cdot \Sigma_{qq}^{u+v+1} \cdot \mathbf{f}'_{rq}$ in (3.43) from higher order cumulants (3.45), is possible if the source signal s_q is i.d. (independently distributed), which is not the case for speech signals. Therefore $\mathbf{d}_{rp}$ in (3.56) is not equal with $\mathbf{H}_{pq} \cdot \Sigma_{qq}^{u+v+1} \cdot \mathbf{f}'_{rq}$ and consequently, the blind deconvolution algorithm in (3.58) is not valid. Furthermore, $\mathbf{H}_{pq} \cdot \Sigma_{qq}^{u+v+1} \cdot \mathbf{f}'_{rq}$ in (3.45), (3.43) is identified from HOS (from (3.2) it should be $u + v > 2$). Hence, SOS , where speech signals are well separable, can not be used. Concluding a modification in (3.45), (3.43) is necessary.

[10]We borrowed this term from factor analysis (FA). For more information about FA look [18, p. 138], [5]

In Subsect. 3.2.2 we extend the blind deconvolution algorithm of (3.59) for BSFS of i.i.d. signals and in Subsect. 3.2.3 we further modify it for separation of speech signals utilizing only SOS.

3.2.1 Problem formulation

Proceeding as in the single channel case the BSFS problem is formulated in matrix notation under the same assumptions of Subsect. 3.1.1, i.e., all signals and systems are real valued and in addition all the systems are casual FIR. First consider the single input single output (SISO) filter $\mathbf{g}_{rq}$ from source s_q, through P microphones, to output y_r as shown in Fig. 3.4.

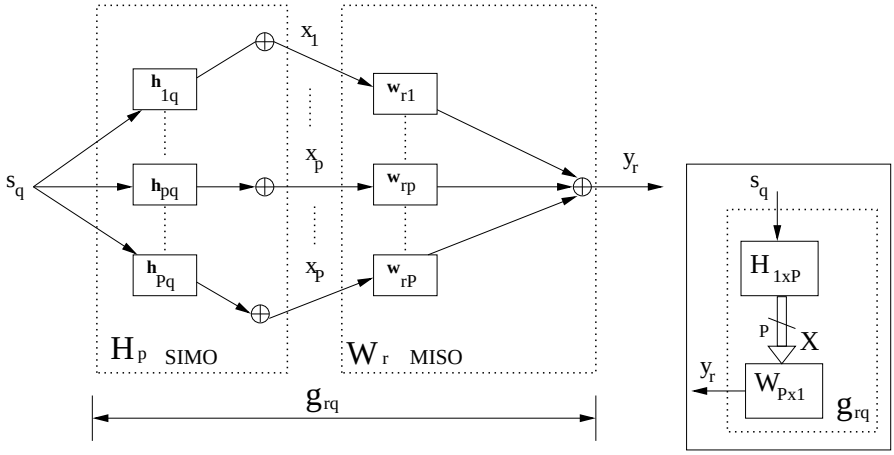

Figure 3.4: Blind deconvolution using multiple sensors.

This filter is the linear combination of the respective subfilters

$$g_{rq,n} = \sum_{p=1}^{P} \sum_{\kappa=0}^{L-1} h_{pq,n-\kappa} \cdot w_{rp,\kappa}, \qquad n = 0, \ldots, K-1\,, \tag{3.66}$$

where $\mathbf{g}_{rq}$, $\mathbf{h}_{pq}$ and $\mathbf{w}_{rp}$ defined as in Subsect. 3.1.1. Writing the linear convolutions as matrix multiplications:

$$\mathbf{g}_{rq} = \sum_{p=1}^{P} \mathbf{H}_{pq} \cdot \mathbf{w}_{rp}\,. \tag{3.67}$$

If we extend the system depicted in Fig. 3.4 to multiple sources we obtain the source factor separation scenario with multiple inputs and one output as shown in Fig. 3.5.

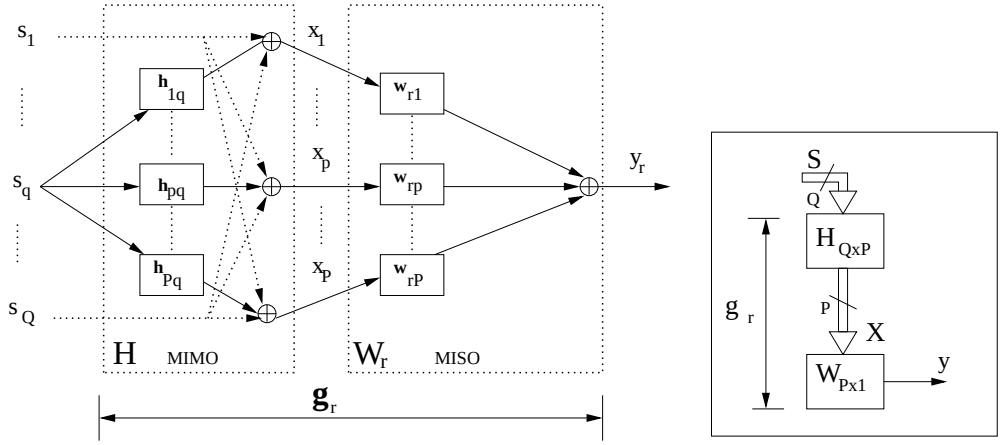

Figure 3.5: Blind source factor separation.

The signal at the p-th microphone is given as

$$x_p(n) = \sum_{q=1}^{Q} \sum_{\kappa=0}^{N-1} h_{pq} \cdot s_q(n-\kappa)\,. \tag{3.68}$$

Concatenating L samples and expressing them using matrix notation results in

$$\begin{aligned} \mathbf{x}_p(n) &= \sum_{q=1}^{Q} \mathbf{H}_{pq} \cdot \mathbf{s}_q(n) \\ &= \mathbf{H}_p \cdot \mathbf{s}(n)\,, \end{aligned} \tag{3.69}$$

where $\mathbf{H}_p$ is a block matrix of size $L \times K \cdot Q$ and $\mathbf{s}_q(n)$ an $K \cdot Q \times 1$ block vector

$$\mathbf{H}_p = [\mathbf{H}_{p1}, \mathbf{H}_{p2}, \ldots, \mathbf{H}_{pQ}]\ , \tag{3.70}$$

$$\mathbf{s}(n) = \left[\mathbf{s}_1^T(n), \mathbf{s}_2^T(n), \ldots, \mathbf{s}_Q^T(n)\right]^T\ . \tag{3.71}$$

Concatenating the vectors $\mathbf{x}_p(n)$, $p = 1, \ldots, P$, leads to one block vector of size $P \cdot L \times 1$

$$\mathbf{x}(n) = \mathbf{H} \cdot \mathbf{s}(n)\,, \tag{3.72}$$

where

$$\mathbf{x}(n) = \left[\mathbf{x}_1^T(n), \mathbf{x}_2^T(n), \ldots, \mathbf{x}_P^T(n)\right]^T\ ,\ , \tag{3.73}$$

and $\mathbf{H}$ is as in (3.62). This is the filtering matrix in the multichannel case consisting of channel-wise submatrices $\mathbf{H}_{pq}$ exhibiting the Sylvester structure shown in

(3.9, 3.12). For perfect reconstruction of the source signals it is assumed that there exists a left inverse of $\mathbf{H}$, i.e. $\mathbf{H}$ is of full column rank [19, p. 882], [26, p. 1499], [23, p. 598]. This implies that the number of microphones should be larger or at least equal to the number of sources[11].

The output signal in matrix notation is

$$\begin{aligned} y_r(n) &= \sum_{p=1}^{P}\sum_{\kappa=0}^{L-1} x_p(n-\kappa)\cdot w_{rp,\kappa} \\ &= \sum_{p=1}^{P} \mathbf{x}_p^T(n)\cdot \mathbf{w}_{rp} \\ &= \mathbf{x}^T(n)\mathbf{w}_r\,, \end{aligned} \tag{3.74}$$

where $\mathbf{w}_r$ is the tap-stacked column vector containing all demixing filter weights

$$\mathbf{w}_r = \left[\mathbf{w}_{r1}^T, \mathbf{w}_{r2}^T, \cdots, \mathbf{w}_{rP}^T\right]^T\,, \tag{3.75}$$

and replacing $\mathbf{x}(n)$ with the expression given in (3.72) results in

$$\begin{aligned} y_r(n) &= \mathbf{s}^T(n)\cdot\mathbf{H}^T\cdot\mathbf{w}_r \\ &= \mathbf{s}^T(n)\cdot\mathbf{g}_r\,, \end{aligned} \tag{3.76}$$

where $\mathbf{g}_r$ is the overall filter as can be seen in Fig. 3.5

$$\mathbf{g}_r = \mathbf{H}^T\cdot\mathbf{w}_r\,, \tag{3.77}$$

Therefore $\mathbf{g}_r$ is a block vector of size $K\cdot Q\times 1$

$$\mathbf{g}_r = \left[\mathbf{g}_{r1}^T, \mathbf{g}_{r2}^T, \ldots, \mathbf{g}_{rQ}^T\right]^T\,, \tag{3.78}$$

and $\mathbf{g}_{rq}$ is defined in (3.5).

Note that in (3.76) each source $s_q(n)$ depends only on the respective filter g_{rq}. This fact is fundamental in the development of the "leading filter" algorithm.

For convenience we summarize the multichannel relations

$$\mathbf{x}(n) = \mathbf{H}\cdot\mathbf{s}(n)\,, \tag{3.79}$$

$$y_r(n) = \mathbf{s}^T(n)\cdot\mathbf{g}_r\,, \tag{3.80}$$

$$\mathbf{g}_r = \mathbf{H}^T\cdot\mathbf{w}_r\,. \tag{3.81}$$

Based on this formulations we can proceed to derive the multichannel algorithms for separation of i.i.d. or speech signals.

[11]The $L\cdot P\times K\cdot Q$ matrix $\mathbf{H}$ is of full column rank implies that $K\cdot Q\le L\cdot P$ and as L can not be larger than K (from Subsect. 3.1.1 it is $K=N+L-1$), P should be larger or the most equal with Q (when $N=1$). Therefore the number of microphones should be more or at least equal to the number of sources.

3.2.2 Blind Source Factor Separation of i.i.d. Signals

In this subsection we consider the application of the leading tap blind deconvolution algorithm (3.59) on the BSFS problem. The mathematical derivations on this subsection are exactly the same with the ones in Sect. 3.1 , extrapolated for the multichannel case. First we introduce the leading tap algorithm for the unaccessible (exponentiated and statistically scaled) taps of the cascade filter $\mathbf{g}_r$. Then using the MSE criterion we estimate the corresponding FIR filters $\mathbf{w}_r$. In the next step the unknown systems, $\mathbf{H}^T \cdot \Sigma \cdot \mathbf{H}$ and $\mathbf{H} \cdot \Sigma \cdot \mathbf{f}'_r$, are estimated from second and higher order cumulants. Finally the realizable algorithm is illustrated in a convenient form.

The multichannel leading tap algorithm

We define the $K \cdot Q \times 1$ block vector $\mathbf{f}'_r$ with blocks of the vectors $\mathbf{f}'_{rq}$, $q = 1, \ldots, Q$ as

$$\begin{aligned} \mathbf{f}'_r &= \left[\mathbf{f}'^T_{r1}, \mathbf{f}'^T_{r2}, \ldots, \mathbf{f}'^T_{rQ}\right]^T \\ &= \left[f'_{r1,0}, \ldots, f'_{r1,K-1}, \ldots, f'_{rQ,0}, \ldots, f'_{rQ,K-1}\right]^T . \end{aligned} \tag{3.82}$$

Let $\mathbf{f}'_r$ given from

$$\mathbf{f}'_r = \Sigma^{-1} \cdot \Sigma^{u+v+1} \cdot \mathbf{g}'_r \, . \tag{3.83}$$

$\mathbf{g}'_r$ is a block vector of length $K \cdot Q \times 1$ and contains the taps of the multichannel filter $\mathbf{g}_r$ exponentiated to the power of $u + v$

$$\begin{aligned} \mathbf{g}'_r &= \left[\mathbf{g}'^T_{r1}, \mathbf{g}'^T_{r2}, \ldots, \mathbf{g}'^T_{rq}\right]^T \\ &= \left[g'_{r1,0}, \ldots, g'_{r1,K-1}, \ldots, g'_{rQ,0}, \ldots, g'_{rQ,K-1}\right]^T \\ &= \left[g^{u+v}_{r1,0}, \ldots, g^{u+v}_{r1,K-1}, \ldots, g^{u+v}_{rQ,0}, \ldots, g^{u+v}_{rQ,K-1}\right]^T \end{aligned} \tag{3.84}$$

Σ is the $K \cdot Q \times K \cdot Q$ correlation matrix of the multichannel source signal of (3.71)

$$\Sigma = \mathbf{E}\left\{\mathbf{s} \cdot \mathbf{s}^T\right\} \, . \tag{3.85}$$

From the assumption that the signals are spatially uncorrelated this matrix is reduced to a block diagonal matrix

$$\Sigma = \begin{bmatrix} \Sigma_{11} & & & \\ & \Sigma_{22} & & \\ & & \ddots & \\ & & & \Sigma_{QQ} \end{bmatrix} , \tag{3.86}$$

where the blocks Σ_{qq} are the $K \times K$ correlation matrices of the source signals $s_q, q = 1, \ldots, Q$. For *temporally i.i.d* source signals their correlation matrices are diagonal as in (3.42). Therefore Σ is also diagonal and similarly its inverse

$$\Sigma^{-1} = \mathrm{diag}\left\{\sigma_1^{-2}, \ldots \sigma_1^{-2}, \ldots, \sigma_Q^{-2}, \ldots, \sigma_Q^{-2}\right\} \tag{3.87}$$

For i.i.d source signals, we define the $K \cdot Q \times K \cdot Q$ diagonal matrix Σ^{u+v+1} as

$$\Sigma^{u+v+1} = \mathrm{diag}\left\{\gamma_1^{u+v+1}, \ldots, \gamma_1^{u+v+1}, \ldots, \gamma_Q^{u+v+1}, \ldots, \gamma_Q^{u+v+1}\right\} . \tag{3.88}$$

where γ_q^{u+v+1} is the $(u+v+1)$-th (central) moment of $s_q,\ q = 1, \ldots, Q$.

Substituting (3.84), (3.87), (3.88), into (3.83), $\mathbf{f}'_r$ becomes

$$\mathbf{f}'_r = \left[\frac{\gamma_1^{u+v+1}}{\sigma_1^2} \cdot g_{r1,0}^{u+v}, \ldots, \frac{\gamma_1^{u+v+1}}{\sigma_1^2} \cdot g_{r1,K-1}^{u+v}, \ldots, \frac{\gamma_Q^{u+v+1}}{\sigma_Q^2} \cdot g_{rQ,0}^{u+v}, \ldots, \frac{\gamma_Q^{u+v+1}}{\sigma_Q^2} \cdot g_{rQ,K-1}^{u+v}\right]^T \tag{3.89}$$

Note that $\mathbf{f}'_r$ contains the statistically scaled and exponentiated taps of the multichannel filter $\mathbf{g}_r$. This convenient form of $\mathbf{f}'_r$ in (3.89) is only possible under the assumption that the source signals are *spatiotemporally i.i.d.* . Then the following iteration scheme can be designed

$$\boxed{\begin{aligned} &for \quad q = 1, \ldots, Q \quad f'_{rq,\kappa} \;=\; \frac{\gamma_q^{u+v+1}}{\sigma_q^2} \cdot g_{rq,\kappa}^{u+v} \qquad \kappa = 0, \cdots, K-1\,, \\ &for \quad q = 1, \ldots, Q \quad f''_{rq,\kappa} \;=\; \frac{f'_{rq,\kappa}}{\sqrt{\mathbf{f}_r'^T \cdot \Sigma \cdot \mathbf{f}'_r}} \qquad \kappa = 0, \cdots, K-1\,. \end{aligned}} \tag{3.90}$$

where (3.90) constitutes one iteration cycle. At the end of each iteration cycle $\mathbf{g}_r$ is replaced with the resulting $\mathbf{f}''_r$. Inouye and Tanebe [19] showed that if the summation of $u + v$ is larger than two, the leading tap condition is satisfied and the source signals are stationary, the cascade filter $\mathbf{g}_r$ will converge to the desired solution, i.e.

$$\mathbf{g}_{rq} = \begin{cases} \boldsymbol{\delta}_{rq_1} & , \quad q = q_1 \\ \mathbf{0}_{K\times 1} & , \quad q \neq q_1 \end{cases} \quad , \qquad q = 1, \ldots, Q \tag{3.91}$$

where $\boldsymbol{\delta}_{rq_1}$ is a $K \times 1$ filter with all taps equal to zero except from one, like in (3.1), and $\mathbf{0}_{K\times 1}$ is a filter with $K \times 1$ zero taps. Then from (3.76) only $s_{q_1}(n)$ will be extracted in the output $y_r(n)$.

MSE criterion for the estimation of $\mathbf{w_r}$

As in the single channel case (Subsections 3.1.3, 3.1.4), the cascade filter $\mathbf{g}_r$ and the statistics of the sources are not known. Hence, $\mathbf{f}'_r$ can not be computed explicitly from (3.89) and the iteration scheme is not realizable. Instead, its estimation is possible. The first iteration step is actually an exponentiation and scaling of the taps of $\mathbf{g}_r$. The filter $\mathbf{g}_r$ is not accessible but as we can see from (3.77) it can be altered by adjusting $\mathbf{w}_r$. This is clearly a MSE problem, to minimize the distance between the vectors $\mathbf{H} \cdot \mathbf{w}_r$ and $\mathbf{f}'_r$ by adjusting the parameter vector $\mathbf{w}_r$

$$\min_{\hat{\mathbf{g}}_r} \|\hat{\mathbf{g}}_r - \mathbf{f}'_r\|^2 = \min_{\hat{\mathbf{w}}_r}((\mathbf{H}^T \cdot \hat{\mathbf{w}}_r - \mathbf{f}'_r)^T \cdot (\mathbf{H}^T \cdot \hat{\mathbf{w}}_r - \mathbf{f}'_r)) \quad (3.92)$$

$$= \min_{\hat{\mathbf{w}}_r}((\mathbf{H}^T \cdot \hat{\mathbf{w}}_r - \mathbf{f}'_r)^T \cdot \Sigma \cdot (\mathbf{H}^T \cdot \hat{\mathbf{w}}_r - \mathbf{f}'_r)) \,, \quad (3.93)$$

which leads to

$$\mathbf{w}'_r = (\mathbf{H} \cdot \Sigma \cdot \mathbf{H}^T)^\dagger \cdot \mathbf{H} \cdot \Sigma \cdot \mathbf{f}'_r \,. \quad (3.94)$$

The symbol † means the pseudoinverse of $\mathbf{H} \cdot \Sigma \cdot \mathbf{H}^T$, because as $\mathbf{H}$ were assumed full column rank the inverse of $\mathbf{H} \cdot \Sigma \cdot \mathbf{H}^T$ is very probable not to exist [19, pp. 882-883]. The tone "′" above $\mathbf{w}_r$ denotes that the estimated filter refers to the first iteration step. Now the product $\mathbf{H} \cdot \mathbf{w}'_r$ is a $K \cdot Q \times 1$ filter whose taps are equal with the exponentiated and statistically scaled taps of $\mathbf{g}_r$ before the estimation, i.e.

$$\mathbf{H}^T \cdot \mathbf{w}'_r = \mathbf{f}'_r \,. \quad (3.95)$$

In the next iteration step, $\mathbf{f}'_r$ is normalized. Again following the same idea to alter $\mathbf{w}'_r$, of the first step, such that the product of the resulting filter $\mathbf{w}''_r$ with $\mathbf{H}$ becomes a normalized version of the cascade filter resulted from the previous step ($\mathbf{H}^T \cdot \mathbf{w}'_r$), i.e.

$$\mathbf{H}^T \cdot \mathbf{w}''_r = \mathbf{f}'' \,. \quad (3.96)$$

The double tone "″" above $\mathbf{w}_r$ denotes that the estimated filter refers to the second iteration step.

This is clearly a scaling of $\mathbf{w}'_r$. Therefore, by first writing the second iteration step of (3.90) as

$$\mathbf{f}'' = \frac{\mathbf{f}'_r}{\|\mathbf{f}'_r\|} = \frac{\mathbf{f}'_r}{\mathbf{f}'^T_r \cdot \mathbf{f}'_r} \,, \quad (3.97)$$

then inserting (3.95), (3.96) into (3.97)

$$\mathbf{H}^T \cdot \mathbf{w}''_r = \frac{\mathbf{H}^T \cdot \mathbf{w}'_r}{\sqrt{\mathbf{w}'^T_r \cdot \mathbf{H} \cdot \Sigma \cdot \mathbf{H}^T \cdot \mathbf{w}'_r}} \,, \quad (3.98)$$

and assuming that there exists a left inverse of $\mathbf{H}^T$ the normalized filter is acquired

$$\mathbf{w}_r'' = \frac{\mathbf{w}_r'}{\sqrt{\mathbf{w}_r'^T \cdot \mathbf{H} \cdot \Sigma \cdot \mathbf{H}^T \cdot \mathbf{w}_r'}} \,. \tag{3.99}$$

Therefore by altering the FIR filter w_r the two step procedure of (3.90) can be realized from (3.94) and (3.99).

Blind system identification

Still the problem is not solved because the matrix products $\mathbf{H} \cdot \Sigma \cdot \mathbf{f}_r'$ and $\mathbf{H} \cdot \Sigma \cdot \mathbf{H}^T$ are not known. We estimate them from cumulants as it is done in subsections 3.1.3 and 3.1.4, for the singe channel case. First the block row $\mathbf{H}_p$ of the multichannel filtering matrix (3.71) is written in terms of its rows

$$\begin{aligned} \mathbf{H}_p^{\kappa_i} &= \left[\mathbf{H}_{p1}^{\kappa_i}, \ldots, \mathbf{H}_{pQ}^{\kappa_i}\right] \\ &= \left[\mathbf{0}_{\kappa_i}^T, \mathbf{h}_{p1}^T, \mathbf{0}_{L-1-\kappa_i}^T, \ldots, \mathbf{0}_{\kappa_i}^T, \mathbf{h}_{pQ}^T, \mathbf{0}_{L-1-\kappa_i}^T\right], \quad \kappa_i = 0, 1, \ldots, L-1 \,, \end{aligned} \tag{3.100}$$

where $\mathbf{H}_p^{\kappa_i}$ denotes the κ_i^{th} row of $\mathbf{H}_p$. Then the κ_i^{th} microphone sample is

$$x_p(n - \kappa_i) = \mathbf{H}_p^{\kappa_i} \cdot \mathbf{s} \tag{3.101}$$

For simplicity we assume zero-mean signals and that $u + v$ equals to two. Then third order expectation between the microphone signals $x_p(n)$ and the output $y_r(n)$ can be used to estimate $\mathbf{H} \cdot \Sigma \cdot \mathbf{f}_r'$

$$\mathbf{E}\left\{x_p(n - \kappa_i) \cdot y_r(n) \cdot y_r(n)\right\} = \mathbf{E}\left\{\mathbf{H}_p^{\kappa_i} \cdot \mathbf{s}(n) \cdot \mathbf{g}_r^T \cdot \mathbf{s}(n)^T \cdot \mathbf{g}_r\right\} \tag{3.102}$$

In similar way with (3.32 - 3.36) we arrive to

$$\mathbf{E}\left\{x_p(n - \kappa_i) \cdot y_r(n) \cdot y_r(n)\right\} = \mathbf{H}_p^{\kappa_i} \cdot \Sigma^3 \cdot \mathbf{g}_r' \tag{3.103}$$

Concatenating L expectations of (3.103) for $\kappa_i = 0, 1, \ldots, L-1$, concatenating the resulting vector expectations for $p = 1, \ldots, P$ to one vector, and generalizing for any order of cumulants we take (3.104), (3.105) and (3.106) respectively

$$\mathbf{E}\left\{\mathbf{x}_p(n) \cdot y_r(n) \cdot y_r(n)\right\} = \mathbf{H}_p \cdot \Sigma^3 \cdot \mathbf{g}_r' \,, \tag{3.104}$$

$$\mathbf{E}\left\{\mathbf{x}(n) \cdot y_r(n) \cdot y_r(n)\right\} = \mathbf{H} \cdot \Sigma^3 \cdot \mathbf{g}_r' \,, \tag{3.105}$$

$$\mathrm{cum}\left\{\mathbf{x}(n) \cdot \underbrace{y_r(n), \ldots, y_r(n)}_{u+v}\right\} = \mathbf{H} \cdot \Sigma^{u+v+1} \cdot \mathbf{g}_r' \,. \tag{3.106}$$

Using the properties of the identity matrix ($\mathbf{I}_{K\cdot Q\times K\cdot Q} = \Sigma\cdot\Sigma^{-1}$) and noting that the right hand side can be expressed in terms of $\mathbf{f}'_r$ (see (3.83)), $\mathbf{H}\cdot\Sigma\cdot\mathbf{f}'_r$ is identified.

$$\begin{aligned}\mathrm{cum}\left\{\mathbf{x}(n)\cdot\underbrace{y_r(n),\ldots,y_r(n)}_{u+v}\right\} &= \mathbf{H}\cdot\Sigma\cdot\Sigma^{-1}\Sigma^{u+v+1}\cdot\mathbf{g}'_r \\ &= \mathbf{H}\cdot\Sigma\cdot\mathbf{f}'_r \end{aligned} \tag{3.107}$$

$\mathbf{H}\cdot\Sigma\cdot\mathbf{H}^T$ is estimated from the second order expectations as

$$\begin{aligned}\mathbf{E}\left\{\mathbf{x}\cdot\mathbf{x}^T\right\} &= \mathbf{E}\left\{\mathbf{H}\cdot\mathbf{s}\cdot\mathbf{s}^T\cdot\mathbf{H}^T\right\} \\ &= \mathbf{H}\cdot\Sigma\cdot\mathbf{H}^T\,, \end{aligned} \tag{3.108}$$

Inserting the (3.107), (3.108) into (3.94), (3.99), the derivation of the realizable algorithm is concluded.

The realizable algorithm

To illustrate conveniently the algorithm we define

$$\mathbf{R} = \mathrm{cum}\left\{\mathbf{x}(n),\mathbf{x}(n)^T\right\}\,, \tag{3.109}$$

$$\mathbf{R}_{ij} = \mathrm{cum}\left\{\mathbf{x}_i(n),\mathbf{x}_j(n)^T\right\}\,, \quad i,j=1,\ldots,P \tag{3.110}$$

Then $\mathbf{R}$ is a block matrix where its blocks are the correlation matrices $\mathbf{R}_{ij}$ between the i-th and j-th channel

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \ldots & \mathbf{R}_{1P} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \ldots & \mathbf{R}_{2P} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{R}_{P1} & \mathbf{R}_{P2} & \ldots & \mathbf{R}_{PP} \end{bmatrix} \tag{3.111}$$

Similarly we define

$$\mathbf{d}_r = \mathrm{cum}\left\{\mathbf{x}(n),\underbrace{y_r(n),\ldots,y_r(n)}_{u+v}\right\}\,, \tag{3.112}$$

$$\mathbf{d}_{rp} = \mathrm{cum}\left\{\mathbf{x}_p(n),\underbrace{y_r(n),\ldots,y_r(n)}_{u+v}\right\}\,. \quad p=1,\ldots,P \tag{3.113}$$

Then $\mathbf{d}_r$ is a block vector where its blocks are the cross-cumulant vectors $\mathbf{d}_{rp}$, with $p=1,\ldots,P$

$$\mathbf{d}_r = \left[\mathbf{d}_{r1}^T,\mathbf{d}_{r2}^T,\ldots,\mathbf{d}_{rP}^T\right]^T \tag{3.114}$$

Therefore (3.107), (3.108) can be written as

$$\mathbf{R} = \mathbf{H} \cdot \Sigma \cdot \mathbf{H}^T \tag{3.115}$$
$$\mathbf{d}_r = \mathbf{H} \cdot \Sigma \cdot \mathbf{f}'_r \tag{3.116}$$

and inserting $\mathbf{R}$, $\mathbf{d}_r$ in (3.94), 3.99 the two steps procedure of (3.90) can be equivalently realized by

$$\boxed{\begin{aligned} \mathbf{w}'_r &= \mathbf{R}^{\dagger} \cdot \mathbf{d}_r \,, \\ \mathbf{w}''_r &= \frac{\mathbf{w}'_r}{\sqrt{\mathbf{w}'^T_r \cdot \mathbf{R} \cdot \mathbf{w}'_r}} \,. \end{aligned}} \tag{3.117}$$

where $\mathbf{w}''_r$ at the end of an iteration will become the $\mathbf{w}_r$ of the next iteration [12] .

This algorithm is not applicable for speech separation because it is based on the assumption that the source signals are i.i.d. . Furthermore, the use of HOS statistics is unavoidable, because from (3.2), $u + v$ should be larger than two. In the next subsection we explain the necessary modification such that the proposed algorithm to become applicable for BSS of speech signals using entirely SOS.

3.2.3 Blind Source factor separation of speech signals

This subsection deals with the extension of the leading tap algorithm to tackle the problem of BSS of speech signals even using only SOS. First we show the inadequacies of the algorithm in its current state and then we intoduce the necessary modification.

Inadequacies of the leading tap algorithm

Speech is SOS and HOS correlated signal. Consequently the fundamental assumption of the presented algorithms, that each source is a temporally independent distributed signal, is violated and thus the derivations are not valid. For instance, consider the mathematical derivations of the previous subsection. Without the assumption that the source signals are temporally i.i.d. the identification of $\mathbf{f}'_r$ as given in (3.91) from the relation in (3.83) is not possible. The reason is discussed below.

As the source signals are spatially but not temporally i.i.d., the correlation matrices Σ_{qq} of s_q will not be any more diagonal but symmetric toeplitz matrices

[12] Then from (3.74) $y_r(n)$ can be computed which is necessary for the estimation of $\mathbf{d}_r$ in (3.112), and the next cycle of iteration can be performed.

as

$$\Sigma_{qq} = \begin{bmatrix} r_{qq}(0) & r_{qq}(1) & \cdots & r_{qq}(K-1) \\ r_{qq}(1) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ r_{qq}(K-1) & \cdots & \cdots & r_{qq}(0) \end{bmatrix}, \quad q = 1, \ldots, Q\,, \tag{3.118}$$

where $r_{qq}(\tau)$ is the $\tau - th$ component of the autocorrelation sequence of the source signal s_q. Therefore the correlation matrix Σ of the multichannel signal s in (3.86) will not be diagonal but only block diagonal. For the same reason Σ^{u+v+1} in (3.88) will also be a non-diagonal matrix. Therefore the convenient structure of $\mathbf{f}'_r$ in (3.91) can not be obtained and consequently an iteration scheme like in (3.92) can not be designed. This reveals the first inadequacy, i.i.e. the algorithm can not be applied to colored signals.

Moreover, if it is desirable to use SOS , where speech signals are well separable, from (3.112) $u + v$ must be equal with one. Therefore the requirement of the leading tap algorithm that the taps $g_{rq,\kappa}$ are exponentiated at each iteration cycle, (3.2), (3.90), is not satisfied and obviously the iterative scheme in (3.117) will not work. This is shown better if we use second order statistics to identify $\mathbf{H} \cdot \Sigma \cdot \mathbf{f}'_r$ (like in (3.102- 3.106) for third order statistics)

$$\begin{aligned} \mathbf{E}\left\{\mathbf{x} \cdot y_r\right\} &= \mathbf{H} \cdot \mathbf{E}\left\{\mathbf{s} \cdot \mathbf{s}^T\right\} \cdot \mathbf{g_r} \\ &= \mathbf{H} \cdot \Sigma \cdot \mathbf{g_r} \end{aligned} \tag{3.119}$$

In the left hand side of (3.119) there is only one filter $\mathbf{g}_r$. Therefore a filter $\mathbf{g}'_r$ with the taps of $\mathbf{g}_r$ exponentiated can never appear. Note that for both inadequacies the problem is located in the estimation of $\mathbf{H} \cdot \Sigma \cdot \mathbf{f}'_r$.

Modifying the algorithm for BSS of speech signals: the leading filter algorithm

The stress in BSS is to separate and not deconvolve. In (3.76) sources have been decoupled in the sense that each source s_q depends only on the corresponding filter $\mathbf{g}_{rq}$. Therefore if all but one filter converge to a zero tap filter, then only one source will appear in the output and the source factor separation problem will be solved, i.e.

$$\mathbf{g}_{rq} = \begin{cases} \mathbf{g}_{rq_1} & q = q_1 \\ \mathbf{0}_{K\times 1} & q \neq q_1 \end{cases}, \qquad q = 1, \ldots, Q \tag{3.120}$$

where $\mathbf{g}_{rq_1}$ is an arbitrary filter with at least one non-zero tap[13]. Based on this idea and the discussion above about the inadequacies of the leading tap algorithm we introduce the necessary modification in the following.

In correspondence with (3.82), (3.83) consider the $K \cdot Q \times 1$ block vector $\mathbf{b}'_r$ with blocks the vectors $\mathbf{b}'_{rq},\ q = 1, \ldots, Q$ as

$$\begin{aligned} \mathbf{b}'_r &= \left[\mathbf{b}'^T_{r1}, \mathbf{b}'^T_{r2}, \ldots, \mathbf{b}'^T_{rQ}\right]^T \\ &= \left[b'_{r1,0}, \ldots, b'_{r1,K-1}, \ldots, b'_{rQ,0}, \ldots, b'_{rQ,K-1}\right]^T . \end{aligned} \tag{3.121}$$

Let $\mathbf{b}'_r$ given from[14]

$$\mathbf{b}'_r = \Sigma^{-1} \cdot \Sigma^{\mathrm{av}} \cdot \mathbf{g}_r \, . \tag{3.122}$$

where Σ^{-1} is the inverse of the correlation matrix Σ of the multichannel signal s (Σ is given in (3.86), (3.119)), $\mathbf{g}_r$ is the block vector corresponding to the cascade system as in (3.78) and Σ^{av} is conceived as the linear sum of all the non-zero shifted version of Σ, as[15]

$$\begin{aligned} \Sigma^{\mathrm{av}} &= \sum_{\tau} \mathbf{E}\left\{\mathbf{s}_q(n-\tau) \cdot \mathbf{s}_q(n)^T\right\} \\ &= \mathbf{E}\left\{\sum_{\tau} \mathbf{s}_q(n-\tau) \cdot \mathbf{s}_q(n)^T\right\} . \end{aligned} \tag{3.123}$$

From the assumption that the signals are mutually spatially uncorrelated, Σ^{av} is reduced to a block diagonal matrix

$$\Sigma^{\mathrm{av}} = \begin{bmatrix} \Sigma^{\mathrm{av}}_{11} & & & \\ & \Sigma^{\mathrm{av}}_{22} & & \\ & & \ddots & \\ & & & \Sigma^{\mathrm{av}}_{QQ} \end{bmatrix} , \tag{3.124}$$

where each $K \times K$ block matrix $\Sigma^{\mathrm{av}}_{qq},\ , q = 1, \ldots, Q,$ is conceived as the linear sum of all non-zero shifted versions of the respective correlation matrix Σ_{qq}, in

[13] Note that the output signal may not be exactly s_{q_1} but a filtered version of it ($y_r = \sum_{\kappa=0}^{L-1} g_{q_1,\kappa} \cdot s_{q_1}(n-\kappa)$).

[14] In (3.122) we see that the taps of $\mathbf{g}_r$ are not exponentiated as in the leading tap algorithm. Therefore the stress now is in Σ^{av}, i.e., the average SOS of the sources. This is more clear from the convergence analysis of the modified algorithm, given in Appendix C.2 .

[15] The reason for introducing (3.122) and Σ^{av} as in (3.123) is in order to arrive in a convenient form for $\mathbf{b}'_r$ and therefore to design an iterative scheme for its subfilters.

(3.119), of the single channel signal s_q. Σ^{-1} and Σ^{av} are block diagonal matrices. Therefore $\mathbf{b}_r$ in (3.122) can be equivalently written in terms of its subfilters

$$\mathbf{b}'_{rq} = \Sigma^{-1}_{qq} \cdot \Sigma^{\text{av}}_{qq} \cdot \mathbf{g}_{rq}\,, \quad q = 1, \ldots, Q\,. \tag{3.125}$$

We first look one subfilter $\mathbf{b}'_{rq}$ and then we generalize for the overall $\mathbf{b}'_r$.

If the source signal s_q is of finite energy and the sum of all its correlation components equals to a non-zero scalar κ_{s_q}, we can write

$$\kappa_{s_q} = \sum_{\tau} r_{qq}(\tau) = \sum_{\tau} r_{qq}(\tau+1) = \cdots = \sum_{\tau} r_{qq}(\tau+K-1)\,,\ \kappa_{s_q} \in Z^* \tag{3.126}$$

Under these assumptions Σ^{av}_{qq} is reduced to a unity matrix scaled by κ_{s_q} as

$$\Sigma^{\text{av}}_{qq} = \begin{bmatrix} \kappa_{s_q} & \kappa_{s_q} & \cdots & \kappa_{s_q} \\ \kappa_{s_q} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \kappa_{s_q} & \cdots & \cdots & \kappa_{s_q} \end{bmatrix}. \tag{3.127}$$

Then from (3.127), (3.78) the product $\Sigma^{\text{av}}_{qq} \cdot \mathbf{g}_{rq}$ becomes

$$\begin{aligned} \Sigma^{\text{av}}_{qq} \cdot \mathbf{g}_{rq} &= \begin{bmatrix} \kappa_{s_q} & \kappa_{s_q} & \cdots & \kappa_{s_q} \\ \kappa_{s_q} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \kappa_{s_q} & \cdots & \cdots & \kappa_{s_q} \end{bmatrix} \cdot \begin{bmatrix} g_{rq,0} \\ g_{rq,1} \\ \vdots \\ g_{rq,K-1} \end{bmatrix} \\ &= \begin{bmatrix} \kappa_{s_q} \cdot \sum_{\kappa=0}^{L-1} g_{rq,\kappa} \\ \kappa_{s_q} \cdot \sum_{\kappa=0}^{L-1} g_{rq,\kappa} \\ \vdots \\ \kappa_{s_q} \cdot \sum_{\kappa=0}^{L-1} g_{rq,\kappa} \end{bmatrix}, \end{aligned} \tag{3.128}$$

and by letting $\check{g}_{rq}$ be the sum of all the taps of the subfilter $\mathbf{g}_{rq}$ as

$$\check{g}_{rq} = \sum_{\kappa=0}^{L-1} g_{rq,\kappa}\,, \tag{3.129}$$

(3.128) becomes

$$\Sigma^{\text{av}}_{qq} \cdot \mathbf{g}_{rq} = \kappa_{s_q} \cdot \check{g}_{rq} \cdot \mathbf{1}_{K\times 1}\,, \tag{3.130}$$

where $\mathbf{1}_{K\times 1}$ is the $K \times 1$ unity vector.

Inserting (3.130) into (3.125) we obtain one subfilter $\mathbf{b}'_{rq}$ and equivalently the block filter $\mathbf{b}'_r$ of (3.122)

$$\mathbf{b}'_{rq} = \Sigma_{qq}^{-1} \cdot \Sigma_{qq}^{\text{av}} \cdot \mathbf{g}_{rq}\,, \tag{3.131}$$

$$\mathbf{b}'_{r} = \Sigma^{-1} \cdot \Sigma^{\text{av}} \cdot \mathbf{g}_{r}\,. \tag{3.132}$$

With this convenient formulation the following iterative scheme can be designed for the subfilters $\mathbf{b}'_{rq}$[16]

$$\boxed{\begin{aligned} &for \quad q = 1, \ldots, Q\,, \quad \mathbf{b}'_{rq} = \Sigma_{qq}^{-1} \cdot \kappa_q \cdot \breve{g}_{rq} \cdot \mathbf{1}_{K\times 1}\,, \\ &for \quad q = 1, \ldots, Q\,, \quad \mathbf{b}''_{rq} = \frac{\mathbf{b}'_{rq}}{\sqrt{\mathbf{b}'^{T}_{rq} \cdot \Sigma_{qq} \cdot \mathbf{b}'_{rq}}}\,. \end{aligned}} \tag{3.133}$$

where $\mathbf{f}'_r$ at the end of the iteration cycle will become the $\mathbf{g}_r$ of the next iteration and $\kappa_{s_q}\,, \breve{g}_{rq}$ are non-zero scalars given from (3.126), (3.129) respectively.

Kawamoto and Inouye [24] showed that for stationary signals the iteration procedure converges to the desired solution, i.e. all but one of the filters $\mathbf{b}_{rq}$ will become equal to a zero tap filter[17]. The convergence analysis is given in the appendix C.2.

Applying exactly the same strategy as in Subsect. 3.2.2, by the following MSE estimation

$$\min_{\hat{\mathbf{g}}_r} \left\| \mathbf{H}^T \cdot \hat{\mathbf{w}}_r - \mathbf{b}'_r \right\|^2\,, \tag{3.134}$$

the unaccessible algorithm of (3.133) is expressed in terms of the FIR filter $\mathbf{w}_r$

$$\mathbf{w}'_r = (\mathbf{H} \cdot \Sigma \cdot \mathbf{H}^T)^{\dagger} \cdot \mathbf{H} \cdot \Sigma \cdot \mathbf{b}'_r\,.\,, \tag{3.135}$$

$$\mathbf{w}''_r = \frac{\mathbf{w}'_r}{\sqrt{\mathbf{w}'^{T}_r \cdot \mathbf{H} \cdot \Sigma \cdot \mathbf{H}^T \cdot \mathbf{w}'_r}}\,. \tag{3.136}$$

The unknown system $\mathbf{H} \cdot \Sigma \cdot \mathbf{H}^T$ is acquired as in (3.108). $\mathbf{H} \cdot \Sigma \cdot \mathbf{b}'_r$ is estimated from the sum of all non-zero cross-correlation vectors between the multichannel microphone signal x and the output y_r

$$\sum_{\tau} \mathbf{E}\left\{\mathbf{x}(n-\tau) \cdot y_r\right\} \quad \text{where,} \quad \tau \in \mathbf{Z}\,. \tag{3.137}$$

[16]Note that now we do not look fi lter taps but a kind of self-averaged subfi lters (averaged due to the second step which is a normalization).

[17]Note that after each iteration cycle the "leading fi lter" becomes of linear phase (LP).

Expanding (3.137) and using the properties of expectations

$$\begin{aligned}\sum_{\tau}\mathbf{E}\left\{\mathbf{x}(n-\tau)\cdot y_r\right\} &= \sum_{\tau}\mathbf{E}\left\{\mathbf{H}\cdot\mathbf{s}_q(n-\tau)\cdot\mathbf{s}_q(n)^T\cdot\mathbf{g}_r\right\}\\ &= \mathbf{H}\cdot\sum_{\tau}\mathbf{E}\left\{\mathbf{s}(n-\tau)\cdot\mathbf{s}(n)^T\right\}\cdot\mathbf{g}_r\\ &= \mathbf{H}\cdot\Sigma\cdot\Sigma^{-1}\cdot\sum_{\tau}\mathbf{E}\left\{\mathbf{s}(n-\tau)\cdot\mathbf{s}(n)^T\right\}\cdot\mathbf{g}_r\\ &= \mathbf{H}\cdot\Sigma\cdot\mathbf{b}_r \qquad (3.138)\end{aligned}$$

the unknown system is acquired. Note that the modification introduced in (3.137) coincides with the one introduced in (3.122).

Therefore the realizable algorithm for colored signals, e.g., like speech, will be

$$\boxed{\begin{aligned}\mathbf{w}_r' &= \mathbf{R}^{\dagger}\cdot\mathbf{d}_r\,,\\ \mathbf{w}_r'' &= \frac{\mathbf{w}_r'}{\sqrt{\mathbf{w}_r'^T\cdot\mathbf{R}\cdot\mathbf{w}_r'}}\,.\end{aligned}} \qquad (3.139)$$

where $\mathbf{R}$ is given from (3.109), (3.110), and $\mathbf{d}_r$ is modified such that the algorithm to be able to handle SOS

$$\mathbf{d}_r = \sum_{\tau}\mathbf{E}\left\{\mathbf{x}(n-\tau)\cdot y_r\right\} \qquad (3.140)$$

$$\mathbf{d}_{rp} = \sum_{\tau}\mathbf{E}\left\{\mathbf{x}_p(n-\tau)\cdot y_r\right\} \qquad (3.141)$$

The algorithm can be generalized to separate signals using HOS as was done for i.i.d. signals in the previous subsections. For more information see [24].

3.3 Conclusions

In this chapter it was investigated the extension of a blind deconvolution algorithm for i.i.d signals ((3.47), (3.59)) to a BSFS one for correlated signals ((3.133), (3.139)), e.g., like speech. For illustration purposes and without loss of generality we assumed real-valued casual FIR signals and systems. The analysis showed that when the signals are i.i.d. the convergence of the algorithm depends on the statistical scaling and exponentiation of the taps of the overall filter $\mathbf{g}_{rq}$ ((3.47), (3.90), (C.5)). In contrary for correlated signals the algorithm relies almost entirely on the statistical scaling of the taps ((3.133), (C.18)). The latter is certified from experimental results as shown in chapter 5.

Another interesting point is that the update step of (3.139) is very similar with the so-called normal equation. Indeed $\mathbf{R}$ is the correlation matrix of the multichannel microphone signal $\mathbf{x}(n)$. The difference is $\mathbf{d}_r$ which is not just the cross-correlation vector between the multichannel signal $\mathbf{x}(n)$ and the output signal $y_r(n)$ but the summation of all non-zero shifted versions of this vector. As will be discussed in the subsequent chapters this similarity may be exploited to design efficient frequency domain algorithms. The only problem is the summation index τ not found in the conventional normal equation.

It is interesting to see also that the second step of (3.139) is just the normalization of the output signal $y_r(n)$ to energy one. This is apparent if we multiply with $\mathbf{x}(n)^T$ from the left.

$$\mathbf{x}^T(n) \cdot \mathbf{w}_r'' = \frac{\mathbf{x}^T(n) \cdot \mathbf{w}_r'}{\sqrt{\mathbf{w}_r'^T \cdot \mathbf{x}(n) \cdot \mathbf{x}^T(n) \cdot \mathbf{w}_r'}}, \tag{3.142}$$

$$y_r(n) \leftarrow \frac{y_r(n)}{|y_r(n)|}. \tag{3.143}$$

Finally it is important to note that the proposed algorithms are designed for stationary signals. Moreover the BSFS method is suitable for separation of only one out of many sources. Therefore the algorithm of (3.139) is still not suitable for BSS of speech signals and some enhancements are necessary. We discuss these problems and propose solutions in the next chapter.

Chapter 4

Extension to Deflationary BSS of Speech Signals

The focus of this thesis is on the BSS of speech signals. Within this frame the method of the previous chapter appears two inadequacies:

- It can extract only one source out of a mixture.
- It assumes that the signals are stationary, which does not hold for speech.

Therefore an enhancement of the method is necessary. The first insufficiency can be solved if the BSFS method is incorporated into a deflationary approach where the signals are extracted one by one [24]. The second problem is solvable under the assumption that the speech signals are short-time stationary, which is usually the case. Then an adaptive scheme of the deflationary algorithm will be able to track the time-variances.

Following these considerations, in the first section a deflationary algorithm is introduced. In addition to the BSFS, which have been already discussed, it includes an adaptive filtering part to estimate the *contribution* of the extracted sources in the mixtures and then remove it. This part is examined in detail, and then the whole algorithm is summarized in bullet form. The resulting algorithm is adequate to extract speech signals.

But still the method is not optimal. The drawbacks of deflationary techniques is that the quality of the separation degrades and the computational complexity increases with the order of the extracted sources. For the solution of these problems it is useful to consider the deflationary BSFS (DBSFS) structure as a Generalized Sidelobe Canceler (GSC) [12]. Then the well studied algorithms of GSC, especially those in frequency domain [15, 14, 16], may be applicable and improve the algorithm.

The similarity of the proposed method with the GSC is studied in the second section. It is shown that the ABM is exactly the deflating filters of the deflationary approach. By comparing the different parts, i.e. the BSFS block and the fixed beamformer, it is concluded that it may be possible to construct similar algorithms with those of GSC. The only obstacle now is to transform the BSFS block in the F.D. This idea seems promising because the fundamental relation of the BSFS method is very similar with the well known normal equation. The only difference is the summation index in the cross-corelation vector (3.141). It was attempted to overcome this problem but it was not concluded and thus it was let for further considerations. [1] Nevertheless, a number of time domain algorithms are proposed in section 4.3. Based on them several experiments were contacted in order to analyse the behavior of the algorithm and the conclusions are given in the next chapter.

4.1 Deflationary BSS for Speech Signals

BSFS methods can retrieve only one signal and thus alone they are insufficient for BSS. A deflationary technique can bypass this limitation. It consists of three iterative steps, one to extract a source, a second to estimate its contribution to the mixtures, and a third to remove, i.e., deflate its contribution from the mixtures. Here, BSFS constitutes the first step. The next step is realized by least squares estimation and the last step is a simple substraction. To retrieve Q sources, $Q-1$ iterations are necessary. Therefore, the particular deflationary BSS method is:

- Iterate $Q-1$ times the following three steps
  - Utilize the proposed BSFS method to extract a filtered version of a source.
  - Solve P least squares problems to estimate the contribution of the extracted source to the mixtures.
  - Perform the deflation by removing the contribution of the source from the mixtures.

The BSFS method has been considered in the previous chapter. The third step is a simple substraction and does not need more attention. In contrary the second step should be examined further.

P filters are necessary to estimate the contribution of the extracted source into the mixtures. Let $\mathbf{b}_{pr}$ denote the filter to deflate an extracted source $y_r(n)$ from the microphone signal $x_p(n)$ as in Figure 4.1.

[1] Another problem that may be signifi cant is that the short-time stationarity of the algorithm may not be enough. This is discussed in the appendix.

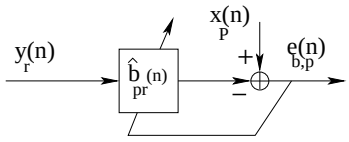

Figure 4.1: Block diagram for the estimation and removal of the contribution of the extracted source y_r to the p-th microphone signal.

Choosing causal FIR filters of length L_b, where L_b is sufficient large, the error at their output is

$$\begin{aligned} e_{b,p}(n) &= x_p(n) - \sum_{\kappa=0}^{L_b-1} y_r(n-\kappa) \cdot b_{pr,\kappa} \\ &= x_p(n) - c_{pr}(n) \quad p = 1, \ldots, P\,. \end{aligned} \tag{4.1}$$

where c_{pr} is the contribution of the r^{th} output signal to the p^{th} microphone signal. As can be seen in (4.1) the component c_{pr} is then subtracted from the p-th mixture leading to a deflation of the extracted source.

Then the P least squares problems to estimate the desired filters are

$$\min_{\hat{b}_{pr}} |e_{b,p}(n)|^2, \quad p = 1, \ldots, P\,, \tag{4.2}$$

or formulated as a recursive least squares cost function

$$J_{b,p}(n) = (1-\lambda) \cdot \sum_{i=0}^{n} \lambda^{n-i} \cdot |e_{b,p}(n)|^2 \quad p = 1, \ldots, P \tag{4.3}$$

where $J_{b,p}(n)$ is the cost function for minimization w.r.t. to $\hat{\mathbf{b}}_{pr}(n)$ and λ is a constant, the so-called forgeting factor. The time-index n is added in order to show that the filter is estimated adaptively (in the next section the adaptive criterion is formulated in block fashion, to estimate the so called adaptive blocking matrix).

Finally, the third step is a simple substraction of the estimated contributions from the respective mixtures. Summarizing, the deflationary BSS algorithm for the r-th output channel ($r = 1, \ldots, Q-1$), as:

1. Extract one source with the BSFS method:

   - Carry out the BSFS iterative scheme till convergence

$$\begin{aligned} \mathbf{w}'_r &= \mathbf{R} \cdot \mathbf{d}_r\,, \\ \mathbf{w}''_r &= \frac{\mathbf{w}'_r}{\sqrt{\mathbf{w}'^T_r \cdot \mathbf{R} \cdot \mathbf{w}'_r}}\,, \end{aligned}$$

where $\mathbf{R}$ and $\mathbf{d}_r$ are given from (3.111), (3.140) respectively. (Due to the non-stationarity of speech signals the adaptive version of BSFS method is necessary which we describe later in subsections 4.3.4 and 4.3.5).

- Perform the filtering of the microphone signals with the estimated filters obtained from the BSFS method

$$y_r(n) = \sum_{p=1}^{P} \sum_{\kappa=0}^{L-1} x_r(n-\kappa) \cdot w_{rp,\kappa}\,, \tag{4.4}$$

where $w_{rp,\kappa}$ are the taps of the filter w_r'' after convergence.

2. Estimate the contribution of the extracted source y_r, into the mixtures x_p, $p = 1, \ldots, P$ with the LS criterion:

- Solve P least square problems to estimate the filters $\hat{\mathbf{b}}_{pr}$

$$\min_{\hat{b}_{pr}} |x_p(n) - \sum_{\kappa=0}^{L_b-1} y_r(n-\kappa) \cdot \hat{b}_{pr,\kappa}|^2 \quad p = 1, \ldots, P \tag{4.5}$$

For speech signals an adaptive criterion as in (4.3) is necessary.

- Compute the contribution of the r-th source into the q-th mixture by using the estimated filter $\hat{\mathbf{b}}_{pr}$

$$c_{pr}(n) = \sum_{\kappa=0}^{L_b-1} y_r(n-\kappa) \cdot b_{pr,\kappa} \quad p = 1, \ldots, P. \tag{4.6}$$

3. Deflate the contribution c_{pr} from the p-th mixture

$$x_p(n) \leftarrow x_p(n) - c_{pr}(n) \quad p = 1, \ldots, P. \tag{4.7}$$

The main advantage of the method is that global convergence is assured as shown in [19, p. 885]. However, there are important disadvantages in comparison with approaches that extract the signals in parallel. As the order of the recovered sources increases, the computational complexity of the method increases and the separation performance degrades. Therefore the method is preferable in cases where only one source should be extracted, for instance in hearing aids or speech recognition applications. Nevertheless, the quality of the extracted signals may be improved and the efficiency of the method may increased if its similarity with the Generalized Sidelobe Canceller (GSC) is exploited, as discussed in the next section.

4.2 Similarity of the Deflationary BSS with the Generalized Sidelobe Canceler (GSC)

GSC techniques have been evolved considerably the last years and very efficient algorithms have been proposed. Therefore it is advantageous to examine the deflationary BSS method from this point of view. Indeed the deflationary BSS algorithm described in the previous section may be viewed similarly to a GSC [12] with the adaptive blocking matrix [17], where the fixed beamformer is replaced by the BSFS block and the deflating adaptive filters constitute the Adaptive Blocking Matrix (ABM)[2] (see Fig. 4.2). Then by cascading this structure $Q-1$ times all the sources may be retrieved.

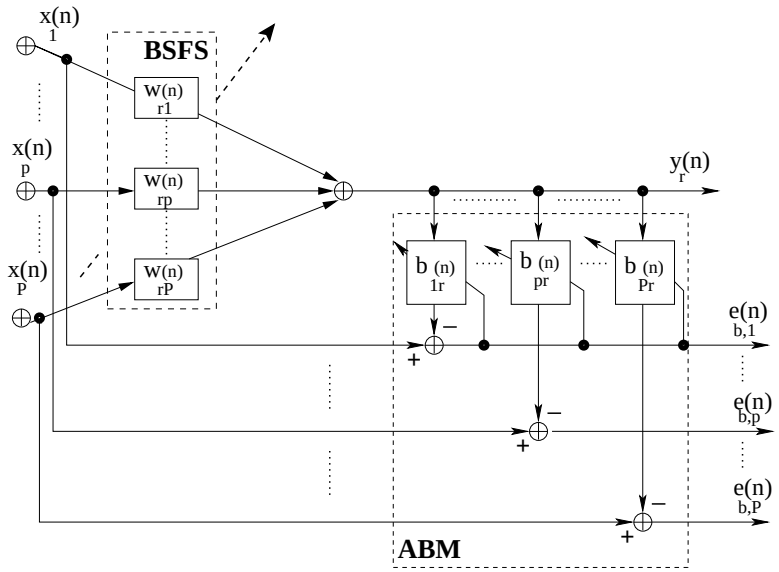

Figure 4.2: The deflationary BSS in GSC fashion.

To show that the deflating filters constitute the ABM is relatively easy. First we concatenate the taps of the filter $\mathbf{b}_{pr}$ and L_b samples of $y_r(n)$ to form the following vectors

$$\mathbf{b}_{pr}(n) = (b_{pr,0}(n), b_{pr,1}(n), \ldots, b_{pr,L_b-1}(n))^T \quad p = 1, \ldots, P\,, \tag{4.8}$$

$$\mathbf{y}_r(n) = (y_r(n), y_r(n-1), \ldots, y_r(n-L_b+1))^T \quad p = 1, \ldots, P\,. \tag{4.9}$$

[2]The Adaptive Interference Canceler (AIC) is not consider here but its addition may be a topic of future research

Then the error at the output of the adaptive filters may rewritten as

$$e_p(n) = x_p(n) - \mathbf{y}_r(n)^T \cdot \mathbf{b}_{pr}(n) \quad p = 1, \ldots, P\,, \tag{4.10}$$

where the linear convolution was transformed to the inner product of the respective vectors

Further we concatenate the P microphone signals and the P error signals (similarly with (3.72)) to create the respective multi-channel signals

$$\mathbf{y}(n) = (x_1(n), x_2(n), \ldots, x_P(n))^T\,, \tag{4.11}$$

$$\mathbf{e}(n) = (e_1(n), e_2(n), \ldots, (e_P(n))^T\,. \tag{4.12}$$

and the multi-channel error is

$$\mathbf{e}(n) = \mathbf{y}(n) - \mathbf{B}(n)^T \cdot \mathbf{y}_r(n)\,, \tag{4.13}$$

where $\mathbf{B}(n)$ is the so called ABM

$$\mathbf{B}(n) = [\mathbf{b}_{1r}(n), \mathbf{b}_{2r}(n), \ldots, \mathbf{b}_{Pr}(n)]\,, \tag{4.14}$$

and the adaptive criterion of (4.3) may be written in multi-channel fashion as

$$J_{\hat{B}}(n) = (1-\lambda) \cdot \sum_{i=0}^{n} \lambda^{n-i} \cdot \mathbf{e}(n)^T \cdot \mathbf{e}(n) \quad p = 1, \ldots, P \tag{4.15}$$

Therefore we arrived to the ABM and to a multichannel criterion for its adaptation as in the GSC (Fig. 4.3).

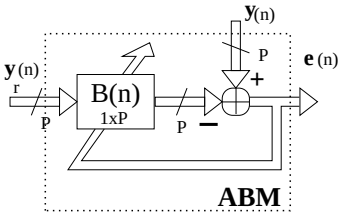

Figure 4.3: The adaptive deflating filters formulated as the ABM.

On the other hand the fixed beamformer (FBF) and the BSFS block are clearly different (Fig. 4.4). By principle the FBF processes the spatial plane waves of the speakers received by the microphone array such that those coming from a particular angle are passed through and the others are attenuated. In a similar way the proposed BSFS processes the statistically scaled filters between the speech

sources and the output of the overall system, and dumps all of them except from one, in order to extract a filtered version of only one source. In the contrary the proposed BSFS is an adaptive blind algorithm while its counterpart has fixed filter weights.

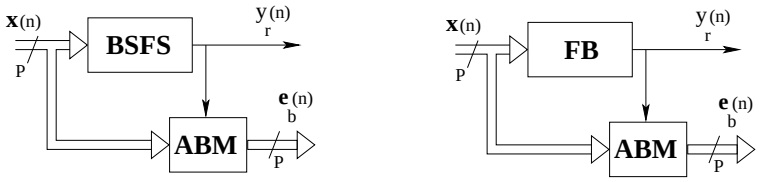

Figure 4.4: Comparison of DBSS (left plot) and GSC (right plot).

The above discussion suggests that GSC algorithms with some modifications may be applicable to the proposed method which may lead to improved performance. The modifications are imposed by the fact that the front end block (BSFS) is not anymore fixed but adaptive and therefore there exist two adaptive blocks. For optimum tracking it is desirable to adapt them simultaneously, however this is not possible. Furthermore, the ABM should be adapted only when the desired speaker at the output of the BSFS is active, otherwise it will eliminate components which may belong to other sources which are not extracted yet. The energy of desired and interfering speech may vary in different frequencies. Therefore, using the same strategy in the fullband may not be optimum. This point reveals the advantage of considering the Deflationary BSS structure as a GSC. Recently very efficient frequency domain adaptive algorithms were proposed in the field of GSC's [15]. Application of such an algorithm allows for adaption in narrow subbands instead of the fullband which may improve the overall performance. In combination with a possible implementation of the BSFS in the frequency domain it will allow to take advantage of well studied algorithms for the GSC. However, we will not go into details here. The interested reader may refer to [15], [16]. Instead several kind of algorithms will be proposed for the deflationary BSFS method in the time domain.

4.3 Proposed algorithms

Following Shalvi and Weinstein [34, pp. 513-514, 518] and Kawamoto and Inouye [24] a class of algorithms is proposed. The fundamental relation of the method is

$$\mathbf{w}_r = \mathbf{R} \cdot \mathbf{d}_r \tag{4.16}$$

where $\mathbf{R}$, as in (3.111), is the block corelation matrix of the microphone signals and $\mathbf{d}_r$, as in (3.112), is a block vector which depends on the statistics of the signals to be separated, i.e. for which order of statistics the signal is more separable. For convenience we rewrite them here

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \dots & \mathbf{R}_{1P} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \dots & \mathbf{R}_{2P} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{R}_{P1} & \mathbf{R}_{P2} & \dots & \mathbf{R}_{PP} \end{bmatrix}, \tag{4.17}$$

$$\mathbf{d}_r = \left(\mathbf{d}_{r1}^T, \mathbf{d}_{r2}^T, \dots, \mathbf{d}_{rP}^T\right)^T, \tag{4.18}$$

where $\mathbf{d}_r$ is

- if SOS are used then

$$\mathbf{d}_{rp} = \sum_{\tau} \mathbf{E}\left\{\mathbf{x}_p(n-\tau) \cdot y_r(n)\right\}, \tag{4.19}$$

- if HOS are used then for the simplest case of third order statistics

$$\mathbf{d}_{rp} = \sum_{\tau} \mathbf{E}\left\{\mathbf{x}_p(n) \cdot y_r^2(n)\right\}. \tag{4.20}$$

4.3.1 Empirical Averages for approximating Expectations

Exact cumulants are unknown. Cumulants of any order are equivalent to some combination of expectations. This is shown in appendix B for first to fourth order cumulants of zero mean random processes. Expectations can be approximated by empirical averages for ergodic processes functioning as the interface between mathematics and reality. From now on assume zero mean processes and use expectations. Consequently, ensemble averages (expectations) can be approximated from empirical averages.

- Batch averaging
  - Subblock correlation matrix

$$\hat{\mathbf{R}}_{ij} = \frac{1}{M \cdot L} \cdot \sum_{n=0}^{ML-1} \mathbf{x}_i(n) \cdot \mathbf{x}_j(n)^T, \quad i,j = 1, \dots, P \tag{4.21}$$

– Subblock cross-correlation vector

$$\text{SOS:} \quad \hat{\mathbf{d}}_{rp} = \frac{1}{M \cdot L} \cdot \sum_{n=0}^{ML-1} \sum_{\tau=0}^{T-1} \mathbf{x}_p(n-\tau) \cdot y_r(n), \tag{4.22}$$

$p = 1, \ldots, P$

$$\text{HOS:} \quad \hat{\mathbf{d}}_{rp} = \frac{1}{M \cdot L} \cdot \sum_{n=0}^{ML-1} \sum_{\tau=0}^{T-1} \mathbf{x}_p(n-\tau) \cdot y_r^2(n), \tag{4.23}$$

$p = 1, \ldots, P$

- Recursive averaging

  – Subblock correlation matrix

$$\hat{\mathbf{R}}_{ij}(n) = (1-\lambda) \cdot \sum_{i=0}^{n-i} \lambda^{n-i} \cdot \mathbf{x}_i(n) \cdot \mathbf{x}_j(n)^T \tag{4.24}$$

$$= \lambda \cdot \hat{\mathbf{R}}_{ij}(n-1) + (1-\lambda) \cdot \mathbf{x}_i(n) \cdot \mathbf{x}_j(n)^T,$$

$i, j = 1, \ldots, P$ (4.25)

  – Subblock cross-correlation vector

SOS: (4.26)

$$\hat{\mathbf{d}}_{rp}(n) = (1-\lambda) \cdot \sum_{i=0}^{n} \lambda^{n-i} \cdot \sum_{\tau=0}^{T-1} \mathbf{x}(i-\tau) \cdot \mathbf{y}_r(i)$$

$$= \lambda \cdot \hat{\mathbf{d}}_{rp}(n-1) + (1-\lambda) \cdot \mathbf{v}(n) \cdot \mathbf{y}_r(n) \tag{4.27}$$

where

$$\sum_{\tau=0}^{T-1} \mathbf{x}(i-\tau) = \mathbf{v}(i) \tag{4.28}$$

HOS: (4.29)

$$\hat{\mathbf{d}}_{rp}(n) = (1-\lambda) \cdot \sum_{i=0}^{n} \lambda^{n-i} \cdot \mathbf{x}(i) \cdot \mathbf{y}_r(i)^2$$

$$= \lambda \cdot \hat{\mathbf{d}}_{rp}(n-1) + (1-\lambda) \cdot \mathbf{x}(n) \cdot \mathbf{y}_r(n)^2 \tag{4.30}$$

where $\mathbf{T}$ is the number of all non-zero cross-correlation vectors $\mathbf{x}(n)$ and the forgeting factor λ corresponds to exponential windowing that includes the effect of past data, and effectively adaptive algorithms may be obtained.

4.3.2 Batch-iterative algorithm

- Initialization
  - Use (4.21) to estimate all the $\mathbf{R}_{ij}(n)$ from the whole set of available samples $x_p(n)$. Then compute the inverse of $\hat{\mathbf{R}}$: i.e. $\hat{\mathbf{R}}^{-1}$.
  - Initialize the demixing filter $\mathbf{w}_r^{[0]}$.
- For $l = 1, \ldots,$ till convergence.
  1. Acquire output samples: Filtering
  $$y_r^{[l]}(n) = \sum_{\kappa=0}^{L-1} x_p(n-\kappa) \cdot w_{rp,\kappa}^{[l-1]}, \quad n = 0, \ldots, ML-1. \tag{4.31}$$
  2. Estimate $\mathbf{d}_r^{[l]}$ from the whole set of available samples:
     - if SOS are used from (4.23)
     $$\hat{\mathbf{d}}_r^{[l]} = \frac{1}{M \cdot L} \sum_{n=0}^{M \cdot L-1} \mathbf{v}(n) \cdot y_r^{[l]}(n) \tag{4.32}$$
     $$\text{where} \quad , \quad \mathbf{v}(n) = \sum_{\tau=0}^{T-1} \mathbf{x}(n-\tau) \tag{4.33}$$
     - If HOS are used from (4.24)
     $$\hat{\mathbf{d}}_r^{[l]} = \frac{1}{M \cdot L} \sum_{n=0}^{M \cdot L-1} \mathbf{x}(n) \cdot z_r^{[l]}(n) \tag{4.34}$$
     $$\text{where} \quad , \quad z_r^{[l]}(n) = (y_r^{[l]}(n))^2 \tag{4.35}$$
  3. Estimate filter:
  $$\mathbf{w}_r'^{[l]} = \hat{\mathbf{R}}^{-1} \cdot \mathbf{d}_r^{[l]} \tag{4.36}$$
  4. Normalize filter:
  $$\mathbf{w}_r''^{[l]} = \frac{\mathbf{w}_r'^{[l]}}{\sqrt{\mathbf{w}_r'^{[l]\,T} \cdot \hat{\mathbf{R}} \cdot \mathbf{w}_r'^{[l]}}} \tag{4.37}$$
  5. Update filter:
  $$\mathbf{w}_r^{[l]} \leftarrow \mathbf{w}_r''^{[l]} \tag{4.38}$$
- Use the final filter to aquire a filtered version of one source.

4.3.3 Recursive-iterative algorithm

Initialization

- Initialize for the recursion the correlation matrix and cross-correlation vector: $\hat{\mathbf{R}}, \hat{\mathbf{d}}_r(0)$.
- Initialize filter for first recursion and first iteration: $\mathbf{w}_r^{[0]}(0)$.

For the m-th block $m = 1, \ldots, M \cdot L$

- Compute from 4.24 a recursive estimate of $\hat{\mathbf{R}}$. Then compute its inverse $\hat{\mathbf{R}}^{-1}$
- For iteration $l = 1, \ldots, l_{\max}$ (until convergence)

  1. Filtering: Acquire one output sample

$$y_r^{[l]}(m) = \sum_{\kappa=0}^{L-1} x(m-\kappa) \cdot w_r^{[l-1]}(m-1) = \mathbf{x}^T \cdot \mathbf{w}_r^{[l-1]}(m-1) \tag{4.39}$$

  2. Estimate recursively $\mathbf{d}_r^{[l]}(m)$ from (4.27), (4.30)

$$\text{SOS} \quad \hat{\mathbf{d}}_r^{[l]}(m) = \lambda \cdot \hat{\mathbf{d}}_r^{[l]}(m-1) + (1-\lambda) \cdot \mathbf{v}(m) \cdot y_r^{[l]}(m) \tag{4.40}$$

$$\text{HOS} \quad \hat{\mathbf{d}}_r^{[l]}(m) = \lambda \cdot \hat{\mathbf{d}}_r^{[l]}(m-1) + (1-\lambda) \cdot \mathbf{x}(m) \cdot z_r^{[l]}(m) \tag{4.41}$$

     where

$$\mathbf{v}(m) = \sum_{\tau=0}^{T-1} \mathbf{x}(m-\tau) \tag{4.42}$$

$$z_r^{[l]}(m) = (y_r^{[l]}(m))^2 \tag{4.43}$$

  3. Estimate filter

$$\mathbf{w}'^{[l]} = \hat{\Phi}(m) \cdot \hat{\mathbf{d}}_r^{[l]}(m) \tag{4.44}$$

  4. Normalize filter

$$\mathbf{w}''^{[l]} = \frac{\mathbf{w}'^{[l]}}{\mathbf{w}'^{[l]\,T} \cdot \hat{\mathbf{R}}(m) \cdot \mathbf{w}'^{[l]}} \mathbf{w}'^{[l]} \tag{4.45}$$

  5. $\mathbf{w}^{[l]} \leftarrow \mathbf{w}''^{[l]}$

- $\mathbf{w}^{[0]} \leftarrow \mathbf{w}^{[l_{\max}]}$

4.3.4 Adaptive algorithm

To convert the recursive-iterative algorithm to a recursive-sequential (as in [10]):

- Replace the iteration index $[l]$ with the block index (m).
- To acquire an output sample $y_r(m)$ use the most current state of the filter, i.e. $w_r(m-1)$

This is a common practice in stochastic approximation. The resulting algorithm is

Initialize for the recursions $\hat{\mathbf{R}}(0), \hat{\mathbf{d}}_r(0), \mathbf{w}_r(0)$

For block index $m = 1, \ldots, M \cdot L$

1. Compute from (4.24) a recursive estimate of $\hat{\mathbf{R}}(m)$

$$\hat{\mathbf{R}}(m) = \lambda \cdot \hat{\mathbf{R}}(m-1) + (1-\lambda) \cdot \mathbf{x} \cdot \mathbf{x}^T \tag{4.46}$$

   and then compute its inverse $\hat{\mathbf{R}}^{-1}(m)$

2. Acquire one output (filtered) sample:

$$y_r(m) = \mathbf{x}^T(m) \cdot \mathbf{w}_r(m-1) \tag{4.47}$$

3. Estimate recursively $\mathbf{d}_r(m)$ from (4.27) (4.30)

$$\text{SOS} \quad \hat{\mathbf{d}}_r(m) = \lambda \cdot \hat{\mathbf{d}}_r(m-1) + (1-\lambda) \cdot \mathbf{v}(m) \cdot y_r^{[l]}(m) \tag{4.48}$$

$$\text{HOS} \quad \hat{\mathbf{d}}_r(m) = \lambda \cdot \hat{\mathbf{d}}_r(m-1) + (1-\lambda) \cdot \mathbf{x}(m) \cdot z_r^{[l]}(m) \tag{4.49}$$

   where

$$\mathbf{v}(m) = \sum_{\tau=0}^{T-1} \mathbf{x}(m-\tau) \tag{4.50}$$

$$z_r(m) = (y_r(m))^2 \tag{4.51}$$

4. Estimate the filter

$$\mathbf{w}' = \hat{\mathbf{R}}^{-1}(m) \cdot \hat{\mathbf{d}}_r(m) \tag{4.52}$$

5. Normalize filter

$$\mathbf{w}'' = \frac{\mathbf{w}'}{\mathbf{w}'^T \cdot \hat{\mathbf{R}}(m) \cdot \mathbf{w}'} \tag{4.53}$$

4.3.5 Adaptive algorithm using the matrix inversion lemma

The averaging rectangular window is converted to an exponential one in order to introduce adaptability. Using HOS for the estimation of $\hat{\mathbf{d}}_{rq}$ and ignoring the summation index (like in [22]), the recursive estimates in the m-th block will be

$$\hat{\mathbf{R}}_{pp}(m) = (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}\mathbf{x}_p(i)\cdot\mathbf{x}_p^T(i) \tag{4.54}$$

$$\hat{\mathbf{d}}_{rq}(m) = (1-\lambda)\sum_{i=0}^{m}\lambda^{m-i}\mathbf{x}_p(i)\cdot y_r^2(i) \tag{4.55}$$

where

$$y_r(m) = \mathbf{x}^T(m)\cdot\mathbf{w}(m-1) \tag{4.56}$$

Let $z_r(m) = y_r^2(m)$ and rewrite in recursive form

$$\hat{\mathbf{R}}_{pp}(m) = \lambda\cdot\hat{\mathbf{R}}_{pp}(m-1) + (1-\lambda)\cdot\mathbf{x}_p(m)\cdot\mathbf{x}_p^T(m) \tag{4.57}$$

$$\hat{\mathbf{d}}_{rq}(m) = \lambda\cdot\hat{\mathbf{d}}_{rq}(m-1) + (1-\lambda)\cdot\mathbf{x}_p(m)\cdot z_r(m) \tag{4.58}$$

It is desirable to acquire a recursive estimate of $\hat{\mathbf{R}}_{pp}^{-1}(m)$. Therefore the matrix inversion lemma is used

$$A = B^{-1} + C\cdot D^{-1}\cdot C^T \tag{4.59}$$

$$A^{-1} = B - B\cdot C(D + C^T\cdot B\cdot C)^{-1}C^T\cdot B \tag{4.60}$$

By comparing (4.57), (4.59) we can write

$$A = \hat{\mathbf{R}}_{pp}(m) \tag{4.61}$$

$$B = \lambda^{-1}\cdot\hat{\mathbf{R}}_{pp}^{-1}(m-1) = \lambda^{-1}\cdot\hat{\Phi}_{pp}(m-1) \tag{4.62}$$

$$C = \mathbf{x}_p(m) \tag{4.63}$$

$$D = (1-\lambda)^{-1}\cdot\mathbf{I} \tag{4.64}$$

Then by (4.60) and simplifying

$$\hat{\Phi}_{pp}(m) = \lambda^{-1}\cdot\hat{\Phi}_{pp}(m-1) - \frac{\lambda^{-2}\cdot(1-\lambda)\cdot\hat{\Phi}_{pp}(m-1)\cdot\mathbf{x}_p(m)\cdot\mathbf{x}_p(m)^T\cdot\hat{\Phi}_{pp}(m-1)}{1+\lambda^{-1}\cdot(1-\lambda)\cdot\mathbf{x}_p(m)^T\hat{\Phi}_{pp}(m-1)\cdot\mathbf{x}_p(m)} \tag{4.65}$$

Therefore the so called kalman gain $\mathbf{k}_p(m)$ of source $x_p(m)$ is

$$\mathbf{k}_p(m) = \frac{\lambda^{-1}\cdot\hat{\Phi}_{pp}(m-1)\cdot\mathbf{x}_p(m)}{1+\lambda^{-1}\cdot(1-\lambda)\cdot\mathbf{x}_p(m)^T\cdot\hat{\Phi}_{pp}(m-1)\cdot\mathbf{x}_p(m)} \tag{4.66}$$

And evaluating

$$\hat{\Phi}_{pp}(m) = \lambda^{-1} \cdot \hat{\Phi}_{pp}(m-1) - \lambda^{-1} \cdot (1-\lambda) \cdot \mathbf{k}_p(m) \cdot \mathbf{x}_p(m)^T \cdot \hat{\Phi}_{pp}(m-1) \quad (4.67)$$

$$\mathbf{k}_p(m) = \hat{\Phi}_{pp}(m) \cdot \mathbf{x}_p(m) \quad (4.68)$$

Now by inserting (4.58) into (4.16) the filter estimate of the first step becomes

$$\begin{aligned} \hat{\mathbf{w}}'_{rq}(m) &= \hat{\Phi}_{pp}(m) \cdot \hat{\mathbf{d}}_{rq}(m) \\ &= \hat{\Phi}_{pp}(m) \cdot [\lambda \cdot \hat{\mathbf{d}}_{rq}(m-1) + (1-\lambda) \cdot \mathbf{x}_p(m) \cdot z_r(m)] \\ &= \lambda \cdot \hat{\Phi}_{pp}(m) \cdot \hat{\mathbf{d}}_{rq}(m-1) + (1-\lambda) \cdot \mathbf{k}_p(m) \cdot z_r(m) \end{aligned} \quad (4.69)$$

and then inserting (4.67) leads to the recursive estimate of $\hat{\mathbf{w}}'_{rq}$

$$\hat{\mathbf{w}}'_{rq}(m) = \hat{\mathbf{w}}'_{rq}(m-1) - (1-\lambda) \cdot \mathbf{k}_p(m) \cdot (\mathbf{x}_p(m)^T \cdot \hat{\mathbf{w}}'_{rq}(m-1) - z_r(m)). \quad (4.70)$$

4.4 Conclusions

In this chapter the BSFS method were extended to a deflationary one, where the signals are extracted one by one, thus succeeding BSS. Each source (actually a filtered version of it) is extracted with the BSFS method and its contribution to the mixture is estimate using the MSE criterion. Then with a simple substraction it is deflated from the mixtures.

Moreover, we showed the link between the proposed deflationary approach and the traditional GSC structure. In particular, the deflating filters coincide with the ABM, and the BSFS block replaces the FBF of the GSC. The similarities of the BSFS block with the ABM were discussed.

Finally, we proposed several time domain algorithms for the proposed deflationary method such as batch, recursive and combination of them. For real time implementation the adaptive algorithms are necessary in order to truck the non-stationaryties of speech signals. On the other hand the combination of them (e.g. batch-recursive) in an off-line and on-line fashion may result to improved performance (e.g. see [10]).

Possible enhancements of the method may be

- Extension of the algorithms in the frequency domain, which may allow for adaption in subbands and thus improved separation rates. The obstacle on this direction is the summation index τ in (4.19).
- Extension of the BSFS method to one where all signals are extracted simultaneously as discussed in appendix D, thus avoiding the problems of the deflationary approaches.

Chapter 5

Experimental Results

Several experiments were conducted in order to examine the behavior of the proposed algorithms. As evaluation parameter were used the signal to interference ratio (SIR). The SIR indicates the separation performance and it is defined as

$$\mathrm{SIR} = \frac{\mathbf{P}_{target}}{\cdot\mathbf{P}_{interferer}}\,, \tag{5.1}$$

where $\mathbf{P}_{target}$ is the direct path energy and $\mathbf{P}_{interferer}$ is the cross-talk energy in the output of interest, and expressed in decibel (dB). Moreover, the signals are normalized in the microphone front-ends, for comparison purposes. The normalization of a signal is done by dividing it with its norm.

The time-frequency algorithm were first verified in the free field case, and then exposed in reverberant enclosures. For the deflationary approach we evaluate two algorithms, one batch and one recursive.

In the next section we present the experimental setup and then we discuss the experimental results of the two different approaches. In the last section of the chapter we give an overview of the experimental work pointing out the most important conclusions.

5.1 Experimental Setup

The experimental setup is depicted in Fig. 5.1. There are ten (10) loudspeaker positions and eleven (11) microphones. The room impulse responses were recorded for each possible constellation for low and high reverberation times T_{60}, $150\,ms$ and $350\,ms$ respectively.

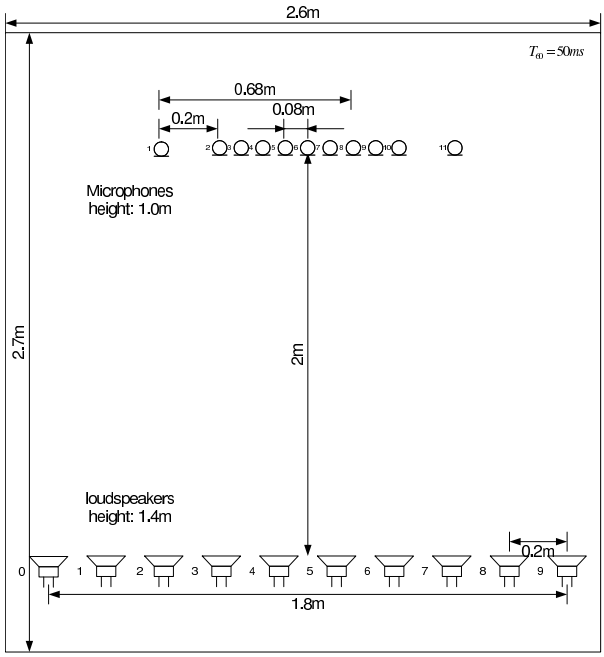

Figure 5.1: Experimental Setup.

For the experiments we used three dry speaker voices, one mail and two female. The time length of the voices were truncated to ten seconds (10 sec.). For the real-world scenarios, we convolved the signals with the room impulse responses.

5.2 Time-Frequency Masking

In this section we present our experiments for the time-frequency masking algorithm (Sect. 2.3) for both anechoic and echoic propagation conditions. The objective of this algorithm is to extract *pairs* of sources utilizing as labeling criterion the DOA of the time-frequency components. We investigated the three sources by two microphones scenario (3×2).

The results for the free-field case were very satisfying. The interferer was almost cancelled and the objective speakers were extracted with small distortion. For the convolutive mixtures this was not the case. Significant energy of the interferer source remained in the pairs and large distortion (musical noise) appeared on the objective sources. Moreover, the enumeration of the sources in the mixture was not possible any more. In this case the application of a quadratic BSS had not effect on the separation performance.

5.2.1 Anechoic mixtures

For the anechoic case non-integer delays were necessary. Therefore we used lagrange filters to create the fractional delay filters [25]. Then the sources were convolved with the appropriate filter to create the necessary constellation. For comparison we simulated the realistic experimental setup (Fig. 5.1). Several experimental results for different speaker positions were contacted.

For instance, in Fig. 5.2 we simulated the cases that the speech signals arrived from angles of $45, 100, 140$ degrees. We observe that the doa is computed correctly, but still there is a small overlap between the sources. For this case a separation mask as in (2.59) with $\Delta = 10^o$ gave very good separation performance and small distortion.

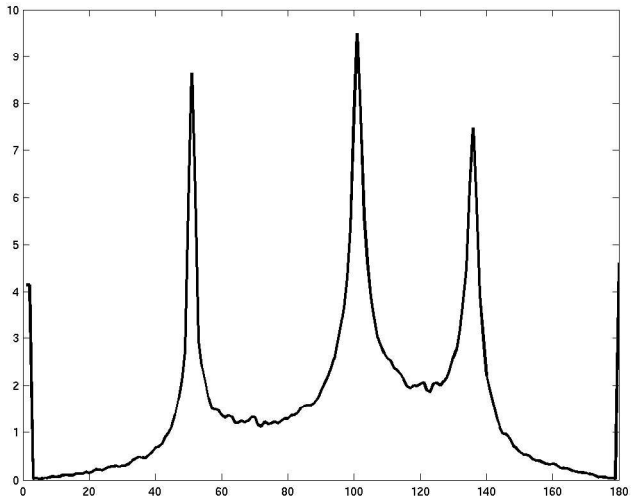

Figure 5.2: Computation of the doa of the sources for the free field case.

For the anechoic model the clusters of the signals are very narrow. This is shown in Fig. 5.3 where we evaluate the algorithm with only one speaker. Therefore even in the case that the speakers are spatially close, the separation of the method for the free-field case is still satisfying.

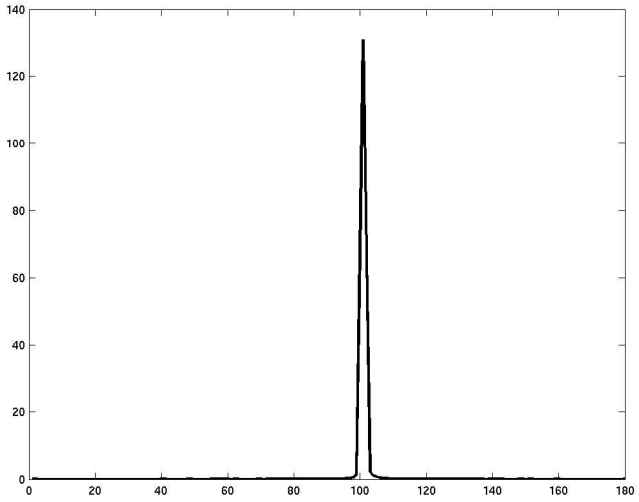

Figure 5.3: Computation of the doa of one source for the free field case.

From the comparison of Fig. 5.3 and Fig. 5.2 we can see that a small overlap exists even in the free-field case. In the next section we see that the overlap is not negligible any more, for convolutive mixtures.

Moreover, in Fig. 5.2 we map all the undetected components $0^o \geq \mathrm{DOA} \geq 180^o$ to the angles of 0^o and 180^o. These components may come from overlapping sources or faults of the spectrum estimation method.

From the experimental results we see that the theoretical considerations of Chapt. 2 were certified. Moreover, the applicability of the method for separation of speech signals in anechoic mixtures were proved.

5.2.2 Echoic mixtures

In this section we conducted reverberant tests ($T_{60} = 150\,ms$). In Fig. 5.4 the speakers arrived from $60,\ 110,$ and 125 degrees. We see that now the overlap between adjacent sources is significantly large. Indeed, the middle speaker can not be separated with any binary time-frequency mask.

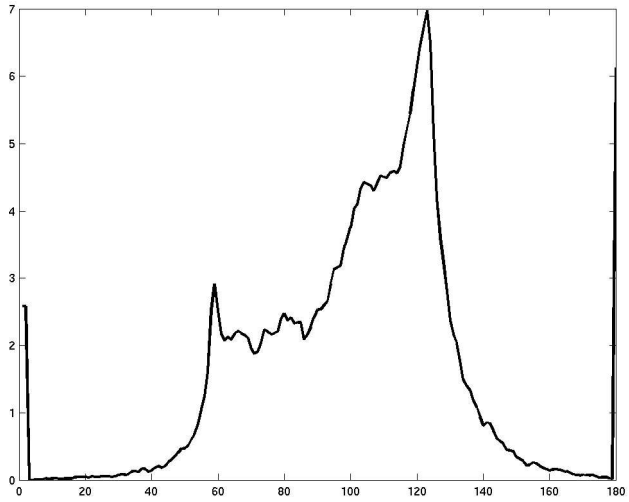

Figure 5.4: Computation of the doa of the sources for $T_{60} = 150\,ms$.

In another experiment we allowed only one source to enter the algorithm Fig. 5.5. We see that even in this case the energy leakage to neighboring DOA's is significant. Moreover, we observe that some components are reflected. From similar experiments we verified that this reflection increases along with the reverberation time (T_{60}).

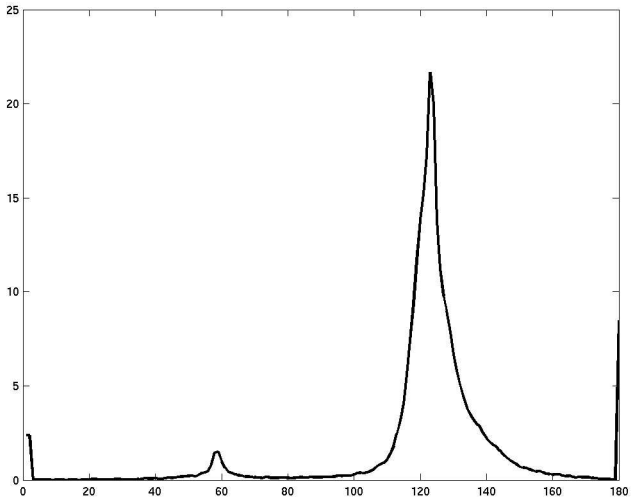

Figure 5.5: Computation of the doa of one source for $T_{60} = 150\,ms$.

5.2.3 Remarks

Here, we point out some general remarks for the sparseness algorithm for both the echoic and anechoic case verified from the experimental results.

- Spatial aliasing depends on the microphone spacing and the sampling frequency. For instance, for $16\,KHz$ sampling frequency the microphone spacing should be less than $4\,cm$. To prevent aliasing we decimated the sources to sampling frequency of $8\,KHz$, which allowed the use of microphone spacing of $4\,cm$. This improved effectively the separation performance of the sparseness method.

- The frequency resolution plays an important role on the performance of the algorithm. The optimal resolution is between ten and twenty Hz ($10 - 20\,\mathrm{Hz}$). It is determined by the sampling frequency and the DFT window length. For $8\,KHz$ sampling frequency a window of 512 frame size is necessary.

- Furthermore the type of the window is important. The rectangular window gave the worst results, while the Hamming was the optimum. The reason is that the latter is more tapered at the sides and therefore spectral smearing is reduced.

- One of the most important factors to improve the efficiency of the algorithm is the introduction of overlap on the frames of the time dependent Fourier transform, and thus to improve the resolution in the frequency domain [32, p. 714], [33, p. 433].

- We observed that the middle speaker is more difficult to separated.

The experimental conclusions coincide with the theoretical considerations of Chapt. 2. Moreover, the application of a quadratic BSS method to the source pairs can not compensate for the distortion of the objective signals and can not cancel the interferer (third source). Therefore we conclude that the enhancement of the method is necessary for BSS of signals in echoic mixtures.

5.3 Deflationary BSS

For the Deflationary approach we examined two different algorithms a batch and a recursive, as described in the previous chapter.

We directly performed experiments for echoic environments. The reverberation time (T_{60}) of the simulation environments were set to $150\,ms$. For the batch

algorithm we investigated the 2×3, 2×4 and 3×4 scenario. In all cases the batch algorithm exhibited good performance but appeared stability problems due to the non-stationarity of speech signals.

Several tests were performed for the recursive algorithm, which gave good separation results. The stability of the algorithm appeared to be very sensitive to the values of τ, block length and λ.

In the following we discuss the experimental results in detail, for both the batch and the recursive algorithm.

5.3.1 Batch algorithm

Several experimental results showed that the batch algorithm is unstable, as expected due to the non-stationarity of speech signals. Nevertheless, even for this algorithm we succeeded good separation performance by choosing carefully the parameters of the method. For instance in Fig. 5.6 the sir is close to twenty (20) dB even after one hundred iterations. From the ripples in the same plot we can clearly observe that the algorithm is unstable.

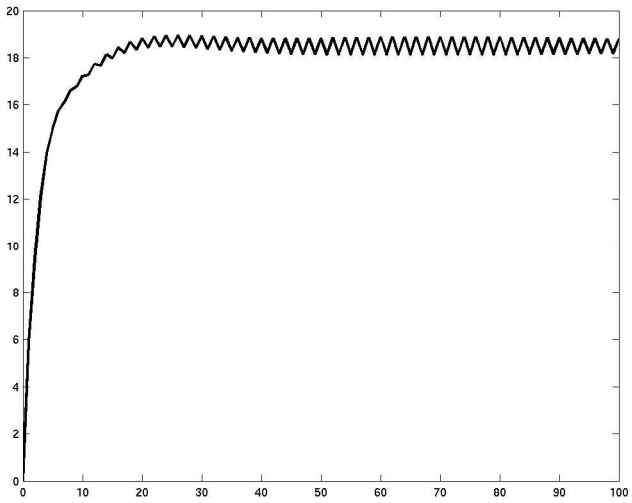

Figure 5.6: Performance of the batch deflationary algorithm ($T_{60} = 150\,ms$).

Moreover, from extensive simulations we concluded the following

- The algorithm converges with a very fast rate, usually after three iterations.
- For many iterations, an audible distortion appears on the separated signal. We concluded that it is better to stop the algorithm after few iterations even if the sir gain is smaller.

- It gives separation gain even in the case of more than two sources.
- Its performance increases along with the order of the microphones.
- The value of the summation index τ is important for the stability of the algorithm. Theoretically the larger the τ the more stable the algorithm is, but experimental results showed that $\tau \approx 256$ is enough.

In order to improve the stability of the method, a recursive algorithm were designed. We discuss it in the next subsection.

5.3.2 Recursive algorithm

The batch algorithm is only applicable for stationary signals and for off-line computations. In order to improve the stability of the method and to allow real time implementation we designed a recursive algorithm. This algorithm were tested for the 2×3 scenario.

Experimental results showed that this algorithm is still very sensitive to changes on the parameters of the method, namely τ, block length, and λ. In particular, it optimizes for one source for a specific time gate, and then changes rapidly to optimize for the other. The sir/iteration diagram of such an experiment is shown in Fig. 5.7.

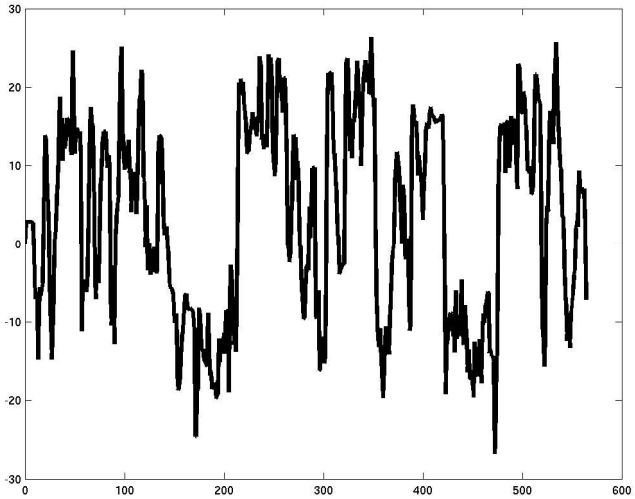

Figure 5.7: Performance of the recursive algorithm ($\tau = 16$, block length $= 256$, $\lambda = 0.1$).

For some specific values we succeeded more stable behavior, as shown in Fig. 5.8. In this particular example the values of τ, block length and λ are significantly larger.

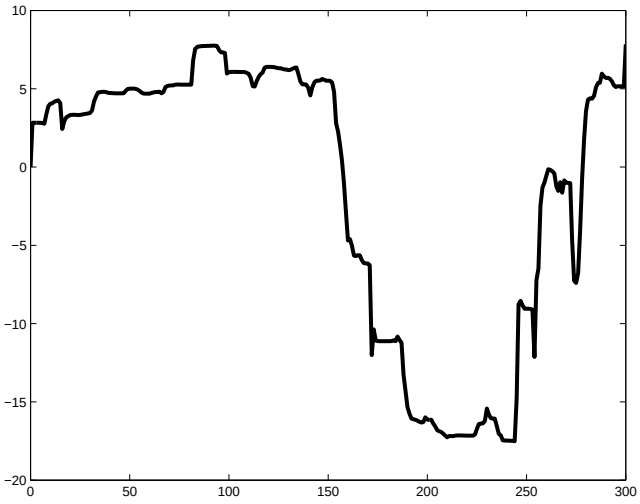

Figure 5.8: Improvement of the stability of the recursive algorithm (τ $=$, block length $= 256$, $\lambda = 256$).

The explanation of the behavior of the algorithm is possible if we consider the convergence analysis of the BSFS method, as given in appendix C.2. There it is shown that the algorithm assumes the following properties for the source signals

- The signals are short-time stationary.
- The algorithm extract the statistical dominant source. Therefore the one source should persistently dominate statistically the other.

As soon as the above properties hold the algorithm will continue to optimize for the same source. Therefore choosing large values for τ, block length and λ, the optimization for one source is preserved for longer time.

5.4 Conclusions

In this chapter we investigated the applicability of the proposed algorithms for BSS of speech signals. The sparseness algorithm exhibited good performance

for instantaneous mixtures but appeared inadequacies when exposed to reverberant enclosures. Therefore an enhancement of the algorithm is necessary as was discussed in Chapt. 2.

For the deflation approach we designed two algorithms, one batch and one recursive. These algorithms were evaluated directly to reverberant conditions. The separation results were very good. On the other hand both algorithms appeared stability problems. Therefore further investigations on this issue are necessary.

Moreover, several other experiments were attempted, such as performance measures of the masks, implemantation of the adaptive blocking matrix (ABM) in the frequency domain for the deflationary method, and other. Due to time limitations these attempts were not concluded or not presented here. We let them for future work.

Concluding we can say that the experimental results proved the applicability of both methods (time-frequency masking and deflationary) for speech signals separation, but also revealed the fact that there is much more work to be done for the application of these approaches to realistic scenarios.

Chapter 6

Conclusions–Future Work

This thesis aimed to examine two approaches for underdetermined BSS of speech sources, in particular one method based on sparseness and one deflationary. The main points of the thesis consisted of

- The investigation of a method combining time-frequency masking and ICA.
- The investigation of a deflationary approach.
- A unified treatment of the deflationary method in order to show how methods initially designed for the extraction of i.i.d. signals can be modified and applied to audio signals.
- Investigation of the similarities between the proposed deflationary approach and the conventional GSC structure.
- Experimental work to show the applicability of the methods for speech signal separation.

Several conclusions arised, which we present in the rest of the chapter along with some suggestions for future considerations.

Time-frequency masking combined with conventional BSS

This algorithm combines time-frequency masking and conventional BSS. Therefore we can say that

- In contrary to conventional sparseness methods, the components of the signals incorrectly detected as coming from neighboring sources are not lost, and consequently the separated sources are less distorted.
- On the other hand, the performance of the algorithm still degrades when exposed to reverberant enclosures.

Experimental work for the three sources by two microphones case proved the theoretical considerations. Moreover, we verified that:

- The spacing between subsequent frequency bins of the signals in the DFT domain affects the performance of the algorithm. The optimum resolution is between 10 and 20 Hz.
- The performance of the algorithm depends on the type of the DFT window. The rectangular window is the worst choice. From the other common windows, Hamming exhibit the best performance.[1]
- The spacing between the microphones should be small enough to prevent spatial aliasing.

Concluding we should point that the method is a combination of beamforming and source localization. Therefore further improvement may come by employing techniques from these fields, e.g., doa estimation [6], exploitation of temporal correlation of speech and other.

A deflationary approach based on a BSFS

The deflationary algorithm consisted of a BSFS method to extract one source, and a MSE criterion to estimate the contribution of the sources and deflate them from the mixtures. From the experimental results we concluded the following

- The largest sir gain observed was around 20 dB, which proves the applicability of the method for speech processing.
- The algorithm converges at very fast rate, usually after three to four iterations.
- Due to the non-stationarity of the speech the batch algorithm was unstable. The recursive algorithm improved partially the stability of the system. More investigations on this matter are necessary.

A unified treatment

The proposed deflationary algorithm is the extension of a single channel blind deconvolution algorithm designed for i.i.d signals. This extension came from the work of several researchers. We replicated their work in order to show how approaches designed for i.i.d. signals can be modified and applied to audio signals. The main modifications were

[1] Probably other special windows may give better results, e.g. cosine tapper.

- The SISO problem formulation were translated to a MISO one, which decouples the sources to have an one-to-one relation with the filters.
- The leading tap algorithm which designed for i.i.d. signals, extrapolated to a leading filter algorithm for the separation of speech sources.
- The cross-cumulant vector of the update equation were replaced by the summation of all the non-zero cross-correlation vectors between the output and the input of the FIR filter. This modification allows the use of SOS to separate colored signals, e.g., speech.

Moreover, from this comprehensive review several other insightful conclusions acquired.

Links with the GSC

We attempted to find links between the proposed deflation algorithm and the GSC. In particular,

- We showed that the ABM is equivalent to the deflating filters of the deflationary structure.
- We pointed the similarities and differences between the FBF and the BSFS structures.

The conclusions arised may be useful to improve the deflationary algorithm.

Future work

The update equation of the FIR filter in the BSFS method, (3.139), is very similar with the so called normal equation. Actually the only difference is the vector $\mathbf{d}_r$ as shown bellow.

$$\mathbf{w}_r = \mathbf{R} \cdot \mathbf{d}_r \tag{6.1}$$

$$\mathbf{d}_r = \sum_{\tau} \mathbf{E}\left\{\mathbf{x}(n-\tau) \cdot y_r(n)\right\} . \tag{6.2}$$

In wiener filtering the summation index in (6.2) is discarded and $y_r(n)$ is replaced by the desired signal. In linear prediction, τ is fixed, larger than zero, and $y_r(n)$ is replaced by the microphone signals $x(n)$. Experimental results showed that the cross-correlation vector at shift $\tau = 0$ is important for the estimation of $\mathbf{d}_r$ in (6.2) and cannot be discarded. Therefore this method is closer to to wiener filtering where the cross-correlation vector is replaced from its average, in order to avoid the ambiguity that the desired signal is not known. This similarity may

be exploited to design efficient algorithms in the frequency domain. The obstacle on this direction is the summation index τ in (6.2).

On the other hand kawamoto et. al [22, p. 1054] used the cumulant version of the proposed algorithm (3.141) discarding the summation index, for the separation of speech signals and acquired satisfying results[2]. In this case a derivation of frequency domain algorithm from a time domain one is straightforward.

Moreover, Shalvi and Weinstein [34, p. 515] proposed an algorithm in the frequency domain for colored signals. This algorithm may be extendable to the MISO case as its time domain counterpart.

We should note also the similarity of the proposed algorithm with subspace methods. Equation (3.72) is very similar with (16.16) in [13, p. 692]. Although the formulation is for the SIMO case, the filtering matrix have similar structure. Using the filtering-matrix rank theorem and regarding the interfering sources as colored noise, a link with subspace methods may be possible.

The deflationary approach is applicable only to the overdetermined case. Therefore it will be interesting to extend this method for underdetermined BSS. In [39] is proposed a time frequency subspace method for the overdetermined case. It is shown that this method may be applicable to the underdetermined problem. The investigation of this method may be of particular interest.

In chapter 3 we presented the extension of the leading tap algorithm designed for i.i.d. signals to a leading filter algorithm for the separation of colored signals. Motivated from this treatment, it may possible with the appropriate problem formulation to extend the leading filter algorithm to a leading matrix one, where the matrices will be the sylvester matrices as in (3.64). This idea has been applied to other algorithms, e.g. , [10].

Concluding we should note the superiority of the original blind deconvolution algorithm over similar algorithms in the field, e.g., predictive deconvolution, godorad algorithm and other [34]. Therefore further investigation of the proposed deflation algorithm seems promising.

[2]Similar algorithms were also proposed from Martone [29, 28]

Appendix A

Abbreviations

BSS	Blind Source Separation
BD	Blind Deconvolution
DSP	Digital Signal Processing
LTI	Linear Time-Invariant
FIR	Finite Impulse Response
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
SISO	Single Input Single Output
SIMO	Single Input Multiple Output
DFT	Discrete Fourier transform
ST-DFT	Short-Time (or Time-Dependent) Discrete Fourier transform
DCT	Discrete Cosine Transform
W-DO	W-disjoint orthogonality
pdf	Probability Density Function

Operators

$\exp\{\cdot\}$	exponential function
diag$\{\cdot\}$	diagonal matrix with the elements $\cdot$
$\min$	absolute minimum
$\mathbf{A}^T, \mathbf{a}^T$	transposed matrix, vector
$\mathbf{A}^*, \mathbf{a}^*$	complex conjugation of matrix, vector
$\lvert \cdot \rvert$	absolute value of a scalar or determinant of a matrix
$\lVert \cdot \rVert$	norm of a vector or matrix
$\delta(\cdot)$	in continuous time domain is the Dirac delta function, in discrete time domain is the u

Principal Symbols

t	continuous time variable
m	block index
n	discrete time variable
κ	discrete time convolution variable or filter tap index
q	integer denoting the source number
p	integer denoting the microphone number
r	integer denoting the number of output of the DSP system
$h_{pq,\kappa}$	$\kappa^t h$ tap of the mixing filter from the source q to the microphone p
$\mathbf{h}_{pq}$	vector of length L containing the taps of h_{pq} which is assumed FIR
H	multichannel filter from the sources to the microphones
$\mathbf{H}_{pq}$	filtering or Sylvester matrix of the filter h_{pq}
$\mathbf{H}$	multichannel filtering matrix from the sources to the microphones
$\mathbf{H}_p$	the p^{th} block row of the multichannel filtering matrix
w_{rq}	the FIR filter from the microphone p to the output r of the DSP system
W	multichannel filter from the sources to the microphones
g_{rq}	the cascade filter from the source q to the microphone r
g_r	the MISO filter from all the sources to the output r of the DSP system
u, v	exponents of the taps used for the leading tap blind deconvolution method
$\{x(n)\}$	sequence formed from the elements of $x(n)$

Appendix B

Properties of Cumulants and Expectations

Cumulants are used to estimate the unknown systems in Chapt 3. The following properties of them are necessary for the derivations [18, p. 40]:

P1) Linearity: $\text{cum}\left\{\sum_n b_n \cdot x(n)\right\} = \sum_n b_n \cdot \text{cum}\left\{x(n)\right\}$,

P2) $\text{cum}_2\left\{x_i(n), x_j(n)\right\} = \begin{cases} 0 & , \quad i \neq j; \\ \text{cum}\left\{x(n)\right\} & , \quad i = j; \end{cases}$ Statistical Independence:

where x is a random process with realizations $x(n)$, and b_n are scalars.

Moreover, the following properties of expectations are necessary [18, p. 20]:

P1) Linearity: $\mathbf{E}\{\sum_i a_i \cdot \mathbf{x}_i\} = \sum_i a_i \cdot \mathbf{E}\{\mathbf{x}_i\}$,

where $\mathbf{x}_i$ are different random vectors.

P2) Linear transformations $\mathbf{E}\left\{\mathbf{A} \cdot \mathbf{x}\right\} = \mathbf{A} \cdot \mathbf{E}\left\{\mathbf{x}\right\}$,

where $\mathbf{A}$ is a linear transformation of suitable dimension.

We should note also that for zero mean random variables, first, second and third order cumulants equal to the respective expectations:

$$\begin{aligned} \text{cum}\left\{x(n)\right\} &= 0 \\ \text{cum}\left\{x_i(n), x_j(n)\right\} &= \text{E}\left\{x_i(n), x_j(n)\right\} \\ \text{cum}\left\{x_i(n), x_j(n), x_k(n)\right\} &= \text{E}\left\{x_i(n), x_j(n), x_k(n)\right\} \end{aligned}$$

Appendix C

Convergence analysis in the *g-domain*

Here we give the convergence analysis of the blind deconvolution algorithm of Sect. 3.1 and the BSFS algorithm of Sect. 3.2.3

C.1 Convergence analysis of the single leading tap algorithm

Here we analyse the convergence analysis of the algorithm of (3.2)

It is helpful to write the iteration scheme in pseudo-code form as follows.

For $l = 1, \ldots,$ until convergence

$$g_{rq,\kappa}^{[l]} \leftarrow (g_{rq,\kappa}^{[l-1]})^{u+v}, \quad \kappa = -\infty, \ldots, \infty \tag{C.1}$$

$$g_{rq,\kappa}^{[l]} \leftarrow \frac{g_{rq,\kappa}^{[l]}}{\left\| \mathbf{g}_{rq}^{[l]} \right\|}, \qquad \kappa = -\infty, \ldots, \infty \tag{C.2}$$

where it is assumed that the LTI filters are real-valued.

Suppose that the tap with index $\tilde{\kappa}$, is the largest of the $l^t h$ iteration. Consider its ratio with the other filter taps.

$$\left| \frac{g_{rq,\kappa}^{[l]}}{g_{rq,\tilde{\kappa}}^{[l]}} \right| = \left| \frac{g_{rq,\kappa}^{[l-1]}}{g_{rq,\tilde{\kappa}}^{[l-1]}} \right|^{u+v} \tag{C.3}$$

This recursion is valid for every iteration, therefore

$$\left| \frac{g_{rq,\kappa}^{[l]}}{g_{rq,\tilde{\kappa}}^{[l]}} \right| = \left| \frac{g_{rq,\kappa}^{[l-1]}}{g_{rq,\tilde{\kappa}}^{[l-1]}} \right|^{u+v} = \left| \frac{g_{rq,\kappa}^{[l-2]}}{g_{rq,\tilde{\kappa}}^{[l-2]}} \right|^{(u+v)^2} = \cdots = \left| \frac{g_{rq,\kappa}^{[0]}}{g_{rq,\tilde{\kappa}}^{[0]}} \right|^{(u+v)^l}. \tag{C.4}$$

As l is chosen arbitrarily this holds for any iteration[1]

$$\left|\frac{g_{rq,\kappa}^{[l]}}{g_{rq,\tilde{\kappa}}^{[l]}}\right| = \left|\frac{g_{rq,\kappa}^{[0]}}{g_{rq,\tilde{\kappa}}^{[0]}}\right|^{(u+v)^l} . \tag{C.5}$$

Therefore if initially the tap with index $\tilde{\kappa}$ is larger than all the other, by this iteration scheme, the algorithm will converge to the desired solution with an exponential rate.

C.2 Convergence analysis of the leading filter algorithm

In this section we present the convergence analysis of the algorithm in (3.133). Now instead of taps the interest is on the filters. For the same reason like in the previous section the iteration scheme is re-written in pseudo-code form

For $l = 1, \ldots,$ until convergence

$$\mathbf{g}_{rq}^{[l]} \leftarrow \Sigma_{qq}^{-1} \cdot \kappa_{s_q} \cdot (\breve{g}_{rq}^{[l-1]})^p \cdot \mathbf{1}_{K\times 1}, \quad q = 1, \cdots, Q\,, \tag{C.6}$$

$$\mathbf{g}_{rq}^{[l]} \leftarrow \frac{1}{\left\|\mathbf{g}_r^{[l]}\right\|} \cdot \mathbf{g}_{rq}^{[l]}, \qquad q = 1, \cdots, Q \tag{C.7}$$

where

$$\breve{g}_{rq}^{[l]} = \sum_{\kappa} g_{rq,\kappa}^{[l]} \tag{C.8}$$

expanding it for the l^{th} iteration it is

$$\breve{g}_{rq}^{[l]} = \rho_q \cdot \kappa_{s_q} \cdot (\breve{g}_{rq}^{[l-1]})^p \tag{C.9}$$

where ρ_{s_q} is the sum of all the elements of Σ_{qq}^{-1}

$$\rho_{s_q} = \sum_{\kappa} \phi_{qq}(\kappa) \tag{C.10}$$

In the beggining we ignore the normalization step. Then the first step is a simple recursion. The same for the unnormalized sum of the taps. Expanding the sum to arrive to the first iteration ($l = 0$)

$$\begin{aligned} \breve{g}_{rq}^{[l]} &= \rho_{s_q} \cdot \kappa_{s_q} \cdot (\rho_{s_q} \cdot \kappa_{s_q} \cdot (\breve{g}_{rq}^{[l-2]})^p)^p \\ &= \cdots = \rho_{s_q} \cdot \kappa_{s_q} \cdot (\rho_{s_q} \cdot \kappa_{s_q} \cdot (\ldots (\rho_{s_q} \cdot \kappa_{s_q} \cdot (\breve{g}_{rq}^{[l-l]})^p)^p \ldots)^p)^p \\ &= (\rho_{s_q} \cdot \kappa_{s_q})^{1+p+p^2+\cdots+p^{l-1}} \cdot (\breve{g}_{rq}^{[0]})^{p^l} \end{aligned} \tag{C.11}$$

[1] This formula can also be proved by inference.

We distinguish two cases $p \neq 1$ and $p = 1$

For $p \neq 1$, i.e., for higher order cumulants

$$\sum_{n=0}^{l-1} p^n = \frac{p^l - 1}{p - 1} \tag{C.12}$$

therefore

$$\breve{g}_{rq}^{[l]} = (\rho_{s_q} \cdot \kappa_{s_q})^{\frac{-1}{p-1}} \cdot (\rho_{s_q} \cdot \kappa_{s_q})^{\frac{p^l}{p-1}} \cdot (\breve{g}_{rq}^{[0]})^{p^l} \tag{C.13}$$

$$= (\rho_{s_q} \cdot \kappa_{s_q})^{\frac{-1}{p-1}} \cdot ((\rho_{s_q} \cdot \kappa_{s_q})^{\frac{1}{p-1}} \cdot \breve{g}_{rq}^{[0]})^{p^l} \tag{C.14}$$

this holds for every l because it was chosen without any restriction[2].

Now if the filter with index $\tilde{q}$ has product $|\rho_{s_{\tilde{q}}} \cdot \kappa_{s_{\tilde{q}}}| \cdot |\breve{g}_{rq}^{[0]}|$,i.e. between the its statistical scaling and the sum of its taps at the first iteration, in comparison with all the other filters, then the following iteration holds

$$\frac{\breve{g}_{rq}^{[l]}}{\breve{g}_{r\tilde{q}}^{[l]}} = \frac{|\rho_{s_{\tilde{q}}} \cdot \kappa_{s_{\tilde{q}}}|^{\frac{1}{p-1}}}{|\rho_{s_q} \cdot \kappa_{s_q}|^{\frac{1}{p-1}}} \cdot \left(\frac{|\rho_{s_q} \cdot \kappa_{s_q}|^{\frac{1}{p-1}}}{|\rho_{s_{\tilde{q}}} \cdot \kappa_{s_{\tilde{q}}}|^{\frac{1}{p-1}}} \cdot \frac{|\breve{g}_{rq}^{[0]}|}{|\breve{g}_{r\tilde{q}}^{[0]}|} \right)^{p^l} \tag{C.15}$$

Note that the normalization step does not affect the iteration. Instead it is important in order to preserve the energy of at least one filter.

$$\tilde{q} = \arg\max_q \left\{ |\rho_{s_q} \cdot \kappa_{s_q}|^{\frac{1}{p-1}} \cdot |\breve{g}_{rq}^{[0]}| \right\} \tag{C.16}$$

For $p = 1$, i.e., for second order statistics, (C.11) becomes

$$\breve{g}_{rq}^{[l]} = (\rho_{s_q} \cdot \kappa_{s_q})^l \cdot \breve{g}_{rq}^{[0]} \tag{C.17}$$

Therefore the recursive relation is

$$\frac{\breve{g}_{rq}^{[l]}}{\breve{g}_{r\tilde{q}}^{[l]}} = \left(\frac{|\rho_{s_q} \cdot \kappa_{s_q}|}{|\rho_{s_{\tilde{q}}} \cdot \kappa_{s_{\tilde{q}}}|} \right)^l \cdot \frac{|\breve{g}_{rq}^{[0]}|}{|\breve{g}_{r\tilde{q}}^{[0]}|} \tag{C.18}$$

Concluding the leading filter is preserved if the signals are stationary and its index will be

$$\tilde{q} = \arg\max_q \left\{ |\rho_{s_q} \cdot \kappa_{s_q}| \right\} \tag{C.19}$$

Concluding the convergence of the algorithm is based on the statistical scaling of the filters $|\rho_{s_q} \cdot \kappa_{s_q}|$ and the sum of the filter taps at the first iteration when $p \neq 1$, and only on the statistical scaling of the filters $|\rho_{s_q} \cdot \kappa_{s_q}|$ when $p = 1$.

[2]The same can be proved by inference

Appendix D

BSS conceived as a Multichannel Blind Deconvolution problem

From a point of view BSS can be directly related to multichannel blind deconvolution, in the same way as in BSFS (see Sect. 3.2) the sources s_q can be decoupled to refer to each output y_r through a different filter g_{rq} (actually this problem is equivalent to Q BSFS problems). Then the *leading filter* method can be applied for each output. This means that Q leading filter problems are solved simultaneously. The idea is depicted in Fig.D.1. The drawback of the multichannel approach is that global convergence is not assured, because the same source may be extracted in more than one outputs.

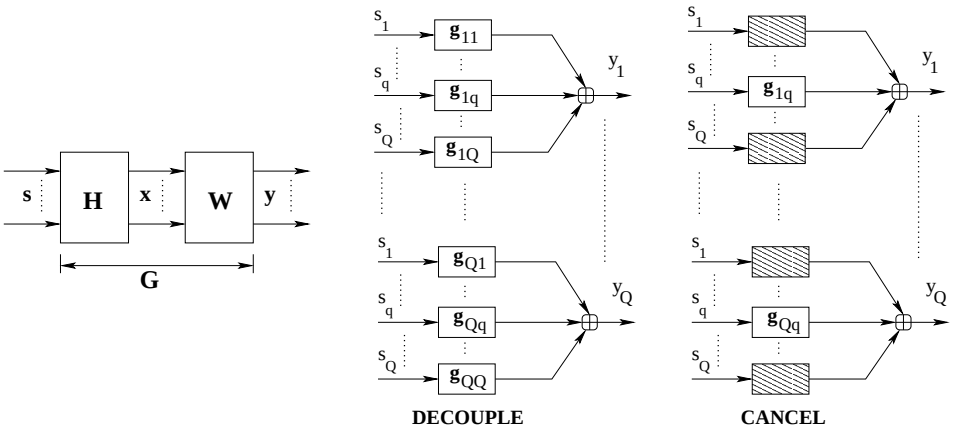

Figure D.1: BSS structure decoupled to Q FIR filters.

In order this method to become applicable the problem of global convergence should be solved. For instance the application of a symmetric orthogonalization

on the demixer **W** after every iteration cycle, may solve the problem [18, p. 194].

Bibliography

[1] J. Anemüller. *Across-Frequency Processing in Convolutive Blind Source Separation*. PhD thesis, Universität Oldenburg, Oldenburg, July 2001.

[2] M. Aoki and K. Furuya. Real-time source separation based on sound localization in a reverberant environment. In *Proc. Neural Networks for Signal Processing*, pages 475–484, Martigny, Switzerland, Sep. 2002.

[3] M. Aoki, M. Okamoto, S. Aoki, H. Matsui, T. Sakurai, and Y. Kaneda. Sound source segregation based on estimating incident angle of each frequency component of input signals acquired by multiple microphones. *Acoust. Sci. & Tech.*, 22(2):149–157, 2001.

[4] S. Araki, S. Makino, A. Blin, R. Mukai, and H. Sawada. Blind separation of more speech than sensors with less distortion by combining sparseness and ICA. In *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, pages 271–274, Kyoto, Japan, Sep. 2003.

[5] H. Attias. Independent factor analysis. *Neural Computation*, 11:803–851, 1999.

[6] J. Benesty. Adaptive eigenvalue decomposition algorithm for passive acoustic localization. *J. Acoust. Soc. Am.*, 107(1):384–391, Jan. 2000.

[7] A. Blin, S. Araki, and S. Makino. Blind source separation when speech signals outnumber sensors using a sparseness-mixing matrix estimation (SMME). In *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, pages 211–214, Kyoto, Japan, Sep. 2003.

[8] P. Bofill and M. Zibulevsky. Underdetermined blind source separation using sparse representations. *Signal Processing*, 81:2353–2362, 2001.

[9] H. Buchner, R. Aichner, and W. Kellermann. A generalization of a class of blind source separation algorithms for convolutive mixtures. In *Proc. Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, Nara, Japan, April 2003.

[10] H. Buchner, R. Aichner, and W. Kellermann. Blind source separation for convolutive mixtures: A unified treatment. In Y.Huang and J.Benesty, editors, *Audio Signal Processing for Next-Genaration Multimedia Communication Systems*. Kluwer Academic Publishers, Boston, Feb. 2004.

[11] J.-F. Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 9(10):2009–2025, Oct. 1998.

[12] L.J. Griffiths and C.W. Jim. An alternative approach to linearly constrained adaptive beamforming. *IEEE Trans. on AP*, AP-30(1):27–34, Jan. 1982.

[13] S. Haykin. *Adaptive Filter Theory*. Prentice Hall Inc., Englewood Cliffs, NJ, 1996.

[14] W. Herbordt, H. Buchner, and W. Kellermann. An acoustic human-machine front-end for multimedia applications. *EURASIP Journal on Applied Signal Processing*, 1(2003):21–31, Jan. 2003.

[15] W. Herbordt and W. Kellermann. Frequency-domain integration of acoustic echo cancellation and a generalized sidelobe canceller with improved robustness. *European Transactions on Telecommunications*, 13(2):123–132, March 2002.

[16] W. Herbordt and W. Kellermann. Adaptive beamforming for audio signal acquisition. In J.Benesty and Y.Huang, editors, *Adaptive signal processing: Application to real-world problems*, pages 155–194. Springer, Berlin, Jan. 2003.

[17] O. Hoshuyama, A. Sugiyama, and Hirano. A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters. *IEEE Trans. Signal Processing*, 47(10):2677–2684, Oct. 1999.

[18] A. Hyvaerinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[19] Y. Inouye and K. Tanebe. Super-exponential algorithms for multichannel blind deconvolution. *Trans. on Signal Processing*, 48(3):881–888, March 2000.

[20] D. H. Johnson and D. E. Dudgeon. *Array signal processing: Concepts and techniques*. Prentice Hall Signal Processing Series, 1993.

[21] D. C. Kammer. Linear algebra for test and analysis. `http://silver.neep.wisc.edu/~kammer/LinearAlg.pdf`.

[22] M. Kawamoto, K. Aoshima, and Y. Inouye. A signal separation technique which can be applied to the ears of robots. In *Proc. Int. Conference on Intelligent Robots and Systems*, pages 1050–1055, Las Vegas, Navada, USA, Oct. 2003.

[23] M. Kawamoto and Y. Inouye. Blind deconvolution of MIMO-FIR systems with colored inputs using second-order statistics. *IEICE Trans. Fundamentals*, E86-A(3):597–604, March 2003.

[24] M. Kawamoto and Y. Inouye. Generalized deflation algorithms for the blind source-factor separation of MIMO-FIR channels. In *Proc. Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, pages 561–566, Nara, Japan, April 2003.

[25] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine. Splitting the unit delay – tools for fractional delay filter design. *IEEE Signal Processing Magazine*, 13(1), Jan. 1996.

[26] A. Mansour, A.K. Barros, and N. Ohnishi. Blind separation of sources: Methods, assumptions and applications. *IEICE Trans. Fundamentals*, E83-A(8):1498–1512, August 2000.

[27] M. Martone. Blind multichannel deconvolution in multiple access spread spectrum communications using high order statistics. In *Proc. IEEE International Conference on Communications*, pages 49–53, Rome, Italy, Feb. 1995.

[28] M. Martone. An adaptive algorithm for antenna array low-rank processing in cellular TDMA base stations. *IEEE Trans. Communications*, 46(5):627–643, May 1998.

[29] M. Martone. On MMSE real-time antenna array processing using fourth-order statistics in the U.S. cellular TDMA system. *IEEE Journal on Selected Areas in Communications*, 16(8):1396–1410, Oct. 1998.

[30] I. A. McCowan. *Robust speech recognition using microphone arrays*. PhD thesis, Queensland University of Technology, Australia, 2001.

[31] J.M. Mendel. Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications. *Proceedings of the IEEE*, 79(3):278–305, Mar. 1991.

[32] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete Time Signal Processing*. Prentice Hall Signal Processing Series, 1999.

[33] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing*. Prentice Hall International, 1996.

[34] O. Shalvi and E. Weinstein. Super-exponential methods for blind deconvolution. *IEEE Trans. Information Theory*, 39(2):504–519, March 1993.

[35] K. Varma. Time-delay-estimate direction-of-arrival estimation for speech in reverberant environments. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Oct. 2002.

[36] S. V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction*. John Wiley & Sons, 2000.

[37] L. Vielva, D. Erdogmus, and J.C. Príncipe. Underdetermined blind source separation using a probabilistic source sparsity model. In *3er International Conference on Independent Component Analysis and Blind Signal Separation, Speech and Signal Processing (ICASSP)*, pages 675–679, San Diego, USA, Dec. 2001.

[38] O. Yilmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Trans. Signal Processing*, 2003. submitted.

[39] Y. Zhang, W. Mu, and M.G. Amin. Subspace analysis of spatial time-frequency distribution matrices. *IEEE Trans. Signal Processing*, 49(4):747–759, April 2001.