Université catholique de Louvain

École polytechnique de Louvain (EPL)

Multi-channel blind deconvolution with short filters

Algorithm, local ill-posedness and applications

Dissertation presented by
Antoine Paris

for obtaining the Master's degree in
Electrical Engineering

Supervisor(s)
Laurent Jacques, Laurent Demanet

Reader(s)
Luc Vandendorpe

Academic year 2017-2018

Acknowledgements

First of all, I would like to offer my special thanks to Prof. Laurent Jacques for the numerous meetings we had, the time he dedicated to me and its continuous help during my master thesis.

Next, I would like to thank Prof. Laurent Demanet that supervised me when this work started at MIT (Imaging and Computing Group, Department of Mathematics) during summer 2017. There, I also benefited from the experience in blind deconvolution of Ali Ahmed. I would like to thank them both for this experience but also for their help and interesting discussions during the whole year that followed.

I would also like to thank Pawan Bharadwaj for his help on questions regarding applications in geophysical imaging.

Finally, I would like to thank my friends that accepted to proofread my master thesis for their valuable comments.

Abstract

This work considers the problem of recovering length-L vectors $\boldsymbol{s}$ and $\{\boldsymbol{w}_n\}_{n=0}^{N-1}$ from their circular convolutions $\boldsymbol{y}_n = \boldsymbol{s} \circledast \boldsymbol{w}_n$. This problem is referred to as multi-channel blind deconvolution. The input signal $\boldsymbol{s}$ is assumed to be random while the filters $\boldsymbol{w}_n$ are assumed to be short, i.e., only the K first elements of $\boldsymbol{w}_n$ are non-zero.

This problem can be reformulated as a low-rank matrix recovery problem and solved by nuclear norm minimization. Numerical experiments show the effectiveness of this algorithm under a certain generative model of the channels. Under this generative model and in the noiseless case, the proposed algorithm achieves successful recovery as soon as $\frac{LN}{L+KN} \geq 1$, i.e., as soon as the oversampling factor is greater than or equal 1. It is also robust in the presence of noise.

The hard constraint on the filters length imposed by our model is however a source of ill-posedness. In this work, sufficient and necessary conditions on the filters that guarantee well-posedness are given and proved. The derived identifiability conditions were already known. To our knowledge, this is however the first time that such conditions are linked to the null space of the Hessian of the cost function around the ground truth.

Two applications of multi-channel blind deconvolution are then discussed. In the context of communications systems, multi-channel blind deconvolution can be used to perform multi-channel blind equalization. A complete blind equalization system is described and its performance is assessed in numerical simulations. In the context of geophysical imaging, multi-channel blind deconvolution can be used to perform Green's functions retrieval. Unfortunately, with the proposed model and algorithm, this last application suffers from ill-posedness.

Contents

Chapter 1

Introduction

Blind deconvolution is an ubiquitous problem that pervades many fields of science and engineering: communications systems (referred to as blind equalization [22]), geophysical imaging [3], medical imaging [14], astronomical imaging [11], computer vision [12], etc.

Before diving into multi-channel blind deconvolution in the next chapters, the simplest form of the blind deconvolution problem is discussed and some "facts" about it – that remains valid for the multi-channel case – are given. The organization of the document is also described.

Let $\boldsymbol{s}$ and $\boldsymbol{w}$ be two unknown vectors in $\mathbb{R}^L$. The vector $\boldsymbol{s}$ can be referred to as the *input signal* while $\boldsymbol{w}$ can be referred to as the *channel* or *filter*. In its simplest form, blind deconvolution is the problem of recovering both $\boldsymbol{s}$ and $\boldsymbol{w}$ from their circular convolution

$$\boldsymbol{y} = \boldsymbol{s} \circledast \boldsymbol{w}$$

or

$$y[l] = \sum_{k=0}^{L-1} w[k]s[l-k \mod L], \qquad l \in [L].$$

This problem will be referred to as single-input single-channel blind deconvolution. In blind deconvolution problems, circular convolution is generally preferred to linear convolution so that the convolution theorem for the Discrete Fourier Transform (DFT) applies. Even though convolutions arising in practical scenarios are not circular, a linear convolution can reasonably be approximated by a circular convolution if the support of $\boldsymbol{w}$ is sufficiently short with respect to L (i.e., if $\boldsymbol{w}$ decays quickly) [19]. In some situations, as in the context of communication systems, a linear convolution can even be transformed in a circular convolution very easily (see Chapter 5). The assumption that the convolution is circular is thus not limiting in most practical situations.

For starters, let's state some important facts about the blind deconvolution problem.

Fact 1 (Additional assumptions are needed)**.** *Without any additional assumptions, the blind deconvolution problem formulated above counts more unknowns than observations. The problem is thus under-determined and ill-posed.*

Fact 1 highlights the need for additional assumptions on $\boldsymbol{s}$ and $\boldsymbol{w}$. Chapter 2 first discusses popular assumptions made in recent related works on blind deconvolution. It starts with the single-input single-channel case as discussed in this introduction, proceeds with the multiple-inputs single-channel case and finally describes the problem studied in this work: single-input multi-channel case with the only assumption that the channels are *short*. Under this assumption, Chapter 3 then explains how the single-input multi-channel blind deconvolution problem can be reformulated as a low-rank matrix recovery problem and describes an algorithm to solve this reformulated problem that was already use in the single-input single-channel case. It then assesses the performances of this algorithm with numerical experiments under a certain random generative model for the channels.

As discussed at the end of Chapter 3 for the particular case of single-input multi-channel blind deconvolution with short channels, the additional assumptions on $\boldsymbol{s}$ and $\boldsymbol{w}$ do not guarantee well-posedness. This is formalized in the next fact.

Fact 2 (Ill-posedness)**.** *Let $\boldsymbol{f}$ be a filter that admits an inverse $\boldsymbol{f}^{-1}$, i.e. $\boldsymbol{f} \circledast \boldsymbol{f}^{-1} = \boldsymbol{\delta}$. If $\boldsymbol{f}$ is such that $\boldsymbol{s}' = \boldsymbol{s} \circledast \boldsymbol{f}$ and $\boldsymbol{w}' = \boldsymbol{f}^{-1} \circledast \boldsymbol{w}$ satisfy the assumptions made on $\boldsymbol{s}$ and $\boldsymbol{w}$ respectively, then $\boldsymbol{s}'$ and $\boldsymbol{w}'$ are also valid solutions of the blind deconvolution problem. Indeed,*

$$\boldsymbol{s}' \circledast \boldsymbol{w}' = \boldsymbol{s} \circledast \boldsymbol{f} \circledast \boldsymbol{f}^{-1} \circledast \boldsymbol{w} = \boldsymbol{s} \circledast \boldsymbol{w}.$$

A unavoidable particular case of Fact 2 is given in the next fact.

Fact 3 (Scalar ambiguity)**.** *For any scalar $\alpha \in \mathbb{R}_0$, $\alpha \boldsymbol{s}$ and $\alpha^{-1}\boldsymbol{w}$ are also valid solutions of the blind deconvolution problem. Indeed,*

$$(\alpha \boldsymbol{s}) \circledast (\alpha^{-1}\boldsymbol{w}) = \boldsymbol{s} \circledast \boldsymbol{w}.$$

As a consequence of Fact 3, one can only hope to recover $\boldsymbol{s}$ and $\boldsymbol{w}$ from $\boldsymbol{y}$ up to a scalar factor. Resolving this scalar ambiguity requires additional information on either $\boldsymbol{s}$ or $\boldsymbol{w}$ (e.g., if the norm of either $\boldsymbol{s}$ or $\boldsymbol{w}$ is known, the scalar ambiguity can be reduced to a sign ambiguity). However, this scalar ambiguity is generally accepted. In the single-input multi-channel case, Chapter 4 aims at deriving conditions on the channels that make this scalar ambiguity the only possible ambiguity, i.e., $\boldsymbol{f} = \alpha\boldsymbol{\delta}$ is the only filter that satisfies the properties given in Fact 2.

Next, Chapter 5 describes an application of single-input multi-channel blind deconvolution in the context of communication systems: multi-channel blind equalization. Chapter 6 then describes another application in the context of geophysical imaging: Green's functions retrieval. Unfortunately, this last application suffers from ill-posedness and does not work despite the simplified settings established.

Finally, Chapter 7 concludes this work by a summary of the strengths and weaknesses of the proposed model and algorithm and discusses some perspectives for future works.

1.1 Notations

Although reminders about the non-trivial notations are disseminated in the text, the paragraph below lists the main notations used.

Circular and linear convolutions are noted as $\circledast$ and $*$, respectively. Upper bold letters and lower bold letters denote matrices and vectors, respectively, e.g. $\boldsymbol{X}$ and $\boldsymbol{x}$. The i-th element of vector $\boldsymbol{x}$ is noted $x[i]$. The transpose and Hermitian transpose of a matrix (or a vector) $\boldsymbol{X}$ are noted $\boldsymbol{X}^T$ and $\boldsymbol{X}^*$, respectively. The notation $\overline{\boldsymbol{X}}$ (or $\overline{\boldsymbol{x}}$ for vectors) denotes entry-wise complex conjugation (i.e. without transposition). For any matrix $\boldsymbol{X}$, $\|\boldsymbol{X}\|_*$ denotes the nuclear norm of $\boldsymbol{X}$ (i.e., the sum of its singular values) while $\|\boldsymbol{X}\|_F = \sqrt{\sum_{ij} |X_{ij}|^2}$ denotes the Frobenius norm. For any pair of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^*\boldsymbol{y}$ denotes their inner product. For any pair of matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ of compatible dimensions, $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \operatorname{trace}(\boldsymbol{X}^*\boldsymbol{Y})$ denotes their trace inner product. The notation $\boldsymbol{I}_{L\times K}$ denotes an $L \times K$ matrix formed by taking the K first columns of the identity matrix. The normalized $L \times L$ Discrete Fourier Transform matrix is noted $\boldsymbol{F}$ and is given by

$$F_{kl} = \frac{1}{\sqrt{L}} e^{-j2\pi kl/L}, \qquad (k,l) \in [L] \times [L]$$

where $[L]$ denotes the set $\{0, 1, \ldots, L-1\}$. Its restriction to its first K columns is written as $\boldsymbol{F}_K$. The set of vectors $\{\boldsymbol{e}_n\}_n$ denotes the standard basis vectors. Vectors in the Fourier domain, e.g. $\boldsymbol{F}\boldsymbol{x}$, are often abbreviated as $\hat{\boldsymbol{x}}$. Finally, the Hadamard (i.e. entry-wise) product is noted $\odot$ while the Kronecker product is denoted $\otimes$.

Chapter 2

Related work and problem setup

This chapter first briefly reviews some recent works on several flavours of the blind deconvolution problem. The common assumptions used are detailed and the methods to effectively solve the problem are briefly mentioned. This is the topic of Sec. 2.1. Next, Sec. 2.2 set up the problem of interest here: single-input multiple-channel blind deconvolution under the assumption that the channels are short.

2.1 Related work

Sec. 2.1.1 starts with single-input single-channel blind deconvolution as discussed in the pioneering paper [2] by Ahmed, Recht and Romberg and in the more recent [13]. In [1], Ahmed and Demanet leverage multiple diverse inputs going through a single-channel to weaken the assumption on the channel. This is discussed in Sec. 2.1.2. Finally, Sec. 2.1.3 concludes with a problem very close to the one discussed in this work that is briefly mentioned in [1] as a "dual" problem of the multiple-input single-channel blind deconvolution.

2.1.1 Single-input single-channel blind deconvolution

This first case is the same as the one described in the introduction. Let $\boldsymbol{s}$ and $\boldsymbol{w}$ be two unknown vectors in $\mathbb{R}^L$. The vector $\boldsymbol{s}$ is referred to as the *input signal* while $\boldsymbol{w}$ is referred to as the *channel* or *filter*. Blind deconvolution is the problem of recovering both $\boldsymbol{s}$ and $\boldsymbol{w}$ from their circular convolution

$$\boldsymbol{y} = \boldsymbol{s} \circledast \boldsymbol{w}.$$

As discussed in the introduction, without any additional assumptions on either $\boldsymbol{s}$ or $\boldsymbol{w}$, this problem counts more unknowns than observations and is thus *ill-posed*. A popular assumption to avoid this ill-posedness, see e.g. [2, 13], consists in writing

$$\begin{aligned} \boldsymbol{s} &= \boldsymbol{B}\boldsymbol{x}, \qquad \boldsymbol{x} \in \mathbb{R}^M \\ \boldsymbol{w} &= \boldsymbol{C}\boldsymbol{h}, \qquad \boldsymbol{h} \in \mathbb{R}^K \end{aligned}$$

for *known* matrices $\boldsymbol{B} \in \mathbb{R}^{L\times M}$ and $\boldsymbol{C} \in \mathbb{R}^{L\times K}$. In other words, $\boldsymbol{s}$ and $\boldsymbol{w}$ are assumed to live in known subspaces whose bases are given by the columns of $\boldsymbol{B}$ and $\boldsymbol{C}$. Recovering $\boldsymbol{s}$ and $\boldsymbol{w}$ is now strictly equivalent to recovering $\boldsymbol{x}$ and $\boldsymbol{h}$ and the number of unknowns has been reduced from $2L$ to $M + K$. How much this assumption is reasonable depends on the nature of the matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ and the underlying application context.

In [2], the authors discuss "multipath channel protection using random code" (this could have been equivalently called "blind channel estimation using random code"). In this application, $\boldsymbol{x}$ is a message to be transmitted, $\boldsymbol{B}$ is a random "encoding" matrix and $\boldsymbol{s} = \boldsymbol{B}\boldsymbol{x}$ is the "protected" message. Next, the multipath communication channel $\boldsymbol{w}$ is assumed to be non-zero at a few *known* locations only and $\boldsymbol{C}$ is thus a subset of the columns of the identity matrix. While the first part of that assumption could be made reasonable in practice by assuming that the transmitter and the receiver agreed on the encoding matrix $\boldsymbol{B}$ beforehand, the locations where the channel impulse response $\boldsymbol{w}$ is non-zero are generally not known in advance. A more reasonable assumption considers that the channel $\boldsymbol{w}$ is short, that is, only its K first elements are non-zero. In this case, $\boldsymbol{C}$ consists in the K first columns of the identity matrix. This assumption is used in the numerical simulations presented in [2] and in [13]. This is also the one used in this work in a multi-channel context, as discussed in Sec. 2.2.

In [2], blind deconvolution is reformulated as a rank-1 matrix recovery problem that is then relaxed to a nuclear norm minimization program. Roughly, the main result of this paper guarantees recovery with high probability when $\frac{L}{\log^3 L} \gtrsim K + M$ and under a certain random generative model of $\boldsymbol{B}$ and $\boldsymbol{C}$.

In [13], the authors propose a sophisticated gradient descent algorithm to solve blind deconvolution. The minimum value of L that allows this algorithm to recover $\boldsymbol{s}$ and $\boldsymbol{w}$ is roughly similar to the one obtained in [2] but the proposed algorithm is claimed to be more computationally efficient.

2.1.2 Multiple-input single-channel blind deconvolution

The single-channel blind deconvolution problem can be extended to deal with multiple unknown inputs $\boldsymbol{s}_n \in \mathbb{R}^L$, $n \in [N]$. In this case, the objective is to recover the channel $\boldsymbol{w} \in \mathbb{R}^L$ and the N unknown inputs $\boldsymbol{s}_n$ from the observed circular convolutions

$$\boldsymbol{y}_n = \boldsymbol{s}_n \circledast \boldsymbol{w}, \qquad n \in [N].$$

This is the problem studied in [1]. In this paper, the N inputs $\boldsymbol{s}_n$ are assumed to belong to *known* subspaces of dimension M, that is

$$\boldsymbol{s}_n = \boldsymbol{B}_n \boldsymbol{x}_n, \qquad \boldsymbol{x}_n \in \mathbb{R}^M, \qquad n \in [N]$$

for *known* matrices $\boldsymbol{B}_n \in \mathbb{R}^{L \times M}$. Furthermore, the matrices $\boldsymbol{B}_n$ are assumed to be independently and identically distributed Gaussian matrices. This naturally leads to an extension of the blind channel estimation using random codes discussed in the previous section where multiple messages are sent over the same channel. When the multiple inputs are sufficiently diverse, the known subspace assumption on $\boldsymbol{w}$ can be dropped and replaced by a weaker assumption. In [1], the channel $\boldsymbol{w}$ is assumed to be S-sparse in a known basis $\boldsymbol{C}$, i.e.,

$$\boldsymbol{w} = \boldsymbol{C}\boldsymbol{h}, \qquad \|\boldsymbol{h}\|_0 \leq S$$

where $\boldsymbol{C} \in \mathbb{R}^{L \times L}$ and $\boldsymbol{h} \in \mathbb{R}^L$. The authors then also recast the problem as a low-rank matrix recovery problem as in [2]. Roughly, the main result of this paper guarantees recovery with high probability when $M + S\log^2 S \lesssim \frac{L}{\log^4(LN)}$ and $N \gtrsim \log^2(LN)$ and under a certain random generative model of $\{\boldsymbol{B}_n\}_{n=0}^{N-1}$ and $\boldsymbol{C}$.

2.1.3 Single-input multi-channel blind deconvolution

Yet another case, briefly discussed in [2] as being "dual" to the previous one, is single-input multi-channel blind deconvolution. In this case, a single-input noted $\boldsymbol{s} \in \mathbb{R}^L$ is going through N

different channels $\boldsymbol{w}_n$

$$\boldsymbol{y}_n = \boldsymbol{s} \circledast \boldsymbol{w}_n, \qquad n \in [N]. \tag{2.1}$$

In this case, the authors propose to use the known random subspace assumption on the channels

$$\boldsymbol{w}_n = \boldsymbol{C}_n \boldsymbol{h}_n, \qquad \boldsymbol{h}_n \in \mathbb{R}^K, \qquad n \in [N]$$

for *known* matrix $\boldsymbol{C}_n \in \mathbb{R}^{L \times K}$ whose columns are a random subset of the columns of the identity matrix. The channels are thus assumed to be non-zero at only K known locations. As for the input, it is assumed to be an unknown "noise source" and is modeled by a length-L Gaussian vector. This last assumption is motivated by *noise imaging* applications [9]. An example of a noise imaging application using multi-channel blind deconvolution is given in the next section.

2.2 Problem setup

The problem studied in this work is a particular case of the one described in Sec. 2.1.3 that seems more reasonable in practical scenarios. Instead of being known random subsets of the columns of the identity matrix, the matrices $\boldsymbol{C}_n$ are all equal and given by the K first columns of the $L \times L$ identity matrix. The channels can thus be written as

$$\boldsymbol{w}_n = \boldsymbol{C} \boldsymbol{h}_n, \qquad \boldsymbol{h}_n \in \mathbb{R}^K, \qquad n \in [N]$$

where $\boldsymbol{C} = \boldsymbol{I}_{L \times K}$. This is equivalent to assuming that the channels are *short*, i.e.

$$\boldsymbol{w}_n = \begin{bmatrix} \boldsymbol{h}_n \\ \boldsymbol{0}_{L-K} \end{bmatrix}.$$

The input signal $\boldsymbol{s}$ is assumed to be a length-L Gaussian vector. This choice is again motivated by noise imaging applications. One such application in the field of geophysical imaging is "seismic-while-drilling" (SWD). In SWD, the objective is to perform sub-surface imaging on a drilling site without disrupting drilling operations. One proposed way of achieving this is to use the vibrations generated by the drill-bit in the subsurface as the noise-like input signal. Multiple receivers recording the seismic traces at several locations would then allow to recover the "channel impulse responses of the earth" using blind deconvolution. This for example discussed by Bharadwaj, Demanet and Fournier in [3] and will also be discussed in Chapter 6.

It is interesting to note that the assumptions made here no longer involve *random* subspaces. Indeed, the subspace in which the filters $\boldsymbol{w}_n$ live is completely deterministic. In a sense, this is what makes this model potentially more realistic in practical contexts. At the same time, this prevents the use of the classical mathematical tools of the field of inverse problems.

From the observations given by equation (2.1), the objective is then to recover the input signal $\boldsymbol{s}$ and the N short channels $\boldsymbol{h}_n$. This problem counts LN observations for $L + KN$ unknowns and there is thus hope for recovery as soon as

$$LN \geq L + KN.$$

This model was already used in [17] and [20]. However, the first paper only shows "example" of successful recovery but lacks a detailed characterization of the algorithm employed. Robustness in the presence of noise is not discussed neither. The second paper makes extra-assumptions belonging to the underwater acoustic field that we want to avoid.

The problem is illustrated in Fig. 2.1.

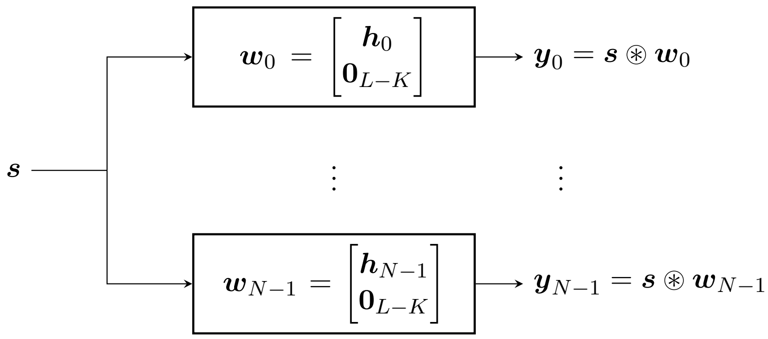


Figure 2.1: Illustration of the multi-channel blind deconvolution problem with short filters.

Chapter 3

Algorithm and numerical experiments

Several algorithms were already proposed in the literature to solve the blind deconvolution problem. For example, and as mentioned in the preceding chapter, blind deconvolution can be reformulated as a rank-1 matrix recovery problem. This is also the approach taken in this work. Sec. 3.1 adapts the reformulation presented in [1] to the case studied here. Sec. 3.2 then leverages the vast available literature in low-rank matrix recovery and describes an algorithm that already proved itself to be successful for single-input single-channel blind deconvolution [2]. The algorithm used here only differs in one particular point that will also be discussed in Sec. 3.2 and that is apparently also used in [20].

Next, Sec. 3.3 presents numerical experiments assessing the performance of the algorithm both in the noiseless case and in the noisy case.

Finally, Sec. 3.4 describes situations where multi-channel blind deconvolution cannot work. This will then lead to Chap. 4 that theoretically explores these situations.

3.1 Reformulation as a low-rank matrix recovery problem

The development to recast the problem as a rank-1 matrix recovery problem, while almost exactly identical to the one presented in [1], is re-written here for completeness.

Let $\boldsymbol{F}$ be the normalized $L \times L$ discrete Fourier transform matrix and let $\boldsymbol{f}_l^*$ denotes the l-th row of $\boldsymbol{F}$. Using the convolution theorem, equation (2.1) in the Fourier domain is given by

$$\hat{\boldsymbol{y}}_n = \sqrt{L}\hat{\boldsymbol{s}} \odot \hat{\boldsymbol{w}}_n, \qquad n \in [N]$$

where $\hat{\boldsymbol{y}}_n = \boldsymbol{F}\boldsymbol{y}$, $\hat{\boldsymbol{s}} = \boldsymbol{F}\boldsymbol{s}$, $\hat{\boldsymbol{w}}_n = \boldsymbol{F}\boldsymbol{w}_n$ and $\odot$ denotes the Hadamard (entrywise) product. This can also be written as

$$\hat{y}_n[l] = \sqrt{L}\langle \boldsymbol{f}_l, \boldsymbol{s}\rangle\langle \boldsymbol{f}_l, \boldsymbol{w}_n\rangle, \qquad (l, n) \in [L] \times [N].$$

Using the fact that $\boldsymbol{w}_n = \boldsymbol{C}\boldsymbol{h}_n$, this becomes

$$\hat{y}_n[l] = \sqrt{L}\langle \boldsymbol{f}_l, \boldsymbol{s}\rangle\langle \boldsymbol{f}_l, \boldsymbol{C}\boldsymbol{h}_n\rangle = \sqrt{L}\langle \boldsymbol{f}_l, \boldsymbol{s}\rangle\langle \boldsymbol{C}^*\boldsymbol{f}_l, \boldsymbol{h}_n\rangle, \qquad (l, n) \in [L] \times [N]. \tag{3.1}$$

By the conjugate symmetry of the complex inner product and because $\overline{\langle \boldsymbol{x}, \boldsymbol{y}\rangle} = \langle \bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}\rangle$, this can again be re-written as

$$\hat{y}_n[l] = \sqrt{L}\langle \boldsymbol{f}_l, \boldsymbol{s}\rangle\overline{\langle \boldsymbol{h}_n, \boldsymbol{C}^*\boldsymbol{f}_l\rangle} = \sqrt{L}\langle \boldsymbol{f}_l, \boldsymbol{s}\rangle\langle \boldsymbol{h}_n, \boldsymbol{C}^T\bar{\boldsymbol{f}}_l\rangle, \qquad (l, n) \in [L] \times [N].$$

The fact that $\boldsymbol{h}_n$ is a real vector was also used. Letting $\boldsymbol{c}_l = \sqrt{L}\boldsymbol{C}^T\bar{\boldsymbol{f}}_l$ to ease notations, this simplifies as

$$\hat{y}_n[l] = \langle \boldsymbol{f}_l, \boldsymbol{s}\rangle\langle \boldsymbol{h}_n, \boldsymbol{c}_l\rangle, \qquad (l,n) \in [L]\times[N].$$

This last expression can now be written as a trace inner product[1]

$$\hat{y}_n[l] = \langle \boldsymbol{f}_l\boldsymbol{c}_l^*, \boldsymbol{s}\boldsymbol{h}_n^*\rangle, \qquad (l,n) \in [L]\times[N].$$

This last expression already contains a rank-1 matrix formed by the outer product of $\boldsymbol{s}$ and $\boldsymbol{h}_n$. Let $\boldsymbol{h} \in \mathbb{R}^{KN}$ be the concatenation of all the unknown filters $\boldsymbol{h}^* = [\boldsymbol{h}_0^*, \dots, \boldsymbol{h}_{N-1}^*]$ and

$$\boldsymbol{\phi}_{l,n} = \boldsymbol{c}_l \otimes \boldsymbol{e}_n, \qquad (l,n) \in [L]\times[N]$$

a length-KN vector of zeroes except in positions indexed by $nK, \dots, (n+1)K-1$ where it is equal to $\boldsymbol{c}_l$. It is then possible to write the observations as

$$\hat{y}_n[l] = \langle \boldsymbol{f}_l\boldsymbol{\phi}_{l,n}^*, \boldsymbol{s}\boldsymbol{h}^*\rangle, \qquad (l,n) \in [L]\times[N].$$

The observations are thus obtained by trace inner products of known measurement matrices $\boldsymbol{f}_l\boldsymbol{\phi}_{l,n}^*$ with a rank-1 given by $\boldsymbol{s}\boldsymbol{h}^*$. Let $\mathcal{A}: \mathbb{R}^{L\times KN} \to \mathbb{C}^{LN}$ be a linear operator defined as

$$\mathcal{A}(\boldsymbol{X}) := \{\langle \boldsymbol{f}_l\boldsymbol{\phi}_{l,n}^*, \boldsymbol{X}\rangle \mid (l,n) \in [L]\times[N]\}. \tag{3.2}$$

Noting $\hat{\boldsymbol{y}} \in \mathbb{C}^{LN}$ the concatenation of all the observations in the Fourier domain $\hat{\boldsymbol{y}}^* = [\hat{\boldsymbol{y}}_0^*, \dots, \hat{\boldsymbol{y}}_{N-1}^*]$, the observations can thus be written as

$$\hat{\boldsymbol{y}} = \mathcal{A}(\boldsymbol{s}\boldsymbol{h}^*).$$

While the observations are nonlinear in both $\boldsymbol{s}$ and $\boldsymbol{h}$, they are linear in their outer product $\boldsymbol{s}\boldsymbol{h}^*$. Multi-channel blind deconvolution can thus be formulated as a linear inverse problem as follows

$$\begin{aligned} &\text{find} && \boldsymbol{X} \\ &\text{subject to} && \hat{\boldsymbol{y}} = \mathcal{A}(\boldsymbol{X}) \\ & && \operatorname{rank}(\boldsymbol{X}) = 1. \end{aligned}$$

This program is however non-convex because the set of rank-1 matrix is non-convex. In addition, it is known to be NP-hard because of the rank constraint. Fortunately, it can be relaxed as a convex program using the nuclear norm heuristic [15]

$$\begin{aligned} &\underset{\boldsymbol{X}}{\text{minimize}} && \|\boldsymbol{X}\|_* \\ &\text{subject to} && \hat{\boldsymbol{y}} = \mathcal{A}(\boldsymbol{X}). \end{aligned} \tag{3.3}$$

where the nuclear norm $\|\boldsymbol{X}\|_*$ is the sum of the singular values of $\boldsymbol{X}$. In what follows, $\boldsymbol{X}_0 = \boldsymbol{s}\boldsymbol{h}^*$ denotes the ground truth solution of the multi-channel blind deconvolution problem.

3.2 Algorithm

Low-rank matrix recovery and nuclear norm minimization under linear equality constraints are well studied problem in the literature [7, 15]. This section describes the algorithm presented in [15, Sec. 5.3] and inspired from [5, 6]. The different steps of the reformulation are summarized in Fig. 3.1.

[1] This can easily be seen using the fact that $\langle \boldsymbol{X}, \boldsymbol{Y}\rangle = \operatorname{trace}(\boldsymbol{X}^*\boldsymbol{Y}) = \sum_{i,j}\bar{X}_{ij}Y_{ij}$.

3.2.1 Low-rank parametrization

Program (3.3) counts LKN unknowns (i.e., the number of entries in the matrix $\boldsymbol{X}$) while the underlying problem only counts $L + KN$ unknowns. However, because $\boldsymbol{X}$ is known to be a low-rank matrix, it can be factorized as

$$\boldsymbol{X} = \boldsymbol{S}\boldsymbol{H}^*$$

with $\boldsymbol{S} \in \mathbb{R}^{L\times r}$ and $\boldsymbol{H} \in \mathbb{R}^{KN\times r}$, reducing the number of unknowns in $\boldsymbol{X}$ to $r(L + KN)$. If $r \geq \operatorname{rank}(\boldsymbol{X}_0) = 1$, problem (3.3) can be shown [15, lemma 5.1] to be equivalent to

$$\begin{aligned} \underset{\boldsymbol{S},\boldsymbol{H}}{\text{minimize}} \quad & \|\boldsymbol{S}\|_F^2 + \|\boldsymbol{H}\|_F^2 \\ \text{subject to} \quad & \hat{\boldsymbol{y}} = \mathcal{A}(\boldsymbol{S}\boldsymbol{H}^*). \end{aligned} \tag{3.4}$$

This new problem is however nonconvex. As a consequence, it might have local minima that do not correspond to global minima of (3.3). In [5, 6], Burer and Monteiro proposed an algorithm to solve (3.4) and showed that the local minima of (3.4) actually corresponds to global minima of (3.3) if $r > \operatorname{rank}(\boldsymbol{X}_0) = 1$. The next section describes this algorithm.

3.2.2 The method of multipliers

The algorithm proposed by Burer and Monteiro [5, 6] builds on the standard method of multipliers and iteratively minimizes the following augmented Lagrangian with respect to $\boldsymbol{S}$ and $\boldsymbol{H}$

$$\mathcal{L}(\boldsymbol{S}, \boldsymbol{H}, \boldsymbol{\lambda}, \sigma) = \frac{1}{2}\left(\|\boldsymbol{S}\|_F^2 + \|\boldsymbol{H}\|_F^2\right) - \langle \boldsymbol{\lambda}, \mathcal{A}(\boldsymbol{S}\boldsymbol{H}^*) - \hat{\boldsymbol{y}}\rangle + \frac{\sigma}{2}\|\mathcal{A}(\boldsymbol{S}\boldsymbol{H}^*) - \hat{\boldsymbol{y}}\|_2^2$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers and σ is a positive penalty parameter. Burer and Monteiro also proposed a deterministic way to update $\boldsymbol{\lambda}$ and σ at each iteration.

In our case, the augmented Lagrangian is minimized using Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) with the MATLAB solver *minFunc* [18]. To this effect, the gradient of the augmented Lagrangian needs to be computed. The partial derivatives of $\mathcal{L}$ are given by

$$\begin{aligned} \nabla_{\boldsymbol{S}}\mathcal{L} &= \boldsymbol{S} - \mathcal{A}^*(\boldsymbol{z})\boldsymbol{H} \\ \nabla_{\boldsymbol{H}}\mathcal{L} &= \boldsymbol{H} - \mathcal{A}^*(\boldsymbol{z})^*\boldsymbol{S} \end{aligned}$$

where $\mathcal{A}^*$ is the adjoint of $\mathcal{A}$ and $\boldsymbol{z} = \boldsymbol{\lambda} - \sigma(\mathcal{A}(\boldsymbol{S}\boldsymbol{H}^*) - \hat{\boldsymbol{y}})$. The calculation of the adjoint operator is detailed in appendix A. The algorithm stops when $\boldsymbol{S}\boldsymbol{H}^*$ is rank deficient (i.e., of rank 1).

The algorithm used to minimize the augmented Lagrangian also needs a starting point. In the absence of a better strategy, the starting point is currently chosen at random, i.e., the initial guesses for $\boldsymbol{S}$ and $\boldsymbol{H}$ are random Gaussian matrices. The deterministic nature of the measurement matrices involved in (3.2) does not allow us to use a smarter initialization scheme such as the spectral method proposed in [13].

Although $r \geq 2$ is needed to guarantee that the local minima of (3.4) are also global minima of (3.3), we empirically observed better results when $r = 1$ (in this case the algorithm stops when a specified tolerance on the constraint violation is achieved). Interestingly, this is also the approach used in [20] for multi-channel blind deconvolution in underwater acoustic channels. A direct consequence of this choice is that our algorithm might get trapped in local minima. To

$$\begin{array}{ll} \text{find} & \boldsymbol{s}, \{\boldsymbol{h}_n\}_{n=0}^{N-1} \\ \text{subject to} & \boldsymbol{y}_n = \boldsymbol{s} \circledast (\boldsymbol{C}\boldsymbol{h}_n) \end{array}$$

$$\Big\downarrow \text{ rank-1 reformulation, } \boldsymbol{X} = \boldsymbol{s}\boldsymbol{h}^*$$

$$\begin{array}{ll} \text{find} & \boldsymbol{X} \\ \text{subject to} & \hat{\boldsymbol{y}} = \mathcal{A}(\boldsymbol{X}) \\ & \operatorname{rank}(\boldsymbol{X}) = 1 \end{array}$$

$$\text{convex relaxation } \Big\downarrow$$

$$\begin{array}{ll} \underset{\boldsymbol{X}}{\text{minimize}} & \|\boldsymbol{X}\|_* \\ \text{subject to} & \hat{\boldsymbol{y}} = \mathcal{A}(\boldsymbol{X}) \end{array}$$

$$\Big\downarrow \text{ rank-}r\text{ parametrization, } \boldsymbol{X} = \boldsymbol{S}\boldsymbol{H}^*$$

$$\begin{array}{ll} \underset{\boldsymbol{S},\boldsymbol{H}}{\text{minimize}} & \|\boldsymbol{S}\|_F^2 + \|\boldsymbol{H}\|_F^2 \\ \text{subject to} & \hat{\boldsymbol{y}} = \mathcal{A}(\boldsymbol{S}\boldsymbol{H}^*) \end{array}$$

Figure 3.1: Illustration of the different problems involved.

avoid that, the following heuristic strategy was implemented. When the squared norm of the constraint violation, i.e.,

$$\|\mathcal{A}(\boldsymbol{s}\boldsymbol{h}^*) - \hat{\boldsymbol{y}}\|_2^2$$

stopped decreasing for a pre-determined number of iterations and is largely above the accepted tolerance, the algorithm decrees to be trapped at a local minima. At this point, the algorithm simply starts again from another starting point. Although clearly not optimal, this approach gives relatively good results for a reasonable average number of attempts in most cases. This will be discussed in more details in Sec. 3.3.

3.3 Numerical experiments

This section demonstrates the effectiveness of the algorithm described above for a certain random generative model of $\boldsymbol{s}$ and $\{\boldsymbol{h}_n\}_{n=0}^{N-1}$. Sec. 3.3.1 first discusses the empirical probability of successful recovery in the noiseless case. The average number of times the algorithm needs to restart with another random guess before succeeding is also discussed. Next, Sec. 3.3.2 discusses the noise robustness of the algorithm.

In the following numerical experiments, the vectors $\boldsymbol{h}$ (i.e., the concatenation of all the filters) and $\boldsymbol{s}$ are standard Gaussian vectors with independent entries. To comply with the existing literature on blind deconvolution using low-rank matrix recovery, the metric used to assess the quality of the recovery is the relative outer product error, i.e,

$$\frac{\left\|\boldsymbol{s}\boldsymbol{h}^* - \tilde{\boldsymbol{s}}\tilde{\boldsymbol{h}}^*\right\|_F}{\|\boldsymbol{s}\boldsymbol{h}^*\|_F}$$

where $\tilde{\boldsymbol{s}}$ and $\tilde{\boldsymbol{h}}$ are the recovered vectors.

3.3.1 Phase transitions

Fig. 3.2a depicts the empirical probability of successful recovery (on the left) and the average number of attempts before successful recovery (on the right) when $\boldsymbol{s}$ is of length $L = 32$, the number of channels N is ranging from 2 to 10 and the channels length K is ranging from 1 to 32. The recovery is considered to be successful if the relative outer product error is less than 2%, that is

$$\frac{\|\boldsymbol{X}_0 - \boldsymbol{s}\boldsymbol{h}^*\|_F}{\|\boldsymbol{s}\boldsymbol{h}^*\|_F} < 0.02.$$

Note that this threshold has been chosen to comply with the existing literature. Most of the time however, the actual relative error is orders of magnitude smaller than that (in the noiseless case). In each cell of Figs. 3.2a and 3.2b (i.e. for each pair of parameters N and K), the success probability has been averaged over 100 experiments involving a new random source and new random channels each time.

On both Fig. 3.2a and Fig. 3.2b, a red curve of equation

$$\frac{LN}{L + KN} = 1$$

is plotted. This curve corresponds to the situation where the oversampling factor is exactly equal to 1. Above this line, recovery is thus not possible.

Fig. 3.2a shows that the recovery is successful with probability 1 almost everywhere when the oversampling is ≥ 1 (i.e, below and on the red curve). This result is particularly interesting when compared to “usual” results in blind deconvolution where the minimal number of observations is larger than the number of unknowns (usually within log factors).

Finally, Fig. 3.2b shows that the average number of attempts before successful recovery is usually small and decreases when the oversampling factor increases. The number of attempts on this diagram is rarely above 3 and never exceeds 10. Our decision to use $r = 1$ in the low-rank parametrization described above, along with the heuristic implemented to avoid getting trapped at local minima thus seems reasonable.

3.3.2 Robustness to noise

In this section, the impact of noise on the relative outer product error is discussed. Each observed convolution $\boldsymbol{y}_n$ is corrupted by an additive noise $\boldsymbol{v}_n$ given by

$$\boldsymbol{v}_n = \sigma \cdot \|\boldsymbol{y}_n\|_2 \cdot \frac{\boldsymbol{\nu}_n}{\|\boldsymbol{\nu}_n\|_2}$$

where $\boldsymbol{\nu}_n \in \mathbb{R}^L$ is a standard Gaussian random vector. In this way, the SNR is simply given

$$\mathrm{SNR}_n = 10 \log_{10} \frac{\|\boldsymbol{y}_n\|_2^2}{\|\boldsymbol{v}_n\|_2^2} = 20 \log_{10} \frac{1}{\sigma}.$$

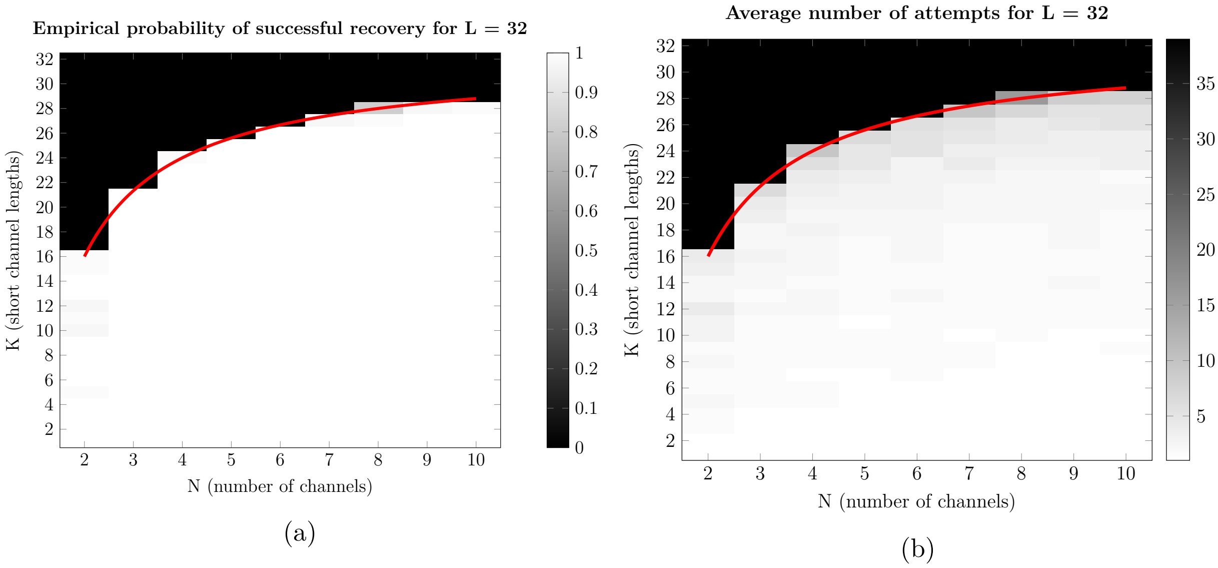


Figure 3.2: (a) Empirical probability of successful recovery for Gaussian vectors On the left, white means a 100% probability while black means a 0% probability. (b) Average number of attempts before recovery. Darker means that more attempts were needed to succeed in average. By convention, the maximum number of attempts (i.e. 40) has been assigned to cases above the red curve.

The SNR is also assumed to be the same for each observed convolution. In all of the following experiments, each data point is the result of an average over 2800 deconvolutions. For each deconvolution, new random instances of $\boldsymbol{s}$, $\boldsymbol{h}$ and the noise was used.

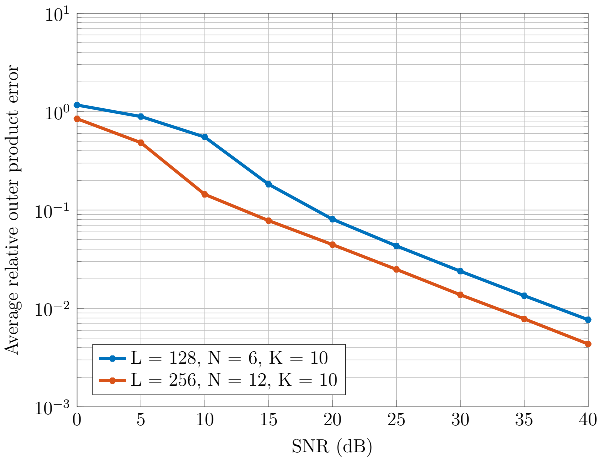
Fig. 3.3a depicts the average relative outer product error against the SNR. Except at low SNR (i.e. less than 10dB), the relative error scales linearly with the noise level as desired. In addition, increasing the oversampling factor helps the algorithm to achieve a better relative outer product error.

Next, Fig. 3.3b shows the average number of attempts before achieving the specified tolerance, against the SNR. At low SNR, the algorithm seems to achieve the specified tolerance after close to a single attempt. Given the performance of the algorithm at low SNR depicted on Fig. 3.3a, this suggests that the algorithm stops too early. However, other experiments show that decreasing the specified tolerance does not particularly help. Beyond that, the average number of attempts does not vary much with the SNR. This figure also shows that the average number of attempts is very low and that increasing the oversampling factor further reduces this number.

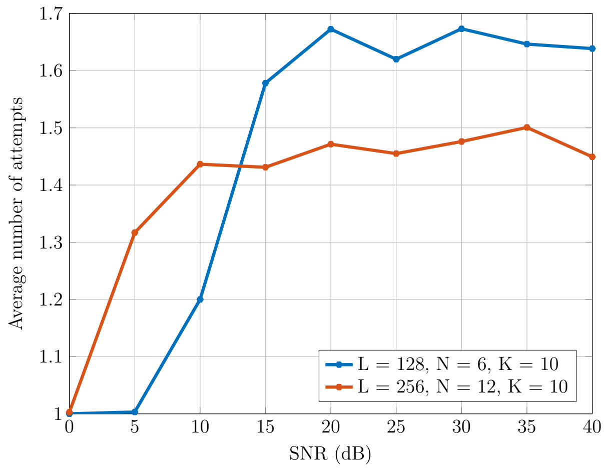
Finally, Fig. 3.3c is a plot of the average relative error against the oversampling factor for a fixed SNR. The oversampling factor was increased by varying the number of channels N from 2 to 16 for $L = 256$ and $K = 10$. This figure again illustrates the effect of the oversampling factor on the robustness to noise.

3.4 When does it fail?

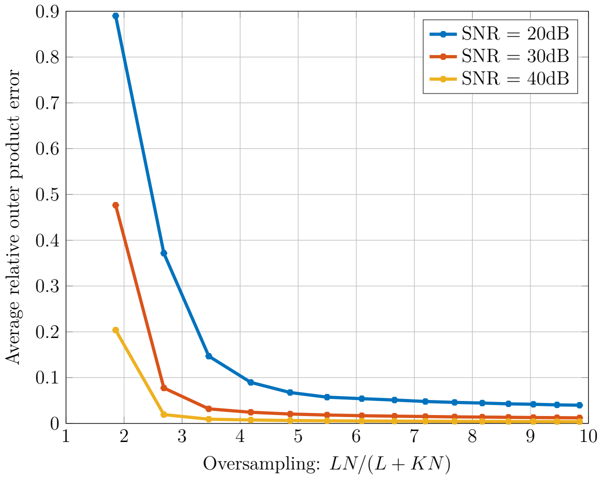
This chapter, particularly Sec. 3.3.1 discussing the noiseless case, seems to suggest that an oversampling factor greater than 1 is the only requirement for multi-channel blind deconvolution to work. However, it turns out that choosing $\boldsymbol{s}$ and $\boldsymbol{h}$ to be Gaussian random vectors is actually very favorable with regard to ill-posedness and Fact 2. For the filters, this is even proved



(a)



(b)



(c)

Figure 3.3: (a) Average relative outer product error against SNR for two sets of parameters. (b) Average number of attempts against SNR for the two same sets of parameters. (c) Average relative outer product error against the oversampling factor for 3 fixed SNRs. The oversampling factor was increased by varying the number of channels N from 2 to 13 for $L = 256$ and $K = 10$.

in Prop. 2 of Chapter 4. This section discusses some particular filters configurations where multi-channel blind deconvolution will fail.

The first obvious situation where multi-channel blind deconvolution fails is when all the channels are exactly identical. In this case, all the observations are redundant and the information count is not favorable. In the case where only a subset of the filters are identical, deconvolution might still work if the effective oversampling factor (i.e., counting the identical filters as a single filter) is sufficient.

Another case, closely related, is when filters are translated versions of each other. Because of the translation equivariance of convolution, this again leads to redundant observations. As above, if only a subset of the filters are translated versions of each other, deconvolution might still work if the effective oversampling factor is sufficient.

The last case is more interesting as it leads to ill-posedness even if the information count is favorable. Consider a set of filters that are neither identical nor translated versions of each other and assume the information count is favorable. If the filters are such that $h_n[K-1] = 0$ for all $n \in [N]$ (i.e., the "end" of the support is not filled), it is possible to find a vector $\boldsymbol{f}$ such that $\boldsymbol{w}_n \circledast \boldsymbol{f}$ is still supported on $[0, K-1]$. In this case, it is possible to find a vector $\boldsymbol{s}'$ that gives the observations $\boldsymbol{y}_n$ when convolved with $\boldsymbol{w}_n \circledast \boldsymbol{f}$. Indeed, $\boldsymbol{s}'$ must satisfy

$$\boldsymbol{y}_n = \boldsymbol{s}' \circledast (\boldsymbol{w}_n \circledast \boldsymbol{f}), \qquad n \in [N]$$

In the frequency domain, this becomes

$$\hat{\boldsymbol{y}}_n = \hat{\boldsymbol{s}}' \odot \hat{\boldsymbol{w}}_n \odot \hat{\boldsymbol{f}}, \qquad n \in [N]$$

or

$$\hat{y}_n[l] = \hat{s}'[l]\hat{w}_n[l]\hat{f}[l], \qquad (l, n) \in [L] \times [N].$$

Assuming $\hat{f}[l] \neq 0$ for all $l \in [L]$, the vector

$$\hat{s}'[l] = \frac{\hat{s}[l]}{\hat{f}[l]}, \qquad l \in [L]$$

is a solution of the above equations. This situation in fact corresponds to Fact 2 applied to the case studied here with short channels. Furthermore, numerical experiments show that this is indeed what is happening when the "end" of the support is not filled. In the numerical experiments presented in the last section, this situation never occurs because the filters are random Gaussian vectors. As such, the probability that the support is not complete is zero. However, this ill-posedness constitutes a limitation of the short channels model proposed in this work, as it directly depends on the choice of K. If K is larger than the actual channels length, the end of the support will not be filled and the problem is thus ill-posed. At the opposite, if K is smaller than the actual channels length, the support will be complete but the algorithm will not be able to explain the observations with channels shorter than the actual channels that generate the observations. This last claim can be confirmed numerically.

The hard constraint on the support size imposed by our model thus only makes sense in contexts where one can *reliably* estimates the support of the filters beforehand. In cases where such an estimate is not available, another way of favoring short filters in the recovery must be used.

Chapter 4

Local ill-posedness and filters identifiability conditions

The last chapter discussed the importance of the end of the support to be filled for the problem to be well-posed. However, it is not yet clear if this condition sufficient. This chapter aims at filling this gap.

First, an objective function to solve multi-channel blind deconvolution is defined. This objective function, while simple, is defined carefully to ease its analysis in the next sections. This is the topic of Sec. 4.1. Sec. 4.2 then makes the link between local[1] ill-posedness of the the problem and the null space of the Hessian of the objective function evaluated at the ground truth. Because of the inherent scalar ambiguity of blind deconvolution, this null space is expected to be at least of dimension 1. As discussed in the preceding chapters, this ambiguity is acceptable. A null space of higher dimension, however, would lead to other ambiguities and more local ill-posedness. Sec. 4.3 then details the computation of the Hessian of the objective function. Finally, Sec. 4.4 derives sufficient and necessary conditions on the filters for the null space of the Hessian to be of dimension 1. The same sufficient and necessary conditions were also derived in [23]. To our knowledge, this is however the first time that these conditions are linked to a notion of local ill-posedness.

Note that the theory developed here is not linked to any algorithm that could be used to solve multi-channel blind deconvolution.

4.1 Objective function definition

Let's start again from equation (3.1), repeated here for convenience

$$\hat{y}_n[l] = \sqrt{L}\langle \boldsymbol{f}_l, \boldsymbol{s}\rangle\langle \boldsymbol{f}_l, \boldsymbol{C}\boldsymbol{h}_n\rangle = \sqrt{L}\langle \boldsymbol{f}_l, \boldsymbol{s}\rangle\langle \boldsymbol{C}^*\boldsymbol{f}_l, \boldsymbol{h}_n\rangle, \qquad (l,n) \in [L]\times[N].$$

Noting $\hat{s}[l] = \sqrt{L}\langle \boldsymbol{f}_l, \boldsymbol{s}\rangle$ and $\boldsymbol{f}_{l,K} = \boldsymbol{C}^*\boldsymbol{f}_l$ a truncated version of $\boldsymbol{f}_l$ to its K first elements, the observations can be written as

$$\hat{y}_n[l] = \hat{s}[l]\langle \boldsymbol{f}_{l,K}, \boldsymbol{h}_n\rangle, \qquad (l,n) \in [L]\times[N].$$

This allows us define a bilinear map mapping any source signal in the Fourier domain $\boldsymbol{p}$ and any set of filters $\{\boldsymbol{q}_n\}_{n=0}^{N-1}$ to the corresponding observations in the Fourier domain. Let

[1]The term local is referring to the fact that it is derived from a local analysis of the cost function around the ground truth.

$\mathcal{B} : \mathbb{C}^L \times \mathbb{R}^{KN} \to \mathbb{C}^{LN}$ be this bilinear map, defined as follows

$$\mathcal{B}(\boldsymbol{p}, \boldsymbol{q}_0, \ldots, \boldsymbol{q}_{N-1}) := \{\boldsymbol{p}[l]\langle \boldsymbol{f}_{l,K}, \boldsymbol{q}_n\rangle \mid (l, n) \in [L] \times [N]\}. \tag{4.1}$$

This operator indeed satisfies

$$\mathcal{B}(\hat{\boldsymbol{s}}, \boldsymbol{h}) = \hat{\boldsymbol{y}}$$

where $\boldsymbol{h}$ and $\boldsymbol{y}$ are again defined as the concatenations of the filters and the observations in the Fourier domain, respectively. Note that this operator is in fact equivalent to the linear operator defined by (3.2). Indeed, one can easily show that

$$\mathcal{B}(\boldsymbol{p}, \boldsymbol{q}) = \mathcal{A}(\boldsymbol{F}^*\boldsymbol{p}\boldsymbol{q}^*)$$

where $\boldsymbol{q}^* = [\boldsymbol{q}_0^*, \ldots, \boldsymbol{q}_{N-1}^*]$. To ease notations in what follows, let $\boldsymbol{x} = [\boldsymbol{p}^*, \boldsymbol{q}^*]^*$ be the concatenation of all the arguments of $\mathcal{B}$. A possible objective function to be minimized to solve multi-channel blind deconvolution is then given by

$$f(\boldsymbol{x}) = \frac{1}{2}\left\|\mathcal{B}(\boldsymbol{x}) - \hat{\boldsymbol{y}}\right\|_2^2$$

In this context, $\boldsymbol{x}$ is a candidate solution. This objective function is actually equivalent to the "natural" ℓ_2-norm data misfit, both in the time and in the frequency domain (up to a scalar factor). This rather intricate way of defining such a simple objective function will ease the analysis of its Hessian in the next sections.

4.2 Local ill-posedness and Hessian null space

Let $\boldsymbol{x}_0 = [\hat{\boldsymbol{s}}^*, \boldsymbol{h}^*]^*$ be the ground truth. To investigate the local ill-posedness of the problem, let's first write the Taylor series of the objective function around the ground truth $\boldsymbol{x}_0$

$$f(\boldsymbol{x}_0 + \epsilon\boldsymbol{v}) = f(\boldsymbol{x}_0) + \epsilon\boldsymbol{\nabla} f(\boldsymbol{x}_0)^T\boldsymbol{v} + \frac{\epsilon^2}{2}\boldsymbol{v}^*\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)\boldsymbol{v} + \cdots .$$

Assuming ϵ is small enough for the higher order terms to be negligible, and because

$$\begin{aligned} f(\boldsymbol{x}_0) &= 0, \\ \boldsymbol{\nabla} f(\boldsymbol{x}_0) &= 0, \end{aligned}$$

this reduces to

$$f(\boldsymbol{x}_0 + \epsilon\boldsymbol{v}) = \frac{\epsilon^2}{2}\boldsymbol{v}^*\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)\boldsymbol{v}.$$

As a consequence, points $\boldsymbol{x}_0 + \epsilon\boldsymbol{v}$ such that

$$f(\boldsymbol{x}_0 + \epsilon\boldsymbol{v}) = 0 \quad \text{or} \quad \boldsymbol{v}^*\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)\boldsymbol{v} = 0 \tag{4.2}$$

also minimize the objective function. This shows how the null space of the Hessian of the objective function is related to local ill-posedness. Indeed, the set of solutions of this last equation is given by the null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$

$$\boldsymbol{v}^*\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)\boldsymbol{v} = 0 \quad \Leftrightarrow \quad \boldsymbol{v} \in \mathrm{Null}(\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)).$$

Because of the inherent scalar ambiguity of blind deconvolution, this null space is expected to be at least of dimension 1. The next section details the computation of the Hessian at $\boldsymbol{x}_0$.

4.3 Hessian at the ground truth

The following lemma is the first step to the characterization of the null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$. It was also the motive behind the definition of the objective function as detailed in Sec. 4.1.

Lemma 1. *Let $\mathcal{A}$ be a non-linear operator such that $\mathcal{A}(\boldsymbol{x}_0) = \boldsymbol{b}$. Let also $f(\boldsymbol{x}) = \frac{1}{2} \|\mathcal{A}(\boldsymbol{x}) - \boldsymbol{b}\|_2^2$. The Hessian of $f(\boldsymbol{x})$ at $\boldsymbol{x}_0$ is then given by*

$$\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0) = \boldsymbol{A}^* \boldsymbol{A}$$

with

$$\boldsymbol{A} = \left. \frac{\delta \mathcal{A}(\boldsymbol{x})}{\delta \boldsymbol{x}} \right|_{\boldsymbol{x}=\boldsymbol{x}_0}$$

the linearization of $\mathcal{A}(\boldsymbol{x})$ around $\boldsymbol{x}_0$ (i.e. the Jacobian of $\mathcal{A}$ at $\boldsymbol{x}_0$).

Lemma 1 can directly be applied to our objective function. Let $\boldsymbol{J} \in \mathbb{C}^{LN \times (L+KN)}$ be the Jacobian of $\mathcal{B}$ at $\boldsymbol{x}_0$. This Jacobian is given by

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{D}_{\hat{\boldsymbol{w}}_0} & \boldsymbol{D}_{\hat{\boldsymbol{s}}} \boldsymbol{F}_K & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{D}_{\hat{\boldsymbol{w}}_1} & \boldsymbol{0} & \boldsymbol{D}_{\hat{\boldsymbol{s}}} \boldsymbol{F}_K & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{D}_{\hat{\boldsymbol{w}}_{N-1}} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{D}_{\hat{\boldsymbol{s}}} \boldsymbol{F}_K \end{bmatrix}$$

where $\boldsymbol{D}_{\boldsymbol{x}} = \operatorname{diag}(\boldsymbol{x})$ and $\boldsymbol{F}_K \in \mathbb{C}^{L \times K}$ is the restriction of $\boldsymbol{F}$ to its K first columns. Using lemma. 1, the Hessian of the objective function at $\boldsymbol{x}_0$ is then given by

$$\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0) = \boldsymbol{J}^* \boldsymbol{J} = \begin{bmatrix} \sum_{n=0}^{N-1} \boldsymbol{D}^*_{\hat{\boldsymbol{w}}_n} \boldsymbol{D}_{\hat{\boldsymbol{w}}_n} & \boldsymbol{D}^*_{\hat{\boldsymbol{w}}_0} \boldsymbol{D}_{\hat{\boldsymbol{s}}} \boldsymbol{F}_K & \cdots & \boldsymbol{D}^*_{\hat{\boldsymbol{w}}_{N-1}} \boldsymbol{D}_{\hat{\boldsymbol{s}}} \boldsymbol{F}_K \\ \boldsymbol{F}_K^* \boldsymbol{D}^*_{\hat{\boldsymbol{s}}} \boldsymbol{D}_{\hat{\boldsymbol{w}}_0} & \boldsymbol{F}_K^* \boldsymbol{D}^*_{\hat{\boldsymbol{s}}} \boldsymbol{D}_{\hat{\boldsymbol{s}}} \boldsymbol{F}_K & & \boldsymbol{0} \\ \vdots & & \ddots & \\ \boldsymbol{F}_K^* \boldsymbol{D}^*_{\hat{\boldsymbol{s}}} \boldsymbol{D}_{\hat{\boldsymbol{w}}_{N-1}} & \boldsymbol{0} & & \boldsymbol{F}_K^* \boldsymbol{D}^*_{\hat{\boldsymbol{s}}} \boldsymbol{D}_{\hat{\boldsymbol{s}}} \boldsymbol{F}_K \end{bmatrix}. \tag{4.3}$$

A first observation on the null space of this matrix can be made. When the Jacobian $\boldsymbol{J}$ is fat, that is when $LN < L + KN$ or the oversampling factor is less than 1, the null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$ is at least of dimension $(L + KN - LN)$. If $L + KN - LN > 1$, there is thus no hope for this null space to be of dimension 1. What is interesting here is that for $L + KN - LN = 1$, or $L = \frac{KN-1}{N-1}$, it is still possible to have a one dimensional null space containing only the scalar ambiguity, even if the oversampling factor is less than 1. Successful recovery is thus still possible in this case[2]. This counter-intuitive result is in fact not that surprising given that a scalar ambiguity in the recovery is allowed. A first necessary condition for the null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$ to be of dimension 1 is $LN \geq L + KN - 1$.

4.4 Hessian null space and identifiability conditions

The null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$ can now be computed by solving (4.2). Letting $\boldsymbol{v} = [\boldsymbol{p}^*, \boldsymbol{q}_0^*, \ldots, \boldsymbol{q}_{N-1}^*]^*$ and using (4.3) this can be developed as

$$\sum_{n=0}^{N-1} \|\boldsymbol{D}_{\hat{\boldsymbol{w}}_n} \boldsymbol{p} + \boldsymbol{D}_{\hat{\boldsymbol{s}}} \boldsymbol{F}_K \boldsymbol{q}_n\|_2^2 = 0 \tag{4.4}$$

[2]Numerical experiments indeed show successful recovery for such cases, e.g. for $N = 2$, $L = 33$, $K = 17$ (i.e., 66 observations for 67 unknowns), in the noiseless case.

or, equivalently

$$\boldsymbol{D}_{\hat{\boldsymbol{w}}_n}\boldsymbol{p} = -\boldsymbol{D}_{\hat{\boldsymbol{s}}}\boldsymbol{F}_K\boldsymbol{q}_n, \qquad n \in [N].$$

A first obvious solution is given by

$$\boldsymbol{p} = \alpha\hat{\boldsymbol{s}} \quad \text{and} \quad \boldsymbol{q}_n = -\alpha\boldsymbol{h}_n, \qquad n \in [N]$$

for a any scalar α. This gives us a first basis vector of the null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$

$$\boldsymbol{v} = [\hat{\boldsymbol{s}}^*, -\boldsymbol{h}_0^*, \dots, -\boldsymbol{h}_{N-1}^*]^*.$$

The space generated by this vector contains the scalar ambiguity inherent to blind deconvolution. Indeed, let's consider the ground truth $\boldsymbol{x}_0$ and another minimizer of the objective function resulting from the scalar ambiguity, $\boldsymbol{x}' = [\alpha^{-1}\hat{\boldsymbol{s}}^*, \alpha\boldsymbol{h}_0^*, \dots, \alpha\boldsymbol{h}_{N-1}^*]^*$ The vector joining $\boldsymbol{x}'$ and $\boldsymbol{x}_0$ is given by

$$\boldsymbol{x}' - \boldsymbol{x}_0 = [\hat{\boldsymbol{s}}^*(1-\alpha^{-1}), \boldsymbol{h}_0^*(1-\alpha), \dots, \boldsymbol{h}_{N-1}^*(1-\alpha)]^*.$$

For $\alpha = 1+\epsilon$, this becomes

$$\boldsymbol{x}' - \boldsymbol{x}_0 = [\frac{\epsilon}{1+\epsilon}\hat{\boldsymbol{s}}^*, -\epsilon\boldsymbol{h}_0^*, \dots, -\epsilon\boldsymbol{h}_{N-1}^*]^*.$$

which for ϵ close to 0 reduces to[3]

$$\boldsymbol{x}' - \boldsymbol{x}_0 = [\epsilon\hat{\boldsymbol{s}}^*, -\epsilon\boldsymbol{h}_0^*, \dots, -\epsilon\boldsymbol{h}_{N-1}^*]^* = \epsilon\boldsymbol{v}.$$

The hope is that the obvious solution to (4.4) is the only one and hence that the null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$ is one dimensional. Indeed, the existence of other solutions would imply a null space of higher dimension which then imply the existence of other ambiguities. The next proposition gives two sufficient and necessary conditions on the filters for the null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$ to be one dimensional.

Proposition 1. *Let $P_{\boldsymbol{x}}(z)$ be the polynomial in z obtained by taking the $\mathcal{Z}$-transform of the vector $\boldsymbol{x}$ and substituting z^{-1} by z. Assume the signal $\boldsymbol{s}$ has no zero in the Fourier domain, $\hat{\boldsymbol{s}}[l] \neq 0$ for all $l \in [L]$ and $LN \geq L + KN - 1$. The null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$ is one dimensional and only contains the scalar ambiguity inherent to blind deconvolution if and only if*

1. *there exists an index $n' \in [N]$ such that $h_{n'}[K-1] \neq 0$, and*
2. *the polynomials $\{P_{\boldsymbol{h}_n}(z)\}_{n=0}^{N-1}$ are coprime, i.e., they do not share any common root.*

Proof. The objective is to prove that the obvious solution to (4.4) is the only one. For other solutions to exist, we would need

$$\boldsymbol{p} = -\boldsymbol{D}_{\hat{\boldsymbol{w}}_n}^{-1}\boldsymbol{D}_{\hat{\boldsymbol{s}}}\boldsymbol{F}_K\boldsymbol{q}_n, \qquad n \in [N]$$

which then implies

$$\boldsymbol{D}_{\hat{\boldsymbol{w}}_n}^{-1}\boldsymbol{D}_{\hat{\boldsymbol{s}}}\boldsymbol{F}_K\boldsymbol{q}_n = \boldsymbol{D}_{\hat{\boldsymbol{w}}_m}^{-1}\boldsymbol{D}_{\hat{\boldsymbol{s}}}\boldsymbol{F}_K\boldsymbol{q}_m, \qquad (n,m) \in [N] \times [N].$$

With the assumption that $\hat{\boldsymbol{s}}[l] \neq 0$ for $l \in [L]$, this simplifies as

$$\boldsymbol{D}_{\hat{\boldsymbol{w}}_m}\boldsymbol{F}_K\boldsymbol{q}_n = \boldsymbol{D}_{\hat{\boldsymbol{w}}_n}\boldsymbol{F}_K\boldsymbol{q}_m, \qquad (n,m) \in [N] \times [N]$$

which in the time domain becomes

$$\boldsymbol{w}_m \circledast \boldsymbol{C}\boldsymbol{q}_n = \boldsymbol{w}_n \circledast \boldsymbol{C}\boldsymbol{q}_m, \qquad (n,m) \in [N] \times [N]. \tag{4.5}$$

[3]Using a first order Taylor series around 0, $\frac{\epsilon}{1+\epsilon} \approx \epsilon$.

The objective is now to prove that

$$\boldsymbol{q}_n = \alpha \boldsymbol{h}_n, \qquad n \in [N]$$

is the only solution to this equation if and only if conditions 1 and 2 are true. Taking the $\mathcal{Z}$-transform on both sides of (4.5) and using the convolution theorem leads to

$$\mathcal{Z}(\boldsymbol{w}_m)\mathcal{Z}(\boldsymbol{C}\boldsymbol{q}_n) = \mathcal{Z}(\boldsymbol{w}_n)\mathcal{Z}(\boldsymbol{C}\boldsymbol{q}_m), \qquad (n,m) \in [N] \times [N]$$

or, more explicitly,

$$\left(\sum_{k=0}^{K-1} \boldsymbol{h}_m[k] z^{-k}\right)\left(\sum_{k=0}^{K-1} \boldsymbol{q}_n[k] z^{-k}\right) = \left(\sum_{k=0}^{K-1} \boldsymbol{h}_n[k] z^{-k}\right)\left(\sum_{k=0}^{K-1} \boldsymbol{q}_m[k] z^{-k}\right), \qquad (n,m) \in [N] \times [N].$$

This last equation holds for all $z \neq 0$ (so that the $\mathcal{Z}$-transform converges). Let us now perform the inconsequential change of variable $z \to z^{-1}$ so that we now deal with polynomials in z

$$\left(\sum_{k=0}^{K-1} \boldsymbol{h}_m[k] z^{k}\right)\left(\sum_{k=0}^{K-1} \boldsymbol{q}_n[k] z^{k}\right) = \left(\sum_{k=0}^{K-1} \boldsymbol{h}_n[k] z^{k}\right)\left(\sum_{k=0}^{K-1} \boldsymbol{q}_m[k] z^{k}\right), \qquad (n,m) \in [N] \times [N].$$

Using the notation defined in Prop. 1, the objective has now been transformed into proving that

$$P_{\boldsymbol{h}_m}(z) P_{\boldsymbol{q}_n}(z) = P_{\boldsymbol{h}_n}(z) P_{\boldsymbol{q}_m}(z), \qquad (n,m) \in [N] \times [N] \tag{4.6}$$

only admits

$$P_{\boldsymbol{q}_n}(z) = \alpha P_{\boldsymbol{h}_n}(z), \qquad n \in [N]$$

as a solution if and only if conditions 1 and 2 are true.

Forward direction Let us first prove the forward direction by contraposition. We want to prove that if the negation of either the first or the second condition true, then there exists other solutions to (4.4) and the null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$ is thus more than one dimensional.

1. Let's first assume that the negation of the first condition is true: there does not exist any index $n \in [N]$ such that $h_n[K-1] \neq 0$. Equivalently, there does not exist any index $n \in [N]$ such that $P_{\boldsymbol{h}_n}(z)$ is of degree $> K-2$. In this case,

   $$P_{\boldsymbol{q}_n}(z) = \alpha(z-\beta) P_{\boldsymbol{h}_m}(z), \qquad n \in [N]$$

   also constitutes a solution to (4.6) for any $\beta \in \mathbb{C}$ and the obvious solution is thus not the only one.

2. Let's now assume that the negation of the second condition is true: there is a root $\beta \in \mathbb{C}$ shared by all the polynomials $P_{\boldsymbol{h}_n}(z)$. Then

   $$P_{\boldsymbol{q}_m}(z) = \alpha \frac{P_{\boldsymbol{h}_m}(z)}{z-\beta}(z-\gamma), \qquad n \in [N]$$

   also constitutes a solution to (4.6) for any $\gamma \in \mathbb{C}$. For $\gamma \neq \beta$, this solution is different from the obvious one and the obvious solution is thus not the only one.

Backward direction Let us now prove the backward direction. We want to prove that if the two conditions are true, then the only solution to (4.4) is the obvious one and the null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$ is one dimensional. A necessary condition for (4.6) to be satisfied is that $P_{\boldsymbol{h}_m}(z)P_{\boldsymbol{q}_n}(z)$ and $P_{\boldsymbol{h}_n}(z)P_{\boldsymbol{q}_m}(z)$ have the same roots for all $(n, m) \in [N] \times [N]$.

1. Let's first assume that the second condition is satisfied: the polynomials $\{P_{\boldsymbol{h}_n}(z)\}_{n=0}^{N-1}$ do not share any common root. In this case, the necessary condition for (4.6) to be satisfied directly translates to $P_{\boldsymbol{q}_n}(z)$ having at least all the roots of $P_{\boldsymbol{h}_n}(z)$, $\forall n \in [N]$. This is however not sufficient to conclude that $P_{\boldsymbol{q}_n}(z) = \alpha P_{\boldsymbol{h}_n}(z)$ for all $n \in [N]$ as we could for example add a root $\beta \in \mathbb{C}$ in every $P_{\boldsymbol{q}_n}(z)$ if the first condition is not true (as was done in the preceding paragraph).
2. Let's next assume that the first condition is also satisfied: there exists an index $n' \in [N]$ such that $h_{n'}[K-1] \neq 0$. Equivalently, there exists an index $n' \in [N]$ such that $P_{\boldsymbol{h}_{n'}}(z)$ is of degree $K-1$. In this case, it is no longer possible to add a root $\beta \in \mathbb{C}$ in every $P_{\boldsymbol{q}_n}(z)$ as it would result in $P_{\boldsymbol{q}_{n'}}(z)$ being of degree K, which is not possible.

This allows us to conclude that, if the two conditions are true, $P_{\boldsymbol{q}_n}(z)$ and $P_{\boldsymbol{h}_n}(z)$ must have *exactly* the same roots for all $n \in [N]$. Because polynomials having exactly the same roots are equal up to a scalar factor, we can conclude that

$$P_{\boldsymbol{q}_n}(z) = \alpha P_{\boldsymbol{h}_n}(z), \qquad n \in [N]$$

is the only solution to (4.4) and that the null space of $\boldsymbol{\nabla}^2 f(\boldsymbol{x}_0)$ is one dimensional and only contains the scalar ambiguity inherent to blind deconvolution. □

The two conditions on the channels given in Prop. 1 are known as *identifiability* conditions. The same sufficient and necessary conditions were also derived in [23]. To our knowledge, this is however the first time that these conditions are linked to the null space of the Hessian of the objective function. In [23], the authors also derive a complicated condition on the input signal $\boldsymbol{s}$. It is not obvious how this condition might appear from the above reasoning.

Obviously, the two conditions given in Prop. 1 are not valid for arbitrary vectors $\boldsymbol{h}_m$ and $\boldsymbol{h}_n$ (or equivalently, polynomials $P_{\boldsymbol{h}_m}(z)$ and $P_{\boldsymbol{h}_n}(z)$ with arbitrary coefficients). The next proposition gives a general class of random vectors $\boldsymbol{h}_m$ and $\boldsymbol{h}_n$ that satisfy these two conditions with probability one.

Proposition 2. *Vectors $\boldsymbol{h}_n \in \mathbb{R}^K$ whose entries are independent random variables whose distribution can be defined by density functions (i.e., continuous random variables) satisfy the two conditions of Prop. 1 with probability one.*

Proof. Let's first prove that the first condition is satisfied with probability one. For all $n \in [N]$, $h_n[K-1]$ is a random variable whose distribution can be defined by a density function. As such, one can write

$$\Pr(h_n[K-1] = 0) \overset{a.s.}{=} 0, \qquad n \in [N]$$

and the first condition is satisfied almost surely. For the second condition, we work with the set of polynomials $\{P_{\boldsymbol{h}_n}(z)\}_{n=0}^{N-1}$. A proof that the probability for two such polynomials to have any common root is zero is given in [8, p. 442]. This proof can be extended to any number of polynomials and the second condition is thus also satisfied with probability one. □

Remark 1. *Prop. 1 shows that the requirement of the end of the support to be filled, as discussed in the previous chapter, is only a particular case of the more general identifiability conditions. The first condition prevents situations where $h_n[K-1] = 0$ for all $n \in [N]$, i.e., situations where the "end" of the support is not filled. As for the second condition, it prevents situations where* *the "beginning" of the support is not filled, i.e.* $h_n[0] = 0$ *for all* $n \in [N]$ *(as 0 would then be a shared root). Thus, the support must be complete on "both sides" to avoid ill-posedness.*

Chapter 5

Application to multi-channel blind equalization

Convolutions naturally arise in communication systems to model the effect of the channel on the transmitted signal. This convolution produces inter-symbol interference at the receiver that needs to be removed to ensure acceptable bit-error-rate (BER). This is typically done through a process called *equalization* Several equalization methods exist, both in the time and in the frequency domain. These methods are almost always *data aided*. Indeed, they first need a *known* sequence of symbols to be transmitted to estimate the channel impulse response. Blind deconvolution thus seems to be a natural fit to perform blind equalization.

Section 5.1 first briefly describes a basic transmission chain to help make the context clear. The classical way of performing equalization in communication systems is also briefly discussed. Then, section 5.2 describes a system that leverages the multi-channel blind deconvolution algorithm described in Chapter. 3 to perform multi-channel blind equalization. This application is partly inspired from the dual problem discussed in [1]. Section 5.3 presents numerical experiments assessing the performance of the proposed system and compares it with the performance of OFDM, an ubiquitous modulation scheme in today's communication systems.

5.1 Background

A real transmission chain can be modeled by a fully digital equivalent chain as given in Fig. 5.1.

A sequence of bits $b[l]$ needs to be transmitted over a communication channel. This sequence of bits is first transformed in a sequence of symbols $s[l]$. Symbols are typically complex numbers that belong to a set known as the *constellation* and noted $\mathcal{C}$ in what follows. The process of transforming bits into symbols is called *symbol mapping*. The simplest digital modulation scheme

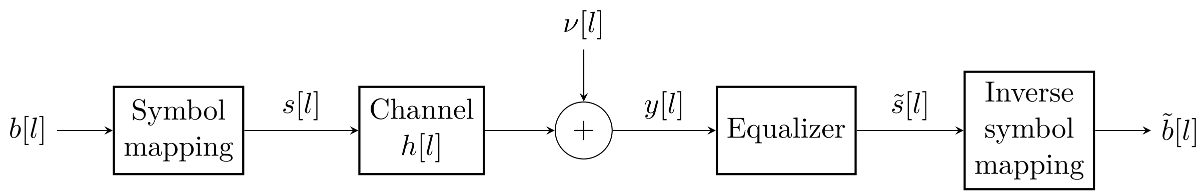


Figure 5.1: Digital equivalent transmission chain.

performs the following mapping

$$s[l] = \begin{cases} -1 & \text{if } b[l] = 0 \\ +1 & \text{if } b[l] = 1. \end{cases}$$

In this case, the constellation is simply given by

$$\mathcal{C} = \{-1, +1\}$$

and a symbol only conveys one bit of information. This digital modulation scheme is known as BPSK (Binary Phase Shift Keying). Another common digital modulation scheme, known as QPSK (Quadrature Phase Shift Keying), uses a complex constellation with four symbols

$$\mathcal{C} = \{1 + j, 1 - j, -1 - j, -1 + j\}$$

and thus maps two bits to each symbol.

The sequence of symbols is then transmitted over the communication channel. In practice, the sequence is converted to the analog domain through a pulse shaping filter, up-converted to the carrier frequency and then emitted by an antenna. On the receiver side, the signal is then down-converted, matched filtered and sampled. In simulations, the up and down-conversion are generally omitted for practical reasons. This leads to the so-called *baseband* equivalent transmission chain. Because both the transmitter and the receiver work in the digital domain in the end, the cascade of the pulse shaping filter, the "physical" channel and the matched filter can be modeled as a digital filter, noted h on Fig. 5.1. In communication systems jargon, an element $h[l]$ of h is called a *tap*. In wireless communications, each tap $h[l]$ of the digital equivalent filter h physically corresponds to a sum of contributions from several physical paths of the analog signal. Furthermore, if these paths can be considered independent and if the number of paths is sufficiently large, by the Central Limit Theorem, each tap $h[l]$ can be modeled as a circularly symmetric complex Gaussian random variables

$$h[l] \sim \mathcal{CN}(0, \sigma_l^2).$$

The variance σ_l^2 of each tap $h[l]$ determines how much energy is expected on the l-th tap. Because $|h[l]|$ follows a Rayleigh distribution, this model is called *Rayleigh fading*. It is reasonable in situations where the number of small reflectors in the environment is sufficiently large and is also often used for its simplicity [21, Chapter 2]. In addition, because the power of an electromagnetic wave decreases as r^{-2}, where r is the traveled distance, it is also reasonable to assume that most of the received power is concentrated in the K first taps of $h[l]$. The channel can thus be though of as being short, i.e. supported on $l \in [K]$.

On the receiving side, the received samples are given by

$$y[l] = \sum_{k=0}^{K-1} h[k]s[l-k] + \nu[l]. \tag{5.1}$$

The sequence of samples $y[l]$ is thus given by the *linear* convolution of the sequence of symbols $s[l]$ with the equivalent digital channel impulse response $h[l]$, corrupted by additive white noise $\nu[l]$. In a baseband equivalent model, the noise samples $\nu[l]$ are generally assumed to be independent and identically distributed according to a circularly symmetric complex Gaussian.

The goal of the receiver is of course to recover the transmitted bit sequence $b[l]$. The first step consists in estimating the transmitted symbols given $y[l]$. This process is known as *equalization* and somehow requires to invert the equivalent channel impulse response $h[l]$. Several equalization methods exist, both in the time and in the frequency domain. These methods are almost always *data aided.* Indeed, they first need a *known* sequence of symbols (typically called a *training sequence* or a *pilot*) to be transmitted to estimate the channel impulse response. In addition, because typical channel impulse responses vary with time (e.g., in mobile communications), this known sequence of symbols needs to be sent periodically so that the receiver can update its channel estimate. By using blind deconvolution to perform blind equalization, training sequences could be replaced by symbols containing information.

Finally, once equalization has been performed, the symbol estimates $\tilde{s}[l]$ are demodulated to give the bit estimates $\tilde{b}[l]$. For the simple BPSK modulation discussed at the beginning of this section, this amounts to looking at the sign of the real part of $\tilde{s}[l]$.

5.2 System description

Leveraging the multi-channel blind deconvolution algorithm described in Chapter 3 to replace the classical data aided equalization by a blind equalization scheme requires some changes to the transmission chain presented in the previous section.

First of all, and as expected, multiple channels are needed. As such, the receiver should be equipped with multiple antennas sufficiently spaced for the channels to be *diverse.* Another possible configuration might involve several independent receivers cooperating in some way to aggregate the received signals and feed them to the deconvolution algorithm. The rather intensive deconvolution task might even be off-loaded to some server in applications where latency is not a critical issue.

For practical reasons, the transmitted symbols should be transmitted in blocks (or packets) of small size to avoid the deconvolution to be too computationally demanding. In addition, the linear convolution in equation 5.1 needs to be transformed in a circular convolution for the deconvolution to work. This transformation is achieved via a classical operation called *cyclic prefix extension.* This will be discussed in section 5.2.1.

Next, the fact that our blind deconvolution algorithm does not work with complex numbers needs to be addressed. Indeed, as discussed in the previous section, the baseband equivalent channel $h[l]$ and the baseband noise $\nu[l]$ are generally complex. Interestingly, this issue turns out to have positive consequences if the chosen constellation is real. This is discussed in section 5.2.2.

Finally, the need for robustness to the scalar ambiguity inherent to blind deconvolution suggests a modulation scheme with differential encoding. For this reason, and because the constellation must be real, DBPSK (differential BPSK) will be used. This is discussed in section 5.2.3.

5.2.1 From linear to circular convolution

A linear convolution can easily be transformed into a circular convolution through a process called *cyclic prefix extension.* This is for example done in the ubiquitous OFDM (Orthogonal Frequency Division Multiplexing) modulation scheme.

Communication systems generally work on *blocks* of symbols. Denoting L the *block size*, the i-th block of symbols is defined as

$$\boldsymbol{s}_i = \begin{pmatrix} s[iL] & s[iL+1] & \cdots & s[(i+1)L-2] & s[(i+1)L-1] \end{pmatrix}.$$

The same notations will be used for blocks on $r[l]$ and $\nu[l]$. To use the blind deconvolution algorithm to perform block per block equalization on $\boldsymbol{y}_i$, the output of the equivalent transmission chain must be given by a circular convolution

$$\boldsymbol{y}_i = \boldsymbol{s}_i \circledast \boldsymbol{h} + \boldsymbol{\nu}_i. \tag{5.2}$$

However, this is not what the receiver gets at the output of the transmission chain presented in the previous section. Indeed, equation 5.1 for the i-th received block can be written as

$$y_i[l] = \sum_{k=0}^{l} h[k] s_i[l-k] + \sum_{k=l+1}^{K-1} h[k] s_{i-1}[L+l-k] + \nu_i[l], \qquad l \in [L].$$

For $l < K-1$, $y_i[l]$ thus depends on both $\boldsymbol{s}_i$ and $\boldsymbol{s}_{i-1}$. This is commonly called *inter-block interference* in communication systems. Fortunately, pre-processing the blocks $\boldsymbol{s}_i$ at the transmitter side can remove inter-block interference. This pre-processing step is called *cyclic prefix extension*. Let L_{cp} be the *cyclic prefix length*. The transmitter builds a vector $\boldsymbol{s}'_i$ given by

$$\boldsymbol{s}'_i = \Big(s_i[L - L_{cp}] \quad \dots \quad s_i[L-1] \quad s_i[0] \quad \dots \quad s_i[L-1]] \Big)$$

Simply stated, $\boldsymbol{s}'_i$ is formed by prefixing $\boldsymbol{s}_i$ with the last L_{cp} symbols of $\boldsymbol{s}_i$ itself. This vector is then transmitted through the channel $h[l]$. The receiver gets

$$y'_i[l] = \sum_{k=0}^{l} h[k] s'_i[l-k] + \sum_{k=l+1}^{K-1} h[k] s'_{i-1}[L + L_{cp} + l - k] + \nu' i[l], \qquad l \in [L + L_{cp}]$$

and throws away the first L_{cp} samples of $\boldsymbol{y}'_i$ to obtain $\boldsymbol{y}_i$

$$y_i[l] = \sum_{k=0}^{L_{cp}+l} h[k] s'_i[L_{cp} + l - k] + \sum_{k=L_{cp}+l+1}^{K-1} h[k] s'_{i-1}[L + 2L_{cp} + l - k] + \nu_i[l], \qquad l \in [L].$$

For $L_{cp} \geq K-1$, this reduces to

$$y_i[l] = \sum_{k=0}^{K-1} h[k] s'_i[L_{cp} + l - k] + \nu_i[l], \qquad l \in [L].$$

Finally, using the definition of $\boldsymbol{s}'_i$, one can write

$$s'_i[L_{cp} + l - k] = \begin{cases} s_i[L + l - k] & l - k < 0 \\ s_i[l-k] & l - k \geq 0 \end{cases}$$

or, equivalently,

$$y_i[l] = \sum_{k=0}^{K-1} h[k] s_i[l - k \mod L] + \nu_i[l].$$

This last equation exactly matches the definition of the circular convolution of $\boldsymbol{h}$ and $\boldsymbol{s}_i$. This demonstrates how a cyclic prefix whose length satisfies

$$L_{cp} \geq K - 1$$

transforms the linear convolution of equation 5.1 into the circular convolution of equation 5.2. This, however, comes at a price: $L + L_{cp}$ symbols are transmitted in each block for only L symbols containing information. This price is already paid by every communication systems using OFDM and is thus not a limitation in practice.

The complete transmission chain that transforms linear convolutions into circular convolutions is given in Fig. 5.2.

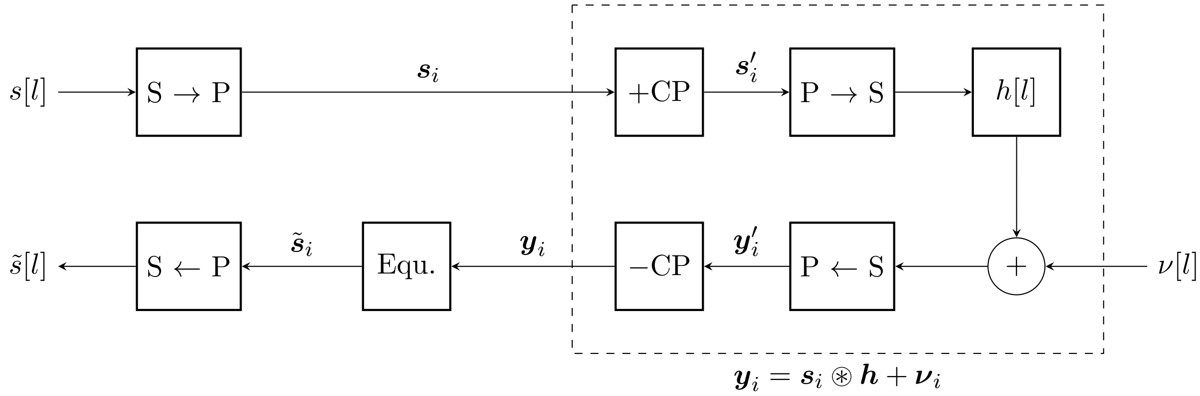


Figure 5.2: Cyclic prefix to transform linear convolution in circular convolution.

5.2.2 Dealing with complex channels

As discussed in section 5.1, Rayleigh fading is a simple yet reasonable model for the channel $h[l]$. However, our blind deconvolution algorithm does not work for complex numbers. Fortunately, this is not really an issue if the symbols are real. Indeed, one can write equation 5.2 as

$$\begin{aligned}\boldsymbol{y}_i &= \boldsymbol{s}_i \circledast (\mathrm{Re}\{\boldsymbol{h}\} + j\,\mathrm{Im}\{\boldsymbol{h}\}) . \\ &= \boldsymbol{s}_i \circledast \mathrm{Re}\{\boldsymbol{h}\} + j\,(\boldsymbol{s}_i \circledast \mathrm{Im}\{\boldsymbol{h}\})\end{aligned}$$

and take the real and imaginary parts of $\boldsymbol{y}_i$ to obtain *two* real channels

$$\begin{aligned}\mathrm{Re}\{\boldsymbol{y}_i\} &= \boldsymbol{s}_i \circledast \mathrm{Re}\{\boldsymbol{h}\} \\ \mathrm{Im}\{\boldsymbol{y}_i\} &= \boldsymbol{s}_i \circledast \mathrm{Im}\{\boldsymbol{h}\} .\end{aligned}$$

In other words, if the symbols are real, a single complex channel is equivalent to two real channels. As a consequence, the multi-channel blind equalizer might even work for a single "physical" channel. This restricts the system to use a real constellation and we thus choose the following constellation

$$\mathcal{C} = \{-1, +1\}.$$

5.2.3 Differential encoding

To demodulate BPSK modulated symbols $\tilde{s}[l]$, the receiver only needs to look at the sign of the real part of $\tilde{s}[l]$. However, if the equalization is performed via blind deconvolution, the estimated symbols can only be equal to the transmitted symbols up to a scalar factor $\alpha \in \mathbb{R}_0$, that is

$$\tilde{\boldsymbol{s}} = \alpha \boldsymbol{s}.$$

If $\alpha > 0$, the signs of the symbols will be preserved and the demodulation will produce the correct sequence of bits $\boldsymbol{b}$. If, however, $\alpha < 0$, the signs of the symbols will be flipped and the demodulation will produce the bit-wise negation of $\boldsymbol{b}$. The inherent scalar ambiguity of blind deconvolution thus becomes a *phase ambiguity* on the estimated symbols $\tilde{\boldsymbol{s}}$. A common approach to solve this ambiguity is to use differential encoding. Instead of the basic BPSK symbol mapping presented in section 5.1, the modulator encodes the information in the sign change between successive symbols as follows

$$s[l] = \begin{cases} s[l-1] & \text{if } b[l] = 0 \\ -s[l-1] & \text{if } b[l] = 1. \end{cases}$$

For $s[0]$, the modulator uses an arbitrary initial state, for example $s[-1] = +1$. On the receiving side, the demodulator only needs to look at the sign change between two successive received symbols to find the corresponding bit. The first symbol, however, needs to be discarded as there is no way to resolve its ambiguity. This digital modulation scheme, known as DBPSK (Differential BPSK), makes the system robust to the scalar ambiguity of the blind equalizer.

5.3 Numerical experiments

This section presents some numerical experiments to assess the performance of the deconvolution method described in the previous section. Rayleigh fading channels with uniform power delay profile (i.e., $\sigma_l^2 = \sigma^2$, $\forall l$) are used in all the simulations that follow. In addition, the channels are independent of each other.

5.3.1 Noiseless case

Even though the noiseless case has no practical interest, it is interesting to look at the empirical probability of successful recovery of our multi-channel blind equalizer in the absence of noise first. This is presented in Fig. 5.3 for a block length $L = 32$, channels lengths K ranging from 1 to 32 and number of complex channels N ranging from 1 to 8. In each case, the cyclic prefix length is such that linear convolutions are correctly transformed into circular convolutions. For each value of N and K, the experiment has been repeated 100 times.

In the communication systems context, the criterion for successful recovery used in Chapter 3 is not really practical. The metric of interest in this case is the bit-error-rate (BER), i.e., the percentage of bits that were not received correctly. Because the goal of the multi-channel blind equalizer is also to obtain an estimate of the channels, channels estimation errors should also be part of the new criterion. Let $\tilde{\boldsymbol{h}}$ be the estimate of the N complex channels (concatenated in a single vector) obtained via blind deconvolution and once the scalar ambiguity resolved. In the noiseless case, recovery is defined as successful if

$$\mathrm{BER} = 0$$

and

$$\frac{\|\boldsymbol{h} - \tilde{\boldsymbol{h}}\|_2}{\|\boldsymbol{h}\|_2} < 0.01.$$

The first part is justified by the fact that a noiseless communication system should not suffer from any transmission error. The second is more arbitrary but seems reasonable in this context. In the noiseless case, the actual relative recovery error is usually lower than 0.01.

Fig. 5.3 shows that, in the noiseless case, the system works perfectly for almost every value of K and N that are above the information theoretic limit (i.e. where the oversampling factor is $\geq$ 1).

5.3.2 Robustness to noise and comparison with OFDM

Fig. 5.4a shows BER curves against the SNR in different configurations. The red curve is the BER achieved by a simple single-carrier system on a channel where additive white Gaussian noise is the only impairment, i.e., the channel is of length 1, $h[l] = \delta[l]$. Such a channel is called an AWGN channel. This corresponds to the best possible configuration. The orange curve is the BER of the same system on multipath channels without any equalization. As expected,

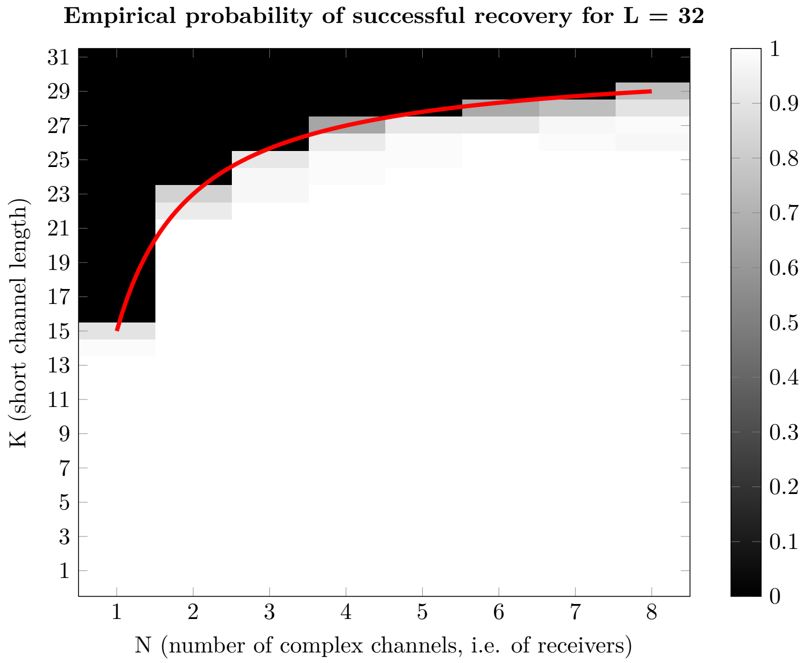


Figure 5.3: Empirical probability of successful recovery for a block size $L = 32$. White is 100% success and black is 0% success. Each success probability in the grid was averaged over 100 transmitted blocks, each time on different random Rayleigh fading channels. The red curve is the limit where the oversampling factor is equal to 1. Above this line, successful recovery is impossible from an information theoretic point of view.

the BER is constant and equal to the worst possible BER, 0.5. This shows the importance of equalization in communication systems. The purple curve is the BER of a system using a very popular modulation scheme, known as OFDM, on multipath channels. OFDM allows easy equalization in the frequency domain but requires a known training sequence to be transmitted in order to estimate the channel impulse response first. OFDM also uses cyclic prefix extension. Finally, the blue curve is the BER of the multi-channel blind equalization system described in this chapter with $N = 4$ receiving antennas. The multipath channels used in the simulations are 10-tap Rayleigh channels. For OFDM and the multi-channel blind equalization system, the packet length $L = 128$ and the cyclic prefix length $L_{cp} = 12$. OFDM uses a complete block for the training sequence.

Fig. 5.4b shows the average relative estimation error on the channels for OFDM and for the multi-channel blind equalizer. The parameters are as in Fig. 5.4a.

In both figures, each data point is the result of an averaging over 7000 blocks, each time with new random symbols to be transmitted and new random Rayleigh channels.

Fig. 5.4a shows that the multi-channel blind equalization system is able to outperform OFDM when the SNR is sufficiently high. At low SNR however, the performance decreases as already observed in Chapter 3. This is also visible on the average relative channel estimation error depicted in Fig. 5.4b.

It is important to note that the comparison with OFDM needs to be nuanced. Indeed, because the two systems are so much different (i.e., one is using a single receiving antenna and a training sequence while the other is using 4 receiving antennas and no training sequence), the comparison might no be totally fair. Comparison with a multiple-antenna OFDM system would be fairer as it would allow both systems to benefit from diversity.

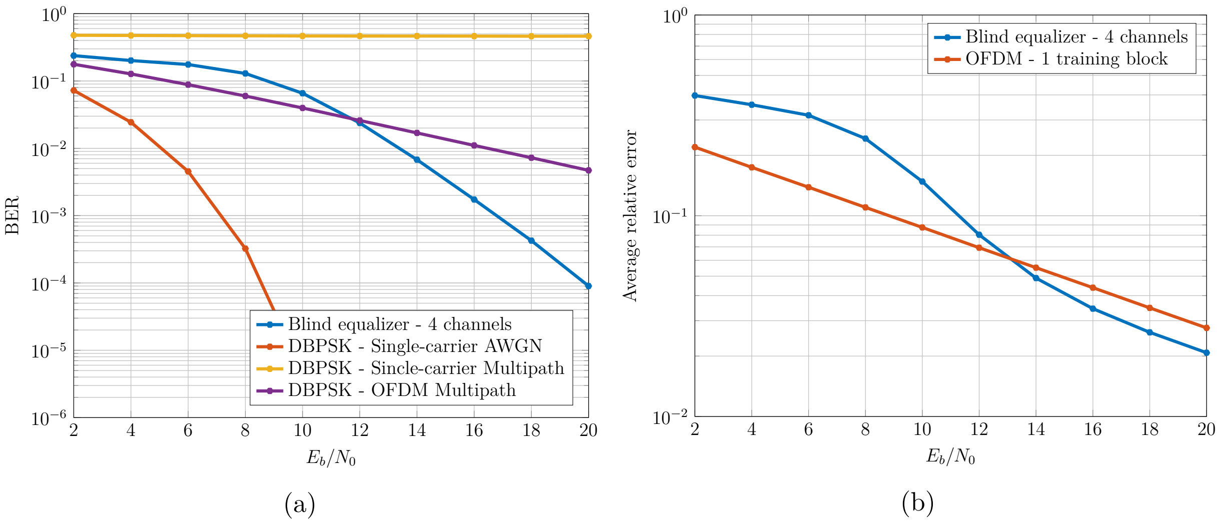


Figure 5.4: (a) Bit-error-rate (BER) curves against the SNR for several systems and channels configuration. (b) Comparison of the average relative estimation error of the channel(s) for both systems.

Chapter 6

Application to geophysical imaging

In this chapter, an application of noise imaging in the context of geophysics is described. Geophysical imaging refers to a set of techniques that can be used to discover the physical properties of the Earth's subsurface. Examples of such physical properties include electrical resistivity, wave velocity, etc. The final objective of seismic imaging is to obtain a map of the different constituents in the subsurface.

Sec. 6.1 first starts by a very short introduction to the use of deconvolution in seismic imaging and motivates the use of blind deconvolution in the context of *seismic-while-drilling* experiments. Next, Sec. 6.2 describes a numerical experiment that uses a seismic imaging toolbox to generates the filters. Unfortunately, this experiment further highlights the limitation of the hard constraint on the support size of the filters imposed by our model.

6.1 Background

Traditional seismic imaging uses a *controlled and known* seismic vibrations source, and infers the properties of the subsurface from recorded seismic traces. This is analogous to the known sequence of symbols transmitted to help equalization in the context of communication systems. Both the source and the geophones (i.e. devices measuring seismic vibrations) can be located either in the subsurface or on the surface. Appendix B shows how the recorded seismic traces can be modeled as linear convolutions between the seismic source signal and the subsurface *Green's functions*. In the context of geophysics, Green's functions can be thought of as the *earth's impulse responses*[1]. For imaging purposes, one is thus interested in recovering the subsurface Green's functions from the recorded seismic traces

$$p_n(t) = (g_n * s)(t)$$

where $g_n(t)$ is the Green's function sampled at the location of receiver n and $s(t)$ is the seismic source signature. When the seismic source signature is known, this can be achieved by performing a simple deterministic deconvolution.

In some situation however, the environment might contain an additional *unknown noisy seismic source*. This is for example the case on drilling sites, where the drilling operation generates so-called *drill-bit noise*. In this context of *seismic-while-drilling* (SWD) imaging, the recorded seismic traces thus contain mixed contributions from both the controlled source and the noisy

[1]As shown in Appendix B, it should be noted that there is in fact a unique Green's function that is sampled at several receivers locations, thus giving one earth's impulse response by receiver.

source convolved with the Green's functions

$$p_n(t) = (g_n * s)(t) + (g_n * b)(t)$$

where $b(t)$ is the drill-bit noise. In [4], the authors propose a deblending procedure based on independent component analysis to separate both contributions. The contribution corresponding to the controlled and known source can then be used to perform imaging. Another approach, proposed in [16] consists in getting rid of the controlled source and using the drill-bit noise only to perform imaging. In this case, the recorded seismic traces are given by

$$p_n(t) = (g_n * b)(t).$$

The proposed method is however not yet blind as the drill-bit signal is measured by a sensor before further processing the recorded seismic traces. In the very recent [3], the authors propose a multi-channel blind deconvolution algorithm based on a least-squares fit of the cross-correlations of the recorded seismic traces. This algorithm, called "Focused Blind Deconvolution", is able to recover the cross-correlated Green's functions (i.e., *interferometric* Green's functions) provided that the Green's functions are sufficiently *short* compared to the recording length and that the recorded seismic traces are sufficiently *diverse*.

In this context of seismic-while-drilling imaging, the next section discusses the applicability of the multi-channel blind deconvolution algorithm described in Chapter 3 to directly recover Green's functions.

6.2 Numerical experiments

This section describes a numerical experiment in the context of blind Green's functions retrieval, using the multi-channel blind deconvolution algorithm described in Chapter 3. The seismic experiments were carried out using PySIT, a Seismic Imaging Toolbox for Python [10].

The subsurface wave velocity model and the source/receivers configuration used in the numerical experiment are given in Fig. 6.1. As is common in geophysics, the y-axis represents the depth in the subsurface and points downwards (i.e., the bottom of the figure is the deepest part of the subsurface). The red triangles represent the receivers positions while the red cross is the drill-bit position. The number of receivers has been fixed to 10. The background velocity is in dark gray. This model only contains two horizontal reflectors, visible as perturbations of the background velocity model in Fig. 6.1.

Given the observed convolutions

$$p_n(t) = (g_n * b)(t), \qquad n \in [N]$$

the objective is to recover both the Green's function sampled at the receiver locations $\{g_n(t)\}_{n=0}^{N-1}$ and the drill-bit noise $b(t)$. To simplify, the drill-bit noise $b(t)$ is assumed to be white and Gaussian, even though it is not the case in practice [3]. The convolution in this last equation is not circular. Even if a linear convolution can reasonably be approximated by a circular convolution if the support of $g_n(t)$ is sufficiently short, this issue will be avoided here by replacing the linear convolution by a circular convolution in the process generating the seismic traces $p_n(t)$.

The generation of the seismic traces $\{p_n(t)\}_{n=0}^{N-1}$ works as follows:

1. Extract the Green's function sampled at the N receivers locations using PySIT and the subsurface model given in Fig. 6.1. The impulse responses associated with two receivers are given in Fig. 6.2.

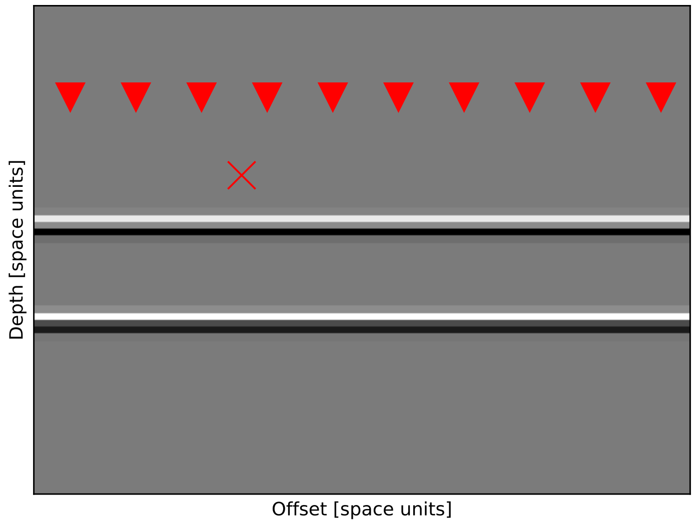


Figure 6.1: Subsurface model used in the numerical experiments. In geophysics, the y-axis represents the depth and points downwards, i.e., the bottom of the figure corresponds to the deepest part of the subsurface. The red triangles represent the receivers positions while the red cross is the drill-bit position. This model only contains two horizontal reflectors, visible as perturbations of the background velocity model.

2. The extracted impulse responses are then downsampled and thresholded to only keep values above a given threshold.

3. The downsampled and thresholded impulse responses are then truncated to their K first non-zero samples. The result of these operations on the impulse responses of Fig. 6.2 are given in Fig. 6.3 for a downsampling factor of 10 and $K = 70$. The resulting length-K vectors are noted $\boldsymbol{g}_n$.

4. The drill bit noise is generated as a length-L Gaussian random vectors with i.i.d entries and is noted $\boldsymbol{b}$.

5. Finally, the seismic traces are length-L vectors $\boldsymbol{p}_n$ generated as follows

$$\boldsymbol{p}_n = \boldsymbol{b} \circledast (\boldsymbol{C}\boldsymbol{g}_n)$$

where $\boldsymbol{C} = \boldsymbol{I}_{L\times K}$. Note that the convolution here is circular.

This way of generating the seismic traces is of course not realistic and far from the seismic-while-drilling experiments presented for example in [3]. It is in fact totally equivalent to the numerical experiments presented in Chapter 3 with a generative model for the filters $\boldsymbol{h}_n$ that is closer to the geophysical imaging context than random Gaussian filters with i.i.d entries. Nevertheless, this experiment can serve as a first test for the applicability of the multi-channel blind deconvolution algorithm in a seismic imaging context.

Results Using the above procedure to generate the seismic traces $\boldsymbol{p}_n$, the algorithm described in Chapter 3 always fails. Other configurations of source/receivers and other subsurface models (such as the Marmousi model, a more realistic model well known in the Geophysics community) were also tested but gave the same results.

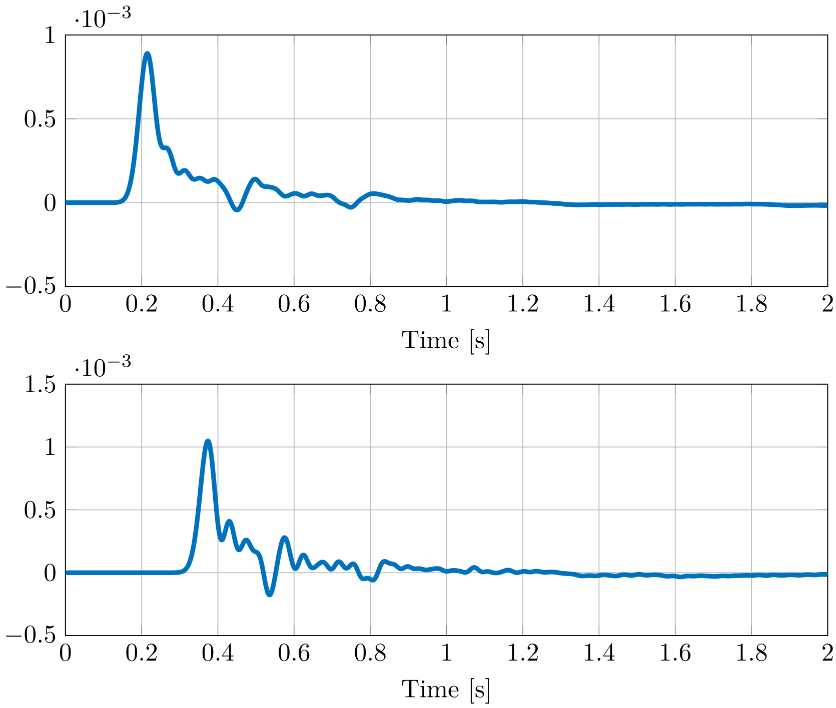


Figure 6.2: Impulse responses $g_1(t)$ and $g_4(t)$ (corresponding to the first and fourth receiver) extracted with PySIT.

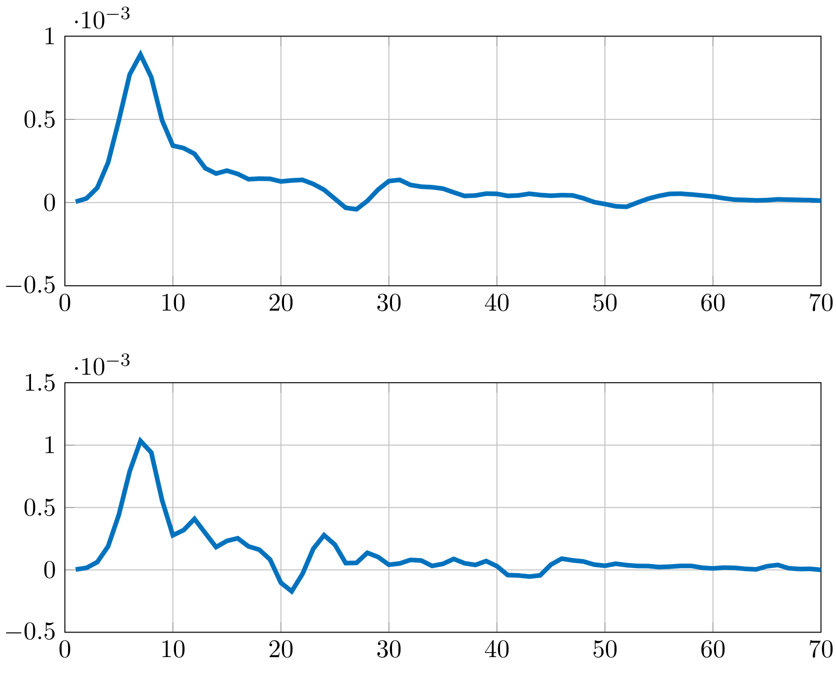


Figure 6.3: Truncated and downsampled version of the impulse responses given in Fig. 6.2.

Compared to the channels model used in the numerical experiment of Chapters 3 and 5, several factors might explain the failure of the algorithm in this context:

- **smoothness**: the channels $\boldsymbol{g}_n$ are smoother than the random Gaussian channels used in the preceding chapters. This comes from the fact that the entries of the channels $\boldsymbol{g}_n$ are no longer independent and identically distributed. Indeed, successive samples are clearly correlated.
- **lack of diversity**: as is apparent from Fig. 6.3, the impulse responses of the different receivers are very similar. This could lead to a reduced effective oversampling factor or even to roots shared by all the channels, as discussed in Chapter 4.
- **support completeness**: Fig. 6.3 shows that the edges of the support are very close to zero (although not strictly equal to zero). Strictly speaking, the support is thus complete. However, having values very close to zero on the edges of the support could make the situation ill-posed (or "almost ill-posed").
- **energy concentration**: Fig. 6.3 also shows that most of the energy of the channels is concentrated in a few elements (in addition of being concentrated in the K first non-zero elements). Note that this issue is in fact closely related to the previous one.

Numerical experiments with random Gaussian channels smoothened by a Gaussian kernel of given standard deviation suggest that smoothness is not an issue. Another experiment consists in repeating the above one with $\boldsymbol{g}_n$ replaced by $\boldsymbol{g}_n - \bar{\boldsymbol{g}}_n$, where $\bar{\boldsymbol{g}}_n$ is the mean value of $\boldsymbol{g}_n$. This makes the support complete and reduces the energy concentration issue while keeping the smoothness and diversity unchanged. In this case, the algorithm sometimes succeed to recover $\boldsymbol{g}_n - \bar{\boldsymbol{g}}_n$. This further indicates that smoothness is not an issue and suggests that the lack of diversity is not the main issue here neither. The main issues thus seems to be support completeness and energy concentration.

As discussed at the end of Chapter 3 and in Chapter 4, the support completeness issue is intrinsic to the hard constraint on the support size imposed by our model. To get rid of this issue would then require to move away from this model and to find another way of favoring short filters in the recovery. This will be briefly discussed in the next and final chapter.

Chapter 7

Conclusion

Chapters 3 and 5 demonstrated the effectiveness of the proposed model and algorithm to solve multi-channel blind deconvolution for Gaussian random channels with independent and identically distributed entries. In the noiseless case, the phase transition diagrams show that an oversampling factor greater than 1 is the only requirement for successful recovery. Except at very low SNR and for small oversampling factors, the recovery is also stable in the presence of noise.

However, the end of Chapter 3 and Chapter 4 tempered these positive results by highlighting the fact that ill-posedness actually depends on the *a priori* choice of K. If K is larger than the actual channels length, the end of the support is not complete and ill-posedness arises. If K is smaller than the actual channels length, the algorithm is not able to explain the observations with channels shorter than the actual channels. This hard constraint on the support size imposed by our model thus only makes sense in contexts where the support size is known or can be estimated reliably. The validity of this assumption in the context of Chapter 5 and the interesting question of estimating the support size given the observations was not discussed in this work.

The failure of the algorithm in the context of Green's function retrieval further emphasized this limitation. It suggested that having a complete support is not sufficient and that, in addition, the values on the edges of the support must not be too small. This is related to the interesting question of the energy concentration inside the support. In the context of Chapter 5 for example, the performance of the multi-channel blind equalizer for channels with non-uniform power delay profile could be studied. In the same context, Ricean channels could also be studied. In such channels, one tap is dominant and contains most of the energy. This physically corresponds a line-of-sight path of the electromagnetic wave being transmitted [21].

A possible solution to avoid this support completeness issue would be to relax the hard constraint on the support size and to find an other way of favoring short filters in the recovery. One possibility would be to minimize the following cost function

$$f(\boldsymbol{s}, \{\boldsymbol{h}_n\}_{n=0}^{N-1}) = \sum_{n=0}^{N-1} \|\boldsymbol{y}_n - \boldsymbol{s} \circledast \boldsymbol{w}_n\|_2^2 + \lambda \sum_{n=0}^{N-1} \sum_{l=0}^{L-1} \gamma(l) |w_n[l]|^2$$

where the second term is a penalty term. For weights $\gamma(l)$ increasing with l, this would favor short filters over longer ones. This approach resembles the one used in [3].

The impact of the channels similarities on the performance of the algorithm is yet another interesting question that was not studied in this work. As mentioned at the end of Chapter 3, similarities between channels are expected to reduce the effective oversampling factor. To characterize this effect, a measure of channels similarities and a way of controlling it must first be devised. Numerical experiments suggest that the condition number of the $K \times N$ matrix whose columns are the length-K channels $\boldsymbol{h}_n$ might be a good candidate to measure channels similarities.

Finally, in terms of computational speed, it should be noted that the performances of our algorithm do not match the great performances achieved in [2] for the single-input single-channel case[1] (although the algorithm is in fact the same). However, as optimizing the computational performances of the algorithm was not a central topic of this work, there is probably room for improvement there.

[1]This explains why the phase transition diagram presented in the numerical experiments of Chapter 3 is done at such a small scale.

Appendices

Appendix A

Calculation of $\mathcal{A}^*$

The adjoint of the linear operator $\mathcal{A}$ defined in (3.2) is the linear operator $\mathcal{A}^* : \mathbb{C}^{LN} \to \mathbb{R}^{L\times KN}$ that satisfies

$$\langle \boldsymbol{X}, \mathcal{A}^*(\boldsymbol{y})\rangle = \langle \mathcal{A}(\boldsymbol{X}), \boldsymbol{y}\rangle$$

for all $\boldsymbol{X} \in \mathbb{R}^{L\times KN}$ and $\boldsymbol{y} \in \mathbb{C}^{LN}$ where $\langle \cdot, \cdot \rangle$ denotes the trace inner product on the left hand side and the standard vector inner product on the right hand side. To ease notations, let

$$y_n[l] = y[nL + l], \qquad (l, n) \in [L] \times [N].$$

The right hand side of the above equation can be developed as

$$\begin{aligned}
\langle \mathcal{A}(\boldsymbol{X}), \boldsymbol{y}\rangle &= \sum_{n=0}^{N-1}\sum_{l=0}^{L-1} \overline{\langle \boldsymbol{f}_l \boldsymbol{\phi}_{l,n}^*, \boldsymbol{X}\rangle} y_n[l] \\
&= \sum_{n=0}^{N-1}\sum_{l=0}^{L-1} \langle \boldsymbol{X}, \boldsymbol{f}_l \boldsymbol{\phi}_{l,n}^*\rangle y_n[l] \\
&= \langle \boldsymbol{X}, \sum_{n=0}^{N-1}\sum_{l=0}^{L-1} y_n[l] \boldsymbol{f}_l \boldsymbol{\phi}_{l,n}^*\rangle.
\end{aligned}$$

By identification, the adjoint is thus given by

$$\mathcal{A}^*(y) = \sum_{n=0}^{N-1}\sum_{l=0}^{L-1} y_n[l] \boldsymbol{f}_l \boldsymbol{\phi}_{l,n}^*.$$

Developing this expression then gives

$$\begin{aligned}
\mathcal{A}^*(y) &= \sum_{n=0}^{N-1}\sum_{l=0}^{L-1} y_n[l] \boldsymbol{f}_l (\boldsymbol{c}_l^* \otimes \boldsymbol{e}_n^T) \\
&= \sum_{n=0}^{N-1} \left(\sum_{l=0}^{L-1} y_n[l] \boldsymbol{f}_l \boldsymbol{c}_l^*\right) \otimes \boldsymbol{e}_n^T \\
&= \sqrt{L} \sum_{n=0}^{N-1} \left(\sum_{l=0}^{L-1} y_n[l] \boldsymbol{f}_l \boldsymbol{f}_l^T C\right) \otimes \boldsymbol{e}_n^T.
\end{aligned}$$

For $N = 1$, this is consistent with the adjoint computed for the single-channel case [13, equation (3.7)]. Finally, the adjoint can be simplied to

$$\mathcal{A}^*(y) = \sum_{n=1}^{N} (\boldsymbol{F}^* \operatorname{diag}(\boldsymbol{y}_n) \overline{\boldsymbol{FC}}) \otimes \boldsymbol{e}_n^T.$$

Appendix B

From the wave equation to convolution

Let $p(\boldsymbol{x};t)$ denote the pressure at position $\boldsymbol{x} \in \mathbb{R}^3$ in the subsurface and time instant t. The evolution of $p(\boldsymbol{x};t)$ through space and time is ruled by the wave equation

$$\frac{1}{c^2(\boldsymbol{x})}\frac{\partial^2 p(\boldsymbol{x};t)}{\partial t^2} - \nabla^2 p(\boldsymbol{x};t) = f(\boldsymbol{x};t) \tag{B.1}$$

where $c(\boldsymbol{x})$ denotes the wave velocity at position $\boldsymbol{x}$, ∇^2 is the Laplacian operator and $f(\boldsymbol{x};t)$ is the seismic source at position $\boldsymbol{x}$ and time instant t. The Green's function $G(\boldsymbol{x},\boldsymbol{x}_s;t)$ is the solution of the wave equation when $f(\boldsymbol{x},t) = \delta(t)\delta(\boldsymbol{x}-\boldsymbol{x}_s)$, that is

$$\frac{1}{c^2(\boldsymbol{x})}\frac{\partial^2 G(\boldsymbol{x},\boldsymbol{x}_s;t)}{\partial t^2} - \nabla^2 G(\boldsymbol{x},\boldsymbol{x}_s;t) = \delta(t)\delta(\boldsymbol{x}-\boldsymbol{x}_s).$$

By linearity of the wave equation, the solution of the general wave equation (B.1) is then given by

$$p(\boldsymbol{x};t) = \int_{\mathbb{R}^3}\int G(\boldsymbol{x},\boldsymbol{y};t-\tau)f(\boldsymbol{y};\tau)\,\mathrm{d}\tau\,\mathrm{d}\boldsymbol{y}.$$

Assuming now that the seismic source is a point source located at $\boldsymbol{x}_s$ and producing a waveform $s(t)$, the pressure field is given by

$$p(\boldsymbol{x};t) = \int G(\boldsymbol{x},\boldsymbol{x}_s;t-\tau)s(\tau)\,\mathrm{d}\tau.$$

In other words, when the seismic source is a point source located at $\boldsymbol{x}_s$, the pressure field $p(\boldsymbol{x},t)$ at any location $\boldsymbol{x}$ and any time instant t can be written as a convolution between $s(t)$ and the Green's function associated to the medium wave velocity $c(\boldsymbol{x})$

$$p(\boldsymbol{x};t) = G(\boldsymbol{x},\boldsymbol{x}_s;t) * s(t).$$

For a geophone located at position $\boldsymbol{x}_n$, the recorded seismic trace would then be given by

$$p(\boldsymbol{x}_n;t) = G(\boldsymbol{x}_n,\boldsymbol{x}_s;t) * s(t)$$

or, in an abbreviated form,

$$p_n(t) = (g_n * s)(t).$$

It should be noted that the convolution in this last equation is not circular. However, as mentioned in Chapter 2, a linear convolution can reasonably be approximated by a circular convolution if the support of the filter is sufficiently short.

Bibliography

[1] Ali Ahmed and Laurent Demanet. Leveraging diversity and sparsity in blind deconvolution. *CoRR*, abs/1610.06098, 2016.

[2] Ali Ahmed, Benjamin Recht, and Justin K. Romberg. Blind deconvolution using convex programming. *CoRR*, abs/1211.5608, 2012.

[3] Pawan Bharadwaj, Laurent Demanet, and Aimé Fournier. Focused blind deconvolution of interferometric green's functions. 2018.

[4] Pawan Bharadwaj, Laurent Demanet, Aimé Fournier, et al. Deblending random seismic sources via independent component analysis. In *2017 SEG International Exposition and Annual Meeting*. Society of Exploration Geophysicists, 2017.

[5] Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, Feb 2003.

[6] Samuel Burer and Renato D.C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, Jul 2005.

[7] J.-F. Cai, E. J. Candes, and Z. Shen. A Singular Value Thresholding Algorithm for Matrix Completion. *ArXiv e-prints*, October 2008.

[8] J. E. A. Dunnage. The number of real zeros of a class of random algebraic polynomials. *Proceedings of the London Mathematical Society*, s3-18(3):439–460, 1968.

[9] Josselin Garnier and George Papanicolaou. *Passive Imaging with Ambient Noise*. Cambridge University Press, New York, NY, USA, 1st edition, 2016.

[10] R.J. Hewett, L. Demanet, and the PySIT Team. PySIT: Python seismic imaging toolbox v0.5, 2013. Release 0.5.

[11] S. M. Jefferies and J. C. Christou. Restoration of Astronomical Images by Iterative Blind Deconvolution. , 415:862, October 1993.

[12] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding blind deconvolution algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2354–2367, Dec 2011.

[13] Xiaodong Li, Shuyang Ling, Thomas Strohmer, and Ke Wei. Rapid, robust, and reliable blind deconvolution via nonconvex optimization. *CoRR*, abs/1606.04933, 2016.

[14] O. Michailovich and A. Tannenbaum. Blind deconvolution of medical ultrasound images: A parametric inverse filtering approach. *IEEE Transactions on Image Processing*, 16(12):3005–3019, Dec 2007.

[15] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[16] James W. Rector and Bruce P. Marion. The use of drill-bit energy as a downhole seismic source. *Geophysics*, 56(5):628, 1991.

[17] Justin Romberg, Ning Tian, and Karim Sabra. Multichannel blind deconvolution using low rank recovery, 2013.

[18] M. Schmidt. minfunc: unconstrained differentiable multivariate optimization in matlab. `http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html`, 2005.

[19] T. Strohmer. Four short stories about Toeplitz matrix calculations. *ArXiv Mathematics e-prints*, October 2000.

[20] Ning Tian, Sung-Hoon Byun, Karim Sabra, and Justin Romberg. Multichannel myopic deconvolution in underwater acoustic channels via low-rank recovery. *The Journal of the Acoustical Society of America*, 141(5):3337–3348, 2017.

[21] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication.* Cambridge University Press, 2005.

[22] Xiaodong Wang and H. V. Poor. Blind equalization and multiuser detection in dispersive cdma channels. *IEEE Transactions on Communications*, 46(1):91–103, Jan 1998.

[23] Guanghan Xu, Hui Liu, Lang Tong, and T. Kailath. A least-squares approach to blind channel identification. *IEEE Transactions on Signal Processing*, 43(12):2982–2993, Dec 1995.

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve www.uclouvain.be/epl