KU LEUVEN

ARENBERG DOCTORAL SCHOOL
FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING

Solving inverse problems in room acoustics using physical models, sparse regularization and numerical optimization

Niccolò Antonello

Supervisor:
Prof. dr. ir. Toon van Waterschoot
Co-supervisors:
Prof. dr. ir. Marc Moonen
Prof. dr. Patrick A. Naylor
(Imperial College London)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Technology (PhD)

August 2018

Niccolò ANTONELLO

Examination committee:
Prof. dr. ir. Bert Lauwers, chair
Prof. dr. ir. Toon van Waterschoot, supervisor
Prof. dr. ir. Marc Moonen, co-supervisor
Prof. dr. Patrick A. Naylor, co-supervisor (Imperial College London)
Prof. dr. ir. Lieven De Lathauwer
Prof. dr. ir. Wim Desmet
Dr. ir. Bram Cornelis (Siemens PLM Software)
Prof. dr. ir. Panagiotis Patrinos (KU Leuven)
Prof. dr. Andreas Jakobsson (Lund University)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Technology

August 2018

© 2018 KU Leuven – Faculty of Engineering Technology
Uitgegeven in eigen beheer, Niccolò Antonello, Kasteelpark Arenberg 10, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Preface

This thesis is the result of more than five years[1] of roaming in the world of signal processing, acoustics and optimization. It has been a pleasant journey, despite its difficulties, that taught me a lot. I am glad that after all there was a "path" that connected the various stages of this roaming, since very often I have felt quite lost. Eventually, I did not completely lose my way thanks to the guide that many people kindly offered me.

Firstly, I would like to thank my supervisor Toon van Waterschoot who has always supported me during this project, that he initially formulated and later wisely guided. Despite during my first interview he found out that I did not know what a least squares problem was, he still decided to trust me and grant me this great opportunity. Later on, I used (regularized) least squares very often, almost obsessively, to convince him that I grasped the concept. Perhaps I have failed to do so, since he offered me to study it for some more time even after the end of my PhD. I cannot express properly the gratitude that I feel for all the things Toon has done for me and for my life. I feel really lucky to have met him and I am glad that I have the opportunity to continue working together.

Marc Moonen also helped me a lot during my PhD and I would really like to thank him. He always read my papers with great care and gave detailed, sometimes sarcastic, comments. I have always appreciated his meticulousness and his black humor and I hope I have learned both of them, at least a little bit!

I would also like to thank Patrick A. Naylor for his contributions and for hosting me in London at the Imperial College. I had a great time in London and I enjoyed very much helping the organization of the Summer Science Exhibition at the Royal Society.

[1] Fortunately, it is not "fifty years" [1–3].

Enzo De Sena has been a great collaborator and office mate. I was really sad when he left KU Leuven but at the same time very happy for his successful career. I still miss the endless discussions we had during late office hours full of creative ideas and crazy stories of his. He still visits Leuven from time to time and I am always happy when he does since it feels like he had never left.

The collaboration with Lorenzo Stella began with very humble objectives: a small git repository where he was supposed to help me implementing a simplified version of an algorithm he developed together with Andreas Themelis and Panos Patrinos. Eventually, the repository rapidly split into four Julia packages! Lorenzo has been extremely helpful: despite he was writing his PhD thesis in the other side of the globe and later working full-time and dealing with early fatherhood, he dedicated so much time to endless online chats full of mathematics and coding issues. I want to thank him for all the things I have learned from him and for the time he dedicated to our project.

I wish to thank Panos Patrinos as well. He has always been helpful and available for intensive meetings full of mathematical formulas. I have to admit that often it has been difficult for me to follow him, similarly to what happened during my last holidays in Greece where I had to follow an alpine guide, named Panos as well, on the top of a rock of Meteora. In both cases it was quite rewarding to follow them.

Besides my collaborators, I would like to thank the chair Bert Lauwers and the members of the examination committee Lieven De Lathauwer, Wim Desmet, Andreas Jakobsson and Bram Cornelis for their feedback on this thesis and the interesting discussion of my preliminary defense.

I would like to thank Università degli Studi di Padova and the Technical University of Denmark (DTU) for the education they provided me. I wish to thank the European Commission and the Marie Skłodowska-Curie Actions for the funding and to all of the people of the DREAMS consortium for organizing the various schools that took place during the first three years of my PhD, in particular Aldona and my fellow PhD researchers Adam, Adel, Ante, Benjamin, Clément, Giacomo, Mathieu, Neo, Pablo (malo será! Gracias a tu familia también Bego e Olivia) and Thomas.

Thanks to all the (ex-)colleges and friends of DSP team, STADIUS, OPTEC and Group T: Ahmed, Alexander, Amin (you have to come back and visit Venice without rain!), Andreas, Bruno, David, Duowei, Fernadno, Filippos, Gert, Giacomo, Giuliano (master of LaTeX& Ti*k*Z), Hanne, Hassan, Ida, Jasper (and spanish girl), Jeroen, Joe, John, Johnny, Jorge, Maja, Marijn, Matthew, Mina, Mohit, Mojtaba, Neetha, Paschalis, Pepe, Puya, Randy (thanks for the English tips and the possibility of a career in the punk album art covers),

Robbe, Rodirigo, Rodolfo, Thomas, Wouter B. and Wouter L.. Thanks to all the office mates at Imperial: Alastair, Christine, Costas, Felicia, Hamza, John, Leo, Richard, Sina and Wei. Thanks to the Salcazzo parties crew and friends in Belgium: Alice, Ana, Bahar, Carlo, Dan, Daniele, Deniz, Ece, Enrico, Federico, Francesco, George, Ignazio, Iman, Juan, Leo, Maria, Marta, Nina and Oreste. Thanks to Medina Mansion and the London crew: Ale M., Diana & Rob, Elia Blackbird, Francesca, Max, Migue and in particular to the combination of Stanghe and Coco that got me in Medina. Thanks to all the friends I have met in Sondrio, in particular to Rosanna, Popi, nonna Anna (best 96 years old I have ever met) and signor Cielo for having accepted me in their family. Thanks to all of my friends in Casteo e dintorni, nonostante ci si veda e senta poco, ogni volta che torno mi fate sentire sempre a casa: Ale, Alessia, Annet, Burri, Chiara, Curto, Dade, Damino, Dido, Dure, Elia, Elisa, Eva, Feda, Fiamma, Giada, Gian, Giulia, Jacopo, Lele, Leo, Massimo, Milan, Monica, Piero, Seba, Sleepy, Stanghe (anche se non siamo piú amici!), Vale, Vera, Zampe.

Thanks to Frank Zappa for the beautiful soundtrack, to my old Dell "KRELIS" for resisting to the many nights of lonely Julia computing and to Alejandro Jodorowsky for the inspiring movies I watched the days before the preliminary defense.

Finally, a really special thanks goes to my family. Without the help, the wisdom and the love della mamma Maria e del papi Matteo I would not be the person that I am and I would have never accomplished this. Thank you for all the phone calls always full of love and kind words della mamma and the "studia!!" and "lavora!!" del papi. Thanks to my brother Jacopo, to Angeliki and to my sister Elettra ciki-bimba. You are the best I could ever wish for.

Last, but definitively not least, I would like to thank Silvia. Thanks to all you have done for me, for being so brave, for all the happiness, the laughter and the love you gave me.

Niccolò Antonello
August 2018

Abstract

Reverberation consists of a complex acoustic phenomenon that occurs inside rooms. Many audio signal processing methods, addressing source localization, signal enhancement and other tasks, often assume absence of reverberation. Consequently, reverberant environments are considered challenging as state-of-the-art methods can perform poorly. The acoustics of a room can be described using a variety of mathematical models, among which, physical models are the most complete and accurate.

The use of physical models in audio signal processing methods is often non-trivial since it can lead to ill-posed inverse problems. These inverse problems require proper regularization to achieve meaningful results and involve the solution of computationally intensive large-scale optimization problems. Recently, however, sparse regularization has been applied successfully to inverse problems arising in different scientific areas. The increased computational power of modern computers and the development of new efficient optimization algorithms makes it possible to tackle inverse problems also in the context of room acoustics.

This thesis explores this novel framework by applying the latest sparse regularization methods and optimization algorithms to develop new audio signal processing methods that are more robust against reverberation and noise. The inverse problems these methods face naturally lead to joint formulations of multiple tasks that are typically treated separately enabling, *e.g.*, simultaneous source localization, sound field reconstruction and dereverberation.

The first part of the thesis is dedicated to optimization algorithms particularly suited for the type of inverse problems under consideration. These are called proximal gradient (PG) algorithms and are capable of minimizing the nonsmooth cost functions that typically arise in optimization problems involving sparse regularization. In addition, PG algorithms can be accelerated using quasi-Newton methods and combined with matrix-free operators allowing them to tackle large-scale problems and reduce the computational burden of many signal

processing methods.

The second part of the thesis addresses acoustic modeling and focuses on sweeping echoes, a particular physical phenomenon that typically does not occur in regular rooms. This phenomenon is studied and attributed to the idealized cuboid geometries employed by many acoustic models. A variation of the image method (IM), a popular acoustic model that is usually restricted to rectangular rooms, is proposed to produce perceptually realistic simulations without the presence of sweeping echos.

The third part of the thesis covers inverse problems that utilize the finite-difference time-domain (FDTD) method. This method, which aims at solving the wave equation numerically, requires precise knowledge of the room geometry and of the acoustic impedances that model the acoustic properties of the walls of the room. Firstly, the problem of acoustic impedance estimation is addressed, also resulting in an inverse problem. Secondly, source localization is jointly formulated with source reconstruction by proposing a two-step method. Once the original sound source location is identified by solving an inverse problem that exploits the spatial sparsity of the sound sources in the room, a reconstruction of the original source signal is performed. Finally, the use of the FDTD method for multi-zone sound field control is envisaged. Solving an inverse problem regularized with spatial sparsity allows to optimally control and place a set of loudspeakers to reproduce a specific sound field inside a highly reverberant room while keeping part of the room silent.

The fourth part of the thesis is dedicated to the use of wave decomposition models, that can represent the sound field of a room only in a limited portion of space but, contrary to the FDTD method, without the knowledge of the room geometry and of the acoustic impedances. Here, firstly the problem of room impulse response (RIR) interpolation is addressed. Since measuring RIRs in a wide space is time-consuming, an effective interpolation of these measurements is often useful in a number of applications. It is shown that the combination of spatio-temporal sparse regularization with a time-domain wave decomposition model can substantially reduce the number of microphones needed to perform RIR interpolation. The fourth part is concluded by the description of a novel method capable of performing source localization and dereverberation jointly. This is achieved by performing a sound field interpolation as well, performed through the solution of an inverse problem that employs a particular combination of a wave decomposition model with sparse regularization. Once a proper sound field interpolation is achieved, the direction of arrival (DOA) of a moving sound source and dereverberated signals can be obtained simultaneously in challenging acoustic environments.

Korte Inhoud

Nagalm of reverberatie is een complex akoestisch fenomeen dat zich voordoet in gesloten ruimtes. In heel wat audiosignaalverwerkingsmethodes voor o.a. bronlokalisatie en signaalverbetering wordt doorgaans verondersteld dat er geen nagalm aanwezig is. Bijgevolg vormen reverberante omgevingen een uitdaging waarvoor state-of-the-art-methodes geen performante oplossing bieden. De akoestiek van een ruimte kan worden beschreven met een breed gamma wiskundige modellen waaronder fysische modellen de meest volledige en nauwkeurige beschrijving bieden.

Het gebruik van fysische modellen in audiosignaalverwerkingsmethodes is vaak niet-triviaal gezien dit kan leiden tot slecht geconditioneerde inverse problemen. Dergelijke inverse problemen vereisen een gepaste regularisatie om betekenisvolle resultaten te bekomen en leiden tot het oplossen van rekenintensieve groteschaaloptimalisatieproblemen. Recent werd een spaarse regularisatie succesvol toegepast bij inverse problemen in verschillende wetenschapsdomeinen. De toegenomen rekenkracht van hedendaagse computers en de ontwikkeling van nieuwe, efficiënte optimalisatiealgoritmes openen een perspectief om ook inverse problemen in de context van ruimteakoestiek aan te pakken. Dit perspectief wordt in dit proefschrift verkend door de meest recente spaarse regularisatiemethodes en optimalisatiealgoritmes toe te passen bij de ontwikkeling van nieuwe audiosignaalverwerkingsmethodes die een hogere robuustheid tegen nagalm en ruis vertonen. Deze methodes leiden op een natuurlijke manier tot inverse problemen waarin meerdere taken, die traditioneel apart worden aangepakt, gezamenlijk geformuleerd en opgelost worden, zoals gelijktijdige bronlokalisatie, geluidsveldreconstructie en dereverberatie.

Het eerste deel van het proefschrift is gewijd aan optimalisatiealgoritmes die in het bijzonder geschikt zijn voor het soort inverse problemen dat hier beschouwd wordt. Deze zogenaamde proximalegradiëntalgoritmes (PG) laten toe om de niet-gladde functies die typisch voortkomen uit optimalisatieproblemen met spaarse regularisatie te minimaliseren. Bovendien kunnen PG-algoritmes

versneld worden met behulp van quasi-Newtonmethodes en gecombineerd worden met matrixvrije operatoren, waardoor deze algoritmes geschikt worden voor groteschaalproblemen en signaalverwerkingsmethodes die traditioneel erg rekenintensief zijn.

Het tweede deel van het proefschrift handelt over akoestische modellering met een focus op zogenaamde sweeping echo's, een bijzonder akoestisch fenomeen dat zich in realistische ruimtes doorgaans niet voordoet. Dit fenomeen wordt bestudeerd en verklaard vanuit de geïdealiseerde rechthoekige geometrie die in veel akoestische modellen wordt gebruikt. Een nieuwe variant van de spiegelbronmethode (IM), een veelgebruikt akoestisch model dat doorgaans enkel voor rechthoekige ruimtes kan worden toegepast, wordt voorgesteld om perceptueel realistische simulaties te maken zonder sweeping echo's.

Het derde deel van het proefschrift behandelt inverse problemen op basis van de eindige-differenties-in-het-tijdsdomeinmethode (FDTD). Deze methode beoogt een numerieke oplossing van de golfvergelijking en vereist een precieze kennis van de geometrie van de ruimte en van de akoestische impedanties die de akoestische eigenschappen van de muren van de ruimte modelleren. Ten eerste wordt het probleem van akoestische-impedantieschatting behandeld, hetwelk resulteert in een invers probleem. Ten tweede worden de problemen van bronlokalisatie en bronreconstructie gezamenlijk geformuleerd door een tweestappenmethode voor te stellen. De locatie van de geluidsbron wordt geschat door een invers probleem op te lossen waarin de spatiale spaarsheid van de geluidsbronnen in de ruimte wordt gebruikt, waarna een reconstructie van het oorspronkelijke geluidssignaal wordt uitgevoerd. Ten laatste wordt het gebruik van de FDTD-methode voor multi-zone geluidsveldbeheersing onderzocht. Het oplossen van een invers probleem met spatiaal spaarse regularisatie laat toe om een verzameling luidsprekers optimaal te plaatsen en aan te sturen om een specifiek geluidsveld te reproduceren in een sterk reverberante ruimte en tegelijk een ander deel van de ruimte stil te houden.

Het vierde deel van het proefschrift is gewijd aan het gebruik van golfontbindingsmodellen die, in tegenstelling tot de FDTD-methode, het geluidsveld in een beperkt deel van een ruimte kunnen voorstellen zonder kennis van de ruimtegeometrie en akoestische impedanties. In eerste instantie wordt de interpolatie van kamerimpulsresponsen (RIRs) onderzocht. Aangezien het opmeten van RIRs in een grote ruimte tijdrovend is, kan een effectieve interpolatie van dergelijke metingen nuttig zijn in verscheidene toepassingen. Er wordt aangetoond dat de combinatie van spatiotemporeel spaarse regularisatie met een tijdsdomein golfontbindingsmodel het aantal microfoons nodig voor RIR-interpolatie aanzienlijk kan reduceren. Dit vierde deel wordt afgerond met de beschrijving van een nieuwe methode waarmee bronlokalisatie en dereverberatie gelijktijdig kunnen worden uitgevoerd. Dit wordt bekomen via

een geluidsveldinterpolatie die wordt uitgevoerd door het oplossen van een invers probleem dat bestaat uit een specifieke combinatie van een golfontbindingsmodel met een spaarse regularisatie. Van zodra een degelijke geluidsveldinterpolatie is bereikt, kan de aankomstrichting (DOA) van een bewegende geluidsbron alsook gedereverbereerde signalen worden bekomen in uitdagende akoestische omgevingen.

Glossary

Acronyms

ADA	adaptive sparse MCLP-based speech dereverberation
ADeLFI	acoustic dereverberation and localization through field interpolation
ADMM	alternating direction method of multipliers
BC	boundary condition
BEM	boundary element method
CS	compressed sensing
DAG	directed acyclic graph
DCT	discrete cosine transform
DFT	discrete Fourier transform
DNN	deep neural network
DOA	direction of arrival
DR	Douglas-Rachford
DSP	digital signal processor

ECOS embedded conic solver

ESM equivalent source method

ESPRIT estimation of signal parameters via rotational invariance technique

FAOs forward-adjoint oracles

FBE forward-backward envelope

FBS forward-backward splitting

FDM finite-difference method

FDTD finite-difference time-domain

FEM finite element method

FFT fast Fourier transform

FIR finite impulse response

FISTA fast iterative shrinkage-thresholding algorithm

FPG fast proximal gradient

FVTD finite-volume time-domain

GN Gauss-Newton

GPU Graphics Processing Unit

HOA higher-order ambisonics

IC initial condition

IIR infinite impulse response

IM image method

IR impulse response

KCV K-fold cross validation

L-BFGS limited memory BFGS

LASSO least absolute shrinkage and selection operator

LS least square

LTI	linear time-invariant
MCLP	multi-channel linear prediction
MIMO	multiple-input multiple-output
MP	matching pursuit
MPC	model predictive control
MSE	mean squared error
MUSIC	MUltiple SIgnal Classification
NFFT	nonequispaced fast Fourier transform
NMSE	normalized mean squared error
NN	neural network
ODE	ordinary differential equation
PANOC	proximal averaged Newton-type algorithm for optimality conditions
PC	Pock-Chambolle algorithm
PCA	principal component analysis
PDE	partial differential equation
PESQ	perceptual evaluation of speech quality
PG	proximal gradient
PM	pressure matching
PWDM	plane wave decomposition method
RIM	randomized image method
RIR	room impulse response
SCS	splitting conic solver
SDM	spatial decomposition method
SDN	scattering delay network
SHDM	spherical harmonic decomposition method

SMARD	single- and multichannel audio recordings database
SNR	signal to noise ratio
SPL	sound pressure level
SQP	sequential quadratic programming
SRP-PHAT	steered response power-phase transform
SS	sweeping spectrum
SSF	sweeping spectrum flatness
sTESM	source-aware time-domain equivalent source method
STOI	short-time objective intelligibility
SVD	singular-value decomposition
TESM	time-domain equivalent source method
TOPS	test of orthogonality of projected subspaces
VMFB	variable metric forward-backward
WASN	wireless acoustic sensor network
WFS	wave-field synthesis
WOLA	weighted overlap-add

Mathematical notation

$\mathbb{N}$	set of natural numbers
$\mathbb{Z}$	set of integer numbers
$\mathbb{R}$	set of real numbers
$\mathbb{C}$	set of complex numbers
x	scalar
$\mathbf{x}$	column vector
$\mathbf{X}$	matrix

$\mathbf{0}$	null matrix or vector
$.^{\mathsf{T}}$	vector or matrix transpose
$.^{\mathsf{H}}$	vector or matrix conjugate transpose
$.^{-1}$	matrix inverse
$\|\cdot\|_0$	l_0-norm
$\|\cdot\|_1$	l_1-norm
$\|\cdot\|_2$	l_2-norm
$\|\cdot\|_F$	Frobenius norm
$\mathbf{vec}(\cdot)$	column-major vectorization operator
$\mathrm{rank}(\mathbf{X})$	rank of a matrix $\mathbf{X}$
$\approx$	approximately equal to
$\ll$	much less than
$\gg$	much greater than
$\forall$	for all
$\in$	set membership
$\notin$	set membership, is not an element of
$\emptyset$	empty set
$\subset$	subset
card	cardinality of a set
$\frac{\partial f}{\partial x}$	partial derivative of the function f with respect to x
∇f	gradient of the function f
$\nabla^2 f$	Hessian of the function f
$\triangle$	Laplace operator
$\mathrm{minimize}_{\mathbf{x}}\, f(\mathbf{x})$	minimize the function f over $\mathbf{x}$
$\mathrm{argmin}_{\mathbf{x}}\, f(\mathbf{x})$	argument of the minimum of the function f
e	Euler's number
c	speed of sound
F_s	sampling frequency

Contents

Preface i

Abstract v

Korte Inhoud vii

Glossary xi

Contents xvii

1 Introduction 1

1.1 Room acoustic models 2

1.1.1 Parametric models 3

1.1.2 Discretized PDE models 3

1.1.3 Reflection path models 5

1.1.4 Wave decomposition models 6

1.2 Room acoustics inverse problems 8

1.2.1 Source localization 10

1.2.2 Sound field reconstruction and dereverberation 10

1.2.3 Sound field control 12

1.2.4 Geometry and impedance estimation 12

1.3 Optimization Algorithms 13
1.4 Overview of the thesis 14

Part I Optimization Algorithms 19

2 Proximal gradient algorithms 21

2.1 Introduction 23
2.2 Modeling 26
2.2.1 Inverse problems 26
2.2.2 Convex and nonconvex problems 29
2.3 Proximal gradient algorithms 31
2.3.1 Proximal mappings 33
2.3.2 Proximal gradient method 36
2.3.3 Forward-backward envelope 38
2.3.4 Newton-type proximal gradient methods 39
2.4 Matrix-free optimization 42
2.4.1 Directed acyclic graphs 43
2.5 General problem formulation 48
2.5.1 Duality and smoothing 50
2.6 A high-level modeling language: StructuredOptimization 53
2.7 Conclusions 56

Part II Modeling 59

3 The randomized image method 61

3.1 Introduction 63
3.2 Background 65

3.2.1 The wave equation 65
3.2.2 Solution of the wave equation for rectangular rooms with rigid walls 66
3.2.3 Image method for a rectangular room 67
3.2.4 Finite-difference time-domain method 68
3.3 Sweeping Echoes in Acoustic Simulations with Rectangular Geometry 69
3.3.1 Sweeping echoes in the IM 70
3.3.2 Sweeping echoes in FDTD 70
3.3.3 Sweeping echo measure 71
3.4 Physical Basis of Sweeping Echoes in Perfectly Rectangular Rooms 73
3.4.1 Simplified case with three walls 74
3.4.2 Simplified case with four walls 76
3.4.3 General case 76
3.5 Room geometries with small out-of-square imperfections 79
3.6 Sweeping Echoes Removal from the Image Method 80
3.7 Effect of Strong Sweeping Echoes in Speech and Audio Processing Applications 83
3.7.1 Multi-Channel Linear Prediction of Reverberant Speech 84
3.7.2 Objective Evaluation of Reverberant Speech Quality . . 86
3.7.3 Pitch Estimation of Monophonic Reverberant Music . . 86
3.8 Conclusions and Future Work 87

Part III Inverse problems using the FDTD method 91

4 Impedance estimation using the FDTD method 93

4.1 Introduction 95
4.2 The finite difference time domain method 96

4.3 The optimization algorithm 99
4.3.1 The Adjoint Method 100
4.3.2 Tikhonov regularization 101
4.4 Simulation Results 103
4.4.1 Simulation Set-up 103
4.5 Conclusions 105

5 Source localization and signal reconstruction using the FDTD method 107

5.1 Introduction 109
5.2 The finite difference time domain method 110
5.3 Matrix formulation of FDTD 112
5.4 Source localization and reconstruction 114
5.4.1 Reconstruction of an impulse 114
5.4.2 Source localization 115
5.4.3 Reconstruction with known position 115
5.4.4 Reconstruction without initial conditions 117
5.5 Simulation results 117
5.6 Conclusions 118

6 Sound field control using the FDTD method 121

6.1 Introduction 123
6.2 The finite difference time domain 124
6.3 The optimization algorithm 126
6.3.1 Computation of $\mathbf{G}$ 127
6.3.2 Newton-type optimization 127
6.3.3 Regularization 128
6.3.4 Dark Zone Regularization 131

6.4 Simulation results 131
6.5 Conclusions and Future Work 134

Part IV Inverse problems using wave decomposition models 137

7 Room impulse response interpolation 139

7.1 Introduction 141
7.2 Plane wave decomposition method 145
7.3 Time-domain equivalent source method 146
7.4 The inverse problem 148
7.5 Optimization algorithm 151
7.6 Computation of the Jacobian 153
7.7 Simulation results 154
7.7.1 Choice of regularization parameter λ 156
7.7.2 Comparison between acoustic models 158
7.7.3 Analysis of weight signals 160
7.8 Experimental results 162
7.9 Conclusions 164

8 Dereverberation and source localization 167

8.1 Introduction 169
8.2 Acoustic models 172
8.2.1 Time domain equivalent source method 172
8.2.2 Plane wave decomposition method 172
8.3 The inverse problem 173
8.4 The ADeLFI algorithm 177
8.4.1 Optimization algorithm 177

8.4.2 Weighted overlap-add procedure 179
8.4.3 Tuning of parameter λ 181
8.4.4 DOA estimation and dereverberation 182
8.4.5 Frequency domain acoustic model 183
8.5 Results 184
8.5.1 Simulation results 184
8.5.2 Results using real measurements 189
8.6 Conclusions 192

9 Conclusion **193**
9.1 Suggestions for future research 197

A StructuredOptimization.jl **203**
A.1 Installation 205
A.2 Standard problem formulation 205
A.2.1 Unconstrained optimization 205
A.2.2 Constrained optimization 206
A.2.3 Using multiple variables 207
A.2.4 Limitations 208

B RIM.jl **211**
B.1 Installation 213
B.2 Usage 213
B.2.1 Changing default parameters with Keyword Arguments 214

Bibliography **217**

Curriculum Vitae **233**

Publication List **235**

Chapter 1

Introduction

Modern electronic devices such as personal computers and smartphones, are equipped with *digital signal processors (DSPs)* to perform a variety of functions. In particular, by combining DSPs with microphones or loudspeakers, various audio digital signal processing tasks such as sound source localization, signal enhancement and sound field reproduction can be accomplished and applied in a variety of contexts including hearing assistance, human-computer interaction, voice control and virtual/augmented reality.

Often, these tasks are performed in enclosed spaces such as rooms or offices. The acoustic environment of these spaces can strongly affect the effectiveness of the tasks performed by the DSP. In fact, in such environments, sound waves are reflected by walls and objects creating a complex acoustic phenomenon known as *reverberation*. Due to its complexity, many of the audio signal processing methods implemented in the DSPs neglect reverberation or use overly simplified *acoustic models*. As a consequence of this, in reverberant rooms these tasks are more challenging and the methods can perform poorly.

Recently, however, several new acoustic models have been developed that can indeed describe the acoustics of reverberant rooms more accurately. *Physical acoustic models* are among the most complete ones as they allow to describe the acoustic phenomenon, *i.e.*, the sound field in entire regions of space. The increased computational power of modern DSPs allows for the use of new methods that can incorporate more advanced acoustic models, thereby increasing robustness against reverberation. These new methods typically consist of solving *inverse problems*, which are generally difficult due to their ill-posed nature. Advances in the field of compressed sensing (CS) have shown that the inclusion of *sparse regularization* in inverse problems can lead to

meaningful solutions. Moreover, recently developed *optimization algorithms* can be used to solve inverse problems efficiently. The objectives of this thesis focus on exploring this new framework in audio signal processing tasks such as *sound source localization*, *sound field reproduction* and *signal enhancement (dereverberation)* in acoustically challenging environments such as reverberant and noisy rooms.

This new framework offers novel methods that could be employed in a broad range of devices such as smartphones, hearing aids, virtual and augmented reality simulators, home cinema and in-car audio systems. Most of the methods that are proposed in the thesis are currently limited to offline applications due to their high computational burden. This prevents their employment in many of the aforementioned devices, although their implementation is envisaged in the near future thanks to the advances in research and in computational power.

In this introduction, firstly an overview of the state-of-the-art acoustic models is given. The various inverse problems that are treated in the thesis are then described and a brief overview of the optimization algorithms that are utilized is given. The introduction ends with an overview of the chapters composing the thesis.

1.1 Room acoustic models

Wave propagation in a homogeneous medium is physically described by the *wave equation* which is a *linear partial differential equation (PDE)* [4]. When a prediction of the sound field inside a room is desired, one has to define the geometry of the room together with the acoustic properties of walls and objects that absorb and reflect the sound waves. This is achieved by defining another set of PDEs called *boundary conditions (BCs)*, where the acoustic properties of the reflectors are modeled through the concept of *acoustic impedance*. Additionally, *initial conditions (ICs)* need to be specified as well, that account for the initial state of the sound field. The PDE together with its BCs and ICs gives a complete physical description of the acoustic sound field in a reverberant room. Alternatively, the wave equation can be formulated in the frequency domain. This implies that the sound field is in a *steady state*, meaning that the definition of the ICs is not needed anymore. The frequency-domain wave equation is often referred to as the *Helmholtz equation*.

In general, solving the wave equation and its frequency-domain counterpart is difficult. In fact analytical solutions of the wave equation exist only for a limited set of simple geometries like cuboids, cylinders or spheres with relatively simple

BCs [4]. This is the reason why many different approximations of the wave equation have been proposed to model the acoustics of a reverberant room.

1.1.1 Parametric models

Room impulse responses (RIRs) provide a *purely data-driven description* of the acoustics of a room and are indeed *parametric models*. They can be used to compute many acoustic parameters such as reverberation time, clarity and many others [5, Ch.6]. As Figure 1.1a illustrates, a RIR is defined between two points in space and in practice it represents the acoustic environment as a linear time-invariant (LTI) system *i.e.*, as a *black box*. This makes it possible to actually measure RIRs using classical system identification techniques through the use of a loudspeaker and a microphone. Ideally, the loudspeaker produces an impulse that fully excites the system, creating a sound wave that together with the reflections of the room boundaries are captured by the microphone. In practice, a broad-band deterministic signal is played by the loudspeaker multiple times; the microphone recordings are then averaged to effectively reduce the background noise and the RIR is obtained through deconvolution techniques [6, 7]. Such measurement techniques are often unpractical and time-consuming when many RIRs are needed in a large region of space.

The main drawback of this parametric model is that RIRs provide a point-to-point description of the sound field, hence their spatial description of the sound field is limited to a small set of discrete positions. In some tasks, this creates problems since RIRs change dramatically with position. Methods that involve RIRs typically rely on the use of *adaptive filters* to continuously track RIRs fluctuations. RIRs are typically stored as finite impulse response (FIR) filters and applied using various convolution algorithms. In highly reverberant environments, the number of parameters defining a FIR filter can become large and make these algorithms computationally challenging. Alternatively, infinite impulse response (IIR) filters can be used to substantially reduce the number of parameters. When multiple RIRs are needed the parameters can be further reduced using the concept of *common acoustical poles* [8, 9].

1.1.2 Discretized PDE models

While parametric models provide a purely data-driven model, *discretized PDE models* give a purely physical model. In fact the aim of these acoustic models is to *numerically* solve the PDE describing the wave equation. This is achieved by a spatio-temporal discretization of the wave equation or, alternatively, a spatial discretization of the Helmholtz equation. In both cases, the discretization

effectively converts the continuous PDE into a discrete set of linear equations that can be solved through linear algebra techniques. There are many different strategies to perform such discretization leading to alternative numerical methods. Among the most famous methods used in acoustics are the finite element method (FEM), the boundary element method (BEM) and the finite-difference method (FDM). The FEM discretizes the continuous PDE in space using meshes and exploits the weak formulation of the PDE leading to a sparse linear system of equations [10]. The BEM alternatively discretizes only the boundaries of the room using the boundary integral formulation of the PDE resulting in a lower-dimensional but dense linear system of equations [11]. The FDM utilizes finite differences to approximate the partial derivatives appearing in the PDE [12] leading to a uniform discretization of space and time. This method also produces a sparse linear system of equations which can be solved using parallel programming techniques. Many recent research works concerning room acoustic simulations have focused on solving the wave equation using the latter method, which in this case is known as the finite-difference time-domain (FDTD) method [13–17]. It has been shown that Graphics Processing Units (GPUs) can be employed in the FDTD method to simulate the acoustics of large rooms [18–20]. In this thesis the FDTD method is used in Chapters 3 to 6. More recently, there has be a renewed interest on the finite-volume time-domain (FVTD) method, which is related to the FDTD method but has the advantage of avoiding a staircase approximation of the room geometry which is inevitable when using the FDTD method [21, 22]. There exist many other techniques for discretizing PDEs such as the discontinuous Galerkin method [23], the Trefftz method [24], the wave-based method [25] and spectral methods [26].

Contrary to the parametric models which provide a point-to-point description of the acoustic phenomenon these methods fully describe the acoustics of a room through a discrete set of points spanning its entire space as shown in Figure 1.1b. In fact, the discretized PDE models can be viewed as a compact collection or *dictionary* of RIRs of all the combination of positions inside the room and, indeed, these methods are often used to produce synthetic RIRs.

The main drawback of these methods is that the discretization procedure introduces numerical errors creating non-physical artifacts. For example, in the FDTD method, wave propagation is corrupted with *numerical dispersion i.e.*, sound waves travel with different speed depending on their frequency and direction. These artifacts can be alleviated by using either finer spatio-temporal discretizations or more sophisticated discretization techniques at the cost of an inevitable increase of the computational complexity. In general, the computational burden of these methods is high, limiting the modeling to low frequencies only. In addition, in order for these acoustic models to provide an accurate description of the acoustic phenomenon a precise definition of the

geometry and of the BCs is needed. This implies that when these acoustic models are used in certain methods, the room geometry must be precisely known together with the acoustic impedance defined in the BCs. A tuning of the acoustic model to properly describe the room under study is therefore necessary and, alternatively, techniques for geometry and acoustic impedance estimation should be pursued.

1.1.3 Reflection path models

Reflection path models do not aim at directly solving the wave equation but instead rely on the observation that sound waves follow particular paths with respect to a position of interest. The image method (IM) is perhaps the most famous representative of this category of acoustic models. The idea behind this method is to substitute the walls of a room with *acoustic mirrors*. In practice, the room walls are virtually removed and replaced by *image sources* positioned symmetrically with respect to the original source and walls and placed in a virtual empty space that exceeds the room dimensions. This process is repeated for many of these *mirrored rooms* creating a large set of image sources whose sound waves reach the position of interest with different rectilinear paths creating reverberation. The standard IM [27] is limited to model cuboid rooms only; variations of the method exists that allow for rooms to have an arbitrary polyhedral geometry [28]. The IM is very popular since it can efficiently generate full-band RIRs with realistic features. However, at specific positions audible artifacts are present. In Chapter 3 the acoustic phenomenon that causes these artifacts, known as *sweeping echo*, is studied and a variation of the IM that effectively removes the artifacts is proposed. An open-source software of this method was also developed, see Appendix B.

Another reflection path model is derived from the behavior of sound waves at high frequencies, namely sound waves travel through space like *rays*. *Ray tracing methods* exploit this feature treating sound waves as particles that are reflected and scattered by the objects and walls [29]. RIRs can be generated by computing the time and energy of the particles when crossing a specific volume around the position of interest. A similar approach is the *beam tracing method* which utilizes beams instead of rays [30]. Another example of a reflection path model is the scattering delay network (SDN) that utilizes only few points of the walls to model reflections in order to improve efficiency [31].

These acoustic models are highly suitable for artificial reverberation and auralization as they provide efficient algorithms that produce perceptually realistic sounds. However, in general they can fail to continuously represent a

sound field (ray tracing methods) or are limited to simple geometries and BCs (IM and SDN).

1.1.4 Wave decomposition models

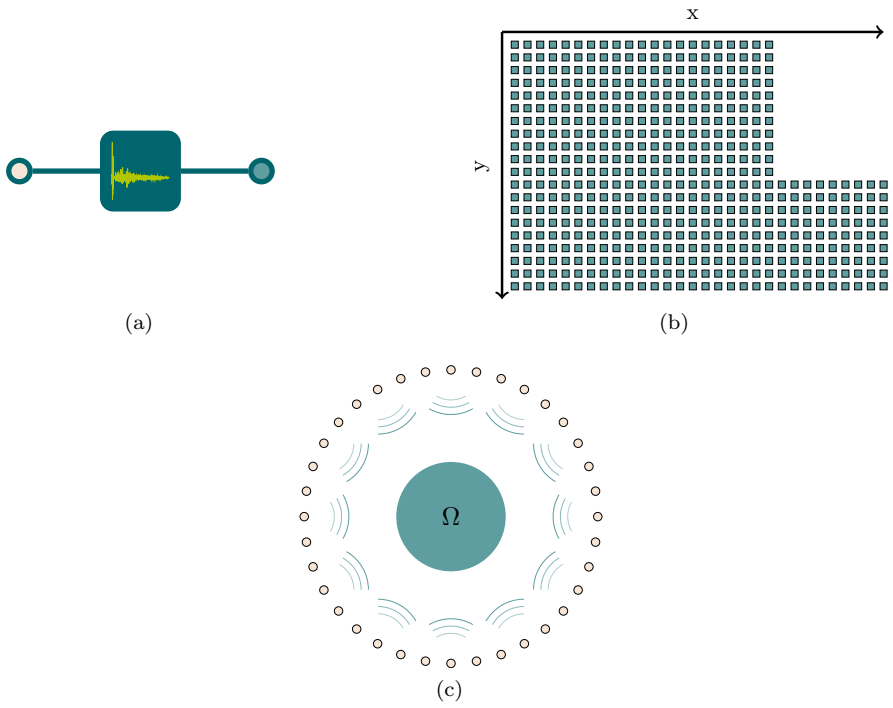


Figure 1.1: (a) Parametric models provide a point-to-point description of the acoustic phenomenon. Inside the black-box a plot of a RIR is shown. (b) Discretized PDE models describe the acoustics throughout the room under study. Here an L-shaped room is depicted with the typical spatial uniform grid imposed by the FDTD method. (c) Wave decomposition models are able to describe the sound field only in a continuous volume Ω. Here the equivalent sources, whose signals must be retrieved from measurements, are placed around Ω and produce waves that approximate the sound field.

Wave decomposition models can be seen as a hybrid between discretized PDE models and parametric models: they typically consist of solutions of the wave equation that can be *"trained"* using measurements. The equivalent source method (ESM) [32] and its time-domain counterpart, the time-domain

equivalent source method (TESM) [33], consist of collections of *particular solutions* of the Helmholtz equation and wave equation respectively in an unbounded domain. These particular solutions are also often referred to as *Green's functions* and in the case of an unbounded domain have simple analytical expressions. The equivalent sources of these methods consist of point sources that produce *spherical waves*. As Figure 1.1c depicts, these equivalent sources can be placed around a volume of interest. If many equivalent sources are used, their sound waves can approximate any sound field *continuously* inside a volume. However, the signals that control the equivalent sources are usually unknown. By placing a set of microphones inside the volume, the measured sound pressure can be matched with the one produced by the equivalent sources and can be used to retrieve the equivalent source signals, which are referred to as *weight signals*. This effectively decomposes the sound field recorded by the microphones into a set of waves arriving to the volume from specific directions. Once the weight signals are known, the acoustic model is effectively "trained" and the sound pressure can be predicted at any position inside the volume. However, as it will be described more in detailed in the next section, estimating the weight signals is not an easy task since it involves the solution of an inverse problem.

A related method, is the plane wave decomposition method (PWDM) which consists of a collection of *plane waves*. Plane waves are *homogeneous solutions* of the Helmholtz equation, also capable of continuously approximating any sound field in a volume [34]. Similarly, the spherical harmonic decomposition method (SHDM) [35] utilizes the homogeneous solution of the Helmholtz equation in spherical coordinates. In this thesis PWDM and TESM are used and compared for various audio signal processing applications in Chapters 7 and 8.

Wave decomposition models offer more flexibility than parametric models described in Section 1.1.1. While RIRs provide only a point-to-point description of the sound field, these acoustic models allow for a spatial description in an entire volume. This comes at the cost of a non-trivial training phase, which is usually computationally involved. Correspondingly, some clear advantages of wave decomposition models with respect to discretized PDE models of Section 1.1.4 can be stated: firstly, the training phase accounts to avoid geometry estimation and impedance identification procedures. Secondly, wave decomposition models provide wider bandwidth since they are not affected by numerical errors and therefore can be more efficient. On the other hand, these acoustic models offer a limited description of the sound field in space in a continuous volume, while discretized PDE models provide a full description in the entire room through a set of discrete points. Hence, indeed wave decomposition models are a tradeoff between parametric models and discretized PDE models.

1.2 Room acoustics inverse problems

In the previous section various acoustic models have been introduced that are all able to approximate the sound field of a room using a wide range of techniques. Most of these acoustic models were developed for *simulation* purposes. Sound sources are virtually placed at particular positions of the room, each driven by a *source signal*. The acoustic models can simulate the sound field typically providing *sound pressure signals* recorded by virtual microphones. Since the acoustic phenomenon in rooms is linear, all of these acoustic models are linear too and can be compactly represented as the following multiple-input multiple-output (MIMO) system:

$$\mathbf{P} = \mathsf{D}\mathbf{S}. \tag{1.1}$$

Assuming the simulation is in the time domain and that it lasts for N_t samples, here $\mathbf{S} \in \mathbb{R}^{N_t \times N_s}$ represents a matrix containing in each of its columns the source signals that control the sound sources which are placed in different positions and generate the sound field. Similarly, $\mathbf{P} \in \mathbb{R}^{N_t \times N_p}$ is the matrix containing in each of its columns the sound pressure signals at the different virtual microphone positions. Despite the resemblance with matrix multiplication, the notation used in (1.1) means that the linear mapping $\mathsf{D} : \mathbb{R}^{N_t \times N_s} \to \mathbb{R}^{N_t \times N_p}$, associated with an acoustic model, is applied to the matrix $\mathbf{S}$ using one of the methods presented in Section 1.1 to obtain $\mathbf{P}$ and simulate a sound field. Evaluating (1.1) for a given $\mathbf{S}$ is also referred to as solving the *forward problem*. This is a *well-posed problem*: as a matter of fact, given that deterministic acoustic models are used, a particular configuration of $\mathbf{S}$ will always generate a unique and stable sound field.

Nevertheless, these acoustic models are also useful in many tasks where $\mathbf{P}$ is only partially known at a few locations and possibly corrupted by noise. This partial knowledge, eventually resulting from real microphone measurements, consists of sound pressure signals that are stored in the columns of $\tilde{\mathbf{P}}$ and must be used to estimate the unknown source signals $\mathbf{S}$. These types of problems are known as *inverse problems*, requiring an inversion of the acoustic model which is often achieved by the solution of an *optimization problem*. For example, the following optimization problem aims at minimizing the distance between the sound pressure signals $\mathbf{P}$ of the acoustic model and the measured ones $\tilde{\mathbf{P}}$, in the least-squares sense:

$$\mathbf{S}^\star = \underset{\mathbf{S}}{\operatorname{argmin}} \|\mathsf{D}\mathbf{S} - \tilde{\mathbf{P}}\|_F^2, \tag{1.2}$$

where $\|\cdot\|_F$ is the Frobenius norm and $\mathbf{S}^\star$ is the optimal solution. Unfortunately, the solution of this inverse problem is often not satisfactory. For example, if D represents the FDTD method, $\mathbf{S}$ could represent the source signals that

control sound sources placed at each grid point. Similarly, if D consists of a wave decomposition model like TESM, $\mathbf{S}$ would contain the weight signals and control different sound waves arriving from particular directions. In both cases, these acoustic models can fit $\tilde{\mathbf{P}}$ using infinitely many configurations of $\mathbf{S}$, or, on the contrary, they may be unable to describe $\tilde{\mathbf{P}}$ at all. In other words, inverse problems are *ill-posed* as their solution can be non-unique or non-existent and, often, unstable. In addition, since $\tilde{\mathbf{P}}$ usually contains noise and the acoustic model may be partially incorrect, perfect fitting is not preferred, as measurement noise and model errors are amplified in the solution $\mathbf{S}^\star$. This issue is known as *over-fitting*.

In order to prevent these obstacles a solution is to *regularize* (1.2) by adding a *prior knowledge* over the nature of $\mathbf{S}^\star$. This prior knowledge can be included in (1.2) by adding an additional term g to the cost function of the optimization problem

$$\mathbf{S}^\star = \underset{\mathbf{S}}{\operatorname{argmin}} \|\mathsf{D}\mathbf{S} - \tilde{\mathbf{P}}\|_F^2 + g(\mathbf{S}). \tag{1.3}$$

The minimization of g promotes particular features on $\mathbf{S}^\star$. For example, the most common regularization is Tikhonov regularization [36] which prevents $\mathbf{S}^\star$ to have high energy. More recently, *sparse regularization*, which promotes a limited number of non-zero elements in the solution, has been successfully used in many applications: an entire framework has been dedicated to their study, called compressed sensing (CS) [37]. CS treats the problem of signal reconstruction with a limited amount of data, *i.e.*, with sub-Nyquist sampling rates. The idea behind these methods is to use a *dictionary* containing *atoms* through which the signal can be fully described. Assuming that the signal can be constructed using only few atoms *i.e.*, that it is sparse, perfect signal reconstruction can be achieved from a limited amount of noisy data through the solution of an inverse problem. Similarly, the sound field can be treated as a signal to be reconstructed from a limited amount of measurements and the acoustic models can be considered dictionaries. The atoms of the dictionary are sound waves that can be activated by the source signals $\mathbf{S}$. Hence a proper estimation of $\mathbf{S}$ can lead to the full knowledge of the sound field. Promoting sparsity on the source signals $\mathbf{S}$ is the key to obtain meaningful results in room acoustic applications as well. Most of the methods treated in this thesis, promote *spatial sparsity*. Indeed, for discretized PDE models, it is often legitimate to assume that sound sources are present only at few locations of the room. Similarly, for wave decomposition models, it can be assumed that sound waves arrive to the volume of interest only from a limited number of directions. Other types of sparse regularization are also possible: *spatio-temporal sparsity* which additionally to spatial sparsity promotes sound sources to have an impulsive behavior and *spatio-spectral sparsity* which enforces few frequencies to be active only at few locations.

1.2.1 Source localization

Many methods for acoustic source localization exist which are applied in a large variety of contexts, such as in speech enhancement. A few examples of classic source localization methods are the MUltiple SIgnal Classification (MUSIC), [38], the estimation of signal parameters via rotational invariance technique (ESPRIT) [39] and their more recent variations [40, 41]. These are purely data-driven approaches that typically assume no reverberation and hence exhibit degraded performance in highly reverberant environments. Typically, these methods are used to estimate the direction of arrival (DOA) of sound sources. The use of acoustic models can both improve the localization in reverberant rooms by actually *exploiting reverberation* rather than treating it as an interference. Recently, many acoustic models have been used for this task: reflection path models [42, 43], discretized PDE models using the FEM [44, 45] or the FDTD method [46, 47], and wave decomposition models using the PWDM [48]. Physical model based source localization methods are also studied for other PDEs *e.g.*, modeling diffusion [49, 50].

In this thesis, Chapters 5 and 8 treat source localization. Discretized PDE models can allow localization in terms of not only the DOA but also the precise coordinate positions even when the source is not in the line of sight of the microphone [44]. Similarly, in Chapter 5, the FDTD method is matched with the signals of a number of microphones scattered around a reverberant room. This is achieved by estimating the source signals through the solution of an inverse problem that promotes their spatio-temporal sparsity. By inspecting the energy of these source signals, it is possible to retrieve the coordinates of the original sound source. However, a precise knowledge of the room geometry and of the BCs is assumed. In Chapter 8 these requirements are relaxed by the usage of wave decomposition models (the PWDM and the TESM) that are trained with the microphone signals to model the sound field only in a limited volume of the room where a microphone array is used. Depending on the acoustic model that is used different types of sparse regularization can be inferred: spatial sparsity, spatio-temporal sparsity and spatio-spectral sparsity with the latter giving the better performance when the sound field is created by a speech source. The weight signals are then retrieved and by inspecting their energy it is also possible to localize the original sound source but, contrary to the FDTD method approach, only in terms of DOA.

1.2.2 Sound field reconstruction and dereverberation

Another advantage of using acoustic models is that source localization can be performed jointly with *sound field reconstruction* and *dereverberation*. Sound

field reconstruction consists of estimating the sound field at positions where no microphone is present, either inside (interpolation) or outside (extrapolation) a volume surrounded by a microphone array. Dereverberation treats instead the problem of removing reverberation from microphone recordings to improve sound quality or speech intelligibility [51]. Despite these two tasks may seem very distant, in the framework treated by this thesis it becomes natural to pursue them jointly. This is discussed in both Chapters 5 and 8 using either discretized PDE models or wave decomposition models, respectively. In particular, in Chapter 5 once the coordinates of the sound source is estimated, the microphone measurements can be readily used to estimate the source signal that generated the sound field, *i.e.*, a *dereverberated* signal, together with the ICs. This is achieved by performing a second inversion of the acoustic model *i.e.*, the solution of an overdetermined system of equations, which becomes feasible due to the knowledge of the sound source position. Although not addressed directly in Chapter 5, when the source signal is retrieved, the acoustic model can provide estimates of the sound field at position where no microphone is present, and hence perform both a *sound field interpolation and extrapolation* of the entire sound field inside the room. Similarly, in Chapter 8, the wave decomposition models allow for a *sound field interpolation* of the sound field inside the volume where the microphone array is used. Likewise, the weight signals that are retrieved from the solution of the inverse problem not only can be used for a localization task but, since reverberation is spatially distributed among them, their isolation can produce dereverberated signals.

These strategies are substantially different from state-of-the-art dereverberation techniques which either perform channel equalization using FIR [52], utilize beamforming techniques [53, 54] or multi-channel linear prediction (MCLP) [55, 56]. Concerning sound field reconstruction, the advantage of using acoustic models combined with sparse regularizations is that they effectively *reduce the number of microphones*. For example, using a 3-dimensional uniform microphone array, correct interpolation can be achieved only if the spacing between the microphones is smaller than $\frac{c}{2F_u}$, where c is the speed of sound and F_u is the cut-off frequency in Hz of the sound source [57]. The use of an acoustic model together with sparse regularization can greatly reduce such a requirement. This is demonstrated in Chapter 7, where the task of RIR interpolation is treated. Since RIRs can be viewed as the measurement of a sound field created by an impulsive source, *i.e.*, a temporally sparse signal, it is shown that effective interpolation can be achieved by combining the TESM with spatio-temporal sparsity to perform interpolation using only a limited number of microphones. As described in Section 1.1.1, measuring RIRs in a volume can be time-consuming and therefore their effective interpolation has been studied in a number of recent works [58–64].

Sound field reconstruction is also pursued in *acoustic holography* which seeks to visualize the radiation patterns of vibrating sources by reconstructing their near-field sound fields. Here wave decomposition models have been used extensively [35] and sparse regularization has also been explored [65].

1.2.3 Sound field control

Sound field reconstruction is strongly linked with *sound field control*. In many cases these two tasks have actually quite similar inverse problem formulations. However, their aim is substantially different: sound field reconstruction can be used as an *acquisition technique* while sound field control is a *reproduction technique*. Sound field control aims at reproducing a sound field recorded in a different room, perhaps interpolated or extrapolated by a sound field reconstruction method or even virtually generated. This is achieved by controlling the signals of a set of loudspeakers. The use of acoustic models in sound field control have a long history. State-of-the art technologies utilize parametric models in channel equalization [52], SHDM in three dimensional ambisonics [66] and the Huygens-Fresnel principle in wave-field synthesis (WFS) [67]. Recent research has been focusing on *multi-zone sound field control* which explores the possibility of controlling the sound field such that some areas of the room are kept silent while sound is being played [68]. Inverse problems with sparse regularization have also been explored [69, 70]. Most of the state-of-the-art methods employ large quantities of loudspeakers and have poor results when the acoustic models assume little or no reverberation of the room where they are employed. Chapter 6 addresses these problems by attempting a sound field reproduction in a highly reverberant room with a limited number of loudspeakers. The FDTD method is combined with spatial sparsity in order to control a multi-zone sound field. The acoustic model makes it possible to control a highly reverberant room while spatial sparsity minimizes the number of active loudspeakers while simultaneously selecting their optimal position.

1.2.4 Geometry and impedance estimation

As described in Section 1.1.2 one of the drawbacks of discretized PDE models is that geometry and BCs must be known precisely. In Chapters 5 and 6 where sound field reconstruction and multi-zone sound field control are treated using the FDTD method, the room geometry and the acoustic properties of the walls are assumed to be known. Only simulation results are presented since in order to make these methods work in real scenarios a fine tuning of the acoustic

model should be performed. This is a difficult task which should preferably be automatized.

Geometry estimation using acoustic measurements is an active branch of research. Many different approaches have been proposed using reflection path models [71–73] and more recently using the PWDM [74]. Of equivalent importance is the estimation of the acoustic properties of the walls which are modeled in the BCs through the usage of *acoustic impedances*. Classical measurement techniques for acoustic impedance measurements require reverberation chambers [4]. These laboratory facilities are however often not practical and therefore in [75] an in-situ measurement technique which relies on the BEM and assumes a known room geometry has been proposed. A similar technique that utilizes the FDTD method instead was proposed in [76]. In both works a single acoustic impedance was used to model entire walls, which, for most rooms, is not enough to provide a realistic description. Chapter 4 extends the idea of [76] by estimating the acoustic impedance on a grid of points of the room walls. The proposed method is also an inverse problem which requires the solution of a nonconvex optimization problem. In order to achieve meaningful results, an additional prior knowledge is also included in the inverse problem, namely that the acoustic impedances should be spatially smooth. Finally, recent research has shown that the acoustic impedances can be estimated jointly with the sound source signals to perform source localization using the FDTD method [47].

1.3 Optimization Algorithms

As it was described in Section 1.2, in general, inverse problems are formulated as *optimization problems*. Similarly to PDEs, many optimization problems do not have analytical solutions and are therefore solved numerically through *optimization algorithms*. There exist a large variety of optimization algorithms and, depending on the type of optimization problem one wants to solve, some are more efficient than others. Most of the inverse problems that have been discussed in Section 1.2 have the common characteristics of leading to *large-scale nonsmooth optimization problems*.

The *large scale* characteristic of these inverse problems is due to the fact that often large quantities of optimization variables are inevitable: for example in source localization with discretized PDEs models, the optimization variables correspond to the source signals in every location of the room. A fine discretization of the PDE, which is necessary to attain a realistic simulation, can easily lead to a number of optimization variables in the order of thousands, millions or even more. This occurs also when using other acoustic models, for

example with wave decomposition models which also need many parameters to be estimated. The other common feature of these inverse problems is the *nonsmoothness* of the cost function. This is typically due to the sparse regularization, included in the cost function by means of a nondifferentiable regularization term, *e.g.*, indicated in (1.3) with g.

Classical optimization algorithms like Newton methods [77] rely on differentiability (smoothness) of the cost function and are not directly applicable to nonsmooth optimization problems. For this reason particular optimization algorithms must be used. Among the most popular ones, used in CS and in many of the work cited in this introduction are the *greedy algorithms e.g.*, matching pursuit (MP) and its variants [78, 79] which, however, are not guaranteed to converge to a local minimum of the cost function. *Interior point methods* [77, Ch. 14] have also been used extensively but are mainly suitable for medium-scale problems. Finally, more recently, *splitting methods* have recently regained popularity [80,81]. These first-order methods address nonsmooth cost functions and are well suited to large-scale optimization problems. A class of splitting methods known as *proximal gradient (PG) algorithms* can be implemented with *matrix-free* operators by utilizing efficient algorithms for the evaluation of D and its adjoint mapping, avoiding matrix decompositions and large memory requirements. In addition, PG algorithms have been recently accelerated through the use of quasi-Newton methods [82,83]. Chapter 2 gives a complete overview of PG algorithms demonstrating many signal processing application examples.

In this thesis, many different optimization algorithms have been used. Classical Newton and quasi-Newton methods have been used in Chapters 4 and 6 where in the latter a nonconvex smooth sparse regularization was used. Interior point methods have been used in Chapter 5 since a two-dimensional room was used as proof of concept for the proposed method leading to a medium-scale optimization problem. Finally, accelerated PG algorithms have been used in Chapters 7 and 8.

1.4 Overview of the thesis

The thesis is divided into four parts. **Part I** addresses optimization algorithms and consists of **Chapter 2** which is an introduction to splitting algorithms with a particular focus on PG algorithms and their application in signal processing. Here many different application examples are shown together with the review of fundamental concepts such as convex and nonconvex optimization and matrix-free optimization. A new optimization software package developed

by the authors is also presented and a quick tutorial guide can be found in **Appendix A**.

Chapter 2 will be submitted for publication as a tutorial paper as:

- **N. Antonello**, L. Stella, P. Patrinos and T. van Waterschoot, "Proximal gradient algorithms: applications in signal processing", *to be submitted for publication*, 2018.

Part II treats the subject of acoustic modeling. It consists of **Chapter 3** which studies *sweeping echoes*, an acoustic phenomenon that occurs in rectangular rooms which are usually not perceived in real rooms. Sweeping echoes are studied and attributed to the perfect cuboid geometry used in the simulation. This enables to design a modified version of the IM, called the randomized image method (RIM), which is capable of generating realistic RIRs without sweeping echoes. A software package has also been developed for the RIM and a quick tutorial is found in **Appendix B**.

Chapter 3 has been published as:

- E. De Sena, **N. Antonello**, M. Moonen, and T. van Waterschoot, "On the modeling of rectangular geometries in room acoustic simulations," *IEEE/ACM Transactions on Audio, Speech Language Processing*, vol. 23, no.6, pp. 774-768, Apr. 2015.

In **Part III** different inverse problems are treated using only discretized PDE models. In particular, all the chapters of this part utilize the FDTD method.

As discussed in Section 1.1.2, one of the drawbacks of these acoustic models is the requirement of the precise knowledge of the geometry and of the acoustic impedances of the room under study. For this reason Part III begins with **Chapter 4** where a novel technique for acoustic impedance estimation is proposed that relies on the assumption of spatial smoothness of the acoustic impedance over the wall surfaces.

Chapter 4 has been published as:

- **N. Antonello**, T. van Waterschoot, M. Moonen, and P. Naylor, "Evaluation of a numerical method for identifying surface acoustic impedances in a reverberant room," in *Proc. of 10th European Congress and Exposition on Noise Control Engineering (EURONOISE 2015)*, Maastricht, the Netherlands, June 2015.

Part III continues with **Chapter 5** which treats the problem of source localization and signal reconstruction. Here a two-step procedure is proposed: firstly source localization is achieved thanks to spatio-temporal sparse regularization and secondly the source signal can be retrieved by solving an overdetermined system of equations.

Chapter 5 has been published as:

- **N. Antonello**, T. van Waterschoot, M. Moonen, and P. Naylor, "Source localization and signal reconstruction in a reverberant field using the FDTD method", in *Proc. 22nd European Signal Process. Conf. (EUSIPCO 2014)*, Lisbon, Portugal, Sep. 2014.

Finally, Part III is concluded by **Chapter 6** which treats the problem of multi-zone sound field control in highly reverberant rooms. Here the FDTD method allows for a compact description of the sound field which is controlled by retrieving the signals of a set of loudspeakers. Meaningful results are obtained by exploiting spatial sparsity which simultaneously reduces the number of active loudspeakers while finding their optimal positions.

Chapter 6 has been published as:

- **N. Antonello**, E. De Sena, M. Moonen, P. Naylor and T. van Waterschoot, "Sound field control in a reverberant room using the Finite Difference Time Domain method," in *AES 60th Int. Conf.*, Leuven, Belgium, Feb. 2016.

Part IV explores the usage of wave decomposition models in room acoustic applications. Here the TESM and the PWDM are used and compared.

Part IV begins with **Chapter 7** where the wave decomposition models are used in order to effectively interpolate measured RIRs in a source-free volume of space. It is shown that the TESM combined with spatio-temporal sparsity gives the best performance with an effective interpolation of RIRs using few microphones, particularly when the impulse response (IR) of the loudspeaker used in the measurements is available. Results are presented using both simulated and real microphone measurements.

Chapter 7 has been published as:

- **N. Antonello**, E. De Sena, M. Moonen, P. A. Naylor and T. van Waterschoot, "Room impulse response interpolation using a sparse spatio-temporal representation of the sound field," *IEEE/ACM Transactions on*

Audio, Speech Language Processing, vol. 25, no. 10, pp. 1929-1941, Oct. 2017.

Chapter 8, which concludes Part IV, proposes a novel method capable of performing joint source localization and dereverberation. The inverse problem results in an optimization problem that is split using a weighted overlap-add (WOLA) strategy enabling the possibility of tracking the DOA of a moving sound source. Different types of sparse regularization are compared. Simulated and real measurements are used to validate the proposed method which shows to be robust to diffuse and localized noise.

Chapter 8 has been submitted for publication as:

- **N. Antonello**, E. De Sena, M. Moonen, P. A. Naylor, and T. van Waterschoot, "Joint acoustic localization and dereverberation by sound field interpolation," *submitted for publication*, 2018.

Finally in **Chapter 9** conclusions are drawn and some directions for future research are explored.

Part I

Optimization Algorithms

Chapter 2

Proximal gradient algorithms

Proximal gradient algorithms: applications in signal processing

Niccolò Antonello, Lorenzo Stella, Panagiotis Patrinos,
and Toon van Waterschoot

To be submitted.

The candidate's contributions as first author include: co-literature study, co-design of software packages, design of the example applications, co-formulation of the conclusions, text co-redaction and co-editing.

Abstract

Advances in numerical optimization have supported breakthroughs in several areas of signal processing. This paper focuses on the recent enhanced variants of the proximal gradient numerical optimization algorithm, which combine quasi-Newton methods with forward-adjoint oracles to tackle large-scale problems and reduce the computational burden of many applications. These proximal gradient algorithms are here described in an easy-to-understand way, illustrating how they are able to address a wide variety of problems arising in signal processing. A new high-level modeling language is presented which is used to demonstrate the versatility of the presented algorithms in a series of signal processing application examples such as sparse deconvolution, total variation denoising, audio declipping and others.

2.1 Introduction

Signal processing and numerical optimization are independent scientific fields that have always been mutually influencing each other. Perhaps the most convincing example where the two fields have met is compressed sensing (CS) [37]. CS originally treated the classic signal processing problem of reconstructing a continuous signal from its digital counterparts using a sub-Nyquist sampling rate. The reconstruction is achieved by solving an optimization problem known as the least absolute shrinkage and selection operator (LASSO) problem. Stemming from the visibility given by CS, LASSO gained popularity within the signal processing community. Indeed, LASSO is a specific case of a *structured nonsmooth optimization problem*, and so representative of a more generic class of problems encompassing constrained and nonconvex optimization.

Developing efficient algorithms capable of solving structured nonsmooth optimization problems has been the focus of recent research efforts in the field of numerical optimization, because classical methods (*e.g.*, Newton-type) do not directly apply. In the context of convex optimization, such problems can be conveniently transformed into conic form and solved in a robust and efficient way using *interior point methods*. These methods became very popular as they are applicable to a vast range of optimization problems [84]. Unfortunately, they do not scale well with the problem size as they heavily rely on matrix factorizations and are therefore efficient for medium-size problems only [85].

More recently, there has been a renewed interest towards *splitting algorithms* [80, 81, 86]. These are first-order algorithms that minimize nonsmooth cost functions with minimal memory requirements allowing to tackle large-scale

problems. The main disadvantage of splitting algorithms is their low speed of convergence, and hence a significant research effort has been devoted to their tuning and acceleration. Notable splitting algorithms are the proximal gradient (PG) algorithm [87], also known as forward-backward splitting (FBS) algorithm, the alternating direction method of multipliers (ADMM) [88], the Douglas-Rachford (DR) and the Pock-Chambolle algorithm (PC) [89]. The first acceleration of PG can be traced back to [90] and is known as the fast proximal gradient (FPG) algorithm or as fast iterative shrinkage-thresholding algorithm (FISTA) [91]. More recent acceleration approaches of PG include the variable metric forward-backward (VMFB) algorithm [92–95] and the application of quasi-Newton methods [82, 83, 96, 97].

Several surveys dedicated to these algorithms and their applications in signal processing have appeared [81, 85, 86, 98], mainly focusing on convex problems only. In fact, only recently some extensions and analysis for nonconvex problems have started to emerge [99, 100]. In convex problems there is no distinction between local and global minima. For this reason, they are in general easier to solve than their nonconvex counterpart which are characterized by cost functions with multiple local minima. Despite this, it was recently shown that nonconvex formulations might either give solutions that exhibit better performance for the specific signal processing application [101], or lead to computationally tractable problems [102], for which the presence of spurious local minima is less pronounced or absent, and thus local optimization coupled with a proper initialization often leads to global minima [100].

This paper will focus on the PG algorithm and its accelerated variants, with the aim of introducing the latest trends of this numerical optimization framework to the signal processing community. The recent advances in the acceleration of the PG algorithm combined with matrix-free operations provide a novel flexible framework: in many signal processing applications such improvements allow tackling previously intractable problems and real-time processing. This framework will be presented in an effective and timely manner, summarizing the concepts that have led to these recent advances and providing easily accessible and user-friendly software tools. In particular, the paper will focus on the following topics:

- *Accelerated variants of PG:* FISTA has received significant attention in the signal processing community. However, more recently, the PG algorithm has been accelerated using different techniques: it has been shown that Quasi-Newton methods [82, 83] can significantly improve the algorithm performance and make it more robust to ill-conditioning.
- *Non-convex optimization:* proximal gradient algorithms can treat both convex and nonconvex optimization problems. While many convex

relaxations increase dimensionality [103] and may result in computationally intractable problems, proximal gradient algorithms are directly applicable to the original nonconvex problem thus avoiding scalability issues. It is indeed of equal importance to have modeling tools and robust, rapidly converging algorithms for both nonconvex and convex nonsmooth problems. This allows to quickly test different problem formulations independently of their smoothness and convexity.

- *Forward-adjoint oracles and matrix-free optimization:* one important feature of proximal gradient algorithms is that they usually only require direct and adjoint applications of the linear mappings involved in the problem: in particular, no matrix factorization is required and these algorithms can be implemented using *forward-adjoint oracles (FAOs)*, yielding *matrix-free optimization algorithms* [104, 105]. Many signal processing applications can readily make use of FAOs yielding a substantial decrease of the memory requirements.

- *A versatile, high-level modeling language:* many optimization frameworks owe part of their success to easily accessible software packages, *e.g.*, [106, 107]. These software packages usually provide intuitive interfaces where optimization problems can be described using mathematical notation. In this paper a new, open-source, high-level modeling language implemented in Julia [108] called `StructuredOptimization` will be presented. This combines efficient implementations of proximal gradient algorithms with a collection of FAOs and functions often used in signal processing, allowing the user to easily formulate and solve optimization problems.

A series of signal processing application examples will be presented throughout the paper in separate frames to support the explanations of various concepts. Additionally, these examples will include code snippets illustrating how easily problems are formulated in the proposed high-level modeling language.

The paper is organized as follows: in Section 2.2 models and their usage in optimization are displayed through the description of inverse problems and the main differences between convex and nonconvex optimization. In Section 2.3 the classical proximal gradient algorithms and their accelerated variants are described. In Section 2.4 the concepts of FAOs and matrix-free optimization are introduced. Section 2.5 describes the types of problems that proximal gradient algorithms can tackle. Finally, in Section 2.6 the proposed high-modeling language is described and conclusions are given in Section 2.7.

2.2 Modeling

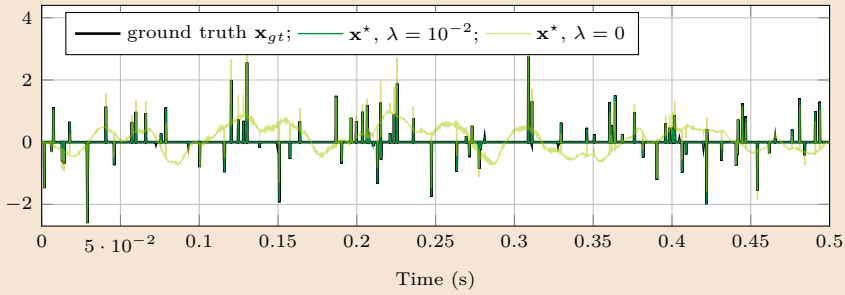
Models allow not only to describe physical phenomena, but also to transform signals and access hidden information that these often carry. Models may be *physical models*, obeying the laws of physics and describing *e.g.*, mechanical systems, electrical circuits or chemical reactions or *parametric models*, which are not necessarily linked to physical phenomena, and are purely defined by mathematical relations and numerical parameters. In general, both categories of models consist of defining an associated mapping that links an input signal $x(t)$ to an output signal $y(t)$. Here t may stand for time, but signals could be also N-dimensional *e.g.*, $x(t_1, \dots, t_N)$, and be functions of many quantities: position, temperature or the index of a pixel of a digital image. For computational convenience if the models are continuous, they must be discretized: the continuous signals involved are sampled and their samples stored either in vectors $\mathbf{x} = [x(t_1), \dots, x(t_n)]^\intercal \in \mathbb{R}^n$, in matrices $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ or in tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times \cdots \times n_N}$ depending on their dimensionality. In the following these vectors, matrices and tensors will be referred to as signals as well. The mapping $\mathcal{A} : \mathtt{D} \to \mathtt{C}$ associated with a model therefore links two (perhaps complex) finite-dimensional spaces $\mathtt{D}$ and $\mathtt{C}$ like, for example, $\mathcal{A} : \mathbb{C}^n \to \mathbb{C}^m$. Notice that in this work a (complex) finite-dimensional space $\mathtt{D}$ induces a Hilbert space with *real* inner product *i.e.*, $\langle \mathbf{x}, \mathbf{y} \rangle = \mathfrak{Re}(\mathbf{x}^{\mathsf{H}}\mathbf{y})$, where ^{H} is the conjugate-transpose operation, and with norm $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. Mappings can also be defined between the Cartesian product of m and n finite-dimensional spaces: $\mathcal{A} : \mathtt{D}_1 \times \cdots \times \mathtt{D}_m \to \mathtt{C}_1 \times \cdots \times \mathtt{C}_n$ for example when dealing with multiple-input multiple-output models.

Depending on the nature of the models, mappings can be either *linear* or *nonlinear*. Such distinction often carries differences on the algorithms where these mappings are employed: as it will be described later, in optimization this often draws the line between convex and nonconvex optimization. For this reason, here the nonlinear mappings are indicated with the notation $\mathcal{A} : \mathtt{D} \to \mathtt{C}$ while the linear mappings with the notation $\mathsf{A} \in \mathscr{L}(\mathtt{D}, \mathtt{C})$, where $\mathscr{L}$ is the set of all linear mappings between $\mathtt{D}$ and $\mathtt{C}$.

2.2.1 Inverse problems

Models are often used to make predictions. For example it is possible to predict how a model behaves under the excitation of an input signal $\mathbf{x}$. The output signal $\mathbf{y} = \mathcal{A}\mathbf{x}$ can be computed by evaluating the mapping $\mathcal{A}$ associated with the model using a specific algorithm. Notice that here the notation $\mathcal{A}\mathbf{x}$ does not necessarily stand for the matrix-vector product. In many signal processing

Example 1 Sparse Deconvolution



Deconvolution seeks to recover the input signal $\mathbf{x}^\star$ from the available output signal $\mathbf{y}$ of a LTI system with FIR $\mathbf{h}$. In a single-channel case with low SNR, deconvolution can easily become an ill-posed problem ($\lambda = 0$) and regularization must be added in order to achieve meaningful results. If $\mathbf{x}^\star$ is assumed to be sparse, a sparsity-inducing regularization function can be included in the optimization problem (LASSO):

$$\mathbf{x}^\star = \underset{\mathbf{x}}{\operatorname{argmin}} \quad \underbrace{\tfrac{1}{2}\|\mathbf{h} * \mathbf{x} - \mathbf{y}\|^2}_{f(\mathbf{x})} + \underbrace{\lambda\|\mathbf{x}\|_1}_{g(\mathbf{x})}, \tag{2.1}$$

where $*$ indicates convolution and λ is a scalar that balances the weight between the regularization function and the data fidelity function.

StructuredOptimization code snippet:

```
Fs = 4000 # sampling frequency
x = Variable(div(Fs,2)) # 'ls' short-hand of '0.5*norm(...)^2'
@minimize ls(conv(x,h)-y)+lambda*norm(x,1)
```

applications, however, one is interested in the opposite problem: an output signal $\mathbf{y}$ is at disposal, and the input signal $\mathbf{x}$ is to be found. Such problems are known as *inverse problems* and involve many signal processing applications some of which will be treated in this paper like denoising, source separation, channel equalization and system identification. Inverse problems are in general *ill-posed problems* and this makes it difficult to solve them. Three main challenges create this difficulty. Firstly the inverse of the mapping is needed. This is rarely available, it must be computed numerically and it may be unstable due to ill-conditioning. Secondly, $\mathbf{y}$ may be corrupted with noise or the model may be inaccurate, thus it is not possible to fully describe $\mathbf{y}$. Its noise and un-modeled features are then interpreted as effects caused by the input signal: this issue is called *over-fitting*. Finally inverse problems have in general non-unique solutions. These challenges are generally solved by *regularizing* the inverse problem: regularization attempts to exploit *prior information* over the

structure of the sought signal which can simultaneously numerically stabilize the inversion of the mapping, find a unique solution and avoid over-fitting.

All of these concepts can be viewed in Example 1. Here a discrete linear time-invariant (LTI) model is used. This is a general parametric model capable of modeling many physical phenomena. Its linear mapping can consist of the discrete convolution between a finite impulse response (FIR) and the input signal: $\mathbf{y} = \mathbf{h} * \mathbf{x}$, where $\mathbf{h}$ is the signal containing the FIR taps. Example 1 treats the case of an inverse problem named *deconvolution*: this finds applications in a large number of signal processing fields and is known with different names such as channel equalization [109] or dereverberation [110]. What deconvolution seeks is to remove the effect of the channel from a signal $\mathbf{y}$ recorded by *e.g.*, a sensor: such signal could be a transmission received by an antenna or some speech recorded by a microphone, but many more examples could be made. The effect of the electromagnetic or acoustic channel corrupts either the sequence of bits of the transmission or the intelligibility of speech and should be therefore removed. This can be achieved by fitting the recorded signal $\mathbf{y}$ with the LTI model. Equation (2.1) shows the type of optimization problem that can bring such result. This consists of the minimization of a *cost function*, here consisting of the sum of two functions f and g. These are functions of $\mathbf{x}$ which in this context indicates the *optimization variables*. When the optimization variables minimize the cost function the *optimal solution* $\mathbf{x}^\star$ is obtained. Typically, solving (2.1) analytically is not possible and the optimal solution is reached numerically through the usage of specific iterative algorithms starting from an initial guess of the optimization variables $\mathbf{x}^0$. Here the functions f and g, associated with a lower case letter, are always assumed to be proper and closed. These always map an element of a finite-dimensional space to a single number belonging to the extended real line $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$.

In Example 1 function f is a *data fidelity term*, representing the error between the model and the signal $\mathbf{y}$ in the least squares sense. The smaller this term is, the closer the output of the linear mapping, driven by the solution $\mathbf{x}^\star$, is to $\mathbf{y}$. On the other hand, g is a l_1-norm *regularization term*: this is a convex relaxation of the l_0-norm, *i.e.*, the norm counting the number of non-zero elements of $\mathbf{x}$, and promotes sparsity in the solution $\mathbf{x}^\star$. The coefficient λ balances the relative weight of the data fidelity and sparsity inducing terms in the cost function. Here, the signal $\mathbf{y}$ is corrupted using white noise as if it was recorded in a noisy environment or by a faulty sensor.

The figure of Example 1 shows what would happen if $\lambda = 0$, *i.e.*, the case of an un-regularized inverse problem. This solution has low-frequency oscillations, a sign of the numerical instability of the inverted linear mapping. Additionally, it is populated by white noise indicating over-fitting. On the other hand, if $\lambda \to \infty$ the prior knowledge would dominate the cost function leading to the

sparsest solution possible, that is a null solution. Generally, λ needs careful tuning, a procedure that can be automatized by means of different strategies which may involve a sequence of similar optimization problems. Here, with a properly tuned λ the ground truth signal is recovered almost perfectly.

2.2.2 Convex and nonconvex problems

As it is important to choose the proper model to describe the available data, so it is to select the most convenient problem formulation. Different problem formulations can in fact yield optimization problems with different properties and one should be able to carefully choose the one that is best suited for the application of interest.

Perhaps the most fundamental distinction is between *convex* and *nonconvex* optimization problems. This distinction is in fact crucial as, in general, nonconvex problems are more difficult to solve than the convex ones. The main advantage of convex problems lies in the fact that every local minimum is a global one. On the contrary, the cost functions of nonconvex problems are characterized by the presence of *local minima*. These are identified as solutions but they are typically *sub-optimal*, since it is usually not possible to determine whether there exist other solutions that further minimize the cost function. As a consequence of this, the *initialization* of the algorithms used in nonconvex optimization becomes crucial as it can lead to distinct local minima with solutions that can greatly differ from each other.

In order to avoid this issue, many nonconvex problems can be re-formulated or approximated by convex ones: it is often possible to relax the nonconvex functions by substituting them with convex ones that have similar properties. The LASSO is a good example of such a strategy: the original problem involves an l_0-norm, which is a nonconvex function. Despite very different mathematically, the l_1-norm behaves similarly by inducing sparsity over the optimization variables it is being applied to. Such relaxation however can have consequences on the solution and Example 2 displays such situation. Here the problem of *line spectral estimation* is treated: this has many applications like source localization [111], de-noising [112], and many others. A signal $\mathbf{y} \in \mathbb{R}^n$ is given and is modeled as a mixture of sinusoidal components. These lie on a fine grid of frequencies belonging to a discrete Fourier transform (DFT) and hence corresponding to the elements of a complex-valued signal $\mathbf{x}^\star \in \mathbb{C}^{sn}$ which must be estimated. The optimization problem seeks for a sparse solution as these components are assumed to be only few.

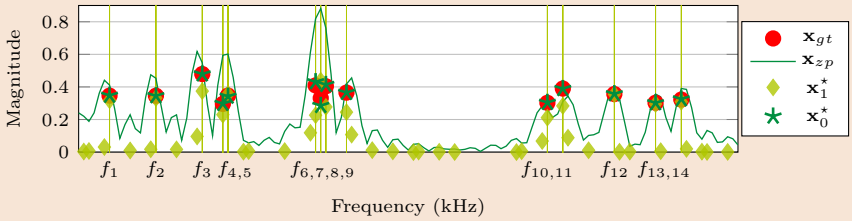
Looking at the figure of Example 2 it is clear that the solution of the nonconvex problem outperforms the one obtained through the LASSO. The solution of

LASSO has in fact many small spurious frequency components. These are not present in the solution of the nonconvex problem which also exhibits amplitudes that are much closer to the ones of the ground truth. This shows that convex relaxations may lead to poorer results than the ones obtained by solving the original nonconvex problem. However, as stated earlier, the presence of local minima requires the problem to be initialized carefully. Indeed, the improved performance of the solution obtained by solving the nonconvex problem in Example 2 would have been very hard to accomplish with a random initialization: most likely a "bad" local minimum would have been reached corresponding to a solution with completely wrong amplitudes and frequencies. Instead, by warm-starting the nonconvex problem with the solution of the LASSO, a "good" local minimum is found.

There are very few nonconvex problems that are not particularly affected by the initialization issue. A lucky case, under appropriate assumptions, is the one of robust principal component analysis (PCA) [113] (Example 4). In general, however, what is typically done is to come up with a good strategy for the initialization. Obviously, the path adopted in Example 2, *i.e.*, initializing the nonconvex problem with the solution of a convex relaxation, is not always accessible. In fact a general rule for initialization does not exist and this is usually problem-dependent: different strategies may involve random initialization using distributions that are obtained by analyzing the available data [114] (Example 3) or by solving multiple times the optimization problems while modifying parameters that govern the nonconvexity (Example 6).

Despite these disadvantages, nonconvex optimization is becoming more and more popular for multiple reasons. Firstly, as Example 2 has just shown, sometimes the quality of the solution of a convex relaxation is not satisfactory. Secondly, convex relaxations may come at the cost of a larger optimization problem with respect to the original nonconvex one [101, 103, 115] and may be prohibitive in terms of both memory and computational power. Finally, sometimes convex relaxations are simply not possible, for example when nonlinear mappings are involved in the optimization problems. These nonlinear mappings are typically derived from complex models which have shown to produce outstanding results and are becoming very common for example in machine learning and model predictive control. For these reasons, having algorithms that are convergent both for convex and nonconvex problems is quite important.

Example 2 Line spectral estimation



Line spectral estimation seeks to accurately recover the frequencies and amplitudes of a signal $\mathbf{y} \in \mathbb{R}^n$ which consists of a mixture of N sinusoids. A simple solution is take the zero-padded DFT of $\mathbf{y}$: $\mathbf{x}_{zp} = \mathsf{F}[\mathbf{y}, 0, \dots 0] \in \mathbb{C}^{sn}$ where s is the super-resolution factor and $\mathsf{F} \in \mathscr{L}(\mathbb{R}^{sn}, \mathbb{C}^{sn})$ the DFT mapping. However, looking at $\mathbf{x}_{zp}$, spectral leakage causes components at close frequencies to merge. This issue is not present if the following optimization problems are used for line spectral estimation:

$$\text{(a) } \mathbf{x}_1^\star = \operatorname*{argmin}_{\mathbf{x}} \tfrac{1}{2}\|\mathsf{S}\mathsf{F}_i\mathbf{x}-\mathbf{y}\|^2+\lambda\|\mathbf{x}\|_1, \quad \text{(b) } \mathbf{x}_0^\star = \operatorname*{argmin}_{\mathbf{x}} \tfrac{1}{2}\|\mathsf{S}\mathsf{F}_i\mathbf{x}-\mathbf{y}\|^2 \text{ s.t. } \|\mathbf{x}\|_0 \leq N, \tag{2.2}$$

here $\mathbf{x} \in \mathbb{C}^{sn}$ consists of the candidate sparse sinusoidal components, $\mathsf{F}_i \in \mathscr{L}(\mathbb{C}^{sn}, \mathbb{R}^{sn})$ is the inverse DFT and $\mathsf{S} \in \mathscr{L}(\mathbb{R}^{sn}, \mathbb{R}^n)$ is a mapping that simply selects the first n elements. Problem (b) is nonconvex which means that it has several local minima and its convex relaxation (a) is typically solved instead (LASSO). Nevertheless, PG methods can solve (a) as well as (b): if a good initialization is given, *e.g.*, the solution of (a), improved results can be achieved.

`StructuredOptimization` code snippet:

```
x = Variable(s*n) # n = 2^8; s = 6;
@minimize ls(ifft(x)[1:n]-y)+lambda*norm(x,1)      # (a)
@minimize ls(ifft(x)[1:n]-y) st norm(x,0) <= N     # (b)
```

2.3 Proximal gradient algorithms

All of the problems this paper treats, including the ones of Examples 1 and 2, can be formulated as

$$\operatorname*{minimize}_{\mathbf{x}} \quad \varphi(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}) \tag{2.3}$$

where f is a smooth function (*i.e.*, it is differentiable, and its gradient ∇f is Lipschitz-continuous), while g is possibly nonsmooth. Despite its simplicity, problem (2.3) encompasses a large variety of applications. For example, constrained optimization can be formulated as (2.3): for a (nonempty) set

$\mathscr{S}$, by setting g to be the *indicator function* of $\mathscr{S}$, that is

$$g(\mathbf{x}) = \delta_{\mathscr{S}}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \mathscr{S}, \\ +\infty & \text{otherwise}, \end{cases} \tag{2.4}$$

then (2.3) is equivalent to minimizing f subject to the constraint $\mathbf{x} \in \mathscr{S}$.

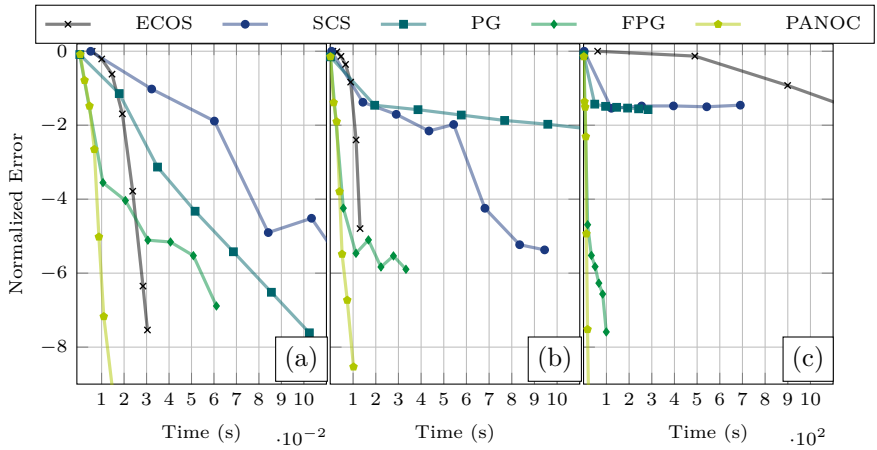


Figure 2.1: Comparison of the performances of different algorithms when solving the LASSO: $\operatorname{argmin}_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1$, where $\mathbf{A} \in \mathbb{R}^{n/5 \times n}$ is a random sparse matrix with $n/4$ non-zero elements and $\lambda = 10^{-3}\|\mathbf{A}^\intercal \mathbf{y}\|_\infty$. Different sizes of the problem are solved using the same random matrices for (a) $n = 10^3$, (b) $n = 10^4$ and (c) $n = 10^5$. Here the normalized error is defined as: $\log(\|\mathbf{x}^k - \mathbf{x}^\star\|_2 / \|\mathbf{x}^\star\|_2)$, where $\mathbf{x}^k$ is the k-th iterate $\mathbf{x}^\star$ the optimal solution.

The presence of a nonsmooth function, like for example the l_1-norm that appeared in the problems encountered so far, prevents from applying classical optimization algorithms such as gradient descent, nonlinear conjugate gradient or (quasi-)Newton methods [77]. These algorithms are in fact based on derivatives and do not apply to the minimization of non-differentiable functions. Although the definition of derivative can be generalized to nonsmooth functions as well through the usage of the *subdifferential*

$$\partial g(\mathbf{x}) = \{\mathbf{v} \mid g(\mathbf{y}) \geq g(\mathbf{x}) + \langle \mathbf{v}, \mathbf{y} - \mathbf{x} \rangle \ \forall \mathbf{y}\}, \tag{2.5}$$

where here g is assumed to be convex, (see [116, Def. 8.3] for the subdifferential definition in the nonconvex case), its usage in these algorithms is often not possible or leads to restrictive convergence properties.

One of the most successful families of algorithms that can deal with nonsmooth cost functions is the one of *interior point methods*. These can in fact be applied

to almost every convex problem by transforming it into a standard problem formulation called *conic form* [84]. Unfortunately, most of these algorithms usually require matrix factorizations and are therefore competitive only for small and medium-sized problems. Figure 2.1 displays this behavior by comparing the time it takes to achieve a specific accuracy of the solution for different sizes of a LASSO problem: here the embedded conic solver (ECOS) algorithm [117], which utilizes a standard path-following interior point method, performs very well for small-scale problems but cannot handle large-scale ones employing quite some time even to reach a solution of low accuracy. In order to overcome this issue while still embracing the large variety of problems that the conic form offers, splitting algorithms have been used also in this context using a variation of ADMM called splitting conic solver (SCS) [118]: as Figure 2.1 shows, this algorithm outperforms standard interior point methods for large-scale problems, reaching a solution of relatively low accuracy but at a much faster rate. However, SCS is in some cases outperformed by PG and FPG, which are introduced later in this section, despite the fact that ADMM has been observed to converge faster in many contexts [81]. This trend reversal of SCS is most likely caused by the transformation of the original problem into its conic form: this changes dramatically the problem structure and introduces additional slack variables which inevitably increase the already large size of the problem. Another advantage of proximal gradient algorithms is their compactness: splitting algorithms like the ADMM or the DR algorithm ultimately require solving large linear systems which often becomes a computational bottleneck. This requires matrix factorizations or subroutines like conjugate gradient methods which are usually not necessary in proximal gradient algorithms. In Section 2.3.4 the results shown in Figure 2.1 will be further discussed.

2.3.1 Proximal mappings

One way to deal with nonsmooth functions in the objective function to be minimized, is through their *proximal mapping* (or *operator*) [119]. For a (possibly nonsmooth) function g, this is defined as

$$\mathbf{z}^\star = \operatorname{prox}_{\gamma g}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{z}} \left\{ g(\mathbf{z}) + \frac{1}{2\gamma}\|\mathbf{z}-\mathbf{x}\|^2 \right\} \tag{2.6}$$

where γ a positive scalar. Here the minimization of g is penalized by the presence of an additional quadratic function that enforces the solution $\mathbf{z}^\star$ to be in the *proximity* of $\mathbf{x}$. Parameter γ controls this proximity and acts as a stepsize: small values of γ will result in $\mathbf{z}^\star$ being very close to $\mathbf{x}$, while large ones will yield a solution close to the minimum of g.

g	$\operatorname{prox}_{\gamma g}$	**Properties**
$\|\mathbf{x}\|_0$	x_i if $\|x_i\| > \sqrt{2\gamma}$, 0 elsewhere $\forall\ i = 1, \dots, n$	nonconvex, separable
$\|\mathbf{x}\|_1$	$\mathcal{P}_+(\mathbf{x} - \gamma) - \mathcal{P}_+(-\mathbf{x} - \gamma)$	convex, separable
$\|\mathbf{x}\|$	$\max\{0, 1 - \gamma/\|\mathbf{x}\|\}\mathbf{x}$	convex
$\|\mathbf{X}\|_*$	$\mathbf{U} \operatorname{diag}(\mathcal{P}_+(\boldsymbol{\sigma} - \gamma))\, \mathbf{V}^{\mathsf{H}}$	convex
$\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$	$(\mathbf{A}^{\mathsf{H}}\mathbf{A} + \gamma^{-1}\mathsf{Id})^{-1}(\mathbf{A}^{\mathsf{H}}\mathbf{b} + \gamma^{-1}\mathbf{x})$	convex
$\mathscr{S}$	$\Pi_{\mathscr{S}}$	
$\{\mathbf{x} \mid \|\mathbf{x}\|_0 \leq m\}$	$\mathcal{P}_m\mathbf{x}$	nonconvex
$\{\mathbf{x} \mid \|\mathbf{x}\| \leq r\}$	$r/\|x\|\mathbf{x}$ if $\|\mathbf{x}\| > r$, $\mathbf{x}$ otherwise	convex
$\{\mathbf{X} \mid \operatorname{rank}(\mathbf{X}) \leq m\}$	$\mathbf{U} \operatorname{diag}(\mathcal{P}_m\boldsymbol{\sigma})\mathbf{V}^{\mathsf{H}}$	nonconvex
$\{\mathbf{x} \mid l \leq \mathbf{x} \leq u\}$	$\min\{u, \max\{l, x_i\}\}\ \forall\ i = 1, \dots, n$	convex, separable
$\{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}\}$	$\mathbf{x} + \mathbf{A}^{\mathsf{H}}(\mathbf{A}\mathbf{A}^{\mathsf{H}})^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x})$	convex

Table 2.1: Table showing the proximal operators of a selection of functions g and indicator function $\delta_{\mathscr{S}}$ with sets $\mathscr{S}$. Here given a n long vector $\mathbf{x}$, $\mathcal{P}_+\mathbf{x}$ returns $[\max\{0, x_1\}, \dots, \max\{0, x_n\}]^{\intercal}$ while $\mathcal{P}_m\mathbf{x}$ returns a copy of $\mathbf{x}$ with all elements set to 0 except for the m largest in modulus. The matrices $\mathbf{U}$ and $\mathbf{V}$ are the result of a SVD: $\mathbf{X} = \mathbf{U} \operatorname{diag}(\boldsymbol{\sigma})\mathbf{V}^{\mathsf{H}}$ where $\boldsymbol{\sigma}$ is the vector containing the singular values of $\mathbf{X}$.

	$g(\mathbf{x})$	$\operatorname{prox}_{\gamma g}(\mathbf{x})$	**Requirements**
Separable sum	$h_1(\mathbf{x}_1) + h_2(\mathbf{x}_2)$	$[\operatorname{prox}_{\gamma h_1}(\mathbf{x}_1)^{\intercal}, \operatorname{prox}_{\gamma h_2}(\mathbf{x}_2)^{\intercal}]^{\intercal}$	$\mathbf{x} = [\mathbf{x}_1^{\intercal}, \mathbf{x}_2^{\intercal}]^{\intercal}$
Translation	$h(\mathbf{x} + \mathbf{b})$	$\operatorname{prox}_{\gamma h}(\mathbf{x} + \mathbf{b}) - \mathbf{b}$	
Affine addition	$h(\mathbf{x}) + \langle \mathbf{a}, \mathbf{x} \rangle$	$\operatorname{prox}_{\gamma h}(\mathbf{x} - \gamma\mathbf{a})$	
Postcomposition	$ah(\mathbf{x}) + b$	$\operatorname{prox}_{a\gamma h}(\mathbf{x})$	$a > 0$
Precomposition	$h(\mathsf{A}\mathbf{x})$	$\mathbf{x} + \mu^{-1}\mathsf{A}^*\left(\operatorname{prox}_{\mu\gamma h}(\mathsf{A}\mathbf{x}) - \mathsf{A}\mathbf{x}\right)$	$\mathsf{A}\mathsf{A}^* = \mu\mathsf{Id}$, $\mu \geq 0$
Regularization	$h(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{x} - \mathbf{b}\|^2$	$\operatorname{prox}_{\tilde{\gamma}h}(\tilde{\gamma}(1/\gamma\mathbf{x} + \rho\mathbf{b}))$	$\tilde{\gamma} = \gamma/(1 + \gamma\rho)$, $\rho \geq 0$
Convex Conjugate	$\sup_{\mathbf{x}}\{\langle \mathbf{x}, \mathbf{u} \rangle - h(\mathbf{x})\}$	$\mathbf{u} - \gamma \operatorname{prox}_{(1/\gamma)h}(\mathbf{u}/\gamma)$	h convex

Table 2.2: Table showing different properties of proximal mappings.

For many functions the correspondent proximal mappings have closed-form solutions and can be computed very efficiently. Table 2.1 shows some examples for functions which are commonly used in applications. For example, the proximal mapping of $g(\cdot) = \lambda\|\cdot\|_1$, consists of a "soft-thresholding" operation of $\mathbf{x}$, while for l_0-norm it is the so-called "hard-thresholding" operation. When g is the indicator function of a set $\mathscr{S}$, cf. (2.4), then $\text{prox}_{\gamma g} = \Pi_{\mathscr{S}}$, the projection onto $\mathscr{S}$. As Table 2.1 shows, many of these projections are also cheaply computable like in the case of boxes, norm balls, affine subspaces.

However, an analytical solution to (2.6) is not always available. For example, given two functions h_1 and h_2, the fact that $\text{prox}_{\gamma h_1}$ and $\text{prox}_{\gamma h_2}$ can be efficiently computed does not necessarily imply that $\text{prox}_{\gamma(h_1+h_2)}$ is efficient as well. Additionally, the proximal mapping of the composition $g \circ \mathsf{A}$ of a function g with a linear operator A, is also not efficient in general. An exception to this is linear least squares: if $g(\cdot) = \frac{1}{2}\|\cdot - \mathbf{b}\|^2$ is composed with matrix $\mathbf{A}$, the proximal mapping of function $g(\mathbf{A}\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ has in fact a closed-form solution, which however requires solving a linear system as Table 2.1 shows. When this linear system is large (*i.e.*, $\mathbf{A}$ has a large number of columns) such inversion may be infeasible to tackle with direct methods (such as QR decomposition or Cholesky factorization), and one may need to resort to iterative algorithms, *e.g.*, using conjugate gradient. In general, composition by a linear mapping results in an efficient closed-form proximal mapping only when A satisfies $\mathsf{A}\mathsf{A}^* = \mu\mathsf{Id}$ where $\mu \geq 0$, Id is the identity mapping and A^* is the adjoint mapping of A (see Section 2.4 for the definition of adjoint mapping). Linear mappings with such properties are called *tight frames*, and include orthogonal mappings like the discrete Fourier transform (DFT) and discrete cosine transform (DCT).

Many properties can be exploited to derive closed-form expressions for proximal operators: Table 2.2 summarizes some of the most important ones [80]. Among these, the separable sum is particularly useful: if h_1 and h_2 have efficiently computable proximal mappings, then so does function $g(\mathbf{x}_1, \mathbf{x}_2) = h_1(\mathbf{x}_1) + h_2(\mathbf{x}_2)$. For example, using the properties in Table 2.2, it is very easy to compute the proximal mapping of $g(x) = \|\text{diag}(\mathbf{d})\mathbf{x}\|_1 = \sum |d_i x_i|$.

If function g is convex then $\text{prox}_{\gamma g}$ is everywhere well defined: (2.6) consists of the minimization of a strongly convex objective, and as such has a unique solution for any $\mathbf{x}$. When g is nonconvex, however, this condition fails to hold. Existence of solutions to (2.6) in this case is guaranteed (possibly for small enough γ) under *prox-boundedness* assumptions on g: this is a rather technical assumption which roughly speaking amounts to g being bounded below by a (possibly nonconvex) quadratic function. The interested reader can refer to [116, Def. 1.23] for its formal definition.

In general, for nonconvex g the operator $\text{prox}_{\gamma g}$ is *set-valued*, *i.e.*, problem

(2.6) may have multiple solutions. As an example of this, consider set $\mathscr{B}_{0,m} = \{\mathbf{x} \mid \|\mathbf{x}\|_0 \leq m\}$, *i.e.*, the ℓ_0 pseudo-norm ball. Projecting a point $\mathbf{x} \in \mathbb{R}^n$ onto $\mathscr{B}_{0,m}$ amounts to setting to zero its $n - m$ smallest coefficients in magnitude. However consider $n = 5$, $m = 3$, and point $\mathbf{x} = [5.7, -2.4, 1.2, 1.2, 1.2]^\intercal$. In this case there are three points in $\mathscr{B}_{0,3}$ which are closest to $\mathbf{x}$. In fact:

$$\Pi_{\mathscr{B}_{0,3}}(\mathbf{x}) = \{[5.7, -2.4, 1.2, 0, 0]^\intercal, [5.7, -2.4, 0, 1.2, 0]^\intercal, [5.7, -2.4, 0, 0, 1.2]^\intercal\}. \tag{2.7}$$

In practice, proximal mappings of nonconvex functions are evaluated by choosing a single element out of its set.

2.3.2 Proximal gradient method

A very popular algorithm to solve (2.3), when φ is the sum of a smooth function f and a (possibly) nonsmooth function g with efficient proximal mapping, is the *proximal gradient (PG)* algorithm: this combines the gradient descent, a well known first-order method, with the proximal mapping described in Section 2.3.1.

Algorithm 1 Proximal Gradient Algorithm (PG)

Set $\mathbf{x}^0 \in \mathbb{R}^n$
Set $\gamma \in (0, 2L_f^{-1}]$
for $k = 0, 1, \ldots$ **do**
    $\mathbf{x}^{k+1} = \text{prox}_{\gamma g}(\mathbf{x}^k - \gamma \nabla f(\mathbf{x}^k))$

The PG algorithm is illustrated in Algorithm 1: here $\mathbf{x}^0$ is the initial guess, γ represents a stepsize and ∇f is the gradient of the smooth function f. The algorithm consists of alternating gradient (or *forward*) steps on f and proximal (or *backward*) steps on g, and is a particular case of the forward-backward splitting (FBS) algorithm for finding a zero of the sum of two monotone operators [120, 121]. The reason behind this terminology is apparent from the optimality condition of the problem defining the proximal operator (2.6): if $\mathbf{z} = \text{prox}_{\gamma g}(\mathbf{x})$, then necessarily $\mathbf{z} = \mathbf{x} - \gamma \mathbf{v}$, with $\mathbf{v} \in \partial g(\mathbf{x})$, *i.e.*, $\mathbf{z}$ is obtained by an *implicit* (backward) subgradient step over g, as opposed to the explicit (forward) step over f.

The steps of the algorithm are visualized in Figure 2.2: the gradient step moves the iterate $\mathbf{x}^k$ towards the minimum of f, while the proximal step makes progress towards the minimum of g. This alternation will ultimately lead to the minimum of the sum of these two functions. In fact, in the convex case (*i.e.*, when both f and g are convex), the iterates $\mathbf{x}^k$ in Algorithm 1 are known to converge under minimal assumptions to a global minimum, for $\gamma \in (0, 2/L_f)$ where L_f is a Lipschitz constant of ∇f, see [80, Cor. 27.9]. Furthermore, in this case the

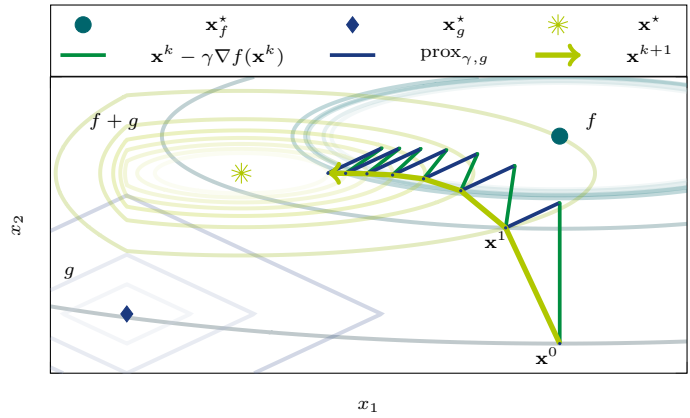


Figure 2.2: Figure showing an example of the path that the PG algorithm creates to reach the optimal value $\mathbf{x}^\star$. Here the minima of the functions f and g are shown in $\mathbf{x}_f^\star$ and $\mathbf{x}_g^\star$ respectively.

algorithm converges with global *sublinear rate*: indicating by $\varphi_\star$ the optimal value of (2.3), the iterates of Algorithm 1 satisfy $\varphi(\mathbf{x}^k) - \varphi_\star \in O(1/k)$ [91]. Notice that when L_f is not available, γ can be adaptively estimated by means of a backtracking procedure [91]. Convergence of Algorithm 1 has also been studied in the nonconvex case (*i.e.*, where both f and g are allowed to be nonconvex): in this case, sublinear convergence to a *local* minimum can be generally expected [99].

Algorithm 2 Fast proximal gradient algorithm (FPG) [91]

Set $\mathbf{v}^0 = \mathbf{x}^{-1} \in \mathbb{R}^n$
Set $\gamma \in (0, L_f^{-1}]$ and $\theta_0 = 1$
for $k = 0, 1, \ldots$ **do**
  $\mathbf{x}^k = \mathrm{prox}_{\gamma g}(\mathbf{v}^k - \gamma\nabla f(\mathbf{v}^k))$
  $\theta_{k+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4\theta_k^2}\right)$
  $\mathbf{v}^{k+1} = \mathbf{x}^k + (\theta_k - 1)\theta_{k+1}^{-1}(\mathbf{x}^k - \mathbf{x}^{k-1})$

Fast variants of the algorithm exist, such as the fast proximal gradient (FPG) algorithm (also known as fast iterative shrinkage-thresholding algorithm (FISTA) [91]), shown in Algorithm 2: this is an extension of the optimal first-order methods for convex smooth problems, pioneered by Nesterov [90], to the case where the additional nonsmooth function g is present.

In addition to the original iterates $\mathbf{x}^k$, FPG computes an extrapolated sequence $\mathbf{v}^k$ by performing a linear combination of previous two iterates. Intuitively, this provides *intertia* to the computed sequence, which improves the convergence speed over PG: in fact, the particular choice of the coefficients of such inertial step ensure that in the convex case the iterates of Algorithm 2 satisfy $\varphi(\mathbf{x}^k) - \varphi_\star \in O(1/k^2)$ [91, 122, 123]. This method is particularly appealing since the extrapolated sequence $\mathbf{v}^k$ only requires $O(n)$ floating point operations to be computed. However, the convergence of FPG has only been proven when both f and g are convex.

2.3.3 Forward-backward envelope

Recently, new algorithms based on the PG algorithm have emerged: these rely on the concept of the *forward-backward envelope (FBE)* which was first introduced in [124]. In order to explain what the FBE is, one should look at the PG algorithm from a different perspective. Using the definition of $\operatorname{prox}_{\gamma g}$, with elementary manipulations the iterations of Algorithm 1 can be equivalently rewritten as,

$$\mathbf{x}^{k+1} = \operatorname*{argmin}_{\mathbf{z}} \big\{ \overbrace{f(\mathbf{x}^k) + \langle \mathbf{z} - \mathbf{x}^k, \nabla f(\mathbf{x}^k) \rangle + \tfrac{1}{2\gamma}\|\mathbf{z} - \mathbf{x}^k\|^2}^{q_\gamma(\mathbf{z}, \mathbf{x}^k)} + g(\mathbf{z}) \big\}, \tag{2.8}$$

that is, the minimization of g plus a *quadratic model* $q_\gamma(\mathbf{z}, \mathbf{x}^k)$ of f around the current iterate $\mathbf{x}^k$. When ∇f is Lipschitz continuous and $\gamma \leq L_f^{-1}$, then for all $\mathbf{x}$

$$\varphi(\mathbf{z}) = f(\mathbf{z}) + g(\mathbf{z}) \leq q_\gamma(\mathbf{z}, \mathbf{x}) + g(\mathbf{z}). \tag{2.9}$$

In this case the steps (2.8) of the PG algorithm are a *majorization-minimization* procedure. This is visualized, in the one-dimensional case, in Figure 2.3. The minimum *value* of (2.8) is the *forward-backward envelope* associated with problem (2.3), indicated by φ_γ:

$$\varphi_\gamma(\mathbf{x}) = \min_{\mathbf{z}} \left\{ f(\mathbf{x}) + \langle \mathbf{z} - \mathbf{x}, \nabla f(\mathbf{x}) \rangle + \tfrac{1}{2\gamma}\|\mathbf{z} - \mathbf{x}\|^2 + g(\mathbf{z}) \right\}. \tag{2.10}$$

The FBE has many noticeable properties: these are described in detail in [82] for the case where g is convex, and extended in [83] to the case where g is allowed to be nonconvex. First, φ_γ is a lower bound to φ, and the two functions share the same local minimizers. In particular, $\inf \varphi = \inf \varphi_\gamma$ and $\operatorname{argmin} \varphi = \operatorname{argmin} \varphi_\gamma$, hence minimizing φ_γ is equivalent to minimizing φ. Additionally, φ_γ is real-valued as opposed to φ which is *extended* real-valued: as Figure 2.3 shows, even

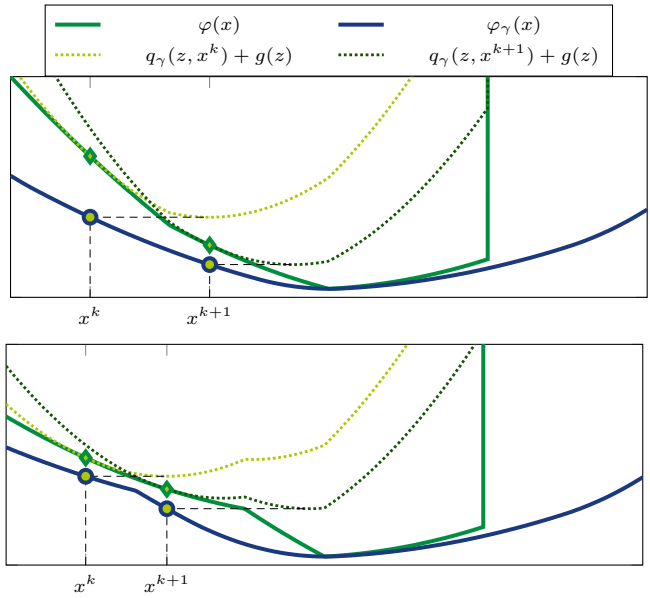


Figure 2.3: One step of PG amounts to a majorization-minimization over φ, when stepsizes γ is sufficiently small. The minimum value of such majorization is the forward-backward envelope φ_γ. Left: in the convex case, φ_γ is a smooth lower bound to the original objective φ. Right: in the nonconvex case φ_γ is not everywhere smooth.

at points where φ is $+\infty$, φ_γ has a finite value instead. Furthermore, when f is twice differentiable and g is convex, then φ_γ is continuously differentiable.

Finally, it is worth noticing that evaluating the FBE (2.10) essentially requires computing one proximal gradient step, *i.e.*, one step of Algorithm 1. This is an important feature from the algorithmic perspective: any algorithm that solves (2.3) by minimizing φ_γ (and thus needs its evaluation) requires exactly the same operations as Algorithm 1. In the next section one of such algorithms is illustrated.

2.3.4 Newton-type proximal gradient methods

In Section 2.3.3 it was shown that minimizing the FBE is equivalent to solving problem (2.3). Algorithm 3 is a generalization of the standard PG algorithm that minimizes the FBE using a backtracking line search.

Algorithm 3 Proximal Averaged Newton-type algorithm for Optimality Conditions (PANOC)

Set $\mathbf{x}^0 \in \mathbb{R}^n$
Set $\gamma \in (0, L_f^{-1}]$ and $\sigma \in (0, \frac{1}{2\gamma}(1-\gamma L_f))$
for $k = 0, 1, \ldots$ **do**
  $\mathbf{v}^k = \mathrm{prox}_{\gamma g}(\mathbf{x}^k - \gamma \nabla f(\mathbf{x}^k))$
  $\mathbf{d}^k = -\mathsf{H}_k(\mathbf{x}^k - \mathbf{v}^k)$ for some nonsingular H_k
  $\mathbf{x}^{k+1} = (1-\tau_k)\mathbf{v}^k + \tau_k(\mathbf{x}^k + \mathbf{d}^k)$,
  where τ_k is the largest value in $\left\{(\frac{1}{2})^i \mid i \in \mathbb{N}\right\}$ such that

$$\varphi_\gamma(\mathbf{x}^{k+1}) \leq \varphi_\gamma(\mathbf{x}^k) - \sigma\|\mathbf{v}^k - \mathbf{x}^k\|^2$$

The *proximal averaged Newton-type algorithm for optimality conditions (PANOC)* was proposed in [125], and the idea behind it is very simple: the PG algorithm is a fixed-point iteration for solving the system of nonsmooth, nonlinear equations $\mathcal{R}_\gamma(\mathbf{x}) = \mathbf{0}$, where

$$\mathcal{R}_\gamma(\mathbf{x}) = \mathbf{x} - \mathrm{prox}_{\gamma g}(\mathbf{x} - \gamma \nabla f(\mathbf{x})), \tag{2.11}$$

is the *fixed-point residual* mapping. In fact, it is immediate to verify that the iterates in Algorithm 1 satisfy

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathcal{R}_\gamma(\mathbf{x}^k). \tag{2.12}$$

It is very natural to think of applying a Newton-type method, analogously to what is done for smooth, nonlinear equations [77, Chap. 11]:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathsf{H}_k\mathcal{R}_\gamma(\mathbf{x}^k), \tag{2.13}$$

where (H_k) is an appropriate sequence of nonsingular linear transformations. The update rule of Algorithm 3 is a convex combination of (2.12) and (2.13), dictated by the stepsize τ_k which is determined by backtracking line-search over the FBE. When $\tau_k = 1$ then (2.13) is performed; as $\tau_k \to 0$ then the update gets closer and closer to (2.12).

Note that Algorithm 3 reduces to Algorithm 1 for the choice $\mathsf{H}_k = \mathsf{Id}$: in this case the stepsize τ_k is always equal to 1. However, by carefully choosing H_k one can greatly improve convergence of the iterations. In [125] the case of *quasi-Newton* methods is considered: start with $\mathsf{H}_0 = \mathsf{Id}$, and update it so as to satisfy the *(inverse) secant condition*

$$\mathbf{x}^{k+1} - \mathbf{x}^k = \mathsf{H}_{k+1}\left[\mathcal{R}_\gamma(\mathbf{x}^{k+1}) - \mathcal{R}_\gamma(\mathbf{x}^k)\right]. \tag{2.14}$$

This can be achieved via the (modified) Broyden method in which case the resulting sequence of H_k can be proven to satisfy the so called *Dennis-Moré condition* which ensures superlinear convergence of the iterates $\mathbf{x}^k$. A detailed convergence proof of PANOC can be found in [125].

Algorithm 4 L-BFGS two-loop recursion with memory M

Set for $i = k - M, \dots, k-1$ $\begin{cases} \mathbf{s}^i = \mathbf{x}^{i+1} - \mathbf{x}^i \\ \mathbf{w}^i = \mathcal{R}_\gamma(\mathbf{x}^{i+1}) - \mathcal{R}_\gamma(\mathbf{x}^i) \\ \rho_i = \langle \mathbf{s}^i, \mathbf{w}^i \rangle \end{cases}$
Set $H = \rho_{k-1}/\langle \mathbf{w}^{k-1}, \mathbf{w}^{k-1} \rangle$, $\mathbf{d}^k = -\mathcal{R}_\gamma(\mathbf{x}^k)$
for $i = k-1, \dots, k-M$ **do**
    $\alpha_i \leftarrow \langle \mathbf{s}^i, \mathbf{d}^k \rangle / \rho_i$
    $\mathbf{d}^k \leftarrow \mathbf{d}^k - \alpha_i \mathbf{w}^i$
$\mathbf{d}^k \leftarrow H \mathbf{d}^k$
for $i = k-M, k-M+1, \dots, k-1$ **do**
    $\beta_i \leftarrow \langle \mathbf{w}^i, \mathbf{d}^k \rangle / \rho_i$
    $\mathbf{d}^k \leftarrow \mathbf{d}^k + (\alpha_i - \beta_i)\mathbf{s}^i$

Using full quasi-Newton updates requires computing and storing n^2 coefficients at every iteration of Algorithm 3, where n is the problem dimension. This is of course impractical for n larger than a few hundreds. Therefore, limited-memory methods such as L-BFGS can be used: this computes directions $\mathbf{d}^k$ using $O(n)$ operations [77], and is thus well suited for large-scale problems. Algorithm 4 illustrates how the L-BFGS method can be used in the context of Algorithm 3 to compute directions: at each iteration, the M most recent pairs of vectors $\mathbf{s}^i = \mathbf{x}^{i+1} - \mathbf{x}^i$ and $\mathbf{w}^i = \mathcal{R}_\gamma(\mathbf{x}^{i+1}) - \mathcal{R}_\gamma(\mathbf{x}^i)$ are collected, and are used to compute the product $\mathsf{H}_k \mathcal{R}_\gamma(\mathbf{x}^k)$ implicitly (*i.e.*, without ever storing the full operator H_k in memory) for an operator H_k that approximately satisfies (2.14).

In Tables 2.3 and 2.4 comparisons between the different proximal gradient algorithms are shown. In most of the cases, the PANOC algorithm outperforms the other proximal gradient algorithms. It is worth noticing that its performance is sometimes comparable to the one of the FPG algorithm: Example 5 is a case when this is particularly evident. Although PANOC requires less iterations than FPG, as Table 2.3 shows, these are more expensive as they perform the additional backtracking line-search procedure. Example 5, which treats the an application of image processing, actually requires a low tolerance to achieve a satisfactory solution. It is in these particular cases, that (fast) PG becomes very competitive with PANOC: this is quite evident also in Figure 2.1 as it can be seen that for low accuracies of the solution the performance of FPG and PANOC is very similar. Of course, these observations are problem-dependent,

		PG	FPG	PANOC
DNN classifier (Ex. 3) $n = 73$, $\epsilon = 10^{-4}$ nonconvex	t	131.8	n/a	7.6
	k	50000	n/a	1370
Robust PCA (Ex. 4) $n = 3225600$, $\epsilon = 10^{-4}$ nonconvex	t	92.8	n/a	38.5
	k	697	n/a	81
Total variation (Ex. 5) $n = 524288$, $\epsilon = 10^{-3}$ convex	t	8.2	4.3	11.2
	k	582	278	259
Audio de-clipping (Ex. 6) $n = 2048$, $\epsilon = 10^{-5}$ nonconvex	t	368.5	n/a	66.2
	k	8908	n/a	732

Table 2.3: Table comparing the time t (in s) and the number of iterations k (mean per frame for Example 6) needed to solve the different examples using proximal gradient algorithms. The value n indicates the number of optimization variables of each problem and $\epsilon = \|\mathcal{R}_\gamma(\mathbf{x}^k)\|_\infty/\gamma$ the stopping criteria tolerance.

and one should always verify empirically which algorithm better performs in the specific application.

2.4 Matrix-free optimization

In all of the algorithms described in Section 2.3 the gradient of f must be computed at every iteration. This operation is therefore fundamental as it can dramatically affect the overall performance of proximal gradient algorithms. Consider the cost functions of Examples 1 and 2: in both cases the data fidelity function f consists of the composition of the squared norm with a linear mapping A. These linear mappings need to be evaluated numerically by means of a specified algorithm. A simple and very versatile algorithm that works for both examples, consists of performing a *matrix-vector product*. The function f can then be written as

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2, \tag{2.15}$$

i.e., the cost function of a linear least squares problem. In Example 1, $\mathbf{A} \in \mathbb{R}^{m\times n}$ would be a Toeplitz matrix whose columns contain shifted versions of the FIR $\mathbf{h}$. Instead, in Example 2, $\mathbf{A}$ would correspond to a complex-valued matrix containing complex exponentials resulting from the inverse DFT. By applying

the *chain rule* the gradient of (2.15) at the iterate $\mathbf{x}^k$ reads:

$$\nabla f(\mathbf{x}^k) = \mathbf{A}^{\mathsf{H}} \left(\mathbf{A}\mathbf{x}^k - \mathbf{y} \right), \tag{2.16}$$

where ^{H} indicates the conjugate-transpose operation. If $\mathbf{A}$ is a dense matrix, evaluating (2.16) then takes two matrix-vector products each one having a complexity $O(mn)$. Moreover $\mathbf{A}$ must be stored and this occupies $O(mn)$ bytes despite the redundancy of the information it carries.

Using a matrix-vector product as an algorithm to perform discrete convolution or an inverse DFT is actually not the best choice: it is well known that there exist a variety of algorithms capable of outperforming the matrix-vector product algorithm. For example, discrete convolution can be performed with a complexity of $O(n \log n)$ by transforming the signals $\mathbf{h}$ and $\mathbf{x}^k$ into the frequency domain, multiplying them and transforming the result back into the time domain. The memory requirements are also lower since now only the $O(n)$ bytes of the FIR need to be stored. When $\mathbf{A}$ represents convolution, its conjugate-transpose matrix-vector product appearing in (2.16), corresponds to a cross-correlation: this operation can also be evaluated with the very same complexity and memory requirements as the convolution. Cross-correlation is in fact the *adjoint mapping* of convolution [126].

In general, given a linear mapping $\mathsf{A} \in \mathscr{L}(\mathtt{D}, \mathtt{C})$ the adjoint mapping is its uniquely associated linear mapping $\mathsf{A}^* \in \mathscr{L}(\mathtt{C}, \mathtt{D})$. In this context A is often called *forward mapping*. Formally, the adjoint mapping is defined by the equivalence of these inner products

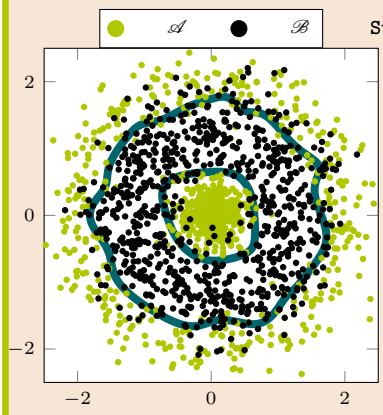
$$\langle \mathsf{A}\mathbf{x}, \mathbf{y} \rangle_{\mathtt{C}} = \langle \mathbf{x}, \mathsf{A}^*\mathbf{y} \rangle_{\mathtt{D}} \ \forall \ \mathbf{x} \in \mathtt{D}, \ \forall \ \mathbf{y} \in \mathtt{C}. \tag{2.17}$$

The adjoint mapping A^* generalizes the conjugate-transpose operation and like A, it can be evaluated using different algorithms. It is now possible to define the *forward-adjoint oracles (FAOs)* of a linear mapping A: these oracles consist of two specific black-box algorithms that are used to compute the forward mapping A and its associated adjoint mapping A^* respectively. Avoiding the use of matrices in favor of FAOs for the evaluation of the linear mappings leads to *matrix-free optimization*. Table 2.4 shows the improvements in terms of computational time with respect to using matrices: clearly, for the aforementioned reasons, solving the optimization problems of Examples 1 and 2 using matrix-free optimization is substantially faster.

2.4.1 Directed acyclic graphs

Being generalizations of matrices, linear mappings share many features with them. For example it is possible to horizontally or vertically concatenate linear

Example 3 Deep Neural Network Classifier



StructuredOptimization code snippet:

```
# W1, W2, W3, b1, b2, b3 are variables
# S1, S2, S3 are sigmoid operators

L1 = S1*(W1* D.+b1) # input  layer
L2 = S2*(W2*L1.+b2) # inner  layer
y  = S3*(W3*L2.+b3) # output layer
# regularization
reg = lambda1*norm(W1,1)+
      lambda2*norm(W2,1)+
      lambda3*norm(W3,1)

@minimize crossentropy(y,yt)+reg
```

A deep neural network is a relatively simple model that, by composing multiple linear transformations and nonlinear *activation functions*, allows obtaining highly nonlinear mappings that can be used for classification or regression tasks [114]. This is achieved by *training* the network, *i.e.*, by finding the optimal parameters (the coefficients of the linear transformations) with respect to a *loss* function and some training data, and amounts to solving a highly nonconvex optimization problem. In this example, three layers are combined to perform classification of data points into two sets $\mathscr{A}$ and $\mathscr{B}$ depicted in the figure above. The following optimization problem is solved to train the network:

$$\underset{\mathbf{W}_1,\mathbf{W}_2,\mathbf{W}_3,b_1,b_2,b_3}{\text{minimize}} \quad \overbrace{-\sum_{n=1}^{N}(\tilde{y}_n\log(y_n)+(1-\tilde{y}_n)\log(1-y_n))}^{f}+\sum_{k=1}^{3}\overbrace{\lambda_k\|\text{vec}(\mathbf{W}_k)\|_1}^{g}$$

$$\text{subject to} \quad \mathbf{y}=\mathcal{S}_3(\mathbf{W}_3\mathbf{L}_2+b_3),\ \mathbf{L}_2=\mathcal{S}_2(\mathbf{W}_2\mathbf{L}_1+b_2),\ \mathbf{L}_1=\mathcal{S}_1(\mathbf{W}_1\mathbf{D}+b_1). \tag{2.18}$$

Here $\mathbf{D}\in\mathbb{R}^{N\times 2}$ are the training data points with binary labels (0 for $\mathscr{A}$ and 1 for $\mathscr{B}$) stored in $\tilde{\mathbf{y}}$. $\mathbf{W}_i$ and b_i are the weights and biases of the i-th layer which combination outputs $\mathbf{y}\in\mathbb{R}^N$. This output is fitted through the usage of a cross-entropy loss function f [114] to the labels $\tilde{\mathbf{y}}$. The nonlinear mappings $\mathcal{S}_i$ are sigmoid functions modeling the activations of the neurons. A regularization function g is added to simultaneously prevent over-fitting while enforcing the weights to be sparse matrices. Contour lines show the classifier obtained after the training.

mappings that share their codomain or domain respectively. Additionally, it is possible to compose linear mappings, *e.g.*, $\mathsf{A}\in\mathscr{L}(\mathtt{K},\mathtt{C})$ can be composed with $\mathsf{B}\in\mathscr{L}(\mathtt{D},\mathtt{K})$ to construct $\mathsf{AB}\in\mathscr{L}(\mathtt{D},\mathtt{C})$. Although conceptually equivalent, these and other *calculus rules* are implemented in a substantially different fashion to what is typically done with matrices. With matrices the application

	Using Matrices			Matrix-Free		
	PG	FPG	PANOC	PG	FPG	PANOC
Sparse Deconvolution (Ex. 1)	1174	520	360	253	127	89
Line Spectral Estimation (Ex. 2)	2773	1089	237	1215	489	108

Table 2.4: Table comparing the time (in ms) that different PG algorithms employ to solve the optimization problems of Examples 1 and 2 using matrices or matrix-free optimization.

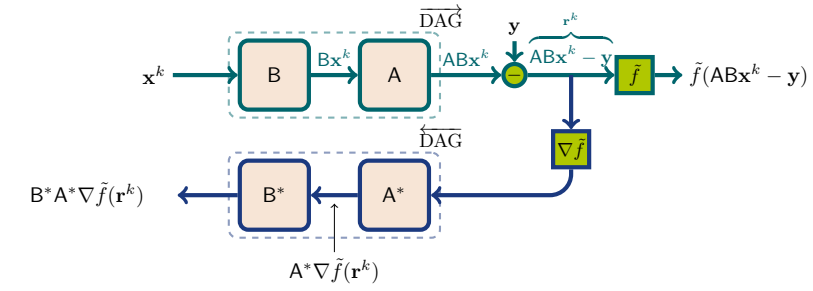
of a calculus rule results in a new and independent matrix which is then used to evaluate the forward and adjoint mappings through matrix-vector products. On the contrary, combinations of linear mappings that are evaluated through FAOs constitute a directed acyclic graph (DAG) which preserves the structure of the calculus rules involved. Each node of these graphs is associated with a particular mapping and the calculus rules are applied according to the way these nodes are connected. It is actually convenient to use these DAGs to evaluate the cost function together with its gradient, a strategy that is known as *automatic differentiation* in numerical analysis [127] and *back-propagation* in machine learning [114].

Figure 2.4a illustrates these concepts using a simple example of composition of two linear mappings. Here the gradient of the function $f(\mathbf{x}) = \tilde{f}(\mathsf{A}\mathsf{B}\mathbf{x} - \mathbf{y})$ is needed. The gradient of this function at $\mathbf{x}^k$ reads:

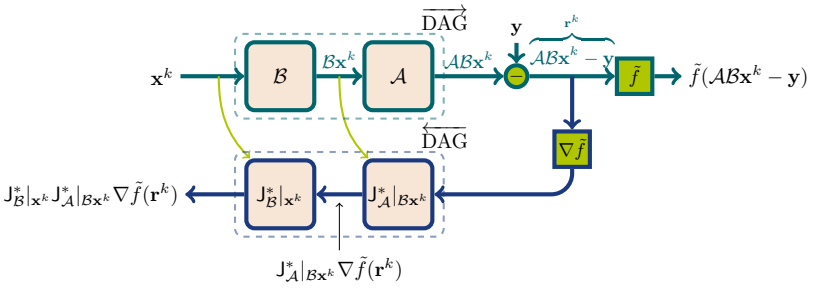
$$\nabla f(\mathbf{x}^k) = \mathsf{B}^*\mathsf{A}^*\nabla \tilde{f}(\mathsf{A}\mathsf{B}\mathbf{x}^k - \mathbf{y}), \tag{2.19}$$

and can be conveniently computed alongside the evaluation of $f(\mathbf{x}^k)$. The iterate $\mathbf{x}^k$ is initially "sent" through what is referred to here as the *forward DAG*: here the linear mappings B and A are applied in series to $\mathbf{x}^k$ using the corresponding forward oracles. After this, the *residual* $\mathbf{r}^k = \mathsf{A}\mathsf{B}\mathbf{x}^k - \mathbf{y}$ can be computed. This can be readily used not only to compute $f(\mathbf{x}^k)$, but also to obtain the gradient: in fact after applying the gradient of $\tilde{f}$ to $\mathbf{r}^k$ the result is "sent back" through the *backward DAG*. This differs from the forward DAG since now the adjoint oracles of A and B are applied in a reversed order.

Similarly to Figure 2.4, in Example 2 two linear mappings were composed. The first linear mapping $\mathsf{F}_i \in \mathscr{L}(\mathbb{C}^{sn}, \mathbb{R}^{sn})$ consists of an inverse DFT. Its forward oracle is an inverse FFT while its adjoint oracle is a non-normalized FFT. The linear mapping $\mathsf{S} \in \mathscr{L}(\mathbb{R}^{sn}, \mathbb{R}^n)$ converts the high resolution signal into a low resolution one. Its FAOs are extremely simple algorithms: the forward oracle selects the first n elements while the adjoint oracle performs the opposite, zero-padding its input.



(a)



(b)

Figure 2.4: Forward and backward DAGs used to evaluate the gradient of a function composed with (a) linear mappings or (b) nonlinear mappings.

So far only linear mappings were considered, but smooth nonlinear mappings can be combined as well using FAOs and DAGs. In fact when nonlinear mappings appear in f, this function and its gradient can be evaluated using an analogous strategy to the one described earlier. The main difference lies in the fact that the adjoint operator of a nonlinear mapping does not exist. However a nonlinear mapping $\mathcal{A} : \mathtt{D} \to \mathtt{C}$ can be linearized by differentiating it and obtaining a *linear* mapping called *Jacobian mapping* for which here the following notation is used: $\mathsf{J}_{\mathcal{A}}|_{\mathbf{x}^k} \in \mathscr{L}(\mathtt{D}, \mathtt{C})$ where $|_{\mathbf{x}^k}$ is used to indicate the point of the linearization. Using again the same example, this time with nonlinear mappings, the chain rule is again applied to $f(\mathbf{x}) = \tilde{f}(\mathcal{A}\mathcal{B}\mathbf{x} - \mathbf{y})$:

$$\nabla f(\mathbf{x}^k) = \mathsf{J}^*_{\mathcal{B}}|_{\mathbf{x}^k} \mathsf{J}^*_{\mathcal{A}}|_{\mathcal{B}\mathbf{x}^k} \nabla \tilde{f}(\mathcal{A}\mathcal{B}\mathbf{x}^k - \mathbf{y}). \tag{2.20}$$

Here, and visually in Figure 2.4b, it can be seen that the main difference with (2.19) and Figure 2.4a, is that the adjoint mappings are replaced by the adjoint Jacobian mappings of $\mathcal{A}$ and $\mathcal{B}$ linearized at $\mathcal{B}\mathbf{x}^k$ and $\mathbf{x}^k$ respectively. These quantities are already available since they are computed during the forward DAG evaluation: if these are stored during this phase they can be used later when

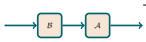
Rule	Input Mappings	Output Mapping	Chain Rule
Composition	$\mathtt{A} \in \mathscr{L}(\mathtt{K},\mathtt{C})$, $\mathtt{B} \in \mathscr{L}(\mathtt{D},\mathtt{K})$	$\mathtt{AB} \in \mathscr{L}(\mathtt{D},\mathtt{C})$	$\nabla(f(\mathtt{AB}\mathbf{x}^k)) = \mathtt{B}^*\mathtt{A}^*\nabla f(\mathtt{AB}\mathbf{x}^k)$
	$\mathcal{A} : \mathtt{K} \to \mathtt{C}$, $\mathcal{B} : \mathtt{D} \to \mathtt{K}$	$\mathcal{AB} : \mathtt{D} \to \mathtt{C}$	$\nabla(f(\mathcal{AB}\mathbf{x}^k)) = \mathtt{J}^*_{\mathcal{B}}\vert_{\mathbf{x}^k}\mathtt{J}^*_{\mathcal{A}}\vert_{\mathcal{B}\mathbf{x}^k}\nabla f(\mathcal{AB}\mathbf{x}^k)$
Horizontal Concatenation 	$\mathtt{A} \in \mathscr{L}(\mathtt{D},\mathtt{C})$, $\mathtt{B} \in \mathscr{L}(\mathtt{K},\mathtt{C})$	$[\mathtt{A},\mathtt{B}] \in \mathscr{L}(\mathtt{D}\times\mathtt{K},\mathtt{C})$	$\nabla(f(\mathtt{A}\mathbf{x}_1^k + \mathtt{B}\mathbf{x}_2^k)) = [\mathtt{A}^*\nabla f(\mathbf{r}^k), \mathtt{B}^*\nabla f(\mathbf{r}^k)]$
	$\mathcal{A} : \mathtt{D} \to \mathtt{C}$, $\mathcal{B} : \mathtt{K} \to \mathtt{C}$	$[\mathcal{A},\mathcal{B}] : \mathtt{D}\times\mathtt{K} \to \mathtt{C}$	$\nabla(f(\mathcal{A}\mathbf{x}_1^k + \mathcal{B}\mathbf{x}_2^k)) = [\mathtt{J}^*_{\mathcal{A}}\vert_{\mathbf{x}_1^k}\nabla f(\mathbf{r}^k), \mathtt{J}^*_{\mathcal{B}}\vert_{\mathbf{x}_2^k}\nabla f(\mathbf{r}^k)]$
Output Multiplication 	$\mathtt{A} \in \mathscr{L}(\mathtt{D},\mathtt{E})$, $\mathtt{B} \in \mathscr{L}(\mathtt{F},\mathtt{G})$	$\mathtt{A}(\cdot)\mathtt{B}(\cdot) : \mathtt{D}\times\mathtt{F} \to \mathtt{C}$	$\nabla(f(\mathtt{A}\mathbf{X}_1^k\mathtt{B}\mathbf{X}_2^k)) = [\mathtt{A}^*\nabla f(\mathbf{R}^k)(\mathtt{B}\mathbf{X}_2^k)^\mathsf{H}, \mathtt{B}^*(\mathtt{A}\mathbf{X}_1^k)^\mathsf{H}\nabla f(\mathbf{R}^k)]$
	$\mathcal{A} : \mathtt{D} \to \mathtt{E}$, $\mathcal{B} : \mathtt{F} \to \mathtt{G}$	$\mathcal{A}(\cdot)\mathcal{B}(\cdot) : \mathtt{D}\times\mathtt{F} \to \mathtt{C}$	$\nabla(f(\mathcal{A}\mathbf{X}_1^k\mathcal{B}\mathbf{X}_2^k)) = [\mathtt{J}^*_{\mathcal{A}}\vert_{\mathbf{X}_1^k}\nabla f(\mathbf{R}^k)(\mathcal{B}\mathbf{X}_2^k)^\mathsf{H}, \mathtt{J}^*_{\mathcal{B}}\vert_{\mathbf{X}_2^k}(\mathcal{A}\mathbf{X}_1^k)^\mathsf{H}\nabla f(\mathbf{R}^k)]$

Table 2.5: Table showing different calculus rules to combine linear and nonlinear mappings. Here $\mathbf{r}^k$ and $\mathbf{R}^k$ indicate the residual inside the parentheses of f. For the output multiplication rule if $\mathtt{E} = \mathbb{R}^{n\times l}$ and $\mathtt{G} = \mathbb{R}^{l\times m}$ than $\mathtt{C} = \mathbb{R}^{n\times m}$.

sending back the residual to compute (2.20) as the small arrows in Figure 2.4b show.

These intuitive examples represent only one of the various calculus rules that can be use to combine mappings. Many other calculus rules can be applied to construct models with much more complex DAGs. Table 2.5 shows the most important calculus rules used to create models. As it can be seen, the *horizontal concatenation* rule is very similar to the horizontal concatenation of two matrices. However this creates a forward DAG that has two inputs $\mathbf{x}_1^k$ and $\mathbf{x}_2^k$ that are processed in parallel using two forward oracles whose result are then summed. This rule can also be used to simply add up different optimization variables by setting the mappings to be identity, like in the least squares function of Example 4. By inspecting the chain rule it is possible to see how its backward DAG would look: this would be a reversed version of the forward DAG, having a single input $\nabla\tilde{f}(\mathbf{r}^k)$ and two outputs where the respective adjoint (Jacobian) mappings would be applied.

The final calculus rule of Table 2.5, called *output multiplication* has a similar forward DAG to the horizontal concatenation, with the summation being substituted with a multiplication. The resulting mapping, even when linear mappings are used, is always nonlinear. Due to this nonlinear behavior its backward DAG is more difficult to picture but still the very same concepts are applied.

Example 3 shows an example of a deep neural network (DNN), a type of nonlinear model which is extensively used in machine learning together with back-propagation. Recently, DNNs have been succesfully used in many areas of signal processing as well [114, 128]. In Example 3, many of the calculus rules of Table 2.5 are used to model a DNN that is trained to perform a nonlinear classification task. DNNs model the behavior of brain neurons. The neurons are divided in sequential groups called layers: these can be distinguished between output, input and inner layers. Specifically, in Example 3 only one inner layer is present. Each neuron belonging to a layer is connected with all of the other neurons of the neighbor layers. Neurons of the output layer are connected to $\mathbf{y}$, while those of the input layer are connected to the input, which in this problem is given and represented by $\mathbf{D}$. The connections between neurons are modeled by the matrices $\mathbf{W}_i$ which contain *weights* representing the importance of each connection and are estimated by solving (2.18). Additionally every layer has a bias term b_i that must also be estimated. Neurons can either be active or inactive and this behavior is modeled by the nonlinear mappings $\mathcal{S}_i$ which consists of sigmoid functions. The DAG of this DNN is not reported here for the sake of brevity, but the constraints of (2.18) well describe it. The addition of the bias term performs a horizontal concatenation while the operation $\mathbf{W}_i\mathbf{L}_i$ represents an output multiplication, which *connects* the different layers.

2.5 General problem formulation

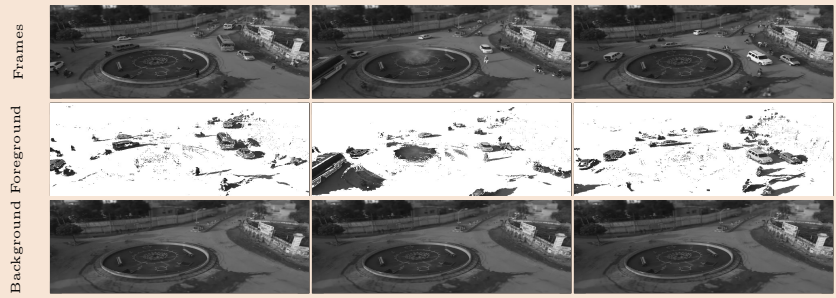
It was already mentioned that the problem formulation (2.3) with its cost function $f(\mathbf{x})+g(\mathbf{x})$ includes a wide variety of optimization problems. However, most of the times problems are formulated without having in mind this particular structure: typically there are M optimization variables representing different signals to estimate which can appear in multiple functions and constraints. This leads to a more general problem formulation which, after converting the constraints into indicator functions, can be summarized as follows [129, 130]:

$$\underset{\mathbf{x}_1,\dots,\mathbf{x}_M}{\text{minimize}} \ \sum_{i=1}^{N} h_i\left(\sum_{j=1}^{M} \mathsf{A}_{i,j}\mathbf{x}_j\right), \tag{2.21}$$

where the N functions $h_i : \mathtt{C}_1 \times \cdots \times \mathtt{C}_M \to \overline{\mathbb{R}}$ are composed with linear mappings $\mathsf{A}_{i,j} : \mathtt{D}_j \to \mathtt{C}_i$. Notice that here the eventual presence of nonlinear mappings is included in h_i.

In order to apply the framework described in the previous sections, (2.21) must be re-structured into (2.3) by splitting the different h_i into two groups: this means, one must appropriately partition the set of indexes $\{1,\dots,N\}$ into two

Example 4 Video background removal



The frames of a video can be viewed as a superposition of a moving foreground to a steady background. Splitting the background from the foreground can be a difficult task due to the continuous changes happening in different areas of the frames. The following optimization problem can be posed to deal with such task:

$$\begin{aligned} \underset{\mathbf{L},\mathbf{S}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{L}+\mathbf{S}-\mathbf{Y}\|^2 + \lambda\|\text{vec}(\mathbf{S})\|_1 \\ \text{subject to} \quad & \text{rank}(\mathbf{L}) \leq 1. \end{aligned}$$

`StructuredOptimization` code snippet:

```
@minimize ls(L+S-Y)+
          lambda*norm(S,1)
st rank(L) <= 1
```

Here $\mathbf{Y} \in \mathbb{R}^{nm\times l}$ consists of a matrix in which the l-th column contains the pixel values of the vectorized l-th frame with dimensions $n \times m$. The optimization problem, also known as robust PCA, decomposes $\mathbf{Y}$ into a sparse matrix $\mathbf{S}$, representing the foreground changes and a rank-1 matrix $\mathbf{L}$ consisting of the constant background, whose columns are linearly dependent.

subsets I_f, I_g, such that $\{1,\dots,N\} = I_f \cup I_g$ and $I_f \cap I_g = \emptyset$, and set

$$f(\mathbf{x}_1,\dots,\mathbf{x}_M) = \sum_{i\in I_f} h_i\left(\sum_{j=1}^{M} \mathsf{A}_{i,j}\mathbf{x}_j\right), \tag{2.22}$$

$$g(\mathbf{x}_1,\dots,\mathbf{x}_M) = \sum_{i\in I_g} h_i\left(\sum_{j=1}^{M} \mathsf{A}_{i,j}\mathbf{x}_j\right). \tag{2.23}$$

In order for f in (2.22) to be smooth, clearly one must have that h_i is smooth for all $i \in I_f$. When this is the case, denoting $\mathbf{r}_i = \sum_{j=1}^{M} \mathsf{A}_{i,j}\mathbf{x}_j$, one has that

$$\nabla f(\mathbf{x}_1, \ldots, \mathbf{x}_M) = \left[\left(\sum_{i \in I_f} \mathsf{A}_{i,1}^* \nabla h_i(\mathbf{r}_i) \right)^{\mathsf{T}}, \ldots, \left(\sum_{i \in I_f} \mathsf{A}_{i,M}^* \nabla h_i(\mathbf{r}_i) \right)^{\mathsf{T}} \right]^{\mathsf{T}}. \tag{2.24}$$

On the other hand, g in (2.23) should have an efficiently computable proximal mapping. This happens, for example, if all of the following conditions are met [130]:

- for all $i \in I_g$, function h_i has an efficient proximal mapping;
- for all $i \in I_g$ and $j \in \{1, \ldots, M\}$, mapping $\mathsf{A}_{i,j}$ satisfies $\mathsf{A}_{i,j}\mathsf{A}_{i,j}^* = \mu_{i,j}\mathsf{Id}$, where $\mu_{i,j} \geq 0$.
- for all $j \in \{1, \ldots, M\}$, the cardinality $\operatorname{card}\{i \mid \mathsf{A}_{i,j} \neq 0\} = 1$.

These rules ensure that the separable sum and precomposition properties of proximal mappings, cf. Table 2.2, are applicable yielding an efficient proximal mapping for g.

Let the cost function of Example 4 be a test case to check these rules. This example treats the problem of robust PCA [113, 131] which has practical applications in surveillance, video restoration and image shadow removal. After converting the rank constraint on $\mathbf{L}$ into an indicator function $\delta_{\mathscr{S}}$, it is easy to see that $g(\mathbf{S}, \mathbf{L}) = \lambda\|\operatorname{vec}(\mathbf{S})\|_1 + \delta_{\mathscr{S}}(\mathbf{L})$ can be written in terms of (2.23). Clearly g consists of a separable sum of functions that have efficient proximal mappings that can be viewed in Table 2.1. Additionally, g satisfies the conditions above and has therefore an efficiently computable proximal mapping: all linear mappings not equal to 0 are identity and satisfy $\mathsf{A}\mathsf{A}^* = \mu\mathsf{Id}$. Moreover, for every variable only one linear mapping is not equal to 0. On the contrary, if an additional constraint on $\mathbf{S}$ or $\mathbf{L}$ appeared or if another nonsmooth function was present in the cost function, *e.g.*, $\|\operatorname{vec}(\mathbf{L})\|_1$, the proximal mapping of g would have been difficult to compute.

2.5.1 Duality and smoothing

Sometimes the rules that ensure that g has an efficiently computable proximal mapping are too stringent. However, even when these rules are not satisfied there are cases where it is still possible to apply proximal gradient algorithms.

Example 5 Total variation de-noising



ground truth $\mathbf{X}_{gt}$ noisy $\mathbf{Y}$ denoised $\mathbf{X}^\star$

StructuredOptimization code snippet:

```
V = Variation(size(Y))
U = Variable(size(V,1)...)

@minimize ls(-V'*U+Y)+
conj(lambda*norm(U,2,1,2))
X = Y-V'*(∼U)
```

Total variation de-noising seeks to remove noise from a noisy image whose pixels are stored in the matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$. This technique relies on the assumption that neighbor pixels of the sought uncorrupted image $\mathbf{X}^\star$ should be similar, namely that $\sqrt{|x^\star_{i+1,j} - x^\star_{i,j}|^2 + |x^\star_{i,j+1} - x^\star_{i,j}|^2}$ should be small, where $x^\star_{i,j}$ is the (i,j)-th component of $\mathbf{X}^\star$. This enforces the image to have sharp edges, namely a sparse gradient. The following optimization problem can be formulated:

(a) $\underset{\mathbf{X}}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|^2 + \lambda\|\mathsf{V}\mathbf{X}\|_{2,1}$ (b) $\underset{\mathbf{U}}{\text{minimize}} \quad \frac{1}{2}\| - \mathsf{V}^*\mathbf{U} + \mathbf{Y}\|^2 + g^*(\mathbf{U})$

Here the operator $\mathsf{V} \in \mathscr{L}(\mathbb{R}^{n \times m}, \mathbb{R}^{nm \times 2})$ maps $\mathbf{X}$ into a matrix having in its j-th column the vectorized forward finite difference gradient over the j-th direction. The operator V appears in the nonsmooth part of the cost function $g(\cdot) = \lambda\|\cdot\|_{2,1}$ and leads to a non-trivial proximal operator. Here the mixed norm $\|\cdot\|_{2,1}$ consists of the sum of the l_2-norm of the rows of $\mathsf{V}\mathbf{X}$. Using Fenchel's duality theorem it is possible to convert the problem into (b) which can instead be solved efficiency using proximal gradient algorithms.

Consider the following problem:

$$\underset{\mathbf{x}}{\text{minimize}} \;\; f(\mathbf{x}) + g(\mathsf{A}\mathbf{x}) \tag{2.25}$$

where f and g are convex and A is a general linear mapping (for example, it is not a tight frame hence $g \circ \mathsf{A}$ does not have an efficient proximal mapping, cf. Section 2.3.1). If f is *strongly convex*, then the *dual problem* of (2.25) has a structure that well suits proximal gradient algorithms, as it will be now shown.

The dual problem can be derived through the usage of the *convex conjugate* functions which are defined as:

$$f^*(\mathbf{u}) = \sup_{\mathbf{x}} \{\langle \mathbf{x}, \mathbf{u} \rangle - f(\mathbf{x})\}. \tag{2.26}$$

Convex conjugation describes f in terms of dual variables $\mathbf{u}$: this conjugation has many properties that often can simplify optimization problems, see [98,132]

for an exhaustive review. Problem (2.25) can be expressed in terms of convex conjugate functions through its associated Fenchel dual problem [80]:

$$\underset{\mathbf{u}}{\text{minimize}} \;\; f^*(-\mathsf{A}^*\mathbf{u}) + g^*(\mathbf{u}). \tag{2.27}$$

Solving the Fenchel dual problem may be particularly desirable when $\mathsf{A} \in \mathscr{L}(\mathbb{R}^m, \mathbb{R}^n)$ and $n \gg m$, in which case the dual variables $\mathbf{u} \in \mathbb{R}^m$ are significantly less than the original ones $\mathbf{x} \in \mathbb{R}^n$. Furthermore, two properties of convex conjugate functions allow for (2.27) to be solved using proximal gradient algorithms. Firstly, proximal mappings and convex conjugate functions are linked by the *Moreau decomposition*:

$$\mathbf{x} = \operatorname{prox}_{\gamma g^*}(\mathbf{x}) + \gamma \operatorname{prox}_{(1/\gamma)g}(\mathbf{x}/\gamma). \tag{2.28}$$

This shows that whenever the proximal mapping of g is efficiently computable, so is that of g^*. Secondly, if f is strongly convex then its convex conjugate f^* has a Lipschitz gradient [133, Lemma 3.2]. This also implies that any solution of (2.27) can be converted back to the one of the original problem through [98]:

$$\mathbf{x}^\star = \nabla f^*(-\mathsf{A}^*\mathbf{u}^\star). \tag{2.29}$$

Therefore, under these assumptions it is possible to solve (2.27) using proximal gradient algorithms of Section 2.3.2. When this is done, the (fast) PG algorithm results in what is also known as (fast) *alternating minimization algorithm* (AMA) [133, 134].

Example 5 treats the classical image processing application of de-noising a digital image. Here in the original problem a linear operator appears in a nonsmooth function preventing its proximal mapping to be efficient. Function f here is the squared norm which is self-dual *i.e.*, $f(\cdot) = \frac{1}{2}\|\cdot\|^2 = f^*(\cdot)$. Hence it is possible to solve the dual problem instead, using proximal gradient algorithms: in the Fenchel dual problem the linear mapping is transferred into the smooth function f^* in terms of its adjoint allowing the usage of an efficient proximal mapping for the nonsmooth function g through (2.28). Once the dual solution $\mathbf{U}^\star$ is obtained, this can be easily converted back to the one of the original problem through the usage of (2.29): $\nabla f(\mathbf{X}^\star) = \mathbf{X}^\star - \mathbf{Y} = -\mathsf{V}^*\mathbf{U}^\star$.

Finally, it was assumed that the functions constructing f are differentiable. When this is not the case, proximal gradient algorithms can be still applied by "smoothing" the nonsmooth functions h_i that appear in f by means of the Moreau envelope [135, 136]:

$$h_i^\beta(\mathbf{x}) = \min_{\mathbf{z}} \left\{ h_i(\mathbf{z}) + \tfrac{1}{2\beta}\|\mathbf{z} - \mathbf{x}\|^2 \right\}. \tag{2.30}$$

Moreau envelopes possess some very important properties related to optimization: similarly to the FBE, when h_i is convex, the function h_i^β is a real-valued,

smooth lower bound to h_i, that shares with h_i its minimum points and values, see [80]. Furthermore, computing the value and gradient of h_i^β essentially requires one evaluation of $\text{prox}_{\beta h_i}$:

$$\nabla h_i^\beta(\mathbf{x}) = \tfrac{1}{\beta}\left(\mathbf{x} - \text{prox}_{\beta h_i}(\mathbf{x})\right). \tag{2.31}$$

However, using the Moreau envelope has the drawback that one has to finetune the parameter β which controls the level of smoothing. This is typically achieved through the usage of a continuation scheme that involves solving the optimization problem multiple times with a decreasing level of β to approach the solution of the original optimization problem with nonsmooth f.

The Moreau envelope can also be used in the dual problem of (2.25) when f is only convex but not strongly convex, meaning that its convex conjugate f^* is not guaranteed to be smooth. Strong convexity can be obtained in f by adding to it a regularization term:

$$f_\beta(\mathbf{x}) = f(\mathbf{x}) + \tfrac{\beta}{2}\|\mathbf{x}\|^2. \tag{2.32}$$

Then, the convex conjugate of f_β becomes the Moreau envelope of the convex conjugate of f, *i.e.*, $(f_\beta^*) = (f^*)^\beta$ [80, Prop. 14.1], which is now differentiable and allows to solve the dual problem (2.27) using proximal gradient algorithms.

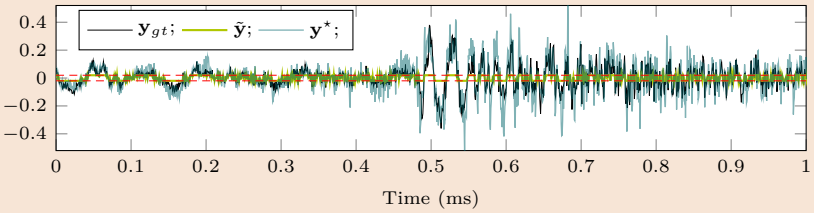
2.6 A high-level modeling language: StructuredOptimization

In all of the examples shown in this paper the formulations of the various optimization problems are placed side by side to some code snippets. This code corresponds to an open-source high-level modeling language implemented in Julia.

Julia is a relatively new open-source programming language which was specifically designed for scientific computing and offers high performance often comparable to low-level languages like C. Despite being a young programming language it has a rapidly growing community and already offers many packages in various fields [108].

The proposed high-level modeling language is provided in the software package `StructuredOptimization` [137]: this utilizes a syntax that is very close to the mathematical formulation of an optimization problem. This user-friendly interface acts as a parser to utilize three different packages that implement many of the concepts described in this paper:

Example 6 Audio de-clipping



When recording an audio signal generated from a loud source the microphone can saturate. This results in a *clipped audio signal* which can be severely corrupted with distortion artifacts. The figure above shows a frame of a clipped signal: the red dashed lines represent the saturation level C of the microphone. The samples of the true signal that are above or below these lines are *lost* during the audio recording. What *audio de-clipping* seeks is to recover these samples and remove the audio artifacts. This can be achieved by solving an optimization problem that combines the uncorrupted samples of the audio signal with the knowledge that the signal is sparse when transformed using the DCT.

$$
\begin{aligned}
\underset{\mathbf{x},\mathbf{y}}{\text{minimize}} \quad & \frac{1}{2}\|\mathsf{F}_{i,c}\mathbf{x} - \mathbf{y}\|^2, \\
\text{subject to} \quad & \|\mathsf{M}\mathbf{y} - \mathsf{M}\tilde{\mathbf{y}}\| \leq \sqrt{\epsilon} \\
& \mathsf{M}_+\mathbf{y} \geq C \\
& \mathsf{M}_-\mathbf{y} \leq -C \\
& \|\mathbf{x}\|_0 \leq N
\end{aligned}
$$

`StructuredOptimization` code snippet:

```
f = ls( idct(x) - y )
for N = 30:30:30*div(N1,30)
 cstr = (norm(M*y-M*yt) <=
              sqrt(fit_tol),
          Mp*y >= C, Mn*y <= -C,
          norm(x,0) <= N)
 @minimize f st cstr
 if norm(idct(∼x) - ∼y) <=
          fit_tol break end
end
```

Here $\mathbf{y}$ and $\mathbf{x}$ are both optimization variables representing the sought de-clipped signal and its DCT transform respectively. M, $\mathsf{M}_\pm$ select the uncorrupted and clipped samples and are used in the first three constraints to keep $\mathbf{y}$ either close to $\tilde{\mathbf{y}}$ at the uncorrupted samples or outside the saturation level respectively. The value N represents the number of active components in the DCT: as the code snippet shows, this value is tuned by solving the optimization problem multiple times by increasing N. As more active components are introduced, the cost function decreases: once its value reaches $\sqrt{\epsilon}$ the solution refinement is stopped.

- `ProximalOperators` is a library of proximal mappings of functions that are frequently used in signal processing and optimization. These can be transformed and manipulated using the properties described in

Section 2.3.1.

- `AbstractOperators` provides a library of FAOs that can be used to create and evaluate DAGs of linear and nonlinear mappings as described in Section 2.4. This package also offers a syntax analogue to the one that is typically used with matrices.
- `ProximalAlgorithms` is a library of optimization algorithms that includes the PG algorithm and its enhanced variants described in Sections 2.3.2 and 2.3.4.

When a problem is provided to `StructuredOptimization` this is automatically analyzed to check whether it falls within the sets of problems described in Section 2.5. Firstly, the various functions and constraints, which are conveniently converted into indicator functions, need to be split into the functions f and g. As it was described in Section 2.5 sometimes multiple splitting configurations are possible: `StructuredOptimization` adopts the simplest strategy possible, splitting the smooth functions from the nonsmooth ones. The nonsmooth functions are then analyzed to verify if the rules described in Section 2.5 are fulfilled to ensure an efficient proximal mapping of g exists. If this is the case, `StructuredOptimization` then provides the necessary inputs to the algorithms of `ProximalAlgorithms` to efficiently solve the problem.

Example 6 can be used as a showcase of the proposed high-level modeling language. This example treats the recovery of an audio signal corrupted by clipping [138, 139]. This recovery is performed using a weighted overlap-add method, *i.e.*, by splitting the audio signal into overlapping frames of length $n = 2^{10}$ and processing them serially, using an initialization strategy analogue to the one proposed in [140].

The high-level modeling language that `StructuredOptimization` provides is designed to be as much natural as possible. Firstly the optimization variables can be defined, *e.g.*, $\mathbf{x} \in \mathbb{R}^n$ is constructed by typing `x = Variable(n)`. By default the variables are initialized by vectors of zeros but it is possible to set different initializations *e.g.*, `Variable([0;1])` will be a variable of two elements initialized by the vector $[0, 1]^\intercal$. The user can also utilize different equivalent notations: for example in the first line of the code snippet of Example 6 the function f could be defined equivalently with `f = 0.5*norm(F*x-y)^2`, by firstly constructing the mapping $\mathsf{F}_{i,c}$ using the notation `F = IDCT(n)`. Similarly, the selection mappings applied to $\mathbf{y}$, *i.e.*, `Mp*y`, could be replaced equivalently by `y[idp]` where `idp` is an array of indexes corresponding to the ones of the selection mapping M_+.

Once the cost function `f` and the constraints `cstr` are defined, the problem can be solved by typing `@minimize f st cstr`. If an efficient proximal mapping is

found, the problem is solved using a proximal gradient algorithm. As can be seen, here this condition is fulfilled despite the fact that multiple constraints over the variable $\mathbf{y}$ are present: these still lead to an efficient proximal mapping since they are applied to non-overlapping slices of the variable $\mathbf{y}$ and are therefore separable.

The standard algorithm, PANOC, is then used to solve the problem, but if a specific one is to be used *e.g.*, the FPG algorithm, one can specify that: `@minimize cf st cstr with FPG()`. As the code snippet of Example 6 shows, the series of problems is set inside a loop: here every problem is automatically warm-started by the previous one, as the variables `x` and `y` are always linked to their respective data vectors which can be accessed by typing `~x`. More details about the software can be found in the documentation online. Finally, in line with the philosophy of reproducible research all the code that was used to create the examples and the various comparison of the algorithms is publicly available online [137].

Many other software packages based on proximal gradient algorithms have been recently developed. There are different MATLAB toolboxes: `FOM` provides several proximal gradient algorithms [141] and `ForBES` implements Newton-type accelerated proximal gradient algorithms [142]. `TFOCS` offers different splitting algorithms that can be used in combination with FAOs through the usage of the toolbox `Spot` [143]. `ProxImaL` [144], also implements different matrix-free splitting algorithms in the Python language with a particular focus to image processing applications.

2.7 Conclusions

The proximal gradient algorithms described in this paper can be applied to a wide variety of signal processing applications. Many examples were presented here to show this versatility with a particular focus on inverse problems of large-scale that naturally arises in many audio, communication, image and video processing applications. Recent enhancements of the PG algorithm have improved significantly its convergence speed. These offer the possibility of using quasi-Newton methods reaching solutions of high accuracy with a speed that was previously beyond the reach of most first-order methods. Additionally these algorithms can be easily combined with fast forward-adjoint oracles to compute the mappings involved leading to matrix-free optimization.

The applications illustrated in this paper are only a small portion of what these algorithms can tackle and it is envisaged that many others will benefit their properties. In fact, proximal gradient algorithms are relatively simple and

they result in very compact implementations, which most of the time do not require additional subroutines, unlike other splitting algorithms *e.g.*, ADMM. This makes them particularly well suited for embedded systems and real-time applications. Additionally, many of the operations involved in this framework are parallel by nature: not only proximal mappings, which in many contexts are separable, but also matrix-free optimization, that utilizes graphs of forward-adjoint oracles, naturally lead to parallelism. This makes these algorithms also particularly fit for wireless sensor networks and many Internet-of-Things applications.

Finally, these algorithms can tackle nonconvex problems: machine learning showed how nonlinear models can reach outstanding results. It is envisaged that these algorithms with their flexibility can be used to create novel nonlinear filters by easily testing the effectiveness of new nonlinear models.

Part II

Modeling

Chapter 3

The randomized image method

On the modeling of rectangular geometries in room acoustic simulations

Enzo De Sena, Niccolò Antonello, Marc Moonen, and Toon van Waterschoot

Published in *IEEE/ACM Transactions of Audio, Speech Language Processing*, vol. 23, no. 4, pp. 774-786, Apr. 2015.

© 2015 IEEE. Reprinted, with permission, from:
E. De Sena, N. Antonello, M. Moonen, and T. van Waterschoot, "On the modeling of rectangular geometries in room acoustic simulations," *IEEE/ACM Transactions of Audio, Speech Language Processing*, vol. 23, no.6, pp. 774-768, Apr. 2015.
Changes include layout, representation, and minor editing aspects.

The candidate's contributions include: co-literature study, software development and simulations using the FDTD method, text co-redaction and co-editing.

Abstract

This paper is concerned with an acoustical phenomenon called *sweeping echo*, which manifests itself in a room impulse response as a distinctive, continuous pitch increase. In this paper, it is shown that sweeping echoes are present (although to greatly varying degrees) in all perfectly rectangular rooms. The theoretical analysis is based on the rigid-wall image solution of the wave equation. Sweeping echoes are found to be caused by the orderly time-alignment of high-order reflections arriving from directions close to the three axial directions. While sweeping echoes have been previously observed in real rooms with a geometry very similar to the rectangular model (*e.g.*, a squash court), they are not perceived in commonly encountered rooms. Room acoustic simulators such as the image method (IM) and finite-difference time-domain (FDTD) correctly predict the presence of this phenomenon, which means that rectangular geometries should be used with caution when the objective is to model commonly encountered rooms. Small out-of-square asymmetries in the room geometry are shown to reduce the phenomenon significantly. Randomization of the image sources' position is shown to remove sweeping echoes without the need to model an asymmetrical geometry explicitly. Finally, the performance of three speech and audio processing algorithms is shown to be sensitive to strong sweeping echoes, thus highlighting the need to avoid their occurrence.

3.1 Introduction

Speech and audio processing research studies often require a room acoustic model that is representative of common acoustical conditions. A simplified model of room geometry that is often used in the literature is the perfectly rectangular box, due to its similarity with many everyday spaces. While this model is only a coarse approximation of real-world rooms, it has a number of appealing properties–it leads to elegant theoretical derivations and simple algorithmic implementations, and it avoids the need to design a detailed model of room imperfections and objects present in the room. These properties are particularly appealing for research studies or applications where the modeling of room acoustics is not the central objective.

In [145], Kiyohara et al. observed that some real-world rooms that are similar to the rectangular model (*e.g.*, a squash court) exhibit a phenomenon that they called *sweeping echo*. This is a different phenomenon from *flutter echo*. A hand-clap in the presence of strong sweeping echoes is perceived with a distinctive, continuous pitch increase. This motivated the term *sweeping* echo. Sweeping echoes cannot be observed directly in the room impulse response (RIR) or the

transfer function and appear in the RIR's spectrogram as straight lines passing through the spectrogram's origin. Examples of spectrograms with sweeping echoes are presented in Section 3.3.

Kiyohara et al. studied the sweeping echoes based on an interesting number theoretic approach, which is capable of predicting various characteristics of the phenomenon [145, 146]. The limitation of this analysis, however, is that it only applies to very specific scenarios, *i.e.*, a cubic room [145], a square tunnel [146], or a rectangular tunnel where the squared ratio of the edge lengths is an integer number [146]. The analysis is further restricted by the assumption that both source and microphone are positioned in the middle of the enclosure.

The main contribution of this paper is a theoretical analysis that is independent of the room dimensions and of the position of source and microphone, showing that sweeping echoes are present (although to varying extents) in all rectangular geometries. The analysis is based on the rigid-wall image solution of the wave equation, *i.e.*, the solution that formally replaces the physical boundaries surrounding an acoustic source with an equivalent lattice of image sources in free-field [27]. The analysis, which is presented in Section 3.4, shows that sweeping echoes are caused by the orderly time-alignment of high-order image sources positioned in proximity of the three axial directions. Section 3.4 also gives an insight as to why more regular setups of room, source, and microphone give rise to stronger sweeping echoes, which is consistent with empirical observations reported in [145].

The simulations in Section 3.3, are obtained using the image method (IM) [27] and finite-difference time-domain (FDTD) method [16]. These room acoustics models, which are briefly described in Section 3.2, correctly predict the presence of sweeping echoes in perfectly rectangular rooms. Audio samples generated with different setups are made available at [147] and in the supplementary downloadable material associated to this paper. The IM samples can also be generated using the Matlab code provided in Appendix. Listening to these audio samples clearly reveals that the cases with regular setups do not correspond to commonly-experienced rooms. This means that particular caution has to be exercised when using the rectangular shape as geometric model since it can yield results that are very far from typical acoustical conditions.

The regularity of the lattice of image sources, which is at the basis of the phenomenon, reduces when real-world imperfections are included in the geometric model. Simulation results in Section 3.5 show, for instance, that small out-of-square asymmetries in the room geometry are sufficient to reduce the sweeping echo phenomenon significantly.

Modeling real-world imperfections requires significant computational resources

and may be too cumbersome for studies that are not centered on room acoustics, or where a high level of accuracy is not required. Section 3.6 shows that by randomizing the image sources' position, the sweeping echoes can be removed with little additional computational burden. Notice that randomization has been used previously in the IM to model diffuse reflections [148] or to reduce coloration [149]. In [149], it is unclear whether "coloration" refers to sweeping echoes or to a more typical stationary phenomenon. The author states that coloration is caused by the regular structure of image sources, which seems to support the former interpretation, but no further analysis or explanation is given.

Section 3.7 presents a performance analysis of three speech and audio processing algorithms. This analysis shows that strong sweeping echoes alter the measured performance significantly, thus highlighting the need to avoid their occurrence in performance studies.

Finally, Section 3.8 concludes the paper and suggests directions for future work.

3.2 Background

3.2.1 The wave equation

The propagation of sound waves in a lossless fluid is governed by the wave equation [4, 5], which is the partial differential equation (PDE) with the form

$$\triangle p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = s \text{ on } \Omega \times \tau \ , \tag{3.1}$$

where Ω and τ denote the spatial and temporal domain, respectively, p is the sound pressure, s is the source term, $c = 343$ m/s is the speed of sound, and $\triangle$ denotes the Laplace operator with respect to the spatial coordinates. The acoustical properties of the walls are described mathematically by the boundary conditions (BCs) [4, 5]:

$$\frac{\partial p}{\partial t} = -cZ_w \nabla p \cdot \mathbf{n} \text{ on } \partial\Omega \times \tau, \tag{3.2}$$

where $\partial\Omega$ denotes the boundary of Ω, $\mathbf{n}$ is the normal vector at the boundary, ∇ denotes the gradient operator with respect to the spatial coordinates, and Z_w is the characteristic wall impedance, which models the energy losses that occur at the walls. Together, the PDE and BCs form a boundary value problem. This problem is very cumbersome to solve in general. However, in the case of rectangular rooms with rigid walls, it admits a closed-form solution. This is discussed in the next subsection.

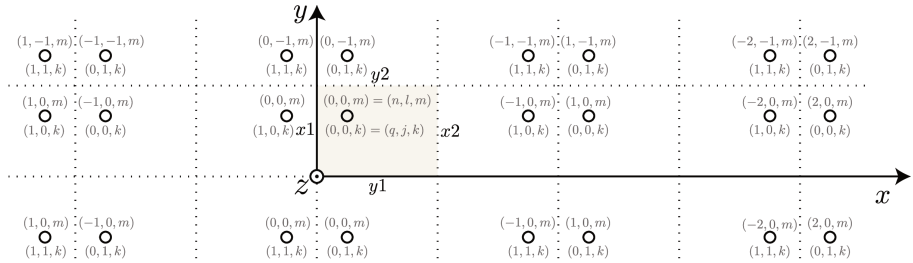


Figure 3.1: Image sources with associated indices. The triplet on top of each image denote (n, l, m), while the triplet on the bottom denote (q, j, k). The labels $x1$, $x2$, $y1$ and $y2$ denote the four vertical walls.

3.2.2 Solution of the wave equation for rectangular rooms with rigid walls

Under the assumption of rigid walls and monochromatic sound pressure with angular frequency $\omega = 2\pi f$, the boundary value problem (3.1)-(3.2) simplifies to [4, 5]:

$$\begin{aligned} &\text{PDE} \quad \triangle\hat{p} + \left(\frac{\omega}{c}\right)^2 \hat{p} = \hat{s} \text{ on } \Omega \\ &\text{BCs} \quad \nabla\hat{p} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega \ , \end{aligned} \tag{3.3}$$

where the hat notation indicates a complex sound pressure and source. If the source is a point source at position $\mathbf{x}$, the sound pressure at observation point $\mathbf{x}'$ can be expressed using separation of variables as [4]:

$$\hat{p}(\mathbf{x}, \mathbf{x}') = \frac{c^2}{V} \sum_{m=0}^{\infty} \frac{\psi_m(\mathbf{x}')\psi_m(\mathbf{x})}{\omega_m^2 - \omega^2}, \tag{3.4}$$

where V is the volume of the domain, and ψ_m are the so-called eigenfunctions (or modes) of the problem. In the case of rectangular rooms, the eigenfunctions can be found analytically and are given by [4]

$$\psi_m(\mathbf{x}) = \cos\left(\frac{m_x\pi}{L_x}x\right)\cos\left(\frac{m_y\pi}{L_y}y\right)\cos\left(\frac{m_z\pi}{L_z}z\right), \tag{3.5}$$

where (L_x, L_y, L_z) are the room dimensions, and, with abuse of notation, m is associated with one of the combinations of the non-negative integers m_x, m_y and m_z. Furthermore, the angular frequencies ω_m take the form

$$\omega_m = c\sqrt{\left(\frac{m_x\pi}{L_x}\right)^2 + \left(\frac{m_y\pi}{L_y}\right)^2 + \left(\frac{m_z\pi}{L_z}\right)^2}. \tag{3.6}$$

Equation (3.4), together with (3.5) and (3.6), is usually referred to as the Green's function for a rectangular room with rigid walls.

3.2.3 Image method for a rectangular room

In [27], Allen and Berkley proved that the Green's function for a rectangular room with rigid walls is equivalent to a time-domain solution based on the image method. The idea behind the image method is that the sound field due to a point source in a half-space bounded by a rigid wall is equal to the sound field without the wall but with an additional point source (called *image*) placed symmetrically on the opposite side of the wall. The equivalence stems from the fact that the position of the two point sources is such that the normal component of the velocity vector field vanishes at the boundary, thus satisfying the BC in (3.3). In rectangular rooms, each image is itself imaged, generating an infinite lattice of equivalent point sources as shown in Figure 3.1 [27]. Allen and Berkley showed that the RIR measured at $\mathbf{x}' = (x', y', z')$ due to a point source positioned at $\mathbf{x} = (x, y, z)$ in a rectangular room with rigid walls can be expressed as[1] [27]:

$$p(\mathbf{x}, \mathbf{x}', t) = \sum_{\mathbf{p}=\mathbf{0}}^{\mathbf{1}} \sum_{\mathbf{r}=-\infty}^{+\infty} \frac{\delta\left(t - \|\mathbf{R_p} + \mathbf{R_r}\|/c\right)}{4\pi\|\mathbf{R_p} + \mathbf{R_r}\|}, \tag{3.7}$$

where the summations are carried out with respect to the integer vectors $\mathbf{p} = (q, j, k)$ and $\mathbf{r} = (n, l, m)$ (six summations in total, three of which only assume binary values), and where

$$\mathbf{R_p} = (x + (2q-1)x', y + (2j-1)y', z + (2k-1)z'),$$

$$\mathbf{R_r} = (2nL_x, 2lL_y, 2mL_z)\ .$$

Figure 3.1 shows the indices combinations associated to some of the image sources. Allen and Berkley referred to (3.7) as the rigid-wall image solution of the wave equation.

While the formulation with rigid walls provides a powerful tool for theoretical derivations, it is seldom used in practical applications since it leads to a non-decaying impulse response. In the non-rigid case, however, the results are no longer mathematically exact. By assuming that image sources remain point sources, and that the wall absorption is angle-independent, the impulse response

[1] This paper uses the same notation of [27] in order to improve readability for those familiar with that paper.

is modified to[2] [27]:

$$h(\mathbf{x},\mathbf{x}',t)=\sum_{\mathbf{p}=\mathbf{0}}^{\mathbf{1}}\sum_{\mathbf{r}=-\infty}^{+\infty}\beta_{x1}^{|n+q|}\beta_{x2}^{|n|}\beta_{y1}^{|l+j|}\beta_{y2}^{|l|}\beta_{z1}^{|m+k|}\beta_{z2}^{|m|}\times\frac{\delta\left(t-\|\mathbf{R_p}+\mathbf{R_r}\|/c\right)}{4\pi\|\mathbf{R_p}+\mathbf{R_r}\|},\tag{3.8}$$

where $\beta_{x1}\cdots\beta_{z2}$ are the wall reflection coefficients, *e.g.*, the wall denoted by $x1$ is the wall adjacent to the origin with the normal vector parallel to the x-axis. The subscript "2", on the other hand, is associated to walls that do not pass through the origin.

3.2.4 Finite-difference time-domain method

Finite-difference time-domain (FDTD) method models room acoustics by means of time and space discretization of the wave equation. The pressure and source signals are sampled in space on a uniform grid with spacing X, and in time with temporal period T. The partial derivatives of the wave equation (3.1) are then approximated using central, second-order, finite differences, thus converting the PDE into a set of linear equations that can be solved iteratively.

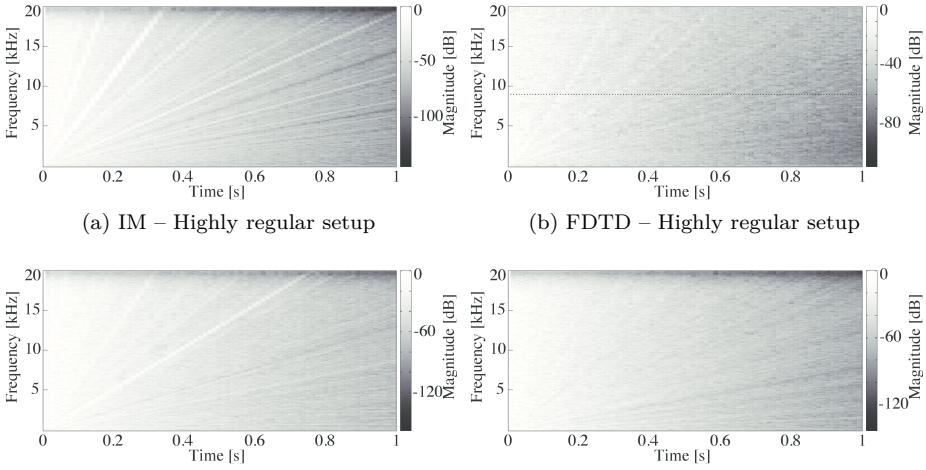
The ratio between spatial and temporal resolution cannot be chosen arbitrarily, due to stability reasons [16]. Indeed, this ratio has to satisfy $\frac{cT}{X}\leq\lambda_{max}$, where λ_{max} depends on the specific type of discretization scheme used. This ratio, which is referred to as Courant number, is usually set with the equality in order to minimize numerical errors while conserving stability of the solution [16]:

$$c\frac{T}{X}=\lambda_{max}.\tag{3.9}$$

The approximations made in replacing the derivatives with finite differences introduce numerical errors that increase with frequency and cause dispersion, *i.e.*, the phenomenon by which sound waves with different frequencies and/or directions propagate with different speeds. The dependence on the direction, in particular, makes it difficult to remove dispersion by means of post-processing operations.

The FDTD simulations in this paper will use the standard leapfrog scheme [16]. With this scheme, only the frequency range $f\in[0,0.075f_s]$ has a relative

[2]The indices at the exponents in this formula are summed, as opposed to [27] where they appear to be erroneously subtracted.



(a) IM – Highly regular setup

(b) FDTD – Highly regular setup

(c) IM – Example in Allen and Berkley's paper [27]

(d) IM – Irregular setup

Figure 3.2: Comparison of the spectrogram of three IM-generated RIRs, 3.2a, 3.2c and 3.2d, and one FDTD-generated RIR, 3.2b, with simulation setups as in Table 3.1. In 3.2b, the area below the dotted line has a relative numerical error smaller than 2%.

numerical error smaller than 2% (*e.g.*, for $f_s = 40$ kHz, $f \in [0, 3]$ kHz) [16]. While various methodologies are available in the literature to reduce dispersion with a given sampling frequency [150], the approach used in this paper is simply to use a very high sampling frequency, at the cost of a significant computational and memory load.

3.3 Sweeping Echoes in Acoustic Simulations with Rectangular Geometry

This section presents simulation results showing the sweeping echo phenomenon using the IM and FDTD method. The section concludes with the definition of a measure that quantifies the presence of sweeping echoes in the spectrogram.

Figure	(L_x, L_y, L_z)	(x, y, z)	(x', y', z')	SSF
3.2a, 3.2b	$(4, 4, 4)$	$(1, 2, 2)$	$(2, 1.5, 1)$	0.5651
3.2c	$(8, 12, 10) \times 343/800$	$(3, 10, 4) \times 343/800$	$(5, 1, 6) \times 343/800$	0.6326
3.2d	$(4.1, 4.2, 4.3)$	$(1.4, 2.5, 2.6)$	$(2.7, 1.8, 1.9)$	0.9627

Table 3.1: Simulation setup and SSF measure of RIRs in Figure 3.2.

3.3.1 Sweeping echoes in the IM

Figure 3.2a, 3.2c, and 3.2d show the spectrogram of the RIRs produced by the IM for the three simulation setups reported in Table 3.1 with $\beta_{x1} = \cdots = \beta_{z2} = 0.93$. The sampling frequency was set to $f_s = \frac{1}{T} = 40$ kHz. The RIRs were filtered with a 2nd-order high-pass Butterworth filter with a cutoff frequency of 50 Hz to remove non-physical behaviour at zero frequency [27]. In order to account for non-integer delays, each pulse was replaced with the impulse response of an ideal low-pass filter with cutoff frequency $f_c = 0.9\frac{f_s}{2}$, windowed with a Hann window forty samples long, as proposed by Peterson in [151]. The spectrograms were calculated over 2^{12} frequency points and used a Hamming window 25 ms long with a 50% overlap ratio. Unless stated otherwise, all remaining simulations in this paper use the above configuration with the same setup of Figure 3.2a.

The spectrograms in Figure 3.2a, 3.2c and 3.2d clearly show the sweeping echo phenomenon. In all three cases, it may be observed that the spectrum scales linearly with time, or, in other words, the spectrum at a given time is similar to a stretched version of the spectrum at a previous time. It may also be observed that simulation setups with a higher degree of regularity result in the phenomenon being more visible. As mentioned in the introduction, impulse responses with sweeping echoes are perceived as a distinctive audible pitch increase[3]. In the case of regular setups, the phenomenon remains clearly visible (and audible) when convolved with anechoic audio material with sharp onsets, as shown in Figure 3.3 for the case of an african bongo.

3.3.2 Sweeping echoes in FDTD

Figure 3.2b shows the spectrogram of a RIR generated using the FDTD method with the same regular setup of Figure 3.2a. In this simulation, the spatial resolution is set to $X = 5$ mm. For the standard leapfrog scheme $\lambda_{max} = \sqrt{1/3}$. The sampling frequency is obtained from equation (3.9) as $f_s = 118$ kHz. With

[3]Audio samples of all spectrograms in Figure 3.2 are available at [147] and in the supplementary downloadable material associated to this paper.

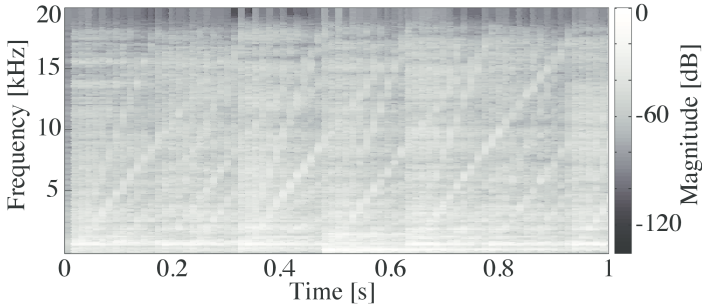


Figure 3.3: Spectrogram of the convolution between the RIR in Figure 3.2a and an anechoic audio sample of an african bongo. The repetitive, bright, vertical bands correspond to the sharp onsets of the audio sample.

this sampling frequency, the standard leapfrog scheme yields a relative numerical error smaller than 2% for frequencies below 8.9 kHz. The room is excited with a physically constrained source as proposed by Sheaffer et al. in [152]. The characteristic impedance is set to $Z_w = 65$ for all walls. All remaining FDTD simulations in this paper use the above configuration.

It is clear from Figure 3.2b that sweeping echoes are present in the FDTD method too, thus confirming that this is not an artifact of the IM. The increasing pitch effect is clearly audible in the FDTD method as well. Notice also that the slope of the main sweeping echo is the same for both methods. Compared to the IM, the pattern appears slightly blurred. This is likely to be attributed to the fact that sweeping echoes, as will become clear in Section 3.4, are critically dependent on phase errors.

3.3.3 Sweeping echo measure

Before proceeding further, it is convenient to define a measure that quantifies how visible sweeping echoes are in the spectrogram. In order to capture the sweeping echoes' pattern, the spectrogram is averaged around a line that intersects the origin with varying slope, as shown conceptually in Figure 3.4. The exact steps carried out to derive the measure are described in the remainder of this subsection.

Let the spectrogram be denoted by $|Z(\xi, 2\pi f)|^2$, where ξ is the center of the analysis window. The first step consists of normalizing the spectrogram such

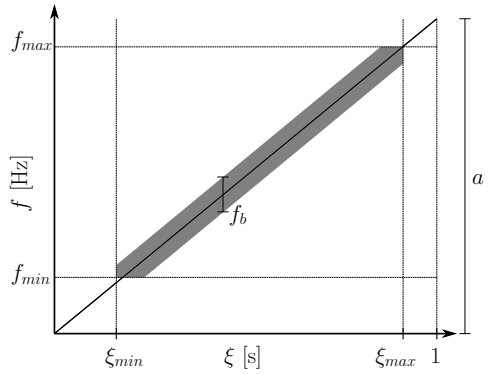


Figure 3.4: Explanation of how the sweeping spectrum (SS) is calculated. The average of the normalized spectrogram is performed in the mask denoted by the gray area. The SS is the result of this averaging as a function of the slope of the line, a.

that the energy is identical in each time bin:

$$|\Phi(\xi, 2\pi f)|^2 = \frac{|Z(\xi, 2\pi f)|^2}{\int_{f_{min}}^{f_{max}} |Z(\xi, 2\pi f)|^2 \, df} \ , \tag{3.10}$$

where f_{min} and f_{max} are the minimum and maximum frequencies considered in the calculation of the proposed measure, respectively. The purpose of this normalization is to remove attenuation, which would cause lower levels of energy at lower line slopes even in absence of sweeping echoes. Let the line be described by the equation $f = a\xi$, where a is the slope of the line. The average of the normalized spectrogram is calculated as

$$\Psi(a) = \frac{1}{K} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} |\Phi(\xi_u, 2\pi f_v)| \ , \tag{3.11}$$

where $\mathcal{U}$ and $\mathcal{V}$ are the set of indices of the time-frequency bins that fall within the masking area shown in Figure 3.4. More specifically, $\mathcal{U} = \{u | \xi_{min} \leq \xi_u \leq \xi_{max}\}$, where ξ_{min} and ξ_{max} are the minimum and maximum time bins of the mask, respectively, and $\mathcal{V} = \{v | \max(a\xi - \frac{f_b}{2}, f_{min}) \leq f_v \leq \min(a\xi_p + \frac{f_b}{2}, f_{max})\}$, where f_b is the bandwidth of the mask (see Figure 3.4), and K is the number of time-frequency bins that fall within the masking area.

The function $\Psi(a)$ will be referred to as sweeping spectrum (SS) in the following. The independent variable, a, is measured in hertz per second; *e.g.*, a strong sweeping echo centered at 10 kHz at time 1 s would be associated to $a = 10$ kHz/s.

Spectrograms with highly visible sweeping echoes have a SS with significant variation. The measure used here to quantify the variability of the SS is the spectral flatness (also known as Wiener entropy), *i.e.*, the ratio of the geometric mean and arithmetic mean of the power spectrum [153]. This measure has values between 0 and 1, with 1 associated to the flat (white-noise-like) case, and 0 associated to the impulsive (tone-like) case. The spectral flatness of the SS, which is termed here SSF, is calculated as:

$$\mathrm{SSF} = \frac{\sqrt[N]{\prod_{i=0}^{N-1} |\Psi(a_i)|^2}}{\frac{1}{N}\sum_{i=0}^{N-1} |\Psi(a_i)|^2} \,, \tag{3.12}$$

where $\Psi(a_i)$ is the SS calculated at discrete values of the slope a_i.

Table 3.1 shows the SSF measure calculated for the simulations in Figure 3.2. These values were obtained using $f_{min} = 50$ Hz, $f_{max} = 0.9 f_s/2$ kHz, $\xi_{min} = 0$ s, $\xi_{max} = 0.5$ s, and $f_b = 400$ Hz. The slope values a_i were five hundred equally spaced values between 5 kHz/s and 150 kHz/s. The lower bound, 5 kHz/s, was chosen such that the line $f = a\xi$ intersects $f_{min} = 50$ Hz at $\xi = 10$ ms, which is approximately the delay of the line-of-sight component in the simulations of this paper. The higher bound, 150 kHz/s, was chosen such that all the significant sweeping echoes observed in Figure 3.2 would be included. Unless stated otherwise, all SSF measures in this paper use the above configuration.

The SSF values associated to Figure 3.2a, 3.2c and 3.2d are 0.56, 0.63, and 0.96, respectively. It is interesting to observe that the example in Allen and Berkley's paper [27] has a relatively low SSF value (*i.e.*, strong sweeping echoes) even though the simulation setup was not chosen on purpose to elicit the sweeping echo phenomenon. The irregular setup has an SSF value, 0.96, which is relatively high. Notice that this value is close to typical SSF values obtained when the setup parameters are drawn from a continuous random distribution. Indeed, the SSF values of one hundred simulations with uniformly distributed room dimensions between 2 m and 8 m, and source and microphone positions uniformly distributed within the room, have an average of 0.96, with 50% of the samples falling between 0.95 and 0.97.

3.4 Physical Basis of Sweeping Echoes in Perfectly Rectangular Rooms

This section gives an insight into the physical cause of sweeping echoes. The analysis is based on the rigid-wall image solution of the wave equation. The first

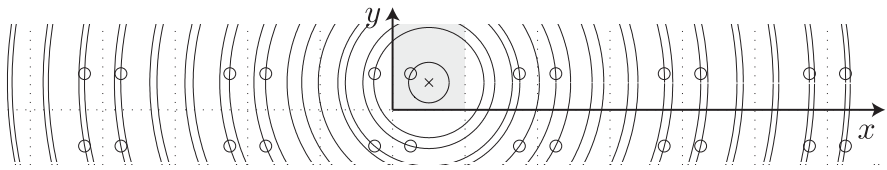


Figure 3.5: Image sources for the simplified case with three walls ($x1$, $x2$, and $y1$). Circles centered at the microphone are drawn to visualize the distance of the image sources. Notice the orderly convergence between far-field pairs' distance with respect to the observation point.

two subsections present simplified cases with three and four walls, respectively. The analysis is then generalized to the case with six walls.

3.4.1 Simplified case with three walls

Consider the simplified case of a rectangular room where the walls denoted by $y2$, $z1$ and $z2$ are placed at a distance large enough for the associated image sources to arrive with a significant delay. Assume also that all the walls are rigid. Figure 3.5 shows the lattice of image sources associated to this problem. Notice that the regularity of image sources' positions leads to a monotonic convergence in the time arrival of far-field image pairs. In this section, it will be argued that this type of converging behaviour–which also happens in the general case–is the root cause of the sweeping echoes.

The signal formed by the converging image pairs with indices $\mathbf{p}_1 = (0, 1, 0), \mathbf{r}_1 = (n, 0, 0)$ and $\mathbf{p}_2 = (0, 0, 0), \mathbf{r}_2 = (n, 0, 0)$ for $n \geq 0$ can be written as

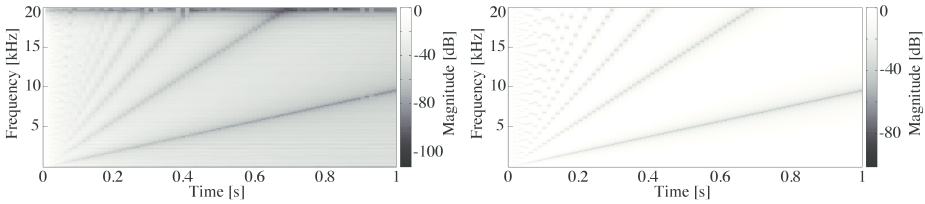
$$x(t) = \sum_{n=0}^{\infty} \left[\delta\left(t - \tau_n\right) + \delta\left(t - \tau_n - \Delta_n\right)\right], \tag{3.13}$$

where τ_n is the delay of $\mathbf{p}_2$, and Δ_n is the difference between the time delay of $\mathbf{p}_1$ and $\mathbf{p}_2$:

$$\begin{aligned}\Delta_n =& \frac{1}{c}\left(\sqrt{(2nL_x + x - x')^2 + (y + y')^2 + (z + z')^2}\right. \\ & \left. -\sqrt{(2nL_x + x - x')^2 + (y - y')^2 + (z + z')^2}\right) .\end{aligned} \tag{3.14}$$

A first-order Taylor expansion of Δ_n for large n gives

$$\Delta_n \simeq \frac{yy'}{nL_x c} . \tag{3.15}$$



(a) Image solution of wave equation (b) Closed-form approximation

Figure 3.6: Simplified case with three walls. The spectrogram of the image solution of the wave equation is shown in 3.6a, while 3.6b shows the approximation derived in equation (3.17).

Let τ_n be approximated by $\tau_n \simeq \frac{2nL_x}{c}$ (see Figure 3.1), which implies $n \simeq \frac{\tau_n c}{2L_x}$. Substituting this expression in (3.15) leads to

$$\Delta_n \simeq \frac{2yy'}{c^2}\frac{1}{\tau_n} \,. \tag{3.16}$$

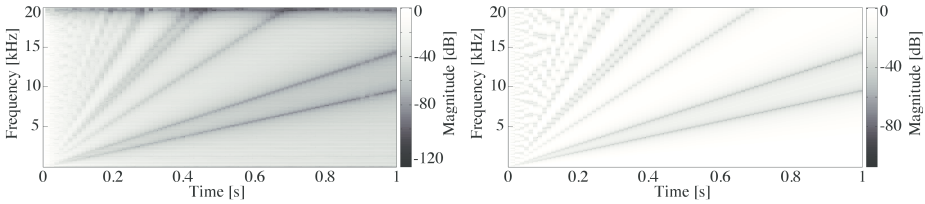
Consider now the windowed signal $z(\xi, t) = w(\xi, t)x(t)$, where $w(\xi, t)$ is a square window centered at ξ. Assuming that the window includes only the two impulses under consideration, the Fourier transform of $z(\xi, t)$ calculated at $\xi = \tau_n$ (*i.e.*, when the window is centered around the delay of $\mathbf{p}_2$) is

$$|Z(\xi,\omega)|^2_{\xi=\tau_n} = \left|1 + e^{-i\omega\Delta_n}\right|^2 \simeq \left|1 + e^{-i\frac{2yy'}{c^2}\frac{\omega}{\xi}}\right|^2 , \tag{3.17}$$

where in the last passage Δ_n was approximated as in (3.16).

Figure 3.6 compares the actual spectrogram obtained for $L_x = 4$ with the approximation given in (3.17), showing a nearly exact match of the sweeping echoes behavior. This result shows that the approximations made in the derivation of (3.17) carry a negligible error, and that, therefore, the sweeping phenomenon is indeed caused by the progressive alignment in time of the far-field image pairs.

In order to gain insight into how more intricate patterns can emerge, the next subsection discusses the case where an additional wall is positioned at a finite distance from the origin.



(a) Image solution of wave equation (b) Closed-form approximation

Figure 3.7: Simplified case with four walls. The spectrogram of the image solution of the wave equation is shown in 3.7a, while 3.7b shows the approximation derived in equation (3.18).

3.4.2 Simplified case with four walls

Consider the case where only the walls denoted by $y2$ and $z2$ are at a large distance from the origin. The difference from the previous example is that the xy-plane (*i.e.*, the plane associated to the $z1$ wall) is at a close distance, and therefore the image sources are positioned on two parallel planes–one above and one below the xy-plane. Sources on each plane are still arranged as in Figure 3.5. It is clear that the number of image sources with converging delay is now four. These are $\mathbf{p}_1 = (0,0,0)$, $\mathbf{p}_2 = (0,1,0)$, $\mathbf{p}_3 = (0,0,1)$, and $\mathbf{p}_4 = (0,1,1)$. Using an approach similar to the case with three walls, one obtains

$$|Z(\xi,\omega)|^2_{\xi=\tau_n} \simeq \left| 1 + e^{-i\frac{2yy'}{c^2}\frac{\omega}{\xi}} + e^{-i\frac{2xx'}{c^2}\frac{\omega}{\xi}} + e^{-i\frac{2(xx'+yy')}{c^2}\frac{\omega}{\xi}} \right|^2 , \qquad (3.18)$$

which is plotted in Figure 3.7, along with the actual spectrogram.

3.4.3 General case

The two examples above have the convenient property that only images with $l = m = 0$ are present. In the general case where all the walls are at a close distance, all the images with order $-\infty < l < \infty$ and $-\infty < m < \infty$ contribute to the patterns observed in Figure 3.2, which makes it more difficult to derive approximate closed-form expressions. A first-order Taylor expansion of the difference between the delay of the images associated to $(\mathbf{p}_1, \mathbf{r}_1)$ and $(\mathbf{p}_2, \mathbf{r}_2)$ with the same order n gives the expression (3.19), where the over-lined indices are $\overline{q} = 2q - 1$, $\overline{j} = 2j - 1$, and $\overline{k} = 2k - 1$.

Observe in (3.19) that for image pairs with $\overline{q}_1 \neq \overline{q}_2$ (*i.e.*, image pairs at opposite sides of the x-axis), Δ does not vanish for large n. All remaining

$$\begin{aligned}\Delta\left[\mathbf{p}_1,\mathbf{r}_1;\mathbf{p}_2,\mathbf{r}_2\right] =& \simeq \frac{1}{n}\frac{1}{L_x c}\big[nL_x(\overline{q}_1-\overline{q}_2)x' + (\overline{q}_1-\overline{q}_2)xx'/2 + \\ &+ (l_1^2-l_2^2)L_y^2 + (l_1-l_2)L_y y + (l_1\overline{j}_1 - l_2\overline{j}_2)L_y y' + (\overline{j}_1-\overline{j}_2)yy'/2 \\ &+ (m_1^2-m_2^2)L_z^2 + (m_1-m_2)L_z z + (m_1\overline{k}_1 - m_2\overline{k}_2)L_z z' + (\overline{k}_1-\overline{k}_2)zz'/2\big]\ ,\end{aligned} \tag{3.19}$$

pairs (*i.e.*, image sources that lie on a plane parallel to the yz-plane) align in an orderly fashion over time for large n. Furthermore, they do so as $1/n$, or, equivalently, as $1/\xi$. This implies that, due to the scaling property of the Fourier transform ($f(\xi t) \xrightarrow{\mathfrak{F}} \frac{1}{|\xi|}F\left(i\omega/\xi\right)$ [154]), the spectrum is stretching over time. More specifically it does so linearly, *i.e.*, proportionally to ξ, which is consistent with the sweeping phenomenon observed in Figure 3.2.

Expressions similar to (3.19) can be obtained along the direction of the y-axis and z-axis for large l and m, respectively. Thus, the time-aligning phenomenon is present along all three axial directions. Notice that the present analysis is independent of the room dimensions, and of the position of the source and microphone. This implies that sweeping echoes are present (albeit to greatly varying extents) in all rectangular rooms.

Equation (3.19) also provides some insight as to why some simulation setups yield stronger sweeping echoes than others: The simulation setup of Figure 3.2a is very regular, in the sense that some factors in (3.19) and in the corresponding expressions along the y-axis and z-axis, are identical, *e.g.*, $L_x = L_y = L_z$, $xx' = zz'$, and others are integer multiples of each other, *e.g.*, $L_x x = 2L_x x'$, $L_z z' = L_y y = 2L_z z$, $yy' = L_y y'$. This causes a high number of complex exponentials to sum up in phase along with their harmonics (associated to integer factors such as $l_1 - l_2$), which, in turn, yield sharper peaks.

It is worth commenting here about the similarity between sweeping echoes and the so-called *chirped echoes* [155, 156]. As opposed to sweeping echoes, a hand clap in the presence of chirped echoes is perceived with a decreasing pitch. Chirped echoes are known to occur, for instance, in front of the staircase of the El Castillo pyramid at the Maya ruins of Chichen-Itza [155]. The periodic geometry of this staircase yields a progressive temporal stretching between the reflected impulses, which, in turn, causes the decreasing pitch [156].

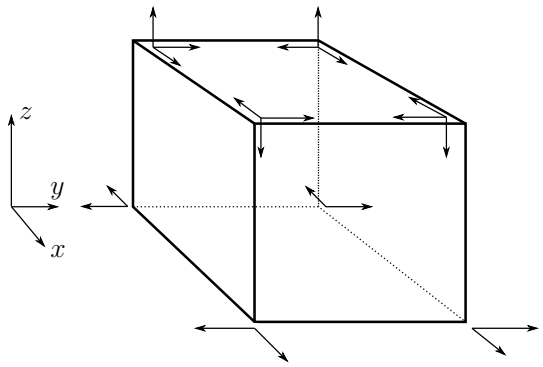


Figure 3.8: Room geometry with small out-of-square distortions added to a cubic room with $L_x = L_y = L_z = 4$ m. The arrows are used to highlight the displacement of the corners. Each arrow is associated with one of the three cardinal axes. At each corner, the presence of an arrow indicates that the corner has been moved along the associated axis by δx cm in the direction of the arrow.

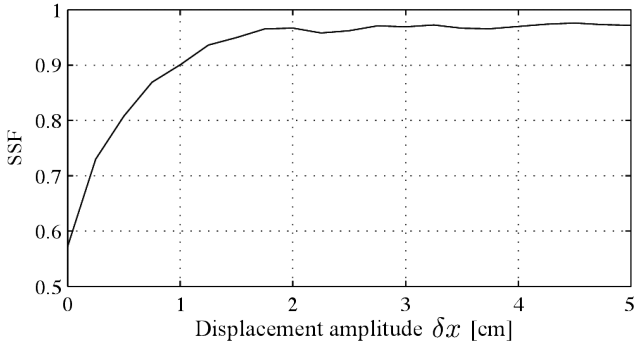
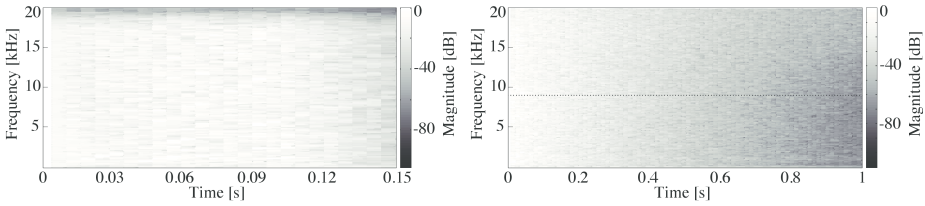


Figure 3.9: Sweeping spectrum flatness (SSF) of the room geometry with small out-of-square imperfections shown in Figure 3.8 as a function of the displacement amplitude δx. The plot was generated using the extended-IM. The position of source and microphone was the same of the regular setup in Figure 3.2a. The case $\delta x = 0$ corresponds to a perfectly cubic room, as in Figure 3.2a.



(a) Extended IM (b) FDTD

Figure 3.10: Spectrogram of the RIRs of the room geometry with small out-of-square imperfections shown in Figure 3.8 with $\delta x = 2$ cm using the extended IM in 3.10a, and FDTD in 3.10b. In both cases, the position of source and microphone is the same of the regular setup in Figure 3.2a. Due to the short length of the RIR in 3.10a, the duration of the analysis window was halved to 12.5 ms. In 3.10b, the area below the dotted line has a relative numerical error smaller than 2%.

3.5 Room geometries with small out-of-square imperfections

In the previous section, it was argued that sweeping echoes are caused by the orderly time-alignment of high-order image sources. This, in turn, is due to the fact that walls opposite to each other are perfectly parallel and that adjacent walls intersect exactly at 90 degrees, which results in an extremely regular lattice of image sources. When tilting the walls by even a small quantity, this regularity starts to break down.

To test the effect of small out-of-square asymmetries, the geometry of the room setup in Figure 3.2a is distorted as described by Figure 3.8. The walls are still perfectly flat, but each corner is moved by a small quantity δx in the direction indicated by the arrows. Figure 3.9 shows the sweeping spectrum flatness (SSF) of this geometry as a function of the displacement amplitude, δx. The simulations were run using the IM extension to arbitrary polyhedra proposed by Borish in [28]. The Matlab toolbox EDB2 was used for this purpose [157, 158]. Specular reflections up to fourteenth order were considered, which resulted in a response around 0.15 seconds long. Orders higher than fourteen could not be considered due to significant memory and computational requirements.

Figure 3.9 shows that a displacement $\delta x = 2$ cm, *i.e.*, 0.5% of the edge length (which is considered an acceptable departure from building specifications [159]), is sufficient to increase the SSF from 0.57 to 0.97. Notice that these values should

not be compared directly with the SSF values presented in Table 3.1. Indeed, the two sets of RIRs have different durations, and the sweeping phenomenon (which is non-stationary) is measured in different parts of the spectrogram.

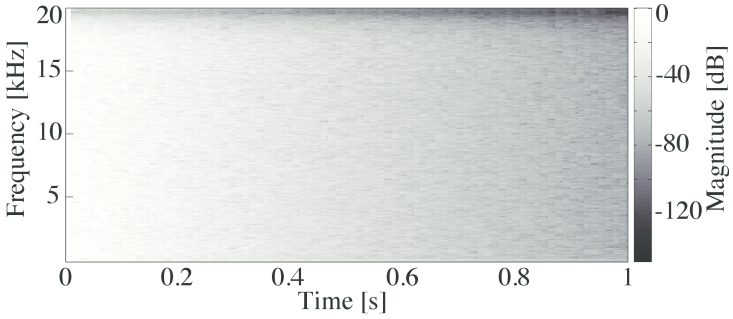
Figure 3.10 shows the spectrograms obtained via the extended-IM and FDTD with $\delta x = 2$ cm. In FDTD, the oblique walls were approximated using a staircase implementation. It may be observed in Figure 3.10 that the sweeping echo phenomenon is significantly reduced in both simulation methods.

The above results did not take into account a number of other real-world imperfections that would contribute to breaking the regularity of the image sources' lattice. These include, but are not limited to, small-scale (roughness) or large-scale (uneveness) imperfections of the walls, temperature gradients in the room, or the presence of objects. A combination of all these real-world imperfections, along with the fact that regular setups of the kind described in the previous section are unlikely in the real world, is probably the reason why sweeping echoes are not commonly experienced.

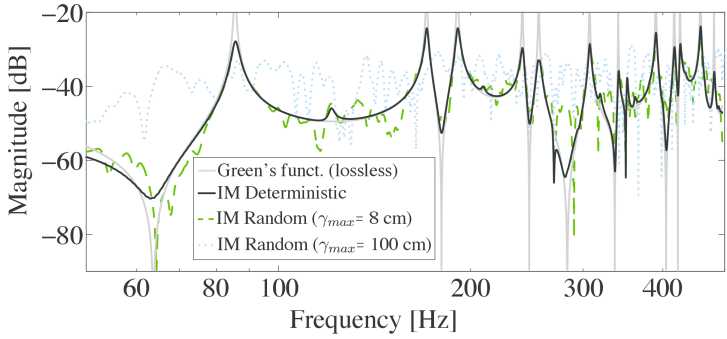
3.6 Sweeping Echoes Removal from the Image Method

Modeling the real-world imperfections that break the sweeping echo phenomenon requires significant memory and computational resources. There are a number of studies and applications where such a high level of accuracy is not required, or where the computational resources are not available. Allen and Berkley's IM algorithm represents an appealing option in these cases. This motivated the analysis in this section, where it is shown how randomizing the image sources' position allows to remove the sweeping echoes with little additional computational burden. Notice that the purpose of this modification is not to model real-world imperfections or diffuse reflections as proposed in [148], but only to remove the phenomenon from the generated RIRs. In [149], Borß proposes to move each image source in a random position within the boundaries of the associated image room. As will be shown here, a much smaller random displacement is already sufficient to remove the sweeping echoes altogether.

The technique used here consists of shifting each image source by a random scalar γ uniformly distributed in $|\gamma| \leq \gamma_{max}$ on the line connecting the image source and the microphone. This is equivalent to adding a random delay, γ/c, to each impulse, which makes the modification of standard IM software implementations extremely simple.



(a) Randomized image method



(b) Low frequency response

Figure 3.11: Spectrogram of the IM (same setup of Figure 3.2a) with the inclusion of a random uniformly-distributed displacement with $\gamma_{max} = 8$ cm in Figure 3.11a, and low-frequency response for $\gamma_{max} = 0, 8, 100$ cm in Figure 3.11b. The SSF for the spectrogram in Figure 3.11a is 0.9945. Notice that the resonance peaks of the Green's function extend to infinity.

Figure 3.11a shows the spectrogram of a simulation with the same setup of Figure 3.2a with $\gamma_{max} = 8$ cm. It may be observed that the sweeping echoes are no longer present. The SSF of this spectrogram is 0.9945. The perceived pitch increase is also removed altogether[4]. This modification of the IM is referred to as randomized image method (RIM) in this paper.

Figure 3.11b shows the low-frequency response for $\gamma_{max} = 0, 8, 100$ cm. The solution of the wave equation (Green's function) for the rigid-walls (lossless)

[4]The audio sample in Figure 3.11a is available at [147] and in the supplementary downloadable material associated to this paper. The same sample can also be generated using the Matlab code provided in appendix.

case is also overlaid, showing that there is a good match with the IM (lossy) deterministic case, *i.e.*, $\gamma_{max} = 0$ cm. The randomized case with $\gamma_{max} = 8$ cm yields a spectrum that is largely similar to the deterministic case $\gamma_{max} = 0$ cm. Notice also that the two spectra are nearly identical around the resonance frequencies. Values of γ_{max} that are too large, on the other hand, may result in a very distorted spectrum, as shown in the figure for the case $\gamma_{max} = 100$ cm.

In order to study what is an appropriate choice for γ_{max} in more general cases, one hundred simulations were run with random room dimensions and random positions of microphone and source, the results of which are shown in Figure 3.12. All random variables were chosen as discrete distributions with a small set of values so as to increase the likelihood of generating regular setups. In regular setups, in fact, sweeping echoes are more challenging to remove. The room dimensions were chosen from a discrete uniform distribution with values between 2 m and 8 m with a step of 1 m. The microphone and source were also positioned at integer distance from the walls with equal probability and were constrained to be at least 1 m away from the walls.

Figure 3.12 shows the SSF and the mean squared error (MSE) of the magnitude response between 50 Hz and 500 Hz as a function of the random displacement γ. As expected, both SSF and MSE increase monotonically with γ. However, while the SSF reaches values very close to unity already at $\gamma_{max} = 8$ cm, the MSE continues to increase for larger values of γ_{max}. The SSF of $\gamma_{max} = 8$ cm, in particular, has a median value of 0.9930 (which is similar to the one in Figure 3.11a), and 50% of the samples are between 0.9905 and 0.9964. The MSE of $\gamma_{max} = 8$ cm has a median value of -52 dB.

In summary, γ_{max} should be chosen based on the intended application and as a compromise between the removal of sweeping echoes and accuracy at low-frequencies. While, in general, it is suggested that the spectrogram is inspected before using a RIR, there are cases where this might be cumbersome, *e.g.*, when the IM is used to generate a large number of RIRs with random scenarios. In these cases, $\gamma_{max} = 8$ cm appears to be a reasonable compromise.

Notice that, in case the accuracy of the magnitude response is not considered important, then other methods could be pursued for generating the late reverberant tail. A common approach with an extremely low computational footprint, for instance, is to use random noise with exponentially decaying envelope [2].

Some caution should be exercised when the room response is sought at multiple microphone positions, and when the spatial information encoded in the difference between microphone signals is important (*e.g.*, in beamforming studies). In these cases, the image sources can be randomized relatively to a

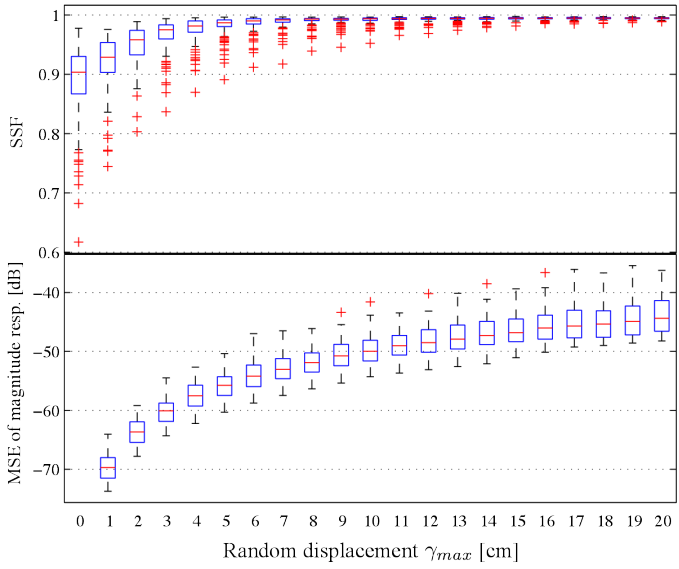


Figure 3.12: Sweeping spectrum flatness (SSF) and mean square error of the low-frequency magnitude response as a function of random displacement γ. The results are calculated from a set of one hundred rooms with randomly generated dimensions, microphone position, and source position. The bottom and top extremes of the box denote the first and third quartile, respectively. The line in the middle of the box denotes the second quartile. The bars represent the extreme values, excluding the outliers. The outliers are denoted by the plus sign.

single microphone position. By using this approach, the wavefront due to each image source originates from a slightly inaccurate point, but the relative time arrivals and directions are correct across microphones.

3.7 Effect of Strong Sweeping Echoes in Speech and Audio Processing Applications

This Section discusses simulation results using the IM and RIM with three different speech and audio processing applications. The purpose of this discussion is to show that the performance of different speech and audio processing algorithms is sensitive to the occurrence of strong sweeping echoes,

without assessing how the phenomenon actually affect the algorithms under consideration (which is beyond the scope of this paper).

The methodology is based on evaluating the performance of three speech and audio processing algorithms for four different sets of simulated RIRs, one of which exhibits considerably stronger sweeping echoes than the other three. It will then be shown that this particular RIR set yields significantly different results compared to the other three RIR sets. The applications considered are (i) multi-channel linear prediction of reverberant speech, (ii) objective evaluation of reverberant speech quality, and (iii) pitch estimation of monophonic reverberant music.

The four RIR sets are obtained by using the IM and RIM for simulating a cubic room of dimensions $(4, 4, 4)$ m with a fixed sound source at position $(1, 2, 2)$ m and two sets of $M = 15$ microphone positions. These two sets are selected in two distinct ways. One set of microphone positions is chosen from a uniform grid with a resolution of 0.25 m in a $(2, 2, 2)$ m box centered in the room. These microphone positions are denoted as *regular* positions, and result in highly regular setups when combined with the given room dimensions and source position. A second set of microphone positions is chosen randomly in a $(3, 3, 3)$ m box centered in the room, and are denoted as *irregular* positions. The wall reflection coefficients are all set to 0.9, and the RIRs are truncated to a length of $0.3f_s$ coefficients. For the RIM, the randomization range is set to $\gamma_{max} = 8$ cm. The SSF values for the resulting four sets of RIRs are given in Tables 3.2 and 3.3 for sampling frequencies $f_s = 8$ kHz and $f_s = 44.1$ kHz. Three out of the four RIR sets have a mean SSF value larger than 0.95, which indicates that the occurrence of sweeping echoes is weak. On the other hand, the RIR set simulated with the image method and regular microphone positions results in considerably lower SSF values, which indicates the presence of strong sweeping echoes. This can also be seen from the RIR spectrograms, which are not given here for the sake of conciseness but can easily be generated given the above simulation parameters.

Please observe that the SSF values of the two sets with different sampling frequencies should not be compared against each another. This is due to the fact that the sweeping phenomenon, which is not equally present across frequency bands, is measured in different parts of the spectrogram.

3.7.1 Multi-Channel Linear Prediction of Reverberant Speech

One of the earliest approaches to speech dereverberation is based on the hypothesis that in a linear prediction (LP) model of reverberant speech, only the LP residual is affected by reverberation while the LP model coefficients

	Image method		Randomized image method	
(x', y', z')	SSF ($f_s = 8$ kHz)	SSF ($f_s = 44.1$ kHz)	SSF ($f_s = 8$ kHz)	SSF ($f_s = 44.1$ kHz)
$(1, 2, 1.50)$	0.831	0.475	0.977	0.984
$(1, 2, 1.25)$	0.920	0.630	0.979	0.991
$(1, 2, 1)$	0.821	0.448	0.980	0.986
$(1, 1.75, 1)$	0.876	0.520	0.959	0.995
$(1, 1.50, 1)$	0.833	0.442	0.955	0.988
$(1, 1.25, 1)$	0.910	0.529	0.966	0.996
$(1, 1, 1)$	0.853	0.445	0.970	0.988
$(1.25, 1, 1)$	0.861	0.512	0.980	0.991
$(1.50, 1, 1)$	0.811	0.536	0.948	0.975
$(1.75, 1, 1)$	0.840	0.502	0.971	0.993
$(2, 1, 1)$	0.879	0.582	0.910	0.948
$(2.25, 1, 1)$	0.836	0.499	0.986	0.995
$(2.50, 1, 1)$	0.805	0.529	0.923	0.972
$(2.75, 1, 1)$	0.833	0.501	0.954	0.993
$(3, 1, 1)$	0.867	0.452	0.981	0.990
mean SSF:	**0.852**	**0.507**	**0.963**	**0.986**

Table 3.2: SSF values for RIRs simulated using regular microphone positions.

	Image method		Randomized image method	
(x', y', z')	SSF ($f_s = 8$ kHz)	SSF ($f_s = 44.1$ kHz)	SSF ($f_s = 8$ kHz)	SSF ($f_s = 44.1$ kHz)
$(0.819, 1.560, 2.272)$	0.961	0.971	0.991	0.994
$(2.250, 1.706, 2.337)$	0.947	0.981	0.985	0.995
$(2.305, 2.057, 0.923)$	0.968	0.983	0.980	0.995
$(0.764, 1.341, 1.624)$	0.921	0.980	0.968	0.994
$(1.584, 2.622, 2.810)$	0.968	0.984	0.978	0.997
$(1.128, 0.834, 2.352)$	0.966	0.981	0.953	0.996
$(0.863, 0.627, 3.169)$	0.947	0.977	0.980	0.994
$(2.624, 3.025, 3.213)$	0.977	0.971	0.981	0.995
$(0.646, 1.085, 1.718)$	0.973	0.973	0.980	0.993
$(3.354, 3.234, 1.694)$	0.959	0.950	0.980	0.993
$(2.921, 1.432, 1.206)$	0.948	0.947	0.947	0.993
$(2.932, 2.105, 1.044)$	0.962	0.966	0.983	0.993
$(1.713, 0.959, 1.676)$	0.966	0.977	0.961	0.993
$(0.747, 2.072, 2.569)$	0.957	0.974	0.966	0.992
$(2.642, 3.077, 3.430)$	0.946	0.973	0.983	0.996
mean SSF:	**0.958**	**0.973**	**0.975**	**0.994**

Table 3.3: SSF values for RIRs simulated using irregular microphone positions.

are the same as those for the clean speech signal [160]. In [161], it was shown that the latter does not hold when considering a single reverberant signal, but only holds when averaging the LP model coefficients estimated for multiple reverberant signals recorded by spatially distributed microphones in the same room.

A 12th order LP model is estimated for the phoneme /ŋ/, which is acquired from a female speaker pronouncing the Dutch word “zang” [162] and downsampled to $f_s = 8$ kHz. The estimation is performed for the clean signal, as well as for the reverberant signals obtained with each of the four RIR sets. The poles corresponding to the estimated LP model coefficients are averaged over the

	Microphone positions	
	Regular	Irregular
Image method	0.0242	0.0158
Randomized image method	0.0126	0.0180

Table 3.4: Mean square pole estimation errors in multi-channel speech LP application.

$M = 15$ microphone positions within each RIR set, and the mean square pole estimation error w.r.t. the poles obtained from the clean signal is calculated for each of the four RIR sets. The results are given in Table 3.4 and indicate that the mean square pole estimation error obtained with the IM and regular microphone positions is considerably larger than for the other RIR sets.

3.7.2 Objective Evaluation of Reverberant Speech Quality

The perceptual evaluation of speech quality (PESQ) measure [163] is one of the objective measures that has been proposed in [164] for quantifying the quality of reverberant speech relative to a clean speech reference. The PESQ implementation of [165] is applied here to evaluate the speech quality of a 6.6 s female speech utterance taken from Track 49 of [166] and downsampled to $f_s = 8$ kHz. The clean speech signal is used as a reference, and the PESQ score of each of the reverberant signals obtained with the four RIR sets is calculated. Box plots of the results for the four RIR sets are shown in Figure 3.13, and indicate that the PESQ score obtained with the image method and regular microphone positions is significantly lower than the scores obtained with the other RIR sets.

3.7.3 Pitch Estimation of Monophonic Reverberant Music

This section applies a pitch estimation algorithm based on approximate nonlinear least squares and maximum a posteriori estimation, included in the Multi-Pitch Estimation Toolbox of [167], to a monophonic piano signal sampled at $f_s = 44.1$ kHz. This signal is created by concatenating the C3 to C5 notes from the Steinway Grand Soft recordings in [168] so as to obtain a 60 s double chromatic scale, in which the length of each note is restricted to 2.4 s. The pitch estimation algorithm operates on non-overlapping frames of 48 ms, and employs a 4096-point FFT grid size and a pitch search range of [0.01,0.1] rad/s. The mean square error, averaged over all frames, between the pitch estimated

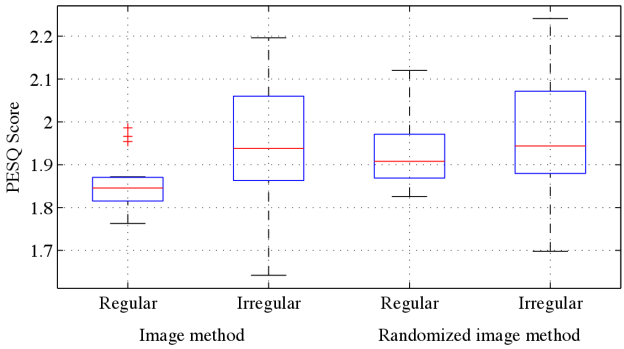


Figure 3.13: PESQ score box plots for objective evaluation of reverberant speech quality.

	Microphone positions	
	Regular	Irregular
Image method	0.335	0.274
Randomized image method	0.296	0.275

Table 3.5: Mean square pitch estimation errors for monophonic reverberant music.

from the clean piano signal and the pitch estimated from the reverberant piano signal obtained from each of the four RIR sets, is given in Table 3.5. Again, the result obtained with the IM and regular microphone positions stands out from the results obtained with the other RIR sets.

3.8 Conclusions and Future Work

This paper was concerned with the modeling of rectangular geometries in room acoustics simulations. It was observed that perfectly rectangular geometries exhibit a phenomenon called sweeping echo. A theoretical analysis based on the rigid-walls image solution of the wave equation showed that the phenomenon is due to the progressive and monotonic convergence of the delay of high-order image sources positioned in proximity of the three axial directions. Simulation results showed that small out-of-square asymmetries are sufficient to reduce the phenomenon significantly.

It was then observed that some room setups cause strong sweeping echoes, which are not experienced in commonly-encountered rooms. One conclusion was then that the rectangular geometry should be used with caution in cases where the objective is to model common acoustical conditions. This was highlighted by the fact that strong sweeping echoes can actually alter the performance of speech and audio processing algorithms. This also shows that previous studies employing a rectangular geometry may have been affected.

While the phenomenon was shown to be present in all rectangular rooms, it was observed that simulation parameters with a higher degree of regularity lead to stronger sweeping echoes. Surprisingly, the example setup chosen by Allen and Berkley in [27] was one of these cases. This points to the fact that manually chosen parameters (which tend to be round numbers) often result in strong sweeping echoes. Irregular setups where the room dimensions, microphone position and source positions are chosen at random from a continuous distribution, on the other hand, lead to weaker sweeping echoes.

A simple modification of the IM was proposed which consisted of delaying each image source by a small random amount. This modification was shown to remove the sweeping phenomenon altogether, while maintaining about the same computational complexity and with only a small degradation of the modal response.

In conclusion, based on the results of this paper, the following recommendation is made. In cases where the simulation setup can be chosen freely, all the parameters (*i.e.*, room dimensions, source position and microphone position) should be drawn from a continuous random distribution. In cases where the simulation setup cannot be chosen freely and the spectrogram shows strong sweeping echoes, then, ideally, one should model the real-world imperfections that would remove the effect. Alternatively, the RIM or other types of randomization methods can be used. These methods can also be used in general cases if one wishes to have a reasonable guarantee that sweeping echoes are not occurring.

A relevant direction for future research involves measuring the perceptual impact of sweeping echoes through formal listening experiments. While it is immediately clear that strong sweeping echoes are perceivable (both in the RIR and when convolved with audio material) and do not correspond to commonly-experienced rooms, the same cannot be said for weak sweeping echoes. Informal listening tests suggest that the pitch increase in the impulse response is still noticeable for SSF values close to 0.96. However, when convolved with audio material, sweeping sounds do not appear to be perceivable. This is consistent with the results of Section 3.7. A relevant research direction would then be to

identify the value of SSF where sweeping sounds start being clearly perceivable.

A further future work would involve investigating alternative randomization criteria. While the criterion used in Section 3.6 has the advantage of being extremely simple, other criteria may be capable of achieving a lower error for a given increase in SSF.

Appendix: Matlab Implementation of the Randomized Image Method

Below is a self-contained Matlab implementation of the RIM method for a single source and single microphone. The only difference with the conventional IM is the instruction `+Nr*(2*rand-1)` at the beginning of the second loop.

```
function h=rim(mi, so, ro, be, Np, Nr, Tw, Fc)
% mi (microphone), so (source) and ro (room) are
% three-dimensional column vectors.
% Np: samples of the RIR.
% Nr: no. of random samples (Nr=0 for original IM).
% Tw: samples of low-pass filter, Fc: cut-off freq.
% All quantities above are in sample periods.
% be: matrix of refl. coeff. [x1,y1,z1;x2,y2,z2]
% Example 1: Fig.2a, h=rim([2;1.5;1]/343*4E4,...
% [1;2;2]/343*4E4, [4;4;4]/343*4E4,...
% 0.93.*ones(2,3), 4E4, 0, 40, 0.9);
% Example 2: Fig.11, h=rim([2;1.5;1]/343*4E4,...
% [1;2;2]/343*4E4, [4;4;4]/343*4E4,...
% 0.93.*ones(2,3), 4E4,0.08/343*4E4, 40, 0.9);
h=zeros(Np,1); ps=perm([0,1],[0,1],[0,1]);
Rps=repmat(so,[1,8])+(2.*ps-1).*repmat(mi,[1,8]);
or=floor(Np./(ro.*2))+1; rs=perm(-or(1):or(1),-or(2):or(2),-or(3):or(3));
for i=1:size(rs'); r=rs(:, i);
 for j=1:8; p=ps(:,j); Rp=Rps(:,j);
  d=norm(2*ro.*r+Rp)+Nr*(2*rand-1);
  if round(d)>Np || round(d)<1; continue; end
  am=be(1,:)'.^abs(r+p).*be(2,:)'.^abs(r);
  if Tw==0; n=round(d); else
    n=(max(ceil(d-Tw/2),1):min(floor(d+Tw/2),Np))';
  end
  s=(1+cos(2*pi*(n-d)/Tw)).*sinc(Fc*(n-d))/2;
  s(isnan(s))=1; h(n)=h(n)+s*prod(am)/(4*pi*d);
end; end;
function res=perm(varargin)
[res{1:nargin}]=ndgrid(varargin{1:nargin});
res=reshape(cat(nargin+1,res{:}),[],nargin)';
```

Dedication

This paper is dedicated to the memory of our friend and colleague, Nejem Huleihel (1989-2014).

Part III

Inverse problems using the FDTD method

Chapter 4

Impedance estimation using the FDTD method

Evaluation of a numerical method for identifying surface acoustic impedances in a reverberant room

Niccolò Antonello, Toon van Waterschoot, Marc Moonen, and Patrick A. Naylor

Published in *Proceedings of 10th European Congress and Exposition on Noise Control Engineering (EURONOISE 2015)*, Maastricht, the Netherlands, June 2015.

© 2015 EAA-NAG-ABAV, ISSN 2226-5147. Reprinted, with permission, from:
N. Antonello, T. van Waterschoot, M. Moonen, and P. Naylor, "Evaluation of a numerical method for identifying surface acoustic impedances in a reverberant room," in *Proceedings of 10th European Congress and Exposition on Noise Control Engineering (EURONOISE 2015)*, Maastricht, the Netherlands, June 2015.
Changes include layout, representation, and minor editing aspects.

The candidate's contributions as first author include: literature study, co-development of the presented method, software implementation and simulations, co-formulation of the conclusions, text redaction and editing.

Abstract

Wave-based room acoustic simulations are becoming more popular as the available compute power continues to increase. The definition of boundary conditions and acoustic impedances is of fundamental importance for these simulations to succeed in representing a realistic acoustical space. Acoustic impedance databases exist in terms of absorption coefficients, which are usually measured in reverberation chambers. In this type of measurements, the sound field is assumed to be diffuse, a condition which is not met in most rooms. In particular at low frequencies, where wave-based simulations are possible, a different approach is sought as an alternative to acoustic impedance measurements. This paper focuses on a recently proposed method for estimating surface acoustic impedances. This method is based on the use of a numerical room model, and does not require the assumption of a diffuse field. Assuming that the geometry of the room is known, a finite-difference time-domain (FDTD) simulation is matched with measured data by solving an optimization problem. The set-up for such a measurement method consists only of a set of microphones and a loudspeaker. This could be applied in every room, removing the need for expensive facilities such as reverberation chambers. The solution of the optimization problem leads to the sought parameters of the acoustic surface impedances. In this paper the adjoint method is used for the computation of the derivative in the optimization problem. This method enables a large number of decision variables in the optimization problem making it possible to account for inhomogeneities of the surface acoustic impedance and hence to avoid the need to specify the different acoustic impedance surfaces beforehand.

4.1 Introduction

Wave-based room acoustic simulations are becoming more popular as the available compute power continues to increase. In particular, the finite-difference time-domain (FDTD) method is receiving a lot of attention especially since its boundary model formulation was improved [16]. However, there is a lack of input parameters for this boundary model, *i.e.*, of measured acoustic impedances. In fact acoustic impedances are usually measured in terms of absorption coefficients either using impedance tubes [169] that rely on one-dimensional standing waves or reverberation chambers where the sound field is assumed to be diffuse [169]. Such ideal sound fields rarely occur in real rooms and in particular diffuse fields are typically not present in low frequency wave-based simulations. Therefore the usage of these absorption coefficients as input parameters for wave-based simulations is questionable.

In [75] a new method for estimating acoustic impedances was proposed. Assuming the room geometry and source distribution to be known, an optimization problem that minimizes the misfit between simulated and measured sound pressure was solved to obtain an estimation of the acoustic impedances. Using the boundary element method (BEM) for the wave-based room acoustic simulation, the optimization problem was posed for singular frequencies, therefore requiring extensive spatial sampling. An alternative approach using the FDTD method was presented in [76] where it was shown that the setup can be greatly simplified and reduced to a loudspeaker and a set of microphones when a narrowband simulation is performed.

In this paper the optimization algorithm of [76] is further developed. Here, the adjoint method is used, which enables the efficient computation of the gradient of the cost function almost independently of the number of the decision variables of the optimization problem. In [75,76] each acoustic impedance surface, *e.g.*, each wall of the room, was modeled using a single acoustic impedance. It is shown here that if inhomogeneities are present in the acoustic impedance surfaces this can lead to a failure of the method. Using the adjoint method it is possible to seek for a different acoustic impedance at each point of the discretized boundary surfaces. Therefore, if inhomogeneities are present, these will be reconstructed. Moreover there is no longer a need to specify the acoustic impedance surfaces manually. Having an increased number of decision variables, however, the main disadvantage of the presented method is that the optimization problem can become ill-posed for certain measured data sets and a regularization is necessary. The adjoint method has been widely used in other fields such as full-waveform inversions in geophysics [170] and imaging techniques [171] but its application for acoustic impedance identification in room acoustic has so far not been studied, to the best of authors' knowledge.

4.2 The finite difference time domain method

The sound field in a room may be predicted by solving the acoustic wave equation, *e.g.*, using the following partial differential equation (PDE), boundary conditions (BCs) and initial conditions (ICs) [4]:

$$\begin{aligned} \text{PDE} \quad & \triangle p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = s \text{ on } \Omega \times \tau \\ \text{BCs} \quad & \frac{\partial p}{\partial t} = -c\xi \nabla p \cdot \mathbf{n} \text{ on } \partial\Omega \times \tau \\ \text{ICs} \quad & \frac{\partial p}{\partial t} = \hat{p}_0, p = p_0 \text{ on } \Omega \end{aligned} \tag{4.1}$$

where $\Omega \subset \mathbb{R}^3$ is the spatial domain defining the room geometry, $\tau \subset \mathbb{R}^+$ is temporal domain, $p(x, y, z, t) : \mathbb{R}^4 \to \mathbb{R}$ is the sound pressure, $s(x, y, z, t)$ is the source distribution, $\triangle$ is the Laplace operator, $\mathbf{n}$ is the normal vector with respect to the boundary surface $\partial\Omega$, $\xi : \partial\Omega \to \mathbb{R}$ is the specific acoustic impedance , c is the speed of sound and $\hat{p}_0$, $p_0 : \mathbb{R}^3 \to \mathbb{R}$ are the ICs.

Analytical solutions of (4.1) exist only for simple geometries and hence it is typically necessary to perform a discretization. The FDTD method discretizes the sound pressure and source distribution in a uniform grid with spatial resolution X, and in time with temporal resolution T, *e.g.*, for the sound pressure

$$p(x, y, z, t) \approx p(lX, mX, iX, nT) = p^n_{l,m,i}. \tag{4.2}$$

Centered finite differences are applied to approximate the second-order derivatives of (4.1). Spatial and temporal resolution are bounded for stability reasons by the ratio $\frac{cT}{X} \leq \lambda_c$, where λ_c is the Courant number, *i.e.*, the maximum ratio between spatial and temporal resolution where stability is ensured and numerical errors are minimized [16]. When explicit FDTD schemes are employed this approximation leads to the update equation

$$p^{n+1}_{l,m,i} = P^n_{l,m,i} - p^{n-1}_{l,m,i} + s^n_{l,m,i}, \tag{4.3}$$

which represent the PDE of (4.1). The term $P^n_{l,m,i}$ represents the approximation of the Laplace operator which consists of the weighted sum of the 26 neighbor samples of $p^n_{l,m,i}$ and $p^n_{l,m,i}$ itself. Different weight choices lead to different FDTD schemes which can be found in [16]. In this paper the 19-samples ISO1 scheme will be employed.

When at the boundary, $P^n_{l,m,i}$ will miss a number of neighbor samples, depending on the nature of the boundary, *e.g.*, walls, inner/outer edges, inner/outer corners and other cases. These missing neighbor samples can be used to enforce the BCs of (4.1) as well as continuity conditions. The modified update equation will become, *e.g.*, for a wall:

$$(1 + \lambda_c/\xi_b)\, p^{n+1}_{l,m,i} = \tilde{P}^n_{l,m,i} + (\lambda_c/\xi_b - 1)\, p^{n-1}_{l,m,i}, \tag{4.4}$$

where $\tilde{P}^n_{l,m,i}$ is the weighted sum having 9 neighbor samples missing and modified weights due to enforced BCs. Notice that the subscript index b indicates the position of ξ_b on the discretization of $\partial\Omega$.

Problem (4.1) is converted into a set of linear equations that is typically solved iteratively using equations (4.3), (4.4) and other equations obtained from other types of BCs. Nevertheless, for illustration purposes, it is convenient to look at the structure of the matrices that arise in such a linear system. By vectorizing the tensors $p^n_{l,m,i}$ and $s^n_{l,m,i}$ for each n, it is possible to group all the equations

and write:

$$\mathbf{Q}_{+}\mathbf{p}_{n+1} - \mathbf{A}\mathbf{p}_{n} - \mathbf{Q}_{-}\mathbf{p}_{n-1} = \mathbf{s}_{n} \tag{4.5}$$

where $\mathbf{Q}_{+}$ and $\mathbf{Q}_{-}$ are $N_x N_y N_z \times N_x N_y N_z$ diagonal matrices having ones at the indexes where the update equation is (4.3) and having coefficients *e.g.*, $(1 + \lambda_c/\xi_b)$ and $(1 - \lambda_c/\xi_b)$ for the wall-modified update equation (4.4), respectively. The vectors $\mathbf{p}_n$ and $\mathbf{s}_n$ are $N_x N_y N_z$ dimensional vectors containing the vectorization of the sampled sound pressure $p^n_{l,m,i}$ and source distribution $s^n_{l,m,i}$ for the time samples n. Here N_x, N_y, N_z are the number of spatial samples used for each cardinal direction. The $\mathbf{A}$ matrix represents the approximation of the Laplace operator and consists of a $N_x N_y N_z \times N_x N_y N_z$ sparse matrix having, for the general explicit scheme, 27 diagonals containing zero elements at the indexes where the BCs are enforced. Notice that the acoustic impedance ξ_b appears only in the matrices $\mathbf{Q}_{\pm}$.

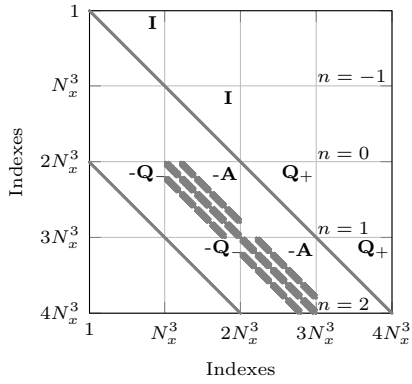


Figure 4.1: Sparsity pattern of the $\mathbf{B}$ matrix.

The final linear system of equations can be obtained by stacking all $\mathbf{p}_n$ and $\mathbf{s}_n$ into two compound vectors and is given as

$$\mathbf{B}\mathbf{p} = \mathbf{s}, \tag{4.6}$$

where $\mathbf{p}$ and $\mathbf{s}$ are now $N_x N_y N_z (N_t + 2)$ dimensional vectors. The sparsity pattern of $\mathbf{B}$ is shown in Fig. 4.1 for a cubic room having $N_x N_y N_z = N_x^3$ spatial samples and $N_t = 2$ temporal samples. It can be noticed that for $n = -1$ and $n = 0$ the ICs (here assumed to be zero) are enforced using an identity matrix $\mathbf{I}$.

4.3 The optimization algorithm

For a given room geometry and the source distribution, K microphones are used to record the sound field in the room for N_t time samples. The impedance vector $\boldsymbol{\xi} \in \mathbb{R}^{N_\xi}$, which contains N_ξ elements that model the acoustic impedance surfaces of the room, can be estimated by solving the following optimization problem [76]

$$\begin{aligned} \min_{\boldsymbol{\xi}} \quad & f = \frac{1}{2}\|\mathbf{F}\mathbf{p} - \tilde{\mathbf{p}}\|_2^2 \\ \text{s. t.} \quad & \mathbf{B}\mathbf{p} = \mathbf{s}, \\ & \boldsymbol{\xi}_{\min} \leq \boldsymbol{\xi} \leq \boldsymbol{\xi}_{\max}, \end{aligned} \tag{4.7}$$

The cost function f represents the misfit function, *i.e.*, the l_2-norm of the residual between the measured sound pressure $\tilde{\mathbf{p}}$ and the sound pressure produced by the FDTD method at the microphone positions. $\mathbf{F}$ is a selection matrix that selects the KN_t samples out of $\mathbf{p}$ that correspond to the measured sound pressure samples of $\tilde{\mathbf{p}}$. The equality constraint is the linear system of equations given by the FDTD method and the inequalities are box constraints to prevent $\boldsymbol{\xi}$ to reach non-physical values *e.g.*, a negative acoustic impedance.

Due to the fact that the acoustic impedances contained in $\boldsymbol{\xi}$ appear in the matrix $\mathbf{B}$, the equality constraint of (4.7) is nonlinear. This optimization problem is therefore non-convex and can be solved using sequential quadratic programming (SQP) [77]. Starting from an initial guess $\boldsymbol{\xi}_0$, and substituting the equality constraint into the cost function, (4.7) can be locally approximated by the following quadratic optimization problem:

$$\begin{aligned} \min_{\mathbf{p}_k} \quad & f(\boldsymbol{\xi}_k) + \nabla f(\boldsymbol{\xi}_k)^T \mathbf{d}_k + \mathbf{d}_k^T \nabla^2 f(\boldsymbol{\xi}_k) \mathbf{d}_k \\ \text{s. t.} \quad & \boldsymbol{\xi}_{\min} \leq \boldsymbol{\xi} \leq \boldsymbol{\xi}_{\max}, \end{aligned} \tag{4.8}$$

where the step $\mathbf{d}_k = \boldsymbol{\xi}_k - \boldsymbol{\xi}_{k-1}$ gives a new value $\boldsymbol{\xi}_k$. After having ensured with a proper line-search method [77] that the step gives a sufficient decrease of the cost function this procedure can be repeated iteratively until a local minimum is found.

At each iteration of the SQP method, the gradient $\nabla f(\boldsymbol{\xi}_k)$ and the Hessian $\nabla^2 f(\boldsymbol{\xi}_k)$ are needed. Typically $\nabla f(\boldsymbol{\xi}_k)$ is computed numerically, *e.g.*, using finite difference, while the Hessian $\nabla^2 f(\boldsymbol{\xi}_k)$ is usually replaced by an approximation using techniques such as Gauss-Newton (GN) or BFGS [77], since its numerical computation is expensive. In [76] $\nabla f(\boldsymbol{\xi}_k)$ was evaluated using finite difference, which required the computation of a FDTD simulation for each

component of $\boldsymbol{\xi}$. However, when the number of sought acoustic impedances is high, such an approach can easily become unfeasible. In the following subsection the adjoint method is described specifically for the FDTD approach. This method enables the calculation of the gradient with the computation of only two FDTD simulations, almost independently of the number of impedances [170].

4.3.1 The Adjoint Method

The derivative of the cost function with respect to an arbitrary acoustic impedance ξ_b may be written as

$$\frac{\partial f}{\partial \xi_b}(\boldsymbol{\xi}) = \frac{\partial \mathbf{p}}{\partial \xi_b}^T \mathbf{F}^T(\mathbf{F}\mathbf{p} - \tilde{\mathbf{p}}). \tag{4.9}$$

Here $\frac{\partial \mathbf{p}}{\partial \xi_b}$ represents the computational bottleneck for the computation of the gradient. Taking the derivative with respect to ξ_b of the equality constraint of (4.7)

$$\frac{\partial \mathbf{B}}{\partial \xi_b}\mathbf{p} + \mathbf{B}\frac{\partial \mathbf{p}}{\partial \xi_b} = 0, \tag{4.10}$$

it can be seen that $\frac{\partial \mathbf{B}}{\partial \xi_b}$ actually represents an extremely sparse matrix: looking at Fig. 4.1, ξ_b appears only in the diagonal matrices $\mathbf{Q}_\pm$ where the BCs are imposed. From (4.10) it follows that

$$\frac{\partial \mathbf{p}}{\partial \xi_b} = -\mathbf{B}^{-1}\frac{\partial \mathbf{B}}{\partial \xi_b}\mathbf{p} \tag{4.11}$$

and substituting this into (4.9), leads to

$$\frac{\partial f}{\partial \xi_b}(\boldsymbol{\xi}) = -\left[\frac{\partial \mathbf{B}}{\partial \xi_b}\mathbf{p}\right]^T \underbrace{\left[\mathbf{B}^{-1}\right]^T \mathbf{F}^T(\mathbf{F}\mathbf{p} - \tilde{\mathbf{p}})}_{\boldsymbol{\lambda}}, \tag{4.12}$$

where $\boldsymbol{\lambda}$ is the solution of the adjoint problem:

$$\boldsymbol{\lambda}^T\mathbf{B} = \mathbf{F}^T(\mathbf{F}\mathbf{p} - \tilde{\mathbf{p}}) = \hat{\mathbf{s}}. \tag{4.13}$$

Here $\hat{\mathbf{s}}$ consists of a new source distribution. The residual $(\mathbf{F}\mathbf{p} - \tilde{\mathbf{p}})$ is expanded into a vector of the same size as $\mathbf{p}$ due to the transpose of the selection matrix $\mathbf{F}$. Hence in the adjoint problem the source distribution consists of point sources appearing at the microphone positions where a particular source signal is given as the residual between measured and simulated signals of the corresponding microphone. It can be noticed that such a problem could be solved iteratively using:

$$-\mathbf{Q}_-\boldsymbol{\lambda}_{n+1} - \mathbf{A}^T\boldsymbol{\lambda}_n + \mathbf{Q}_+\boldsymbol{\lambda}_{n-1} = \hat{\mathbf{s}}_n. \tag{4.14}$$

However, if a time inversion is applied the system of equations becomes the same as the one in (4.5) with the only difference that $\mathbf{A}$ is transposed and that the source distribution consists of a set of sources positioned at the microphone positions and where the source signals are now the time-reversed residuals. Finally, looking back at equation (4.12), it can be seen that in order to obtain the full gradient $\nabla f(\boldsymbol{\xi})$, each of the sparse matrices $\frac{\partial \mathbf{B}}{\partial \xi_b}$ for $i = 1 \dots N_\xi$ must be multiplied by $\mathbf{p}$ and these products are then multiplied by the solution of the adjoint problem $\boldsymbol{\lambda}$. Hence only two FDTD simulations are needed to obtain $\mathbf{p}$ and $\boldsymbol{\lambda}$ for each iteration of the SQP procedure.

A more natural choice of the decision variables of the optimization problem would be the admittance, *i.e.*, the inverse of the acoustic impedance, $\upsilon = 1/\xi$. In fact, in the computation of $\frac{\partial \mathbf{B}}{\partial \upsilon_b}\mathbf{p}$ the derivative of (4.4) with respect to υ_b will simply be $\lambda_c(p^{n+1}_{l,m,i} - p^{n-1}_{l,m,i})$. If the derivative with respect to ξ_b is computed, the acoustic impedance ξ_b would still be present in the last expression. For this reason, in the following the impedances will be replaced by admittances and the decision variables $\boldsymbol{\xi}$ will be represented by $\boldsymbol{\upsilon}$, the vector containing the admittances used to model the admittance surfaces. Hence, for the calculation of $\mathbf{p}$, it is only necessary to save the sound pressure difference $\lambda_c(p^{n+1}_{l,m,i} - p^{n-1}_{l,m,i})$ at the boundary positions, which represents computing $\frac{\partial \mathbf{B}}{\partial \upsilon_b}\mathbf{p}$. This data can then be used in the iterative procedure of solving the adjoint problem, to directly compute the product $(\frac{\partial \mathbf{B}}{\partial \upsilon_b}\mathbf{p})^T\boldsymbol{\lambda}$ of (4.12). Therefore neither $\mathbf{p}$ nor $\boldsymbol{\lambda}$ have to be fully stored.

4.3.2 Tikhonov regularization

The optimization problem (4.7) is an inverse problem and depending on the measured sound pressure $\tilde{\mathbf{p}}$ it can become ill-posed. This condition can occur when the microphone signals contain redundant information or when the system is not fully excited, *e.g.*, when a narrowband signal is used in the source distribution. In fact, at high frequencies the solution of the FDTD method is corrupted by numerical errors. In particular, the isotropic scheme used in this paper has a numerical relative error inferior to 2% up to $0.175 f_s$, where f_s is the sampling frequency. Therefore the data fitting should be performed below this frequency upper limit. Moreover it is well known that admittances are frequency-dependent, meaning that their estimation should be performed in narrow bands where they can be assumed to be frequency-independent. In order to compensate for the ill-posed nature of the problem, the cost function can be modified by adding a Tikhonov regularization [172]

$$\hat{f} = f + \frac{\lambda_x}{2}\left(\|\mathbf{W}_x\boldsymbol{\upsilon}\|_2^2 + \|\mathbf{W}_y\boldsymbol{\upsilon}\|_2^2 + \|\mathbf{W}_{xy}\boldsymbol{\upsilon}\|_2^2\right), \tag{4.15}$$

where f is the misfit function found in (4.7), $\hat{f}$ is the regularized cost function and $\mathbf{W}_i$ are weighting matrices that approximate the gradient $\nabla_i \boldsymbol{\upsilon}$ on the admittance surface over the directions $i = (x, y, xy)$ using finite difference operators. This type of regularization enforces smoothness on the estimated admittance surfaces by minimizing their gradient. The parameter λ_x weights this smoothing operation over the misfit function. If there is a lack of information in $\tilde{\mathbf{p}}$ the regularization gives higher preference to a smooth solution. The weight of the smoothing operator should be carefully chosen not to be stronger than the weight of the misfit function, otherwise the optimization would not fully exploit all the information contained in the measured data.

The gradient of the regularized cost function becomes

$$\nabla \hat{f}(\boldsymbol{\upsilon}) = \nabla f(\boldsymbol{\upsilon}) + \lambda_x \left(\mathbf{W}_x^T \mathbf{W}_x \boldsymbol{\upsilon} + \mathbf{W}_y^T \mathbf{W}_y \boldsymbol{\upsilon} + \mathbf{W}_{xy}^T \mathbf{W}_{xy} \boldsymbol{\upsilon} \right), \tag{4.16}$$

where $\nabla f(\boldsymbol{\upsilon})$ is obtained using the adjoint method. The $\mathbf{W}_i$ matrices can be constructed as follows. Let the vector $\boldsymbol{\upsilon}$ be the vectorization of the admittances of the various surfaces of the room, *e.g.*, for a cubic room

$$\boldsymbol{\upsilon} = [\boldsymbol{\upsilon}_{lw}^T, \boldsymbol{\upsilon}_{rw}^T, \boldsymbol{\upsilon}_{fw}^T, \boldsymbol{\upsilon}_{rew}^T, \boldsymbol{\upsilon}_{c}^T, \boldsymbol{\upsilon}_{f}^T]^T, \tag{4.17}$$

where the subscripts indicate left wall, right wall, front wall, rear wall, ceiling and floor. Let $\boldsymbol{\upsilon}_{lw}$ be an $N_x N_y$ vectorization of an $N_x \times N_y$ admittance surface obtained by stacking the N_x long columns of the admittance surface. A one-dimensional finite difference matrix can be constructed

$$\mathbf{D}_{x,1D} = \begin{pmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{pmatrix} \in \mathbb{R}^{(N_x - 1) \times N_x} \tag{4.18}$$

which can be used to obtain the gradient of an N_x long vector. Let $\mathbf{I}_{N_y}$ be the $N_y \times N_y$ identity matrix, if a two dimensional finite difference operator over the x direction is wanted, this can be obtained by the following Kronecker product

$$\mathbf{D}_{x,2D} = \mathbf{I}_{N_y} \otimes \mathbf{D}_{x,1D}. \tag{4.19}$$

Similarly the y direction gradient and the transverse direction gradient can be obtained by $\mathbf{D}_{y,2D} = \mathbf{D}_{y,1D} \otimes \mathbf{I}_{N_x}$ and $\mathbf{D}_{xy,2D} = \mathbf{D}_{y,1D} \otimes \mathbf{D}_{x,1D}$, respectively. The final $\mathbf{W}_i$ is given by the block diagonal matrix of the various $\mathbf{D}_{i,2D}$ for all the different admittance surface vectors contained in (4.17).

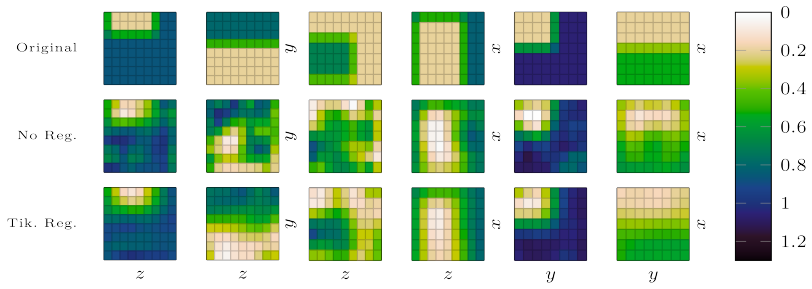


Figure 4.2: Surface admittances of the six faces of the cuboid room: left wall, right wall, rear wall, front wall, floor and ceiling. First row figures are the original surface admittances, second and third row is the estimated admittances for each wall using a narrowband signal as source without regularization and with Tikhonov regularization, respectively. All admittances are normalized by the maximum value of the original admittance for each wall.

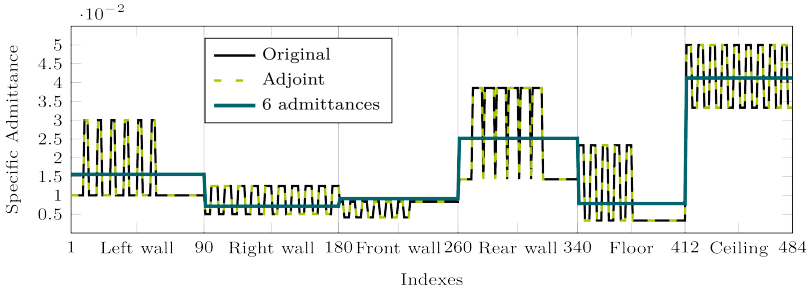


Figure 4.3: Plot of the original $\boldsymbol{\upsilon}$ vector and the estimated ones using the adjoint method and the GN method of [76] where only 6 admittances are used during the inversion procedure. The vertical lines divide the indexes corresponding to the different admittance surfaces. The original admittance values can also be viewed in 2D on the top row of Fig. 4.2.

4.4 Simulation Results

4.4.1 Simulation Set-up

The sound field of a relatively small cubic room of dimensions $4.4 \times 4.8 \times 5.3$ m^3 is first simulated using the FDTD method for a duration of 2 seconds ($N_t = 2f_s$), much longer then the room reverberation time. A low sampling frequency of $f_s = 891$ Hz is chosen to limit the computational burden of the simulations,

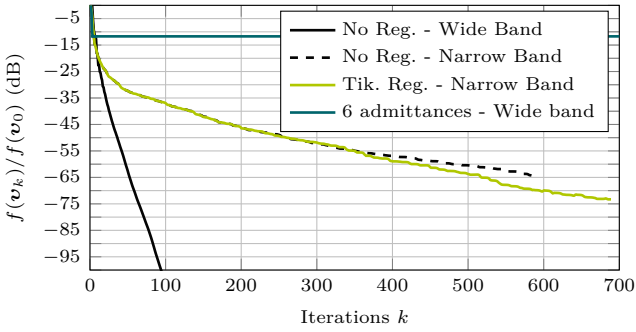


Figure 4.4: Convergence curve of the relative misfit function using wide band and narrow band source signals with no regularization, Tikhonov regularization using the adjoint method and using the method of [76].

resulting in a spatially uniform grid of dimension $(N_x \times N_y \times N_z) = (10 \times 11 \times 12)$. Each wall has different admittance surfaces with inhomogeneities: these are shown in the top row of Fig. 4.2 and resemble the presence of objects on the walls such as windows, curtains, carpets or furniture. A point source excites the system and 6 microphones at fixed random positions are used to obtain $\tilde{\mathbf{p}}$. This $\tilde{\mathbf{p}}$ is then used to invert the system and reconstruct the admittance surfaces as explained in section 4.3. The optimization procedure is stopped either when the reduction of the relative misfit function satisfies $f(\boldsymbol{\upsilon}_k)/f(\boldsymbol{\upsilon}_0) < \epsilon = 10^{-10}$, corresponding to -100 dB, or if line search failure occurs. The optimization problem is initialized for $\upsilon = 1/40$ for all values of $\boldsymbol{\upsilon}$, a value that gives reverberation time that is close to the one of the original room. The sought $\boldsymbol{\upsilon}$ vector is a $2((N_x - 2 + N_y - 2)(N_z - 2) + (N_x - 2)(N_y - 2)) = 484$ dimensional vector. The -2 is due to the fact that the admittances parameters υ_b that appear at corners and edges are actually copied from the neighbor ones belonging to the admittance surfaces. This is done to enforce continuity between two or three intersecting admittance surfaces.

Fig. 4.3 shows the original $\boldsymbol{\upsilon}$ vector and the estimated one using the adjoint method described in section 4.3.1 (without regularization) and using the method presented in [76], where only 6 admittances are used to model each of the 6 admittance surfaces. The SQP is solved using damped BFGS for the adjoint method while GN [77] is used for the 6 admittances case. The source signal consists of wide band bandpass filtered white noise, between 20-440 Hz.

It can be noticed that when only 6 decision variables are used, the estimated admittances are good averages of the admittances of the corresponding admittance surfaces. However, as it can be seen in Fig. 4.4, the method

fails to fit the data: the minimization is stopped due to line search failure after 10 iterations only and relative reduction of the misfit function reaches only -11 dB. On the other hand the adjoint method reaches the local minimum in 95 iterations and a perfect reconstruction is achieved, as can be seen in Fig. 4.3.

When a wideband signal is used, the optimization problem is well-posed and perfect reconstruction is achieved. This is not the case when a narrowband signal is used which, for the reasons described in section 4.3.2, would be the case when applying the method with real measured signals. In the following cases the source signal consists of bandpass filtered white noise between 44-88 Hz. If no regularization is applied, looking at the second row of Fig. 4.2, the estimated admittance surfaces resemble the original ones only partially. Moreover, in Fig. 4.4, it can be seen how the convergence rate of the SQP is reduced and a relative reduction of -64 dB is achieved. Nevertheless, using Tikhonov regularization, this situation can be slightly improved: after tuning the regularization parameter to $\lambda_x = 2 \cdot 10^{-3}$, it can be noticed how in Fig. 4.2 the estimated surfaces are indeed smoother and resemble much more the original ones. The convergence rate is also improved and the misfit reaches a relative reduction of -73 dB, as shown in Fig. 4.4.

4.5 Conclusions

The adjoint method was used to solve an optimization problem that can estimate the acoustic impedance surfaces. The room acoustics were modeled using the FDTD method. Compared to previous works [75, 76], it has been shown that using only one decision variable to model each of the acoustic impedance surfaces can lead to the failure of the methods of [75, 76]. The adjoint method enables the usage of large numbers of decision variables so that the acoustic impedance surfaces can be accurately estimated even when inhomogeneities are present. Nevertheless, when narrowband signals are employed in the source distribution, a condition that is required due to the numerical errors that the FDTD method introduces at high frequencies and due to the assumption of frequency-independent impedances, the problem can become ill-posed leading to a worse convergence of the optimization method and reduced estimation accuracy. The usage of Tikhonov regularization can then help to obtain better results. Future work will focus on the usage of more effective regularizations and application of the method using real data.

Chapter 5

Source localization and signal reconstruction using the FDTD method

Source localization and signal reconstruction in a reverberant field using the FDTD method

Niccolò Antonello, Toon van Waterschoot, Marc Moonen, and Patrick A. Naylor

Published in *Proc. 22nd European Signal Process. Conf. (EUSIPCO 2014)*, Lisbon, Portugal, Sep. 2014.

© 2014 IEEE. Reprinted, with permission, from:
N. Antonello, T. van Waterschoot, M. Moonen, and P. Naylor, "Source localization and signal reconstruction in a reverberant field using the FDTD method", in *Proc. 22nd European Signal Process. Conf. (EUSIPCO 2014)*, Lisbon, Portugal, Sep. 2014.
Changes include layout, representation, and minor editing aspects.

The candidate's contributions as first author include: literature study, co-development of the presented method, software implementation and simulations, co-formulation of the conclusions, text redaction and editing.

Abstract

Numerical methods applied to room acoustics are usually employed to predict the sound pressure at certain positions generated by a known source. In this paper the inverse problem is studied: given a number of microphones placed in a room, the sound pressure is known at these positions and this information may be used to perform a localization and signal reconstruction of the sound source. The source is assumed to be spatially sparse meaning it can be modeled as a point source. The finite difference time domain method is used to model the acoustics of a simple two dimensional square room and its matrix formulation is presented. A two step method is proposed. First a convex optimization problem is solved to localize the source while exploiting its spatial sparsity. Once its position is known the source signal can be reconstructed by solving an overdetermined system of linear equations.

5.1 Introduction

Source localization and signal reconstruction represent a challenge for researchers in many fields. One of the most popular approaches for source localization is beamforming of which robustness in a reverberant environment is still a topic of research [173]. Source signal reconstruction is also a challenging problem that has been studied in the context of dereverberation [51].

In [49] a method was introduced for the estimation of a static field using wireless sensor network measurements combined with the finite element method (FEM). The source was assumed to be a point source and its spatial sparsity was exploited. It was shown that the field could be accurately estimated by solving a convex optimization problem with ℓ^1-norm regularization. A similar approach involving the wave equation was proposed by other authors: here multi-channel microphone measurements combined with greedy algorithms were used to localize a sound source in a two-dimensional room with a frequency domain approach using the FEM [44] or with a time domain approach using finite differences [174].

In this paper a new method is presented that aims not only to localize the sound source but also to reconstruct the source signal once its position is determined. This is achieved by solving an inverse problem which exploits the reverberation of the room. In contrast to [44, 49] the finite difference time domain (FDTD) method is used to model the sound field. This is a method which is less computationally expensive than the FEM and which is a subject

of intensive study [15, 152]. Moreover, convex optimization is employed instead of the greedy algorithms used in [44, 174].

The paper is organized as follows: In Section 2 the FDTD method is briefly reviewed. In Section 3 the matrix formulation of the FDTD method is derived. In Section 4 the two step method for source localization and signal reconstruction is described. Finally, in Session 5 simulation results are shown.

5.2 The finite difference time domain method

In this section the FDTD method is briefly introduced. The sound field in a two-dimensional room can be described by the following partial differential equation (PDE) [4]:

$$\begin{aligned} &\text{PDE} \quad \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = s \text{ on } \Omega \times \tau \\ &\text{BCs} \quad \frac{\partial p}{\partial t} = -c\xi \nabla p \cdot \mathbf{n} \text{ on } \partial\Omega \times \tau \\ &\text{ICs} \quad p(x, y, t_0) = \tilde{p}_0 \text{ and } \frac{\partial p}{\partial t}(x, y, t_0) = \tilde{u}_0 \text{ on } \Omega \end{aligned} \tag{5.1}$$

Here, $\Omega \in \mathbb{R}^2$ represents the spatial domain, which defines the room geometry, τ represents the temporal domain, $p(x, y, t)$ and $s(x, y, t)$ represent the sound pressure and the source signal, respectively, which are both functions $p, s : \Omega \times \tau \to \mathbb{R}$, with $(x, y) \in \Omega$, $t \in \tau$ and $\Omega \times \tau \subset \mathbb{R}^3$ and c is the speed of sound. Notice that all the results presented in this paper can be easily extended to the three-dimensional case.

The boundary conditions (BCs) are used to model acoustical properties of the walls which introduce damping in the system. ξ defines the impedance of the walls and is assumed here to be frequency independent. The operator ∇ is defined as $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ and $\mathbf{n}$ is the normal vector with respect to the boundaries. The initial conditions (ICs) provide the initialization of the problem, both in the sound pressure and its time derivative, namely the initial particle velocity $\tilde{u}_0$. Notice that $\tilde{u}_0$ and $\tilde{p}_0$ are defined in $\Omega \subset \mathbb{R}^2$.

Finding the solution p of the problem (5.1) for a given s is often a difficult task. Numerical approaches have to be used and these usually consist in discretizing the domain $\Omega \times \tau$ in order to reduce the problem to a linear system of equations. In the FDTD method this discretization is performed in a simple fashion: p

and s are sampled uniformly in space and time,

$$p(x, y, t) = p(lX, mY, nT) = p^n_{l,m}, \tag{5.2}$$

where X and Y represent the spatial steps with respect to the x and y axis respectively and T the time step. The same notation is used for the discretized source term $s^n_{l,m}$. In the following X is set to be equal to Y.

The simplest FDTD scheme is obtained by approximating the space and time second order derivatives as follows, *e.g.*, for the time derivative:

$$\frac{\partial^2 p}{\partial t^2} \approx \frac{p^{n+1}_{l,m} - 2p^n_{l,m} + p^{n-1}_{l,m}}{T^2}. \tag{5.3}$$

By substituting these approximations in the wave equation (5.1) it is possible to write the *standard leapfrog* (SLF) scheme update equation

$$\begin{aligned} p^{n+1}_{l,m} =& \lambda_c^2 (p^n_{l+1,m} + p^n_{l-1,m} + p^n_{l,m+1} + p^n_{l,m-1}) + \\ & 2(1 - 2\lambda_c^2) p^n_{l,m} - p^{n-1}_{l,m} + s^n_{l,m}, \end{aligned} \tag{5.4}$$

where $\lambda_c = cT/X$ is the Courant number.

The update equation (5.4) reveals the iterative nature of the FDTD method. The sound pressure at time index $n+1$ is evaluated by using samples from the previous two time indices. In particular, multiple spatial samples of the sound pressure at time index n are employed; in the SLF scheme only the axially neighboring samples are used.

The Courant number is an important parameter in terms of stability and numerical errors [15]. For the SLF scheme in a two-dimensional case the Courant number is chosen to be $1/\sqrt{2}$. Lower values increase the numerical errors while higher values make the resulting solution unstable. This implies that the resolution in time and space are inherently connected by the maximum of the achievable Courant number.

Numerical errors in the FDTD method result in dispersion, *i.e.*, waves travel with different speed depending on their frequency and direction in space. The most common approach to reduce numerical errors is to use a different scheme to discretize the derivatives of the wave equation. The general update equation is

$$\begin{aligned} p^{n+1}_{l,m} =& d_1 \left(p^n_{l+1,m} + p^n_{l-1,m} + p^n_{l,m+1} + p^n_{l,m-1} \right) + \\ & d_2 (p^n_{l+1,m+1} + p^n_{l+1,m-1} + p^n_{l-1,m+1} + p^n_{l-1,m-1}) + \\ & d_3 p^n_{l,m} - p^{n-1}_{l,m} + s^n_{l,m} \end{aligned} \tag{5.5}$$

where the parameters d_1, d_2 and d_3 define the scheme (*e.g.*, for SLF $d_1 = 1/2$ and $d_2 = d_3 = 0$). A collection of the currently known stable schemes can be found in [15].

The update equation (5.5) must be modified at the boundaries, as it would require samples at grid points that lie outside the domain. These points are known as *ghost points* and can be used to model the BCs in (5.1). The modified update equation that has to be used for the boundary grid points is then given (*e.g.*, for a right wall) by

$$\begin{aligned}(1+\lambda_c/\xi_x)\,p_{l,m}^{n+1} =& d_1(2p_{l-1,m}^{n} + p_{l,m+1}^{n} + p_{l,m-1}^{n}) \\ & 2d_2(p_{l-1,m+1}^{n} + p_{l-1,m-1}^{n}) + d_3 p_{l,m}^{n} \\ & -(1-\lambda_c/\xi_x)\,p_{l,m}^{n-1}.\end{aligned} \tag{5.6}$$

Note that when a corner is present more grid points need to be eliminated. The update equation (5.5) can be further modified to include frequency-dependent boundaries using a digital filter at each boundary point [15].

Finally, initial conditions must be specified as well. The first-order derivative is approximated with a finite difference,

$$\frac{p_{l,m}^{+1} - p_{l,m}^{-1}}{2T} = \tilde{u}_{l,m}^{0}, \tag{5.7}$$

and $p_{l,m}^{0} = \tilde{p}_{l,m}^{0}$.

5.3 Matrix formulation of FDTD

In Section 2 it has been shown that the wave equation can be converted into a set of linear equations with FDTD. Usually this linear system is solved iteratively. However, since this set of equations will be used as a constraint in an optimization problem, it is more convenient to write it in the following manner:

$$\mathbf{B}\mathbf{p} = \mathbf{s} \tag{5.8}$$

where $\mathbf{B} \in \mathbb{R}^{N_X^2(N_T+2)\times N_X^2(N_T+2)}$ is a matrix containing geometry and boundaries information and $\mathbf{p}$ and $\mathbf{s} \in \mathbb{R}^{N_X^2(N_T+2)}$ are vectors containing respectively sound pressure and source signal samples at all positions and time samples. Here, N_X indicates the number of spatial samples along one axis (in the following it is assumed this is the same for the x and y axis) and $N_T + 2$ is the number of time samples plus 2 initial conditions.

The vector $\mathbf{p}$ is defined as follows:

$$\mathbf{p} = [\mathbf{p}_{-1}^T, \mathbf{p}_0^T, \dots \mathbf{p}_{N_T}^T]^T, \tag{5.9}$$

where $\mathbf{p}_n$ is the vector containing all the sound pressure samples at time indices n. The vector $\mathbf{s}$ has the same structure but includes also the initial conditions

$$\mathbf{s} = [\tilde{\mathbf{u}}_0^T, \tilde{\mathbf{p}}_0^T, \mathbf{s}_1^T, \dots \mathbf{s}_{N_T}^T]^T. \tag{5.10}$$

The update equation (5.5) can be written as:

$$p_{l,m}^{n+1} - \mathbf{p}_n^T \mathbf{a}_{l,m} + p_{l,m}^{n-1} = s_{l,m}^n \tag{5.11}$$

where $\mathbf{a}_{l,m} \in \mathbb{R}^{N_X^2}$ is a vector that selects the samples at position (l, m) and its neighbor points. Writing $\mathbf{p}_n$ as

$$\mathbf{p}_n = [\dots \overbrace{p_{l,m-1}^n, p_{l+1,m-1}^n \cdots}^{N_X} p_{l,m}^n, p_{l+1,m}^n \dots]^T, \tag{5.12}$$

it can be seen that $p_{l,m-1}^n$ is N_X samples away from $p_{l,m}^n$, and so if $\mathbf{p}_n[i] = p_{l,m}^n$ then $\mathbf{p}_n[i + N_X] = p_{l,m+1}^n$ and $\mathbf{p}_n[i - N_X] = p_{l,m-1}^n$. Hence $\mathbf{a}_{l,m}$ can be defined as:

$$\mathbf{a}_{l,m} = [\dots \overbrace{d_2, d_1, d_2 \dots}^{N_X} d_2, \overbrace{d_3}^{(l,m)}, d_2 \overbrace{\dots d_2, d_1, d_2}^{N_X} \dots]^T. \tag{5.13}$$

where dots represent zero values.

Now the matrix $\mathbf{A} \in \mathbb{R}^{N_X^2 \times N_X^2}$ can be defined as:

$$\mathbf{A} = [\mathbf{a}_{0,0}, \mathbf{a}_{1,0}, \dots \mathbf{a}_{N_X-1,N_X}, \mathbf{a}_{N_X,N_X}]. \tag{5.14}$$

This matrix consists of 9 diagonals. Equation (5.5) may be rewritten as

$$\mathbf{I}\mathbf{p}_{n+1} - \mathbf{A}\mathbf{p}_n + \mathbf{I}\mathbf{p}_{n-1} = \mathbf{s}_n \tag{5.15}$$

where $\mathbf{I}$ is the identity matrix.

The initial conditions are defined in the first $2N_X^2$ samples of the vector $\mathbf{s}$, see (5.10). For the initial particle velocity, the first N_X^2 samples must satisfy (5.7) which in vector notation can be written as $(\mathbf{I}\mathbf{p}_{+1} - \mathbf{I}\mathbf{p}_{-1})/2T = \tilde{\mathbf{u}}_0$ and similarly for the initial sound pressure $\mathbf{I}\mathbf{p}_0 = \tilde{\mathbf{p}}_0$.

As for the boundary conditions, the above equations have to be modified according to (5.6) and similar equations for other walls and corners. Due to these modifications, the $\mathbf{A}$ matrix will have zero elements in the diagonals since the ghost points must be removed. Moreover, the boundary samples $\mathbf{p}_{n+1}$

and $\mathbf{p}_{n-1}$ are multiplied by a constant representing the impedance and hence the identity matrices in (5.15) have to be slightly modified. The FDTD matrix can finally be written as

$$\mathbf{B} = \begin{pmatrix} \mathbf{I}/2T & & -\mathbf{I}/2T & \\ & \mathbf{I} & & \\ \mathbf{I}_{+1} & -\mathbf{A} & \mathbf{I}_{-1} & \\ & \ddots & \ddots & \ddots \\ & \mathbf{I}_{+1} & -\mathbf{A} & \mathbf{I}_{-1} \end{pmatrix}, \tag{5.16}$$

where $\mathbf{I}_{+1}$ is an identity matrix multiplied by the coefficients $(1+\lambda_c/\xi)$ in each row that belongs to a boundary sample (notice that the coefficient is different for corners) and $\mathbf{I}_{-1}$ is similarly defined with coefficients $(1-\lambda_c/\xi)$. Note that $\mathbf{B}$ can become a huge matrix for high spatial/temporal sampling rates. However, it has a sparse multi-diagonal structure which allows an efficient storage and computation.

5.4 Source localization and reconstruction

5.4.1 Reconstruction of an impulse

In the following it will be assumed that the sound field can be simulated correctly at each position of the room, *i.e.*, that FDTD returns an accurate physical model. Numerical methods are usually employed to predict the sound pressure from a known source signal $\mathbf{s}$. Here the inverse problem is considered: the sound pressure is known at a subset of the FDTD grid points where microphones are present, hence $\mathbf{p}$ is partially known and $\mathbf{s}$ must be estimated.

A convex optimization problem can be formulated as follows [49],

$$\begin{aligned} \underset{\mathbf{p},\mathbf{s}}{\text{minimize}} \quad & \sum_{i=1}^{K} \|\mathbf{D}_i\mathbf{p} - \hat{\mathbf{p}}_i\|_2^2 + \lambda\|\mathbf{C}\mathbf{s}\|_1 \\ \text{subject to} \quad & \mathbf{B}\mathbf{p} = \mathbf{s} \end{aligned} \tag{5.17}$$

where K is the number of microphones, $\hat{\mathbf{p}}_i$ is a vector containing the signal measured in the microphone i and $\mathbf{D}_i$ is a selection matrix that selects the samples at the microphone position in the vector $\mathbf{p}$. The equality constraints correspond to the FDTD method. The cost function consists of two terms: The squared error norm forces the sound pressure represented by $\mathbf{p}$ to be close to the sound pressure of the measurements in the microphones. The term

$\lambda\|\mathbf{Cs}\|_1$ is a *regularisation* term inducing sparsity in the source signal vector. This term is crucial for the problem to be solvable since the FDTD system of linear equations is heavily underdetermined. λ is a regularization parameter that can increase the importance of the sparsity-inducing term at the cost of a decreased accuracy between measured and simulated sound pressure values. $\mathbf{C}$ is a selection matrix that removes the first $2N_X^2$ samples corresponding to the initial conditions, where the source signal vector is no longer sparse but can be dense.

By imposing sparsity in such a manner, the source signal $s_{l,m}^n$ is enforced to be non-zero for only few indices l, m and n. Hence the regularization encourages solutions with *impulsive* sources. Simulations show that when the source signal is indeed an impulse in both time and space, reconstruction is possible even with only one microphone. In the next subsections the more interesting case of reconstructing and localizing a more general source signal will be studied.

5.4.2 Source localization

When a general point source signal is used instead of an impulse, problem (5.17) does not allow source signal reconstruction to be performed but it can achieve a correct source localization.

In fact, if the estimated source signal, without the ICs part, is squared and summed over time for all l,m, its maximum value over l, m is found to reveal a good estimate of the position $[l_s, m_s]$ of the source

$$[l_s, m_s] = \underset{l,m}{\operatorname{argmax}} \sum_{n=1}^{N_T} (s_{l,m}^n)^2. \tag{5.18}$$

5.4.3 Reconstruction with known position

It is possible to exploit the estimated source position to perform the source signal reconstruction. The source signal vector $\mathbf{s}$ may be written as

$$\mathbf{s} = \mathbf{F}\mathbf{s}_T \tag{5.19}$$

where $\mathbf{s}_T \in \mathbb{R}^{N_T}$ is a vector containing the time signal at the estimated source position and $\mathbf{F} \in \mathbb{R}^{N_X^2(N_T+2)\times N_T}$ is an expansion matrix. Let k be the index of the estimated source position corresponding to $[l_s, m_s]$ in the vectors $\mathbf{s}_i$ for $i = 1, \ldots, N_T$. The upper $2N_X^2$ rows of $\mathbf{F}$ will have only zeros, since these correspond to the initial conditions $\tilde{\mathbf{u}}_0$ and $\tilde{\mathbf{p}}_0$ that are not present in $\mathbf{s}_T$. The next N_X^2 rows correspond to $\mathbf{s}_1$ and have only one non-zero element in position

$[k + 2N_X^2, 1]$. Then the next N_X^2 rows have a 1 in position $[k + 3N_X^2, 2]$ and so on until the second index reaches N_T.

The sound pressure vector can then be written as:

$$\mathbf{p} = \mathbf{Z}\mathbf{s}_T \tag{5.20}$$

where $\mathbf{Z} = \mathbf{B}^{-1}\mathbf{F} \in \mathbb{R}^{N_X^2(N_T+2)\times N_T}$.

The vector $\mathbf{s}_T$ can thus be obtained by simply solving:

$$\mathbf{D}\mathbf{Z}\mathbf{s}_T = \mathbf{D}\mathbf{p} \tag{5.21}$$

in the least square sense, which is always an overdetermined system. Here $\mathbf{D}$ represent the selection matrix that selects all the rows where microphone measurements are present.

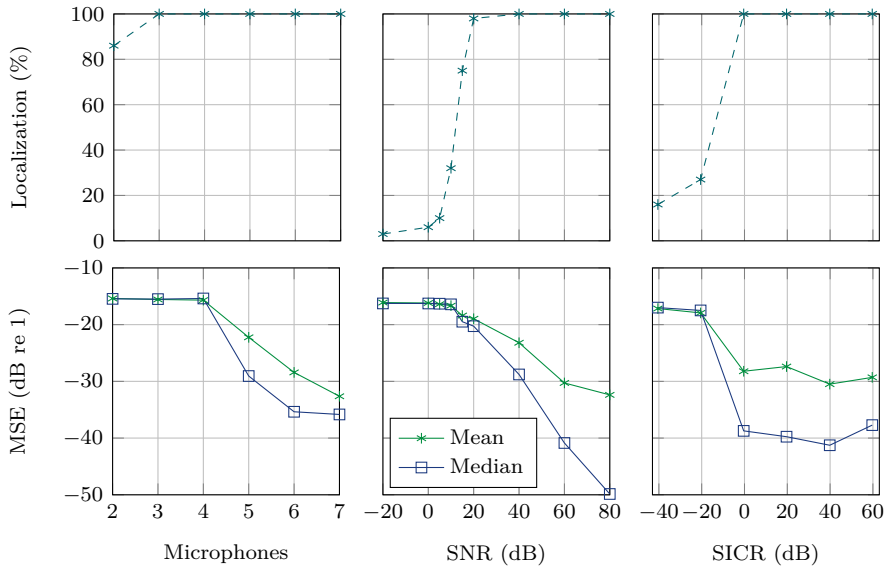


Figure 5.1: Localization of source in probability and reconstruction error for different numbers of microphones, different values of SNR and SICR.

5.4.4 Reconstruction without initial conditions

It is also possible to create an expansion matrix that reconstructs the ICs. The matrix $\mathbf{F}$ then has to be modified in the following manner:

$$\mathbf{F}_{\mathrm{IC}} = \begin{pmatrix} \mathbf{I}_{2N_X^2 \times 2N_X^2} & \vert & \\ & \vert & \mathbf{F} \\ \mathbf{0}_{N_T N_X^2 \times 2N_X^2} & \vert & \end{pmatrix}. \tag{5.22}$$

$\mathbf{F}_{\mathrm{IC}}$ then becomes a $(N_T + 2)N_X^2 \times (2N_X^2 + N_T)$ matrix and the signal can be reconstructed by solving (5.21), which is an overdetermined set of equations only if $KN_T \geq 2N_X^2 + N_T$ since now $\mathbf{DZ} \in \mathbb{R}^{KN_T \times (2N_X^2 + N_T)}$. This problem manages to partially reconstruct the initial conditions. As simulation results in Section 5 will show, the source signal can be reconstructed with high accuracy, even though the first few samples should be discarded since these are usually corrupted by the inaccurate reconstruction of the initial conditions.

Being able to have a method that does not need the initial conditions means that the signal can be reconstructed using shorter successive time windows. This reduces the numerical cost since the matrices that have to be computed become smaller.

5.5 Simulation results

An FDTD simulation is performed in a small 1 m^2 2D room using a spatial resolution of $N_X = 10$ and $N_T = 500$ time samples, meaning a sampling frequency of $f_S = 4.4$ kHz, with the SLF scheme. The small size of the room is chosen to have a relatively small-scale optimization problem. The impedance is chosen to be $\xi = 200$, corresponding to a reverberation time of $T_{60} = 1.2$ s, and initial conditions are set to zero. The system is excited using a physically constrained source [152] of filtered white noise, from 20 to 600 Hz: the low and high frequency cutoffs are chosen in order to roughly model a realistic acoustic source without very low frequency components and to avoid numerical errors that are present in the FDTD at high frequencies, respectively. The K microphone signals are then simulated by adding Gaussian white noise (GWN) to the FDTD field values at the microphone positions. The source localization problem (5.17) is then solved with `CVX` [106]. Here the parameter λ is set to 10^{-4} meaning sparsity is not strongly enforced.

Figure 5.1 shows the results of Monte Carlo simulations. For each point in the graph 100 simulations were run using random positions of the microphones and source. The figures on top show the probability of correct localization (*i.e.*,

finding the correct grid point) while the ones on the bottom show the mean squared error (MSE) between the original and reconstructed signal. Both the mean and median of the MC simulations for the MSE are shown in order to emphasize the fact that few outliers significantly decrease the average of the MSE. The plots on the left show simulation results for different numbers of microphones. Here, GWN was added to each microphone measurement with 40 dB signal to noise ration (SNR). Localization of the source works efficiently with only three microphones but at least 5 are needed to perform an accurate signal reconstruction. The graphs on the middle show results for different values of SNR, using 5 microphones. Again, the performance of the localization is somewhat better than the signal reconstruction performance: at 20 dB SNR the source is localized correctly with a 98% probability while 40 dB SNR is needed for good signal reconstruction. Finally, the plots on the right show the behavior when ICs are non-zero. The initial particle velocity and initial sound pressure are set to $\tilde{u}^0_{l,m} = \sin(2\pi 2l/N_X)$ and $\tilde{p}^0_{l,m} = -10^{-3}\sin(2\pi m/N_X)$, roughly modeling a steady-state sound field. The field that these ICs generate is then added to the one generated by the sound source and multiplied by a constant. In fact here simulations are repeated for different values of signal to initial conditions ratio (SICR), *i.e.*, the ratio between the sound pressure power due to the sound source and the sound pressure power due to the ICs recorded by the microphones. In these simulations GWN was added in each microphone measurement corresponding to a SNR of 60 dB. The ICs can be thought as the source of *noise coming from the past*. This type of noise can be removed efficiently when the ICs can be estimated correctly. The resulting localization and signal reconstruction performance is much more robust when most of the noise is coming from the previous sound field created by the ICs. Good accuracy can be achieved even at 0 dB SICR, meaning that the sound field due to the sound source and the one due to the ICs have the same magnitude.

5.6 Conclusions

A new two step method for localization and signal reconstruction using the FDTD method has been proposed. First the sound source is localized by solving a convex optimization problem that exploits spatial sparsity and then signal reconstruction is achieved by solving an overdetermined system of linear equations. Simulations have shown that the method can be robust against noise in particular when this is coming from a past sound field that can be partially estimated.

The main issue for the applicability of the proposed method in a real world scenario is that the geometry and boundary conditions of the room are assumed

to be known. Matching room acoustic properties with simulations can be a difficult task especially due to the modeling of the boundary conditions. Moreover, the computational cost of the FDTD method and the dispersion error it suffers can be problematic. These errors can be removed or reduced by either using very high resolutions, which would inevitably lead to a huge increase of the computational cost, or by limiting the method to the low frequency range.

Chapter 6

Sound field control using the FDTD method

Sound field control in a reverberant room using the Finite Difference Time Domain method

Niccolò Antonello, Enzo De Sena, Marc Moonen, Patrick A. Naylor, and Toon van Waterschoot

Published in *Proceedings AES 60th International Conference*, Leuven, Belgium, Feb. 2016

© 2016 Audio Engineering Society (AES). Reprinted, with permission, from:
N. Antonello, E. De Sena, M. Moonen, P. Naylor, and T. van Waterschoot, "Sound field control in a reverberant room using the Finite Difference Time Domain method," in *Proceedings AES 60th International Conference*, Leuven, Belgium, Feb. 2016.
Changes include layout, representation, and minor editing aspects.

The candidate's contributions as first author include: literature study, co-development of the presented method, software implementation and simulations, co-formulation of the conclusions, text redaction and editing.

Abstract

In this paper a novel approach for sound field reproduction and multi-zone sound field control in a reverberant room is proposed. Sound field reproduction and multi-zone sound field control are jointly formulated as an optimization problem. The optimization problem relies on a physical model of the room acoustics which is computed numerically using the finite-difference time-domain (FDTD) method. Furthermore a novel regularization of the optimization problem is proposed. A non-convex sparsity-inducing term is used to obtain convenient control source positions and control source signals simultaneously. Finally, simulation results are presented: the proposed method is used to recreate an anechoic sound in a highly reverberant room while keeping part of the room silent.

6.1 Introduction

Sound field reproduction techniques aim to reproduce a measured or synthesized acoustic wave field in a different acoustic environment. Many attempts to improve this technology have been pursued among which the most popular are ambisonics [175] and wave field synthesis [67] which try to physically reconstruct the sound field. Most of these techniques require large amounts of loudspeakers and assume low reverberation of the room where they are applied or even free field conditions. Other techniques focus on perceptual sound field reproduction [176].

Another field of research in acoustics is *multi-zone sound field control*: here the aim is to be able to play sound in a specific area of the room (bright zone) while keeping other locations silent (dark zone) [68]. Multi-zone sound field control shares many features with sound field reproduction. In particular, multi-zone sound field control pressure matching and sound field reproduction can both be formulated as optimization problems, more specifically as a regularized least squares (LSs) problems [68, 177, 178], also known as multi-channel equalization problems [52, 179]. It has been recently shown that the least absolute shrinkage and selection operator (LASSO), which seeks sparse solutions, can perform better than LS in both sound field reproduction [69] and multi-zone sound field control [70]. Here convenient placement of the control source positions can be first obtained solving the LASSO and a second optimization is then performed to obtain the control source signals once the optimal position is known.

In many of these approaches, the room impulse responses (RIRs) are usually unknown and need to be estimated. Having a physical room model can

compensate for this lack of information and a numerical method can be used to model the room acoustics and to generate the RIRs. In particular, the finite-difference time-domain (FDTD) method is increasingly becoming popular among the numerical methods for room acoustics simulations. This is due to the recently improved boundary conditions (BCs) formulation [16].

In this paper a novel approach for sound field reproduction and multi-zone sound field control is proposed. The novel aspects of this paper can be summarized as follows: (i) a non-convex sparsity-inducing term will be used in order to promote sparsity more efficiently than the LASSO. This will make it possible to optimize for the control source positions and control source signals simultaneously; (ii) the optimization problem is modified including an equality constraint that represents the physical model of the room where the algorithm is applied. This equality constraint is obtained through the FDTD method. This enables to compute the RIRs necessary for the sound field control and reproduction.

The paper is structured as follows: in Sec. 6.2 a brief review of the FDTD method is presented. In Sec. 6.3 the optimization algorithm is presented where a novel non-convex sparsity-inducing term is used and the FDTD method is used to compute the RIRs. Finally, in Sec. 6.4 simulation results will be presented. The proposed algorithm will be tested in a highly reverberant two-dimensional room creating an anechoic impulse while keeping the dark zone silent.

6.2 The finite difference time domain

The FDTD method is a numerical method which seeks a solution of a partial differential equation (PDE) through spatiotemporal discretization. It is known that solutions of PDEs with specific BCs and initial conditions (ICs) can be derived analytically only for simple geometries [4]. In the context of room acoustics, the wave equation models the acoustic wave phenomena [4]

$$\begin{aligned} \text{PDE} \quad & \triangle p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = s \text{ on } \Omega \times \tau \\ \text{BCs} \quad & \frac{\partial p}{\partial t} = -c\xi \nabla p \cdot \mathbf{n} \text{ on } \partial\Omega \times \tau \\ \text{ICs} \quad & \frac{\partial p}{\partial t} = \hat{p}_0, p = p_0 \text{ on } \Omega, \end{aligned} \tag{6.1}$$

where $\triangle$ is the Laplace operator, p and s are the sound pressure and source distribution, respectively, defined as scalar functions with spatial domain $\Omega \in \mathbb{R}^2$ (for a two-dimensional room) and temporal domain $\tau \in \mathbb{R}$. Here c is the speed

of sound, ξ is the specific acoustic impedance which models the acoustic losses occurring at the walls, $\mathbf{n}$ is the normal vector with respect to the boundary surface, p_0 is the initial sound pressure and $\hat{p}_0$ the initial sound pressure temporal derivative.

The idea of the FDTD method is to approximate the continuous scalar functions p and s on a uniform grid, *i.e.*, the sound pressure is sampled such that

$$p(x, y, t) \approx p(lX, mX, nT) = p^n_{l,m}, \tag{6.2}$$

where X is the spatial resolution and T is the temporal resolution. Furthermore the second order derivatives appearing in (6.1) are approximated using centered finite differences, *e.g.*, for the spatial second-order derivative over x that appears in the Laplace operator

$$\frac{\partial^2 p}{\partial x^2} \approx \frac{p^n_{l+1,m} - 2p^n_{l,m} + p^n_{l-1,m}}{X^2}. \tag{6.3}$$

Using these approximations the wave equation can be converted into a set of linear equations. For explicit FDTD schemes [16] the resulting update equation is given by

$$p^{n+1}_{l,m} = \Sigma P^n_{l,m} - p^{n-1}_{l,m} + s^n_{l,m,i}, \tag{6.4}$$

where the sound pressure sample at position (l, m) and time $n + 1$ is computed using the previous sample $p^{n-1}_{l,m}$ and a weighted sum, $\Sigma P^n_{l,m}$, of the samples of the 8 neighbor samples of $p^n_{l,m}$ and itself. The weights used in $\Sigma P^n_{l,m}$ determines the explicit scheme used to approximate the Laplace operator. Spatial and temporal resolutions are bounded by a stability constraint

$$\frac{cT}{X} \leq \lambda_c \tag{6.5}$$

where λ_c is the Courant number [16]. The ratio between T and X is usually chosen at the stability limit where numerical errors are minimized [16]. In order to take into account the BCs, the update equations must be modified at the boundaries by enforcing the approximation of the BCs in (6.1), which is performed again using finite differences [16].

Typically the set of linear equations given by the FDTD method is solved iteratively. However for easier readability, it is possible to write this system in vectorized notation:

$$\mathbf{Bp} = \mathbf{s} \tag{6.6}$$

where $\mathbf{B}$ is a sparse $N_x N_y (N_t + 2) \times N_x N_y (N_t + 2)$ matrix [180] and $\mathbf{p}$ and $\mathbf{s}$ are $N_x N_y (N_t + 2)$ vectors containing the $p^n_{l,m}$ and $s^n_{l,m}$ for all $l = 0 \dots N_x$, $m = 0 \dots N_y$ and $n = -2, -1, 0 \dots N_t$, where N_x and N_y are the number of

divisions of the uniform grid in the x and y directions respectively and N_t is the number of time samples. The $N_t + 2$ is due to the two ICs which here will be assumed to be zero. Notice that the bandwidth of the RIRs generated with the FDTD method is usually limited to a low frequency range due to the heavy computational load and the numerical errors of the FDTD method, which both increase with frequency.

6.3 The optimization algorithm

Sound field reproduction can be posed as a multi-channel equalization problem. This is an optimization problem that can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{s}_c} \quad & f = \frac{1}{2}\|\mathbf{F}_p\mathbf{p} - \tilde{\mathbf{p}}\|_2^2 \\ \text{s. t.} \quad & \mathbf{B}\mathbf{p} = \mathbf{F}_s^T\mathbf{s}_c, \end{aligned} \tag{6.7}$$

where f is the cost function which represents the squared l_2-norm of the difference between the vector containing the wanted sound pressure signals $\tilde{\mathbf{p}}$ and the room sound pressure signals $\mathbf{p}$ at some specific positions in the room which will be referred here as *control points*. Both vectors are $N_t \cdot K_p$ long, where K_p is the number of control points. The matrix $\mathbf{F}_p$ is a selection matrix which selects the samples of $\mathbf{p}$ that correspond to the control points. The equality constraint consists of the set of linear equations given by the FDTD method. Here $\mathbf{s}_c$ is a $N_s \cdot K_s$ vector containing the signals of the sought *control sources* at K_s positions having N_s samples each. The expansion matrix $\mathbf{F}_s^T$ expands $\mathbf{s}_c$ onto a vector having the same dimension as $\mathbf{p}$.

The optimization problem (6.7) is convex and can be solved analytically using LS. If the equality constraint is substituted into the cost function, one can write

$$\min_{\mathbf{s}_c} \frac{1}{2}\| \underbrace{\mathbf{F}_p\mathbf{B}^{-1}\mathbf{F}_s^T}_{\mathbf{G}}\mathbf{s}_c - \tilde{\mathbf{p}}\|_2^2 \tag{6.8}$$

where $\mathbf{G}$ is a rectangular matrix of dimensions $K_pN_t \times K_sN_s$. The solution of (6.8) is given by:

$$\mathbf{s}_c^* = -(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\tilde{\mathbf{p}}. \tag{6.9}$$

The LS problem is usually ill-posed: the matrix $\mathbf{G}^T\mathbf{G}$, which represents the Hessian of the cost function, can be non-invertible. In order to avoid this, a regularization term can be added to the cost function (see Sec. 6.3.3).

6.3.1 Computation of G

As described in the previous section, $\mathbf{G}$ is given by:

$$\mathbf{G} = \mathbf{F}_p \mathbf{B}^{-1} \mathbf{F}_s^T. \tag{6.10}$$

$\mathbf{G}$ actually consists of a collection of Toeplitz matrices of the RIRs between each of the control points and control source positions and its dimensions are $K_p \cdot N_t \times K_s \cdot N_s$. The matrices $\mathbf{F}_s^T$ and $\mathbf{F}_p$ contain ones at the indexes corresponding to the spatial and temporal positions of the control source positions and control points respectively.

$\mathbf{G}$ can be computed by solving a number of FDTD simulations. In the simple case where $K_p = K_s = 1$, $\mathbf{G}$ will consist of a single Toeplitz matrix and its first column $\mathbf{g}_0$ can be computed by solving

$$\mathbf{B}\mathbf{h}_s = \mathbf{f}_{s,0}, \tag{6.11}$$

where $\mathbf{f}_{s,0}$ is the first column of $\mathbf{F}_s^T$ which has only one non-zero element at the index of the control source position and time step $n = 0$. $\mathbf{g}_0$ can then be obtained using the selection matrix

$$\mathbf{g}_0 = \mathbf{F}_p \mathbf{h}_s. \tag{6.12}$$

The second column of $\mathbf{G}$ will actually consist of the vector $\mathbf{g}_0$ being time shifted by one sample. That is because the second column of $\mathbf{F}_s^T$ has only one non-zero element at the index of the control source position and time step $n = 1$.

Hence if $K_p > 1$ and $K_s = 1$, the FDTD simulation represented by (6.11) needs to be solved only once and multiple RIRs between control points and control source position can be generated simultaneously using (6.11). In general, if $K_p > K_s$, K_s FDTD simulations are needed to generate $\mathbf{G}$. On the other hand, if $K_p < K_s$ it is more efficient to swap control points and control source positions. The generated RIRs are in fact equivalent due to the reciprocity principle of the Green's function [4]. In this case only K_p FDTD simulations will be necessary.

6.3.2 Newton-type optimization

Since a non-convex regularization term will be used to regularize the ill-posed optimization problem (6.8), a closed-form solution cannot be achieved. Instead, an iterative method can be used to obtain a solution. The idea is to start from an initial guess of the control source signals, $\mathbf{s}_c^0$, compute the Newton step [77]

$$\mathbf{d}^i = -(\nabla^2 f(\mathbf{s}_c^i))^{-1} \nabla f(\mathbf{s}_c^i), \tag{6.13}$$

obtain new control source signals $\mathbf{s}_c^1$, and repeat iteratively this procedure until a local minimum is reached. A line search is also performed at each iteration to ensure sufficient decrease on the cost function [77].

An efficient computation of the gradient of the cost function is fundamental to ensure fast convergence. The gradient of (6.8) can be written as

$$\nabla f = \mathbf{G}^T(\mathbf{p}^i - \tilde{\mathbf{p}}), \tag{6.14}$$

where $\mathbf{p}^i$ is the room sound pressure at the control points for a given $\mathbf{s}_c^i$. As stated in the previous section, $\mathbf{G}$ consists of a collection of Toeplitz matrices. These actually resemble linear convolution operations, and ∇f can then be written alternatively as

$$\nabla f = \sum_{k=1}^{K_s} \sum_{m=1}^{K_p} \mathbf{F}_{N_s} \left(\hat{\mathbf{h}}_{k,m} * (\mathbf{p}_m^i - \tilde{\mathbf{p}}_m) \right), \tag{6.15}$$

where $*$ stands for linear convolution, $\hat{\mathbf{h}}_{k,m}$ is the time reversed RIRs between the control point position m and control source point position k, $\mathbf{p}_m^i$ is the room sound pressure at the ith Newton step and $\tilde{\mathbf{p}}_m$ is the wanted sound pressure, both at the control point m. $\mathbf{F}_{N_s}$ is a selection matrix that takes only the first N_s samples of the linear convolution. When the size of $\mathbf{G}$ is large, expression (6.15) can be more efficient than computing (6.14) since FFT can be used to compute the linear convolution. Finally the Hessian $\nabla^2 f$ can be computed using

$$\nabla^2 f = \mathbf{G}^T\mathbf{G}. \tag{6.16}$$

While the gradient has to be evaluated at each Newton step due to its dependence of $\mathbf{p}^i$ in expression (6.15), the Hessian has to be computed only once. The matrix inversion performed in (6.13) represents the computational bottleneck of this method.

6.3.3 Regularization

In most application cases, the inverse problem in (6.7) is ill-posed. In particular, if $N_s \cdot K_s > N_t \cdot K_p$, *i.e.*, if the problem is underdetermined, there are many combinations of control source signals that can recreate $\tilde{\mathbf{p}}$ at the control points. One approach is to look for a unique solution by adding a regularization term $\phi(\mathbf{s}_c)$ to the cost function f. The new regularized cost function will be called f_r. A simple choice is to use Tikhonov regularization

$$\phi_T(\mathbf{s}_c) = \frac{\lambda_T}{2}\|\mathbf{s}_c\|_2^2, \tag{6.17}$$

where λ_T is a scalar which weights the importance of this term inside the cost function. Here the regularization term will force the problem to look for a solution which penalizes large values in the control source signals. Unfortunately, Tikhonov regularization will also evenly distribute the energy through all the control source signals. Therefore if the number of active control sources is to be minimized, a different regularization should be used. One could in fact try to minimize the number of active control source positions by actually adding this number to the cost function. Such a regularization would be achieved using the following regularization term:

$$\phi_0(\mathbf{s}_c) = \lambda_0 \sum_{k=1}^{K_s} \left\| \|\mathbf{F}_k \mathbf{s}_c\|_2 \right\|_0, \tag{6.18}$$

where $\mathbf{F}_k$ selects the samples of the kth control source signal whose energy is evaluated by the l_2-norm. The l_0-norm will give 1 if the control source is active and a 0 if it is not.

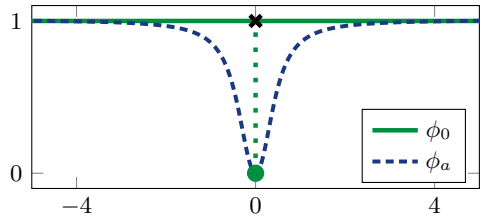


Figure 6.1: l_0-norm of a scalar variable and its relaxation using (6.19) with $\gamma = 10$ and $\lambda_a = 0.12$.

This term is a *group sparsity-inducing regularization* since it forces the solution to be spatially sparse. However, solving an optimization problem that involves the l_0-norm is very difficult: its discontinuity at 0, which can be seen in Fig. 6.1, makes Newton-type optimization not feasible and yields an NP-hard combinatorial problem. In general, a relaxation of the l_0-norm is used to avoid this problem. A typical choice is the l_1-norm, which leads to the LASSO. Here, however, the following relaxation will be used:

$$\phi_a(\mathbf{s}_c) = \sum_{k=1}^{K_s} \frac{2\lambda_a}{\gamma\sqrt{3}} \left(\operatorname{atan}\left(\frac{1 + \gamma \|\mathbf{F}_k \mathbf{s}_c\|_2^2}{\sqrt{3}} \right) - \frac{\pi}{6} \right), \tag{6.19}$$

where λ_a is the weighting scalar and γ is a function parameter. As it can be seen in Fig. 6.1, this relaxation approximates well the behavior of the l_0-norm and has the advantage that it is continuous and differentiable at 0. Notice that

$$\nabla\phi_a(\mathbf{s}_c) = \sum_{k=1}^{K_s} \frac{4\lambda_a}{3} \frac{1}{\left(\frac{1+\gamma\|\mathbf{F}_k\mathbf{s}_c\|_2^2}{\sqrt{3}}\right)^2 + 1} \mathbf{F}_k^T\mathbf{F}_k\mathbf{s}_c \tag{6.22}$$

$$\nabla^2\phi_a(\mathbf{s}_c) \approx \sum_{k=1}^{K_s} \frac{4\lambda_a}{3} \left[\frac{1}{\left(\frac{1+\gamma\|\mathbf{F}_k\mathbf{s}_c\|_2^2}{\sqrt{3}}\right)^2 + 1} \mathbf{F}_k^T\mathbf{F}_k - \frac{4\gamma}{\sqrt{3}} \frac{\left(\frac{1+\gamma\|\mathbf{F}_k\mathbf{s}_c\|_2^2}{\sqrt{3}}\right)}{\left[\left(\frac{1+\gamma\|\mathbf{F}_k\mathbf{s}_c\|_2^2}{\sqrt{3}}\right)^2 + 1\right]^2} \mathbf{F}_k^T\mathbf{F}_k\mathbf{s}_c^2 \right] \tag{6.23}$$

for $\gamma \to 0$ the ϕ_a will tend to Tikhonov regularization. Instead, for high values of γ and a specific λ_a this function will approach the l_0-norm.

It was recently shown that a similar relaxation can enhance sparsity better than the l_1-norm relaxation [181,182]. The usage of a non-convex regularization such as (6.19) can keep the overall problem still convex for a certain set of values of γ [181, 182]. Nevertheless, in this work non-convex optimization will be used, leaving its convex formulation to future work.

Tikhonov regularization will still be used to limit control source signals being too large. The gradient and Hessian of the regularized cost function f_r will become

$$\nabla f_r(\mathbf{s}_c^i) = \nabla f + \lambda_T\mathbf{s}_c^i + \nabla\phi_a(\mathbf{s}_c^i), \tag{6.20}$$

and

$$\nabla^2 f_r(\mathbf{s}_c^i) = \nabla^2 f + \lambda_T\mathbf{I} + \nabla^2\phi_a(\mathbf{s}_c^i), \tag{6.21}$$

where $\mathbf{I}$ is the identity matrix, and $\nabla\phi_a(\mathbf{s}_c^i)$ and $\nabla^2\phi_a(\mathbf{s}_c^i)$ are the gradient and Hessian of (6.19) respectively. These can be computed analytically and are shown in equations (6.22) and (6.23) respectively.

When the optimization problem is solved, the group sparsity term will indicate which control sources are active out of the candidate control source positions. The active control sources are then used to generate the wanted sound field. Notice that compared with the LASSO approach, the proposed algorithm gives control source positions and control source signals simultaneously. In the case of LASSO, active control sources are usually biased towards smaller values [181]. This bias can be removed by applying once more the optimization algorithm using LS or LASSO with the control source positions obtained from the first solution [69,70]. Thanks to the atan regularization, sparsity is induced more effectively and non-active control sources signals are unbiased making it possible to optimize simultaneously for control source positions and control source signals.

6.3.4 Dark Zone Regularization

A dark zone can be enforced in a similar fashion as the regularization performed in the previous section. A dark zone inducing term can be added to the cost function

$$\phi_{dz}(\mathbf{p}) = \frac{\lambda_{dz}}{2} \|\mathbf{F}_{dz}\mathbf{p}\|_2^2, \tag{6.24}$$

which forces the sound pressure being small at K_{dz} specific positions, referred here as *dark zone points*. Here λ_{dz} weights the cost of this term with respect to the other ones appearing in the cost function. The room sound pressure samples at the dark zone points are selected by $\mathbf{F}_{dz}$. If the equality constraint of (6.7) is substituted into (6.24) the gradient and Hessian of this term can be computed as in Sec. 6.3.1 and 6.3.2. The gradient of (6.24) will be

$$\nabla\phi_{dz}(\mathbf{p}^i) = \lambda_{dz}\mathbf{G_{dz}}^T\mathbf{p}^i, \tag{6.25}$$

where $\mathbf{G_{dz}}^T$ is now the collection of Toeplitz matrices of the RIRs between the control source positions and dark zone points. The gradient can be computed more efficiently using linear convolution

$$\nabla\phi_{dz}(\mathbf{p}^i) = \lambda_{dz} \sum_{k=1}^{K_s} \sum_{m=1}^{K_{dz}} \mathbf{F}_{N_s} \left(\hat{\mathbf{h}}_{k,m} * \mathbf{p}_m^i \right). \tag{6.26}$$

The Hessian will be given by

$$\nabla^2\phi_{dz} = \lambda_{dz}\mathbf{G_{dz}}^T\mathbf{G_{dz}}. \tag{6.27}$$

By adding (6.26) and (6.27) to (6.20) and (6.21) respectively, and using these results into (6.13) the iterative procedure can be used to find the spatially sparse control sources which recreate $\tilde{\mathbf{p}}$ at the control points while keeping silent the dark zone points.

6.4 Simulation results

The performance of the algorithm described in Sec. 6.3 is evaluated in a two-dimensional highly reverberant room. A low spatial resolution of $X = 28$ cm is used, resulting in a sampling frequency of $F_s = 1426$ Hz. A sketch of the room is shown in Fig. 6.2: here the $K_p = 4$ control points, the $K_{dz} = 9$ dark zone points and $K_s = 22$ candidate control source positions are shown. The room's reverberation time is $T_{60} = 2.8$ seconds.

Typically, a sound field measured in a different room is imposed at the control points. Here instead, an impulse will be enforced simultaneously at all the

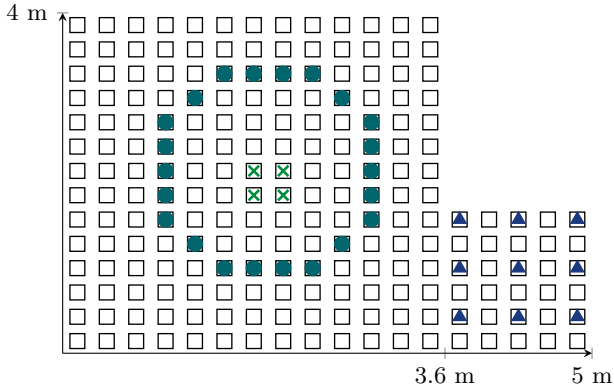


Figure 6.2: Sketch of the L-shaped room used in the simulations. Control points are shown with crosses, control source positions in circles and dark zone points with triangles.

control positions. This rather unnatural sound field is chosen in order to show that, despite the high reverberation of the room, an anechoic sound can be reproduced. Moreover it is possible to test the algorithm performance for different frequency ranges. Fig. 6.3 shows two different *mexican hat* wavelets, which are the impulses that will be used in $\tilde{\mathbf{p}}$. The parameter σ controls the bandwidth of the wavelet as Fig. 6.3 shows. A broad frequency range is achieved using $\sigma_h = 15 \cdot 10^{-4}$ while a low frequency range is obtained using $\sigma_l = 10^{-2}$.

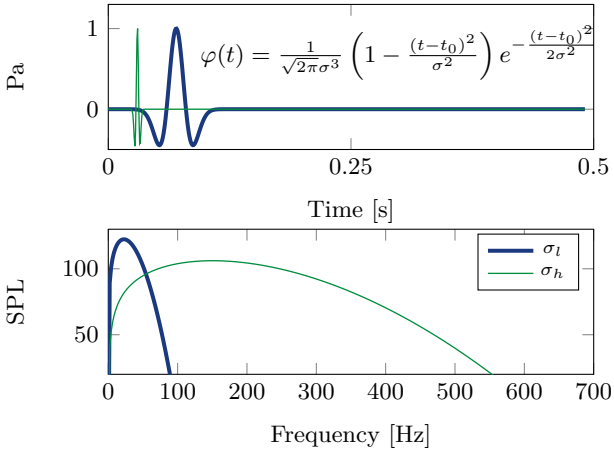


Figure 6.3: Wavelets imposed at the control points.

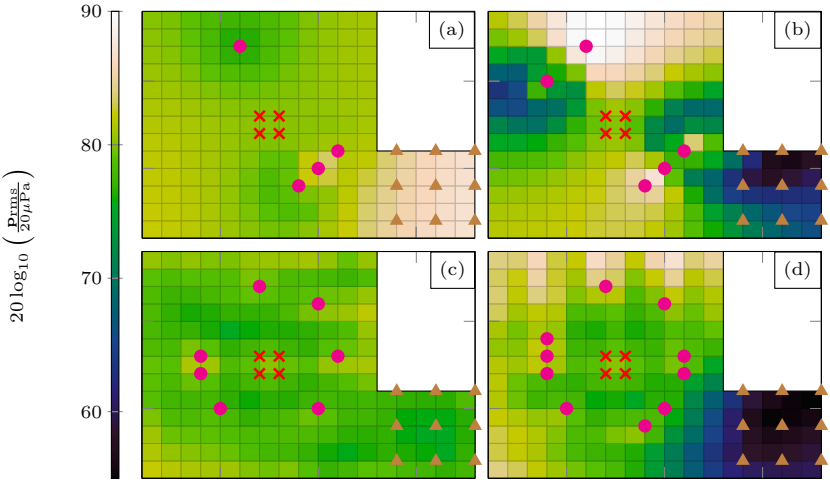


Figure 6.4: Surface plot of the SPL integrated over time at each position of the room. (a) low frequency wavelet ($\lambda_a = 0.18\gamma$, $\lambda_{dz} = 0$, $\epsilon = -40.3$ (dB)), (b) low frequency wavelet with dark zone ($\lambda_a = 0.17\gamma$, $\lambda_{dz} = 0.05$, $\epsilon = -42$ (dB)), (c) high frequency wavelet ($\lambda_a = 0.027\gamma$, $\lambda_{dz} = 0$, $\epsilon = -54.7$ (dB)) and (d) high frequency wavelet with dark zone ($\lambda_a = 0.024\gamma$, $\lambda_{dz} = 0.05$, $\epsilon = -48.1$ (dB)). Normalization was applied in order to have 80 dB SPL at the control points. Video animations can be seen in [183].

Notice that for all the results presented here the parameter γ is set to $5 \cdot 10^3$ and $\lambda_T = 10^{-5}$. The parameter γ is chosen empirically to be large enough for ϕ_a to be a representative relaxation of the l_0-norm while being small enough to ensure a weak non-linearity of ϕ_a. The simulations are performed for 0.5 seconds ($N_t = 700$) while restricting control sources to be 0.18 seconds long ($N_s = 250$). The Newton-type optimization is initialized with $\mathbf{s}_c^0 = \mathbf{0}$. The optimization is stopped when $\|\nabla f_r\|_2 < 10^{-6}$.

Fig. 6.4 shows the SPL integrated over time in the room when the different wavelets are imposed at the control points with and without the dark zone regularization. Fig. 6.4 (a) and (b) shows the results when the low frequency wavelet is imposed. In Fig. 6.4 (a) only 4 control sources are active. At these frequencies the modal behavior of the sound field is prominent. The SPL is not spatially uniform and this is particularly visible at the bottom right corner where the highest SPL is present. When the dark zone regularization is applied in this area, as Fig. 6.4 (b) shows, it can be noticed that the SPL is reduced on average by 23 dB in the dark zone points with respect to the previous case.

This is achieved by adding only one control source. The error at the control points can be seen in the value of ϵ displayed on top of the figures and defined as:

$$\epsilon = 10 \log_{10} \left(\frac{\|\mathbf{F}_p \mathbf{p} - \tilde{\mathbf{p}}\|_2^2}{\|\tilde{\mathbf{p}}\|_2^2} \right). \tag{6.28}$$

Notice that in both cases by relaxing the weight of the sparsity regularization λ_a, the error would be further reduced but more control sources would be activated.

Fig. 6.4 (c) and (d) show the results for the case of the high frequency wavelet. With respect to the low frequency case, in order to obtain convergence, the weight of λ_a had to be relaxed substantially, leading to a less sparse solution with more active control sources: 7 without dark zone regularization and 10 with dark zone regularization. This leads to an increased accuracy at the control points with the error ϵ being smaller compared to the low frequency case. Here, the SPL is reduced on average by 19.5 dB in the dark zone points.

As stated before, the room where the simulations were performed is highly reverberant. These simulations show that an anechoic sound can be reproduced in such an acoustic environment. This is possible because the optimization algorithm actually seeks not only for control sources but also for *control sinks* capable of canceling out the direct sound and reflections created by the other control sources. This is particularly visible in the videos which can be viewed at [183].

6.5 Conclusions and Future Work

A novel method for sound field reproduction and multi-zone sound field control which relies on a physical model of the room acoustics has been proposed. This method has the advantage of being capable of computing all of the RIRs involved in the optimization algorithm by performing multiple FDTD simulations. Using a non-convex sparsity regularization term in the optimization problem, convenient placement of the control sources can be achieved while simultaneously obtaining the control source signals, avoiding a two-step optimization that would be required by the LASSO approach. The simulation results suggest that the number of active control sources increases with the bandwidth of the control point signals and if a dark zone is applied. It was also shown that this algorithm can be used when high reverberation is present in the reproduction room and that anechoic sounds can be reproduced.

Further research will focus on the extension of the proposed method for its application in a real scenario. The computational cost of the FDTD method and the optimization algorithm should be reduced and a realistic model of the

room using the FDTD method should be achieved. Moreover the control sources used here are omnidirectional point sources which are a valid assumption for loudspeakers only at low frequencies. The proposed method will be extended to include directional sound sources.

Part IV

Inverse problems using wave decomposition models

Chapter 7

Room impulse response interpolation

Room impulse response interpolation using a sparse spatio-temporal representation of the sound field

Niccolò Antonello, Enzo De Sena, Marc Moonen, Patrick A. Naylor, and Toon van Waterschoot

Published in *IEEE/ACM Transactions of Audio, Speech Language Processing*, vol. 25, no. 10, pp. 1929-1941, Oct. 2017.

© 2017 IEEE. Reprinted, with permission, from:
N. Antonello, E. De Sena, M. Moonen, P. A. Naylor, and T. van Waterschoot, "Room impulse response interpolation using a sparse spatio-temporal representation of the sound field," *IEEE/ACM Transactions of Audio, Speech Language Processing*, vol. 25, no. 10, pp. 1929-1941, Oct. 2017.
Changes include layout, representation, and minor editing aspects.

The candidate's contributions as first author include: literature study, co-development of the presented method, software implementation and computer simulations, text redaction and editing.

Abstract

Room Impulse Responses (RIRs) are typically measured using a set of microphones and a loudspeaker. When RIRs spanning a large volume are needed, many microphone measurements must be used to spatially sample the sound field. In order to reduce the number of microphone measurements, RIRs can be spatially interpolated. In the present study, RIR interpolation is formulated as an inverse problem. This inverse problem relies on a particular acoustic model capable of representing the measurements. Two different acoustic models are compared: the plane wave decomposition model and a novel time-domain model that consists of a collection of equivalent sources creating spherical waves. These acoustic models can both approximate any reverberant sound field created by a far field sound source. In order to produce an accurate RIR interpolation, sparsity regularization is employed when solving the inverse problem. In particular, by combining different acoustic models with different sparsity promoting regularizations, spatial sparsity, spatio-spectral sparsity and spatio-temporal sparsity are compared. The inverse problem is solved using a matrix-free large scale optimization algorithm. Simulations show that the best RIR interpolation is obtained when combining the novel time-domain acoustic model with the spatio-temporal sparsity regularization, outperforming the results of the plane wave decomposition model even when far fewer microphone measurements are available.

7.1 Introduction

Room impulse responses (RIRs) play a fundamental role in room acoustics. They not only provide useful parameters that describe sound fields qualitatively, such as reverberation time or clarity, but also their knowledge is fundamental in many applications such as channel equalization and sound field reproduction. A RIR between two points in a room is typically measured by generating a deterministic signal, *e.g.*, a sine sweep, using a loudspeaker at one point and recording the sound pressure with a microphone at the other point [6, 7]. Measuring RIRs over a large volume can be a tedious and time-consuming task unless many microphones or a moving microphone [184] are available. Typically, one has to repeat the measurement multiple times for all the source and microphone positions of interest. In order to avoid this, an alternative lies in the spatial interpolation of the measured RIRs to obtain estimates of the RIRs at positions where no microphone measurements are available. In [57] the space-time spectrum of the *plenacoustic function* is studied. In practice, the plenacoustic function represents the collection of all the RIRs associated with

a room as a function of space and time. It is shown that the plenacoustic function space-time spectrum is band limited, implying that the Nyquist sampling theorem can be applied. This spatio-temporal interpolation however still requires many microphone measurements. For example, if a 3-dimensional uniform microphone array is used, the RIRs between the microphones can be interpolated if the microphone measurements are spaced by the distance

$$X < \frac{c}{2F_u}, \tag{7.1}$$

where F_u is the cut-off frequency in Hz of the sound source generating the sound field. Here c represents the speed of sound for which in this paper 343 m/s is used.

In general, it is necessary to seek a milder criterion than that given by the Nyquist sampling theorem. Compressed sensing (CS) represents such an alternative [37]: this framework consists of solving *optimization problems* where the lack of information due to a limited number of available measurements is compensated for by exploiting *prior knowledge*. In particular, in CS this prior knowledge consists of the fact that the sought solution is *sparse*. In this context, CS represents a particular case of an *inverse problem*. As a matter of fact, inverse problems also seek for variables that are not directly measurable [185]. Inverse problems rely on describing the physical phenomenon under study using a model that is partially unknown. These problems are in general *ill-posed*, which implies that meaningful solutions can only be obtained when the inverse problem is *regularized* using a specific prior knowledge. As in CS, inverse problems are posed as optimization problems and the CS framework in fact represents an inverse problem posed with a sparsity promoting regularization.

RIR interpolation has been posed in a wide variety of fashions: different acoustic models have been used, each one having its own specific parameters to estimate, and different algorithms have been used to solve the underlying optimization problems. For example, in [75, 76, 180] RIR interpolation is achieved by solving a classical inverse problem where the acoustic impedances of the walls are the parameters to be estimated. It is assumed that the geometry of the room is known and the wave equation is discretized using wave-based numerical methods such as the finite-difference time-domain (FDTD) method [76, 180] or the boundary element method (BEM) [75]. The acoustic impedance of the walls is reconstructed by solving a non-convex optimization problem. Once the acoustic properties of the walls are determined, the RIRs can be generated at every position of the room using the underlying wave-based numerical method. Nevertheless, these wave-based numerical methods suffer from numerical errors and are only accurate at low frequencies [16]. Moreover, they require a precise knowledge of the room geometry and lead to non-convex optimization problems.

Other approaches that do not require the knowledge of the room geometry have been proposed. For example many approaches rely on a spatial parametrization of the sound field, obtained as well from physical models, *i.e.*, from the spherical harmonic solutions of the wave equation [61, 62, 186, 187]. The main idea is to extrapolate a finite set of parameters out of the measured RIRs such that the parametrization allows the sound field to be predicted at positions where no microphone measurement was made. Such a parametrization is strongly linked to another widely used acoustic model, the plane wave decomposition method (PWDM). In [34] it is shown that, as in the case of spherical harmonics, an acoustic sound field can be correctly approximated using a finite number of *plane waves* independently of the boundaries. Based on this theoretical result, many authors have chosen the PWDM as their acoustic model. For example, in [59] the room modes are identified using common acoustical poles techniques and then reconstructed using the PWDM using a fixed number of plane waves. Still in [59], alternatively a limited number of damped plane waves is chosen out of a large dictionary of damped plane waves using a greedy optimization algorithm. A similar technique is used in [60], where for each frequency an optimization problem is solved in order to find a sparse set of plane waves from a large dictionary that can interpolate the RIRs needed for multi-zone sound field reproduction. A different acoustic model is used in [58] where a wide-band RIR interpolation of the early part of the RIRs is performed. Here the sound field is modeled using the image method (IM) [27]. This inverse problem can be seen as a localization problem: if the location of the image sources can be found, the sound field can be reconstructed. Inverse problems appear also in the acoustic holography field. Here yet another acoustic model of importance is the equivalent source method (ESM), which consists of a collection of equivalent sources generating spherical waves [32, 65, 188].

These techniques and acoustic models are very much related to those used in sound field reproduction, for example wave-field synthesis (WFS) [189], the spatial decomposition method (SDM) [190], and pressure matching (PM) [178, 179] where sparsity promoting regularizations have been used as well [69, 70]. While the aim of these methods is to reproduce a sound field, RIR interpolation techniques seek to estimate the sound field and require different treatments and formulations.

In many of these contexts, inverse problems lead to large scale optimization problems hence requiring specific optimization algorithms. For example, in [58] and [59], the optimization problems are solved using greedy algorithms, specifically with a modified version of matching pursuit (MP). An alternative way of solving large scale problems relies on first-order optimization algorithms, where matrix inversions are avoided. The most well-known first-order optimization algorithm is the gradient descent algorithm, which exhibits a

low convergence rate and is not well suited for non-smooth cost functions that typically appear with sparsity promoting regularizations. Recent interest in large scale optimization problems has led to the development of accelerated first-order optimization algorithms which enjoy faster convergence and can deal with non-smooth cost functions. A successful family of first-order algorithms is referred to as forward-backward splitting (FBS) also known as the proximal gradient method [81]. These algorithms have recently been substantially accelerated using quasi-Newton methods [82, 83]. An important aspect of large scale problems is the memory requirement which can easily become prohibitive. Matrix-free optimization avoids the storage of large matrices through the usage of the adjoint operators [104, 105].

In this paper a novel method for RIR interpolation is proposed. A time-domain acoustic model called time-domain equivalent source method (TESM) is described, which is a modified time-domain version of the ESM. This acoustic model is used to approximate the sound field in a source-free volume. The main differences with respect to ESM are that the source signal can be included in the model to improve RIR interpolation and that the formulation is in the time-domain, which allows *spatio-temporal sparsity* regularization to be applied. Due to the large number of equivalent sources the optimization problem becomes large-scale and the computational aspect becomes relevant. In particular matrix-free optimization is used in order to reduce the memory requirement and the algorithm proposed in [83] is used to speed up the convergence. The spatio-temporal sparsity regularization is compared with other regularizations: it is shown that when using the PWDM in the inverse problem, it is possible to impose either spatial sparsity or spatio-spectral sparsity, with the former being a better choice. Still, numerical simulations show that the choice of promoting spatio-temporal sparsity with TESM always outperforms the RIR interpolation obtained with PWDM and leads to a significant decrease in the number of microphone measurements required. In this study RIR interpolation is limited to the far-field case.

The paper is organized as follows: in Sections 7.2 and 7.3 the PWDM and the novel TESM are presented respectively. In Section 7.4, these acoustic models are used in the inverse problems that rely on different regularizations involving a sparse representation: spatial sparsity, spatio-spectral sparsity and spatio-temporal sparsity. In Section 7.5 the optimization algorithm used to solve these inverse problems is briefly described. In Section 7.6 the efficient computation of the derivatives needed to solve the optimization problems is given. Here an interpretation of the adjoint operators of the acoustic models needed in this computation is also given. In Section 7.7 the simulation results are presented: firstly in 7.7.1 the tuning of the level of the regularization is described, secondly in 7.7.2 the RIR interpolation performances are compared

for different acoustic models and regularizations. In Section 7.7.3 the solutions of the inverse problems are visualized. In Section 7.8 experimental results are presented. Finally, conclusion in Section 7.9 are given.

7.2 Plane wave decomposition method

In this section, the plane wave decomposition method (PWDM) is briefly reviewed. Plane waves are defined as

$$\hat{\phi}_{f,l}(\mathbf{x}) = e^{i\mathbf{k}_{f,l}^T\mathbf{x}} \tag{7.2}$$

and they are solutions of the homogeneous Helmholtz equation, that is

$$\triangle\hat{p}_f(\mathbf{x}) + k_f^2\hat{p}_f(\mathbf{x}) = 0 \text{ on } \mathbb{R}^3, \tag{7.3}$$

which assumes a time-harmonic behavior of the acoustic field, *i.e.*, $p(\mathbf{x},t) = \mathfrak{Re}(\hat{p}_f(\mathbf{x})e^{i\omega_f t})$ for $\omega_f = 2\pi f F_s/N_f$. Here f is the frequency index, $\hat{p}_f(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{C}$ is the complex sound pressure at a particular frequency f and $\triangle$ is the Laplace operator over the spatial variables $\mathbf{x}$. The wave number k_f is defined as $k_f = \omega_f/c$ and $\mathbf{k}_{f,l} = k_f\mathbf{n}_l \in \mathbb{R}^3$ is the wave vector, with $\mathbf{n}_l$ a unit vector that defines the direction of the lth plane wave. Equation (7.3) can be written for $f = 0,\ldots,N_f - 1$ discrete frequencies. In the following $\hat{\phi}_{f,l,m}$ will indicate $\hat{\phi}_{f,l}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_m}$ where $\mathbf{x}_m \in \mathbb{R}^3$ is a point in space.

A sound field in a source-free spatial domain $\Omega \subset \mathbb{R}^3$ can be well represented by a finite weighted sum of plane waves coming from N_w different directions [187]:

$$\hat{p}_f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_m} \approx \sum_{l=0}^{N_w-1} \hat{\phi}_{f,l,m}\hat{w}_{f,l} \text{ for } \mathbf{x}_m \in \Omega, \tag{7.4}$$

where the *weight* $\hat{w}_{f,l}$ is a complex scalar that weights the (f,l)-th plane wave. Under the assumptions that Ω is source-free, that the source generating the sound field and that the reflecting surfaces are in the far field, this acoustic model, known as the PWDM, gives a good approximation of any sound field [34, 187]. If these conditions are met, this approximation holds independently of the boundary conditions, the room geometry and the sound source type that generates the field outside Ω. Suppose that the weights $\hat{w}_{f,l}$ are available, if one wants to predict the sound pressure at N_m discrete positions $\mathbf{x}_m \in \Omega$ for $m = 0,\ldots,N_m - 1$, (7.4) can be generalized for discrete spatial points by

$$\hat{\mathbf{P}} = D_{pw}(\hat{\mathbf{W}}), \tag{7.5}$$

where $\hat{\mathbf{P}}$ is a matrix containing the complex sound pressure for different positions and frequencies:

$$\hat{\mathbf{P}} = \begin{pmatrix} \hat{p}_{0,0} & \cdots & \hat{p}_{0,N_m-1} \\ \vdots & \ddots & \vdots \\ \hat{p}_{N_f-1,0} & \cdots & \hat{p}_{N_f-1,N_m-1} \end{pmatrix} \in \mathbb{C}^{N_f \times N_m}, \tag{7.6}$$

where the notation $\hat{p}_{f,m} = \hat{p}_f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_m}$ was used. In particular, each column of $\hat{\mathbf{P}}$ can be thought of as a N_f-point discrete Fourier transform (DFT) of a discrete-time signal. Similarly, the matrix $\hat{\mathbf{W}} \in \mathbb{C}^{N_f \times N_w}$ contains the weights $\hat{w}_{f,l}$, where the l-th column has the weights of the l-th plane wave for the different frequencies. Notice that the weights and the sound pressures can be transformed into temporal signals using an inverse DFT. The linear operator $D_{pw} : \mathbb{C}^{N_f \times N_w} \to \mathbb{C}^{N_f \times N_m}$ maps these weights $\hat{w}_{f,l}$ to the complex sound pressures $\hat{p}_{f,m}$ using (7.4) and actually represents a dictionary of N_w plane waves with N_f frequencies. This linear operator is separable for each row of $\hat{\mathbf{P}}$ and $\hat{\mathbf{W}}$ since every frequency is independent. Moreover, each column of these matrices is Hermitian symmetric, and this redundancy can be exploited during evaluation to reduce the computational cost. When the weights $\hat{w}_{f,l}$ are given, (7.5) represents the *forward problem*. Nevertheless, typically the weights are unknown and an *inverse problem* must be solved to estimate them.

7.3 Time-domain equivalent source method

Following the same logic of the previous section, a similar time-domain approximation of the sound field is proposed. The time-domain Green's function is defined as

$$\phi_l(\mathbf{x}, t) = \frac{1}{4\pi d_l} \delta\left(t - \frac{d_l}{c}\right), \tag{7.7}$$

where δ is the Dirac delta function and $d_l = \|\mathbf{x}_l - \mathbf{x}\|_2$ is the distance between $\mathbf{x}$ and $\mathbf{x}_l$. In the following $\phi_{l,m}(t)$ will indicate $\phi_l(\mathbf{x}, t)|_{\mathbf{x}=\mathbf{x}_m}$. The time-domain Green's function is a particular solution of the non-homogeneous wave equation [191]

$$\triangle p(\mathbf{x}, t) - \frac{1}{c^2} \frac{\partial^2 p(\mathbf{x}, t)}{\partial t^2} = \delta(\mathbf{x} - \mathbf{x}_l, t) \text{ on } \mathbb{R}^{3\times 1}, \tag{7.8}$$

with null initial conditions and unbounded domain. The wave equation is the time domain counterpart of Helmholtz equation (7.3). The sound pressure $p(\mathbf{x}, t)|_{\mathbf{x}=\mathbf{x}_m}$ evaluated at a given point in space $\mathbf{x}_m$ is equivalent to a RIR between $\mathbf{x}_l$ and $\mathbf{x}_m$, and $\delta(\mathbf{x} - \mathbf{x}_l, t)$ describes a point source positioned at $\mathbf{x}_l$

that emits a pulse at time $t = 0$. Equation (7.7) represents what is referred here as an *equivalent source*, *i.e.*, a sound source that emits a spherical wave. In an unbounded or anechoic domain, an equivalent source positioned at $\mathbf{x}_l$ generates a spherical wave which arrives at a position $\mathbf{x}_m$ after traveling the distance $d_{l,m} = \|\mathbf{x}_l - \mathbf{x}_m\|_2$. Similarly to Section 7.2, where the index l indicates the l-th plane wave $\hat{\phi}_{f,l,m}$ with direction $\mathbf{n}_l$, here this index refers to the l-th spherical wave $\phi_{l,m}$ generated at $\mathbf{x}_l$ also arriving to the microphone at a position $\mathbf{x}_m$ from an l-th specific direction. In addition, here a microphone positioned at $\mathbf{x}_m$ would capture a pulse delayed by $d_{l,m}/c$ seconds and attenuated by a factor of $1/(4\pi d_{l,m})$ due to the spherical spreading of energy. The equivalent source can be discretized over time at a sampling frequency F_s using a *fractional delay* filter with impulse response (IR) $h_{l,m}$:

$$\phi_{l,m}(n) = \frac{1}{4\pi d_{l,m}} h_{l,m}(n), \tag{7.9}$$

where now n is the discrete time index. Here, the fractional delay filters are evaluated using Thiran all pass filters, which consist of infinite impulse response (IIR) filters [192]. Spherical waves can be thought of as a generalization of plane waves [187], *i.e.*, when $d_{l,m}$ is large, spherical waves can effectively represent plane waves. Therefore also a finite sum of equivalent sources in the far field can approximate well any sound field in a source-free volume Ω, under the same assumptions as in the previous section. As a consequence, the acoustic model presented in this section can be seen as a generalization of the PWDM. The following expression shows this time-domain acoustic model which is referred to as the time-domain equivalent source method (TESM):

$$p(\mathbf{x}, n)|_{\mathbf{x}=\mathbf{x}_m} \approx \sum_{l=0}^{N_w-1} \delta(n) * \phi_{l,m}(n) * w_l(n), \tag{7.10}$$

for $\mathbf{x}_m \in \Omega$. Here $w_l(n)$ is signal of dimension N_t that *weights* the equivalent sources through linear convolution (represented here with the symbol $*$). This signal will be referred as the *weight signal*. If the anechoic source signal that generates the sound field is known, this signal, $s(n)$, can be included in the TESM to increase the quality of the RIR interpolation:

$$p(\mathbf{x}, n)|_{\mathbf{x}=\mathbf{x}_m} \approx \sum_{l=0}^{N_w-1} s(n) * \phi_{l,m}(n) * w_l(n), \tag{7.11}$$

for $\mathbf{x}_m \in \Omega$. The signal $s(n)$ could correspond to the IR of a loudspeaker measured in an anechoic chamber. When this approximation is used it will be referred to as source-aware time-domain equivalent source method (sTESM). The abbreviation (s)TESM will be used to indicate TESM and sTESM simultaneously.

Similar to (7.5) in Section 7.2, (7.10) and (7.11) can be generalized for N_m discrete positions $\mathbf{x}_m \in \Omega$:

$$\mathbf{P} = D_{(s)t}(\mathbf{W}) \tag{7.12}$$

where $\mathbf{P} \in \mathbb{R}^{N_t \times N_m}$ is a matrix in which the m-th column is the sound pressure signal $p(\mathbf{x}, n)|_{\mathbf{x}=\mathbf{x}_m}$ and $\mathbf{W} \in \mathbb{R}^{N_t \times N_w}$ is a matrix in which the l-th column is the weight signal $w_l(n)$. Therefore the linear operator $D_{(s)t} : \mathbb{R}^{N_t \times N_w} \to \mathbb{R}^{N_t \times N_m}$ maps the weight signals to the sound pressures and represents a dictionary of equivalent sources. This can be computed simply by using (7.9) with (7.10) or (7.11) for $l = 0, \ldots, N_w - 1$ and for $m = 0, \ldots, N_m - 1$.

Another common approach to evaluate the linear operator $D_{(s)t}$ and also the D_{pw} presented in Section 7.2 is to create a linear system of equations by vectorizing $\mathbf{P}$ and $\mathbf{W}$:

$$\mathbf{p} = \mathbf{D}\mathbf{w}, \tag{7.13}$$

where $\mathbf{p} = \text{vec}(\mathbf{P})$ and $\mathbf{w} = \text{vec}(\mathbf{W})$ with vec() indicating the column-major vectorization operator. This has the advantage that if the linear operator has to be evaluated several times, as when solving optimization problems, the fractional delays appearing in (7.9) or the plane waves appearing in (7.2) can be computed only once and stored in the matrix $\mathbf{D}$. Nevertheless, looking at the dimensions of $\mathbf{D}$, $N_t N_m \times N_t N_w$, it is clear that storing such a matrix can easily become intractable. As an example, for TESM, consider a time window of 0.1 seconds sampled with $F_s = 8$ kHz ($N_t = 800$). If $N_w = 400$ and $N_m = 12$, $\mathbf{D}$ is a $96 \cdot 10^2 \times 32 \cdot 10^4$ matrix which for a 64 floating point format would result in a memory requirement of approximately 24.6 GB. In such cases, it will be necessary to sacrifice computational power for memory by directly evaluating the linear operators when needed. This strategy will lead to *matrix-free* optimization.

7.4 The inverse problem

Figure 7.1 shows the measurement set-up needed for the RIR interpolation. A far field sound source, represented in the figure in the front left corner of the room, generates a reverberant sound field. A spherical microphone array of N_m microphones (light green dots) is placed in the middle of the room and used to measure the RIRs at these positions. Notice that the framework described this far does not assume any particular geometry for the position of the microphone measurements: it is only assumed that the relative distances between the microphone measurements are known. The choice of a spherical microphone array is arbitrary. The aim is to interpolate these RIRs inside the volume Ω. In this volume, it is assumed that the models presented in

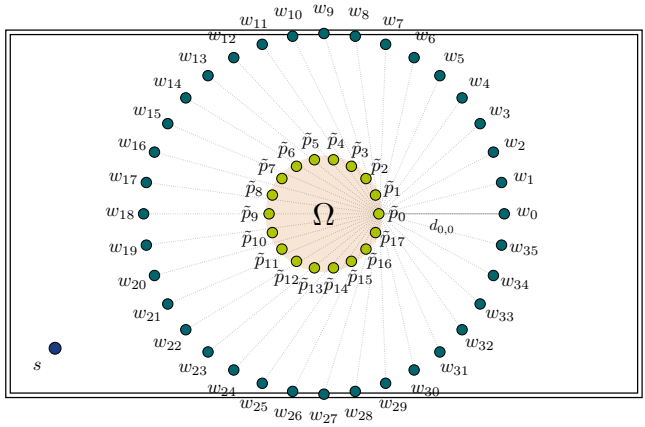


Figure 7.1: A cross-section of a room viewed from above. A sound source placed near the front left corner creates a reverberant sound field that is captured by a spherical microphone array (light green dots). The sound pressure is then matched using a set of equivalent sources $\phi_{l,m}$ (dark green dots). This makes it possible to perform RIR interpolation over the shaded volume Ω surrounded by the microphones. The distance $d_{0,m}$ between the first microphone and the equivalent sources is shown with dotted lines.

Sections 7.2 and 7.3 can well approximate the sound field. What is sought by the inverse problem is to extrapolate out of the microphone measurements the optimal weight signals that lead to the best sound field approximation possible. This inverse problem can be formulated as an optimization problem:

$$\mathbf{W}^\star = \underset{\mathbf{W}}{\operatorname{argmin}} \quad f(\mathbf{W}) = \frac{1}{2}\|D(\mathbf{W}) - \tilde{\mathbf{P}}\|_F^2, \tag{7.14}$$

where $\|\cdot\|_F$ is the Frobenius norm defined as

$$\|\mathbf{A}\|_F = \|\mathrm{vec}(\mathbf{A})\|_2. \tag{7.15}$$

Notice that here the linear operator $D(\cdot)$ has no subscript, which implicitly means that any of the acoustic models presented in the previous sections can be used. The columns of the matrix $\tilde{\mathbf{P}}$ contain the microphone measurements, *i.e.*, the N_t-long measured RIR signals. The aim of this optimization problem is to minimize the *un-regularized cost function* $f(\mathbf{W})$, which consists of the distance between the measured RIRs and the sound pressure of the acoustic model. However, problem (7.14) is heavily *ill-posed*: if many spherical (plane) waves are used to construct $D(\cdot)$, multiple solutions for $\mathbf{W}$ can minimize the cost function effectively. This will in general lead to *over-fitting*: the measured

RIRs will coincide with the sound pressure of the acoustic model but only at the microphone positions, leading to a poor RIR interpolation. To avoid this, it is necessary to regularize problem (7.14). A common regularization method is *Tikhonov regularization* which consists of modifying the cost function by adding a regularization term:

$$\mathbf{W}_T^\star = \underset{\mathbf{W}}{\operatorname{argmin}} \quad f(\mathbf{W}) + \frac{\lambda}{2}\|\mathbf{W}\|_F^2. \tag{7.16}$$

Here the parameter λ controls the level of the regularization and in practice balances the *prior knowledge* induced by the regularization with the information provided by the microphone measurements that appears in $f(\cdot)$. For the case of Tikhonov regularization, this prior knowledge represents the expectation that the components of $\mathbf{W}$ are small. As simulations will show later, this regularization does not produce good results as it does not include any spatial information.

If it is expected that the matrix $\mathbf{W}$ is *sparse* instead, *i.e.*, that the majority of the components of $\mathbf{W}$ are zero, a sparsity promoting regularization can be used, *e.g.*, the l_1-*norm regularization*:

$$\mathbf{W}_1^\star = \underset{\mathbf{W}}{\operatorname{argmin}} \quad f(\mathbf{W}) + \lambda\|\mathrm{vec}(\mathbf{W})\|_1. \tag{7.17}$$

Problem (7.17) is also known as the least absolute shrinkage and selection operator (LASSO) and has been widely used in CS. As opposed to Tikhonov regularization, the l_1-norm regularization, when it is used with the (s)TESM, promotes *spatio-temporal sparsity*. This becomes clear by looking at the structure of $\mathbf{W}$, which in the case of (s)TESM contains the time-domain weight signals that control equivalent sources placed at different positions. Spatio-temporal sparsity can be justified physically: in a reverberant sound field generated by an impulsive source, sound waves arrive from specific directions at specific times. On the other hand, when the l_1-norm regularization is used with the PWDM, *spatio-spectral sparsity* is promoted. Once more this becomes clear by looking at the structure of $\hat{\mathbf{W}}$, although a physical interpretation becomes more difficult. As simulations will show, the l_1-norm regularization does indeed produce good results for the (s)TESM and moderate results for the PWDM.

Another possible regularization is the *sum of* l_2*-norms regularization*:

$$\mathbf{W}_{\Sigma l_2}^\star = \underset{\mathbf{W}}{\operatorname{argmin}} \quad f(\mathbf{W}) + \lambda \sum_{l=0}^{N_w-1} \|\mathbf{W}_{:,l}\|_2. \tag{7.18}$$

where $\mathbf{W}_{:,r}$ indicates the rth column of $\mathbf{W}$. This regularization is aimed to have only few columns of $\mathbf{W}$ to have non-zero coefficients. In the PWDM

	l_1-norm	$\sum$ l_2-norms	Tikhonov
$g(\mathbf{W})$	$\lambda\|\text{vec}(\mathbf{W})\|_1$	$\lambda\sum_{l=0}^{N_w-1}\|\mathbf{W}_{:,l}\|_2$	$\frac{\lambda}{2}\|\mathbf{W}\|_F^2$
$\text{prox}_{\gamma g}(\mathbf{W})$	$\text{sgn}(\mathbf{W})\max(\mathbf{0},\|\mathbf{W}\|-\gamma\lambda)$	$\max(0,1-\gamma\lambda/\|\mathbf{W}_{:,l}\|_2)\mathbf{W}_{:,l}$ for $l=0\dots N_w-1$	$\frac{1}{\gamma\lambda+1}\mathbf{W}$
$\lambda_{\max}$	$\|\text{vec}(D^a(\mathbf{P}))\|_\infty$	$\left\|\left[\|D^a(\mathbf{P})_{:,0}\|_2,\dots,\|D^a(\mathbf{P})_{:,N_w-1}\|_2\right]^T\right\|_\infty$	$\|D^a(\mathbf{P})\|_2^2$

Table 7.1: Table showing the different regularizations used in this paper and their equivalent proximal mappings. The last row shows the maximum value of λ used for the different regularizations.

and (s)TESM case this regularization imposes solely *spatial sparsity*, and it is a special case of *group sparsity*. When the l_1-norm regularization is used with the PWDM, the optimization problem becomes fully separable. Since all frequencies are independent in the linear operator of the PWDM and the same is true for the l_1-norm regularization it is possible to cast individual optimization problems per frequency. This property is no longer valid for sum of l_2-norms regularization, which implies that the optimization problem is not separable anymore. Despite this disadvantage, simulations will show that the sum of l_2-norms regularization outperforms the l_1-norm one when applied to PWDM. Clearly, when a wide band sound source generates the sound field, imposing sparsity in the frequency domain is not a good choice and therefore spatial sparsity provides better results.

Finally, an important aspect is the choice of the equivalent source positions in (s)TESM. In a reverberant environment, sound waves can arrive from every direction but the dictionaries must be of finite dimension. As Fig. 7.1 shows, for the 2-dimensional case the equivalent sources can be placed uniformly on a circle (dark green dots). Nevertheless, in 3-dimensions the uniform sampling of the surface of a sphere is not unique and for instance sampling uniformly the azimuthal and the polar angles leads to a concentration of equivalent sources near the poles. In order to avoid this, Fibonacci lattices, which provide nearly uniform sampling of the surface of a sphere, are used [193, 194]. The azimuthal and polar angles obtained with this lattice can be used also for the candidate directions of the plane waves in the PWDM.

7.5 Optimization algorithm

In this section firstly the FBS algorithm is described. Secondly, the quasi-Newton accelerated FBS is briefly presented. Further details about these methods can be found in [81–83].

The FBS generalizes the well-known gradient descent algorithm to a class of non-smooth cost functions. The FBS can be used to solve optimization problems with the following structure:

$$\mathbf{W}^\star = \underset{\mathbf{W}}{\operatorname{argmin}} f(\mathbf{W}) + g(\mathbf{W}), \tag{7.19}$$

where $f(\cdot)$ is convex and smooth, such as the un-regularized cost function (7.14), and $g(\cdot)$ can be non-smooth, such as the l_1-norm in (7.17). Clearly, all of the optimization problems presented in Section 7.4 have cost functions which can be split in such a fashion.

Starting from an initial guess $\mathbf{W}^0$, one can iterate the expression

$$\mathbf{W}^{k+1} = T_\gamma(\mathbf{W}^k) = \operatorname{prox}_{\gamma g}\left(\mathbf{W}^k - \gamma \nabla f(\mathbf{W}^k)\right), \tag{7.20}$$

to obtain a solution. Here $\nabla f(\cdot)$ is the Jacobian operator of $f(\cdot)$, $T_\gamma(\cdot)$ is the *forward-backward operator* and $\operatorname{prox}_{\gamma g}(\cdot)$ is the *proximal mapping* of the function $g(\cdot)$, defined as:

$$\operatorname{prox}_{\gamma g}(\mathbf{W}) = \underset{\mathbf{U}}{\operatorname{argmin}} \frac{1}{2\gamma}\|\mathbf{U} - \mathbf{W}\|_F^2 + g(\mathbf{U}). \tag{7.21}$$

The argument inside the parenthesis of $\operatorname{prox}_{\gamma g}(\cdot)$ in (7.20) corresponds to the gradient descent iteration for the un-regularized cost function $f(\cdot)$. Therefore a simple interpretation of (7.20) is the following: a new iterate is obtained using the gradient descent for $f(\cdot)$ (forward step)

$$\mathbf{W}_{gd}^{k+1} = \mathbf{W}^k - \gamma \nabla f(\mathbf{W}^k), \tag{7.22}$$

and then a second optimization problem, cf. (7.21), is solved to project the iterate $\mathbf{W}_{gd}^{k+1}$ into a modified solution *close* to the minimum of $g(\cdot)$ (backward step). The scalar γ is the *step-size* which must be sufficiently small to ensure convergence. One of the main advantages of the FBS algorithm is that often the proximal mapping can be computed very cheaply. For example, when $g(\cdot) = \lambda\|\cdot\|_1$ the proximal mapping (7.21) has an analytical solution and reduces to a soft-thresholding of the elements of $\mathbf{W}_{gd}^{k+1}$ [81]. Table 7.1 summarizes the proximal mappings for the different regularizations used in this paper which all have analytical solutions and are cheap to compute. As stated earlier, a solution to (7.19) can be obtained by iterating (7.20) leading to a simple algorithm with a convergence speed comparable to the gradient descent algorithm.

In [83] a novel algorithm that dramatically accelerates the FBS algorithm has been proposed. The idea is to modify the FBS with the following iterations:

$$\mathbf{W}^{k+1} = T_\gamma(\mathbf{W}^k) + \tau \mathbf{S}^k, \tag{7.23}$$

where $\mathbf{S}^k$ is a corrective direction and τ is a step-size. It can be proven that the optimality condition for $\mathbf{W}^\star$ to be the optimal solution of (7.19) is given as [83]

$$R_\gamma(\mathbf{W}^\star) = \mathbf{W}^\star - T_\gamma(\mathbf{W}^\star) = \mathbf{0}, \tag{7.24}$$

where $R_\gamma(\cdot)$ represents the *fixed-point residual*. At each iteration, the corrective direction $\mathbf{S}^k$ is computed using curvature information of the fixed-point residual obtained through quasi-Newton limited memory BFGS (L-BFGS) updates to accelerate convergence. The step-sizes γ and τ are chosen adaptively with two separate line-search procedures to ensure global convergence. These line-search procedures only rely on the very same predictions of the FBS which, together with the L-BFGS updates, have minimal memory requirement. Further details on the algorithm are left out here for brevity and the interested reader can find more information in [83]. An implementation of the algorithm is also available online [137, 195].

7.6 Computation of the Jacobian

The Jacobian $\nabla f(\cdot)$ appears in the FBS iterations (7.20), and consequently also in the accelerated version (7.23). Moreover the un-regularized cost function $f(\cdot)$ must also be computed as it is needed in the line-search procedures. The efficient computation of $f(\cdot)$ and $\nabla f(\cdot)$ are therefore crucial since these are required in every iteration of the optimization algorithm.

Regarding the computation of $f(\cdot)$, by inspecting its definition

$$f(\mathbf{W}) = \frac{1}{2} \| \underbrace{D(\mathbf{W}) - \tilde{\mathbf{P}}}_{\mathbf{R}} \|_F^2, \tag{7.25}$$

it can be noticed that the most computationally expensive operation lies in the evaluation of $D(\mathbf{W})$. Here the residual $\mathbf{R}$ is the difference between the measured sound pressure and the sound pressure provided by the acoustic model of choice. It was already mentioned at the end of Section 7.3 how avoiding the storage of $D(\cdot)$ into a matrix $\mathbf{D}$ and using a recursive computation instead can be beneficial for a large scale optimization problem, as it minimizes the memory requirements.

The same strategy will be used in the computation of the Jacobian $\nabla f(\cdot)$. The Jacobian can be written as:

$$\nabla f(\mathbf{W}) = D^a(D(\mathbf{W}) - \tilde{\mathbf{P}}) \Rightarrow \mathbf{J} = D^a(\mathbf{R}), \tag{7.26}$$

where $D^a(\cdot) : \mathbb{U}^{N_m N_t \times N_w N_t} \to \mathbb{U}^{N_w N_t \times N_m N_t}$ is the *adjoint operator* [105] of $D(\cdot)$, $\mathbf{J} : \mathbb{U}^{N_w N_t \times N_m N_t}$ is the Jacobian matrix and $\mathbb{U}$ is either $\mathbb{R}$ or $\mathbb{C}$. If the

linear operator $D(\cdot)$ consisted of a matrix multiplication the adjoint operator would have been the conjugate-transpose operator of that matrix. It can be noticed that the adjoint operator is applied to the residual $\mathbf{R}$ which is readily available after the computation of $f(\cdot)$.

For the (s)TESM the linear operator $D_{(s)t}(\cdot)$ is computed by recursively convolving the weight signals $w_l(n)$ with $(s(n)) \ * \ \phi_{l,m}(n)$ as shown in (7.10) and (7.11). Since the adjoint operator of convolution is the cross correlation [126], the lth column of the Jacobian matrix $\mathbf{J}$ can be computed as:

$$j_l(n) = \sum_{m=0}^{N_m-1} (s(n) * \phi_{l,m}(n)) \odot r_m(n) \tag{7.27}$$

where $\odot$ indicates the cross correlation operator and $r_m(n)$ is the residual signal of the m-th microphone measurement, *i.e.*, the m-th column of $\mathbf{R}$. This operation can be repeated iteratively to compute $\mathbf{J}$ as in the case of the evaluation of $D_{(s)t}(\cdot)$.

Analyzing (7.27) from a different perspective, the adjoint operator can be thought of as a new swapped forward problem with N_m equivalent sources positioned at $\mathbf{x}_m$ having as weight signals the time reversed residual $r_m(-n)$. This sound field is then captured by N_w microphone measurements positioned at $\mathbf{x}_l$. Equation (7.27) is in fact equivalent to:

$$j_l(-n) = \sum_{m=0}^{N_m-1} (s(n) * \phi_{l,m}(n)) * r_m(-n) \tag{7.28}$$

which in practice is used in the computation since it requires the very same fractional delay filters as the ones used in $D_{(s)t}(\cdot)$.

Similarly, for the PWDM the adjoint operator can be obtained using the following equation:

$$\hat{j}_{f,m} = \sum_{m=0}^{N_m-1} \hat{r}_{f,l} \hat{\phi}^*_{f,l,m}, \tag{7.29}$$

where $\hat{j}_{f,m}$ is the (f,m)-th element of the complex Jacobian matrix $\hat{\mathbf{J}}$, $\hat{r}_{f,l}$ is the (f,l)-th element of the residual $\hat{\mathbf{R}}$ and with * denoting the complex conjugate operation.

7.7 Simulation results

In this section the RIR interpolation performance is analyzed using the different acoustic models described in Sections 7.2 and 7.3 and the different

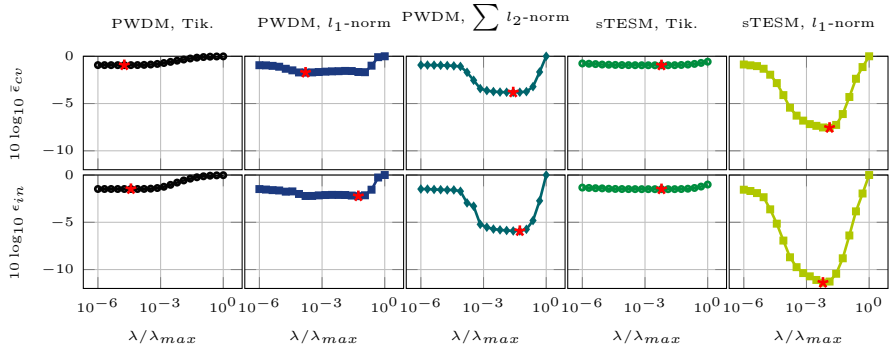


Figure 7.2: Simulated averaged cross validation error curve (top plots) and interpolation error curve (bottom tops) as a function of λ (normalized by $\lambda_{\max}$) for different types of acoustic models and regularizations using $N_m = 12$ microphones.

regularizations presented in Section 7.4. The microphone signals used in the inverse problem are generated using the randomized image method (RIM), a modified version of the IM that avoids the presence of sweeping-echos in the simulated RIRs [196]. As Fig.7.1 shows, the reverberant acoustic environment consists of a common-size box-shaped room with dimensions $[L_x, L_y, L_z] = [6, 3.5, 4]$ m. A sampling frequency of $F_s = 8$ kHz is used in order to have an optimization problem with feasible size and a time window of 70 ms ($N_t = N_f = 560$), long enough to capture all relevant late reflections.

Frequency dependent impedances are used for the walls: this is achieved by using IIR filters for the image sources which are then self-convolved for the higher order image sources. These filters are obtained from measured absorption coefficients found in [197] using the same procedure described in [31], *i.e.*, optimizing the coefficients of the IIR filters via a damped Gauss-Newton method. The fractional delays of the RIM are modeled using finite impulse response (FIR) filters as in [151]. Software is available at [198]. Only one material is used to model all of the room acoustic impedances. The resulting sound field has a spatially averaged reverberation time of $T_{30} = 0.065$ s.

An omnidirectional sound source is placed at $\mathbf{x}_s = [0.75,\ 0.43,\ 2]$ m, near an edge in order to fully excite many room modes, producing a source signal $s(n)$, an impulse filtered using a bandpass 4th order Butterworth filter from 20 Hz to 3.2 kHz that roughly models the IR of a loudspeaker. Microphones are placed on a spherical array of radius 0.35 m, in order to perform the RIR interpolation in a relatively wide volume, with center in the middle of the room at $\mathbf{x}_c =$

$[L_x, L_y, L_z]/2$. The microphone positions form a Fibonacci lattice: this is a convenient arbitrary choice since the different variants of the RIR interpolation method are tested using different number of microphones. All the microphone signals are corrupted with additive white noise with a SNR = 15 dB. When comparing performances with different regularizations and acoustic models the same noise is added. Such a high SNR is motivated by the fact that it is assumed that most of the noise is reduced in post-processing by averaging the RIR microphone measurements. The RIRs are interpolated in the volume (0.18 m^3) enclosed by the spherical microphone array. Nevertheless, since it is not possible to compute the true RIR at every position inside the volume, their quality is evaluated on an *interpolation volume*, that is a cuboid volume of dimensions $[0.43, 0.43, 0.63]$ m which is spatially uniformly sampled ($N_{in} = 300$). and placed inside the spherical microphone array.

For the (s)TESM $N_w = 700$ equivalent sources are positioned in a Fibonacci lattice of radius 1.75 m centered at $\mathbf{x}_c$. The radius of the equivalent sources was chosen in order for the sound waves to have a limited spherical behavior in Ω, in practice resulting very similar to plane waves. Since it is assumed that the sound source is in the far-field, positioning the equivalent sources too close to the microphone array would reduce the spatial sparsity assumption of the weight signals: the resulting optimal weight signals would implicitly apply Huygens-Fresnel principle and many equivalent sources would be simultaneously active to represent a far field sound wave. Similarly, for the PWDM, $N_w = 700$ directions are used and these are equivalent to those used in the (s)TESM. The number of plane wave directions and equivalent sources was obtained empirically: a low N_w produced a reduction of the performances while a higher one did not particularly change the results but increased the computational load of the algorithms. Simulations were performed also for different configurations of source position, room dimensions and reverberation time, which led to similar results as the ones presented in the following subsections and are therefore omitted here for brevity. Notice that all of the simulations presented here are reproducible [199].

7.7.1 Choice of regularization parameter λ

As described in Section 7.4, the parameter λ controls the level of the regularization. The regularization can be viewed as imposing a particular prior knowledge on $\mathbf{W}$. This prior knowledge is not always available and often one wants to extrapolate it out of the available microphone measurements. This requires finding the best λ which controls the balance between how much a model fits the available microphone measurements and how important the prior knowledge is. The optimal balance will simultaneously avoid over-fitting ($\lambda \to 0$)

and predictions based on pure inference (large λ), which both produce poor results.

In this paper, λ is tuned using K-fold cross validation (KCV) [114] in which scheme the available measurements are split into K folds, namely K different groups of equal size. Here these folds consist of K scrambled groups of measured RIRs. For a given λ, the acoustic model is *trained* using only $K-1$ folds by solving the inverse problem using a particular regularization. The fold left outside is used to test the training performance and to produce the cross validation error:

$$\epsilon_{cv} = \frac{1}{N_{cv}} \sum_{m=1}^{N_{cv}} \|\mathbf{P}_{:,m} - \tilde{\mathbf{P}}_{:,m}\|_2^2 / \|\mathbf{P}_{:,m}\|_2^2, \tag{7.30}$$

i.e., the average of the normalized mean squared error (NMSE) of the N_{cv} RIRs belonging to the fold not used in the training. Here $\mathbf{P}$ is reconstructed from the optimal solution $\mathbf{W}^\star$ by means of either (7.5) or (7.12). This procedure is repeated K times, each time leaving out a different fold to produce a new cross validation error. The K cross validation errors are further averaged to produce $\bar{\epsilon}_{cv}$, the averaged cross validation error. A set of values for λ is tested using this procedure and the λ that gives the minimum averaged cross validation error is then chosen. Finally, the model is trained using all of the available microphone measurements with the chosen λ.

The KCV starts with an over-regularized inverse problem, with $\lambda = \lambda_{\max}$, and continues by logarithmically decreasing λ. Here $\lambda_{\max}$ is chosen as the maximum value for which the solution of the inverse problem is null. Table 7.1, reports the values of $\lambda_{\max}$ for each regularization [81, 200]. Notice that, for Tikhonov regularization, the solution is not null for $\lambda < \infty$, however the $\lambda_{\max}$ given in the table gives still an over-regularized solution.

In a simulated environment, it is actually possible to check if the KCV indeed returns the best λ since the *interpolation error* ϵ_{in} can be computed, *i.e.*, the average of the NMSE between the true RIRs and the interpolated RIRs, which in many real scenarios would not be available. Specifically, the interpolation error is computed using the same formula (7.30) on the interpolation volume. Figure 7.2 shows the averaged cross validation error and the interpolation error for $N_m = 12$ microphones as a function of λ for different types of regularization and acoustic models. Here and in the following a $K = 4$ KCV is used. In general the minima of the errors coincide and when they do not they are either very close to each other or they belong to flat regions of the error curves. This condition was verified for all the results presented here and in the following, the only exception being when $N_m = 4$ microphones are used. For this reason,

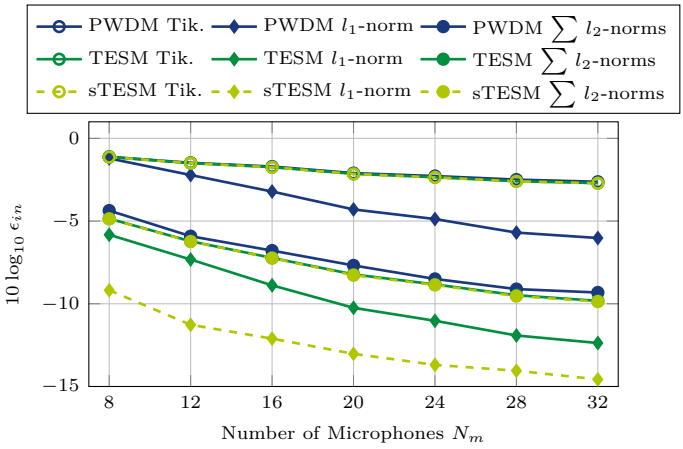


Figure 7.3: Simulated interpolation error for different types of acoustic models and regularizations as a function of the number of microphones.

these results are not shown. In conclusion, the results show that KCV is a good strategy for tuning λ if sufficient microphone measurements are available.

7.7.2 Comparison between acoustic models

In Fig. 7.3 the interpolation error is shown as a function of the number of microphone measurements used in the spherical array, while in Fig. 7.4 the worst-case interpolated RIR and its equivalent DFT are shown. Here worst-case indicates the position with maximum error in the interpolation volume. Looking at Fig. 7.3, it is clear that Tikhonov regularization gives almost equivalent performances with either PWDM or (s)TESM. This is expected, as discussed in Section 7.4, since Tikhonov regularization simply avoids large energy of the weight signals without imposing any spatial information. Here only the low frequencies are correctly interpolated as Fig. 7.4(a) shows. An improvement is achieved with the PWDM and l_1-norm regularization where spatio-spectral sparsity is promoted. In Fig. 7.4, it can be seen that the direct component becomes more pronounced but many artifacts are present. Clearly, as Fig. 7.3 shows, promoting sparsity in the frequency domain when the sound field is generated by a wide band signal is not a good choice. When only spatial sparsity is promoted, using the sum of l_2-norms regularization, better results are achieved. With this type of regularization all the acoustic models show similar performance: the TESM and the sTESM are practically equivalent while the interpolation errors of the PWDM are on average 0.45 dB higher with

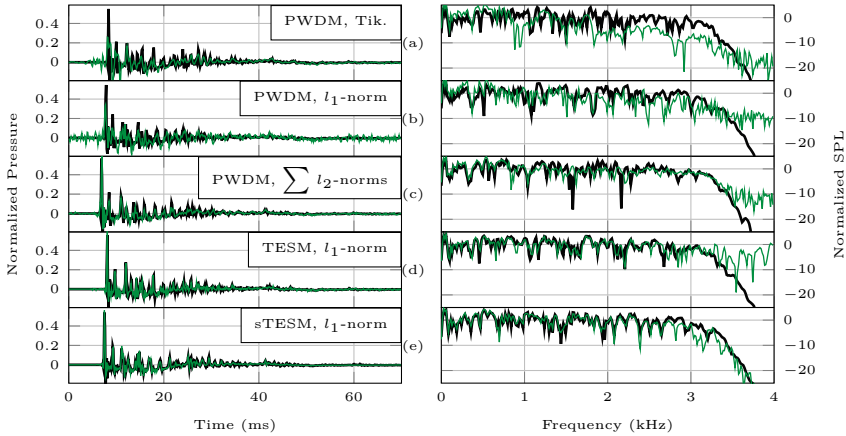


Figure 7.4: Simulated worst-case RIR interpolation for different types of acoustic models and regularizations using $N_m = 12$ microphones. Black thick line shows the original RIR while the green thin line shows the interpolated RIR.

respect to the (s)TESM. Finally, all the results discussed so far are outperformed using the (s)TESM and promoting spatio-temporal sparsity with the l_1-norm regularization. In Fig. 7.3, it can be seen that the same interpolation error obtained with $N_m = 32$ microphones and spatio-spectral sparsity (PWDM with l_1-norm) can be achieved with only $N_m = 8$ microphones using the TESM and spatio-temporal sparsity. A similar comparison can be made between the PWDM with spatial sparsity that reaches -9.3 dB interpolation error with $N_m = 32$ microphones and the sTESM that reaches -9.2 dB using only $N_m = 8$ microphones. When comparing the TESM with respect to the sTESM instead, it can be noticed that the sTESM significantly outperforms the TESM, particularly when the number of microphone measurements is smaller as Fig. 7.3 shows. Comparing Fig. 7.4(d) with Fig. 7.4(e) it can be noticed that the sTESM provides a more accurate representation of the high frequencies. This is due to the fact that the equivalent sources of the sTESM are already *shaped* with the spectrum of $s(n)$.

Finally, looking at Fig. 7.4 it is interesting to compare the RIR interpolation accuracy for the early reflections and late reverberation. Table 7.2 summarizes the worst-case NMSEs between the interpolated and the original RIRs appearing in Fig. 7.4. Here, in the last two rows, the NMSEs of the early reflections and late reverberation are computed separately. In general early reflections are interpolated more accurately than late reverberation for all the different types of regularizations and acoustic models except for the case of the PWDM with

	PWDM, Tik.	PWDM, l_1-norm	PWDM $\sum l_2$-norms	TESM, l_1-norm	sTESM, l_1-norm
NMSE (dB)	−0.58	−0.93	−4.42	−6.14	−9.37
NMSE (dB) Early Reflections	−0.52	−1	−4.5	−6.37	−9.8
NMSE (dB) Late Reverberation	−2.57	1	−1.88	−1.71	−3.39

Table 7.2: Simulated worst-case NMSEs of the interpolated RIRs shown in Fig. 7.4. The NMSEs are also shown for the early reflections of the RIRs (up to 30 ms) and the late reverberation (starting from 30 ms).

Tikhonov regularization, where the opposite happens. It can be seen that despite the higher reflection density in the late part of the RIRs, the sTESM with spatio-temporal sparsity still achieves the best results. Nevertheless it is difficult to conclude which regularization may achieve the best accuracy for the late reverberation. The early reflections constitute the most energetic part of the RIRs and therefore the un-regularized cost function $f(\cdot)$ is unbalanced towards them. Using a weighted norm in $f(\cdot)$ would balance the fitting and different regularizations could be used for different parts of the RIRs as well, with the disadvantage of complicating the regularization tuning procedure. A thorough analysis of these matters goes beyond the scope of this paper and is left for future work.

7.7.3 Analysis of weight signals

The weight signals are not only useful to perform the RIR interpolation but can also be used to provide a novel spatio-temporal visualization of the reverberant sound field, similar to the one proposed in [202]. It is important to outline that compared to the visualizations of [202], here a wider volume is represented instead *i.e.*, the volume where the RIR interpolation is performed. Figure 7.5 shows such a visualization, where for each regularization and acoustic model the weight signals are plotted simultaneously using spherical coordinates: the radius indicates the absolute value $20\log_{10}|w_l(n)|$ of the l-th weight signal (in dB) with its corresponding specific direction identified by the azimuthal and polar angle. Time is represented with color, with lines becoming darker and thinner as time proceeds. This enables one to view the direction of arrival of a specific reflection at a specific time. Moreover the signals are normalized by the maximum absolute value of $\mathbf{W}^\star$. In all the figures a single line, the thickest and with lighter color, reaches 0 dB in the direction associated to the sound source, indicating that all the methods are capable of reconstructing the line-of-sight component correctly. In the left hand figures, these lines point to $\theta \approx 90^\circ$ indicating that the sound source has the same height as the center of the microphone array. In the right hand figures instead, these lines point to

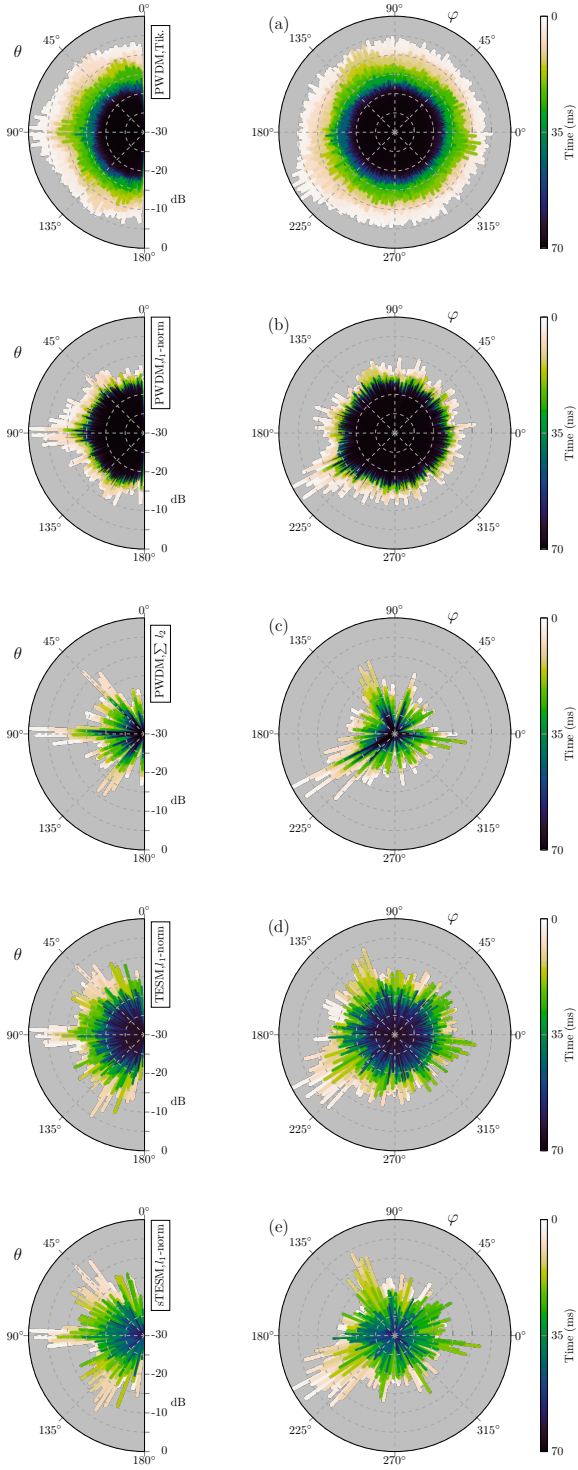


Figure 7.5: Visualization of the weight signals for different types of acoustic models and regularizations using $N_m = 12$ microphones. The radius in dB indicates the absolute value of the weight signal for a given direction. Directions are shown using spherical coordinates. Color indicates time: as time proceeds darker colors and smaller linewidth are used to plot the weight signals. Animated versions of these figures can be found at the following in [201].

$\varphi \approx 220°$ showing that the sound source is located in the front left corner of the room as shown in Fig. 7.1. Clearly, Tikhonov regularization manages to predict only the line-of-sight component as Fig. 7.5(a) shows: reflections keep arriving from all directions as time proceeds. This is not the case for the other results shown here, where the direction of arrival of higher order reflections can be clearly distinguished. In the left hand figures, where the weight signals are plotted as a function of the polar angle θ, it is possible to clearly identify the strongest reflections coming from the ceiling ($\theta \approx 45°$), walls ($\theta \approx 90°$) and floor ($\theta \approx 150°$). On the other hand, the right hand figures give a top view of the reflections. By combining each pair of figures and looking at the radius and color of the lines, one can understand the direction and time of arrival of a specific reflection. Notice that there is a similar trend as observed in the analysis of the performance of the RIR interpolations, with spatial and spatio-temporal sparsity giving clearer spatial information. For the PWDM, the weight signals obtained with spatio-spectral sparsity, Fig. 7.5(b), never decay completely to zero as time proceeds, while this is not the case for Fig. 7.5(c) where spatial sparsity minimizes the number of directions. Here one can clearly distinguish between different directions of arrival of higher order reflections, particularly the first order reflections coming from the walls, ceiling and floor. The same can be stated for the weight signals of (s)TESM with sTESM having much sparser signals with respect to TESM, as expected. Although many more directions of arrival are present in the (s)TESM solutions when compared to the PWDM solution with spatial sparsity, this does not necessarily mean that the PWDM solution is sparser. In fact, in the latter case, having many directions of arrival goes against the regularization which encourages grouping different reflections with similar directions of arrival and bundles them into a single one. This is obviously not beneficial for either the RIR interpolation or for the spatio-temporal visualization. Finally, the level of sparsity of the weight signals obtained with sTESM is maximal when compared with the other results. In the example shown in Fig. 7.5, for $N_m = 12$, the sTESM matrix $\mathbf{W}^\star$ has only 0.4% of its components non-zero, while TESM has 1.4%. The PWDM with spatial sparsity has 29.14% both in the frequency and time domain: despite the appearance of Fig. 7.5(c) the active weight signals decay over time to low levels but are always non-zero. On the other hand, PWDM with spatio-spectral sparsity has a dense matrix in the time-domain, while its frequency domain counterpart has 27% of its components non-zero.

7.8 Experimental results

In this section experimental results are presented. The RIR interpolation is performed using measured RIRs from the single- and multichannel audio

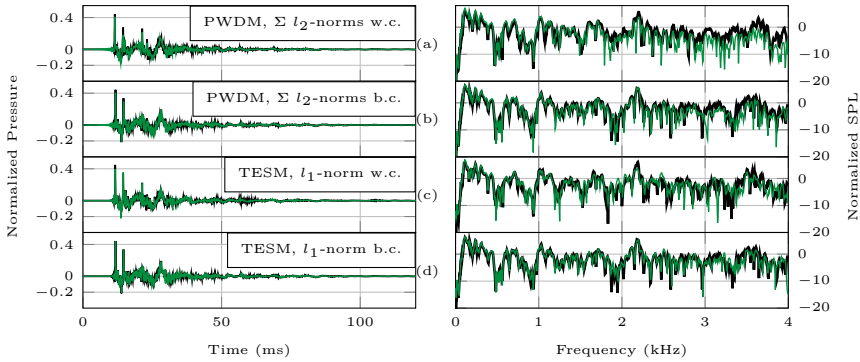


Figure 7.6: Experimental (w.c. and b.c.) results of RIR interpolation for different types of acoustic models and regularizations using $N_m = 15$ measured RIRs. Black thick line shows the original RIR while the green thin line shows the interpolated RIR. NMSE: (a) −5.5 dB (b) −9.43 dB (c) −7.11 dB (d) −11 dB.

recordings database (SMARD) [203]. The RIRs were measured in a box-shaped listening room with dimensions $[L_x, L_y, L_z] = [7.3, 8.1, 2.9]$ m and a reverberation time of $T_{30} = 0.097$ s. The database provides RIRs measured for different configurations of loudspeakers and microphone arrays. In this experimental study, configuration 1002 was used: here a Brüel & Kjær OmniSource 4295 loudspeaker unit positioned at $\mathbf{x}_s = [2.0,\ 6.5,\ 1.4]$ m and an orthogonal microphone array were used. The orthogonal microphone array consists of 3 uniform linear microphone arrays which share their origins at $\mathbf{x}_m = [4.4,\ 3.1,\ 1.5]$ m and are placed orthogonally to each other. Each uniform linear microphone array has 7 microphones spaced 5 cm apart. Here, only the $N_m = 15$ microphones belonging to the endings of the orthogonal microphone array's branches are used for the training, while the remaining $N_{in} = 6$ microphones closer to the origin are used to compute the interpolation error ϵ_{in} and to tune λ, thus avoiding the need of using a KCV scheme. The $N_w = 700$ equivalent sources are positioned in a Fibonacci lattice of radius 2.87 m centered at $\mathbf{x}_m$. All of the RIRs are down-sampled to a sampling frequency of $F_s = 8$ kHz. Note that all of the experiments presented here are reproducible using the code found in [199] and the RIR database available online at [203].

Table 7.3 shows the interpolation errors obtained using different types of acoustic models and regularizations. Note that the sTESM cannot be applied since the IR of the loudspeaker is not available. As in the simulation results, the TESM combined with spatio-temporal sparsity achieves the best performance. Looking at Fig. 7.3, it can be observed that these values are quite close to the ones

	PWDM, l_1-norm	PWDM $\sum$ l_2-norms	TESM, l_1-norm
ϵ_{in} (dB)	-5.47	-7.22	-8.51

Table 7.3: Experimental interpolation error for different types of acoustic models and regularizations using 15 measured RIRs.

obtained in the simulation results using $N_m = 16$ microphones. Notably, for the two cases where the PWDM was used, the experimental results are slightly better than the simulation results despite the fact that these were obtained using one microphone less. This is probably due to higher SNR of the experimental RIRs compared to the ones used in the simulations. Finally, Fig. 7.6 compares the interpolated RIRs with the original ones for the worst-case and best-case results: similarly to Fig. 7.4, where only the worst-case results are shown, early reflections and low frequencies are reconstructed with higher accuracy with respect to the late reverberation and high frequencies respectively.

7.9 Conclusions

In this paper the problem of spatially interpolating measured RIRs is posed as an inverse problem. Two different acoustic models that are able to approximate any sound field in a source-free volume generated by a far field source are presented and compared: the PWDM which is a widely known frequency domain model and (s)TESM, a novel time-domain method that can also incorporate the knowledge of the source signal generating the sound field. It is shown that various sparsity promoting regularizations can be used to cope with the ill-posed nature of the inverse problem. Spatio-spectrally and spatially sparse solutions can be obtained with the PWDM and with the l_1-norm and sum of l_2-norms regularization respectively, while a spatio-temporally sparse solution can be encouraged using (s)TESM with l_1-norm regularization.

These inverse problems turn out to be large scale optimization problems and so great care must be taken in finding a computationally tractable algorithm. For this reason, the acoustic models are described as linear operators that can be computed with minimal memory storage. Their adjoint operators, that share the same property, are used in the computation of the derivatives needed in the optimization algorithm and their physical interpretation is also given. This makes it possible to use matrix-free optimization. The optimization problems are solved using an accelerated version of the FBS.

Numerical simulations are then presented where RIRs generated with a

modified version of the IM are interpolated using the described algorithms. The KCV method is used to select the regularization parameter and it is shown that this strategy works correctly if enough microphone measurements are used. The different approaches to perform RIR interpolation are then compared. The comparison shows that the novel spatio-temporally sparse representation outperforms the others particularly when the source signal is available. This provides a good RIR interpolation even when only few microphone measurements are available. Finally, the different approaches are applied using actual measured RIRs reaching similar results to the one obtained using simulated data.

Acknowledgment

The authors would like to thank Lorenzo Stella, Andrea Themelis and Panagiotis Patrinos for their suggestions and precious help in the algorithm design and coding.

Chapter 8

Dereverberation and source localization

Joint acoustic localization and dereverberation by sound field interpolation

Niccolò Antonello, Enzo De Sena, Marc Moonen, Patrick A. Naylor, and Toon van Waterschoot

Submitted for publication, June 2018.

The candidate's contributions as first author include: literature study, co-development of the presented method, software implementation and computer simulations, text redaction and editing.

Abstract

Acoustic source localization and dereverberation are formulated jointly as an inverse problem. The inverse problem consists of the interpolation of the sound field measured by a set of microphones. The recorded sound pressure is matched with that of a particular acoustic model. This model is based on a collection of equivalent sources creating either spherical or plane waves. In order to achieve meaningful results, spatial, spatio-temporal and spatio-spectral sparsity can be promoted in the signals originating from the equivalent sources. The large-scale optimization problem resulting from the inverse problem formulation is solved using a first order matrix-free optimization algorithm combined with a weighted overlap-add procedure. It is shown that once the equivalent source signals capable of effectively interpolating the sound field are obtained, they can be readily used to localize a moving sound source in terms of direction of arrival (DOA) and to perform dereverberation in a highly reverberant environment. Results from simulation experiments and from real measurements show that the proposed algorithm is robust against both localized and diffuse noise exhibiting a noise reduction in the dereverberated signals.

8.1 Introduction

While there are many source localization algorithms that work well in free-field acoustic scenarios, source localization in highly reverberant environments is challenging [204, 205]. Reverberant environments are also problematic for speech intelligibility and significant research efforts have been focusing on dereverberation [51]. Dereverberation and source localization are often connected: for example many dereverberation algorithms require the knowledge of the direction of arrival (DOA) of the sound source [53, 54]. Instead, other algorithms rely either on channel equalization which requires estimation of the room impulse responses (RIRs) [206] or on multi-channel linear prediction (MCLP) which requires no a priori knowledge of the acoustics but is non-robust to additive noise [55, 56].

Recently, source localization has been posed as an inverse problem where physical acoustic models are used to reconstruct and localize the sound source [44, 46, 207]. This is achieved by exploiting compressed sensing (CS) techniques [37] *i.e.*, by including sparse regularization in the inverse problems. Such methods allow precise localization of the source position inside the room but require detailed knowledge of the room geometry and boundary conditions. Alternatively, the plane wave decomposition method (PWDM) has been shown to approximate well any sound field in source-free volumes [34]. This allows

sound sources to be localized without knowledge of the room geometry but requires a large number of microphones, particularly when these are scattered in a large volume [48, 208]. CS techniques have also been employed with the PWDM, or with the related spherical harmonic decomposition method (SHDM), to perform DOA estimation [65] and combined with channel equalization [209] or beamforming [210] algorithms to achieve dereberveration.

In this paper, a recently proposed RIR interpolation algorithm [33], is reformulated and modified to be able to perform joint source localization and dereverberation. The proposed algorithm, called acoustic dereverberation and localization through field interpolation (ADeLFI), relies on the interpolation of the sound field recorded by a set of microphones which is formulated as a regularized inverse problem. This consists of an optimization problem that matches the sound pressure measured by the microphones with the sound pressure predicted by an acoustic model. Here two acoustic models are compared: the PWDM and the time-domain equivalent source method (TESM) which are both capable of approximating the sound field in a source-free volume where the microphones are positioned [33]. Both models are based on a large collection of *equivalent sources* positioned in a *spherical lattice*. Each equivalent source contributes to the sound field from a particular direction and is associated with a DOA. The equivalent sources produce either spherical (TESM) or plane waves (PWDM) controlled by signals, named *weight signals*, that are estimated through the optimization problem. It is shown that, by combining a specific sparsity-inducing regularization with a specific model, three kinds of sparse priors can be promoted in the weight signals: spatial sparsity, spatio-temporal sparsity and spatio-spectral sparsity. Spatial sparsity promotes only few weight signals to be active, hence limiting the number of directions from which the sound waves can arrive. In addition to that, spatio-temporal and spatio-spectral sparsity lead to weight signals that have either an impulsive behavior or a spectrum composed by few frequency components, respectively. A novel procedure for tuning the level of regularization is presented that requires an additional validation microphone. The resulting optimization problem has a large scale and nonsmooth cost function: this is solved using a matrix-free accelerated version of the proximal gradient (PG) algorithm [83] and combined with a weighted overlap-add (WOLA) procedure.

Once the interpolation step is achieved, the weight signals can be used to estimate the DOA of a sound source. It is observed that the weight signal with *strongest energy* is associated with a particular DOA which is more likely to correspond to the DOA of the original sound source. This enables to localize the sound source and as a consequence a dereverberated signal can be readily obtained by selecting the weight signal corresponding to the estimated DOA. Alternatively, one can compute the sound pressure inside

the source-free volume with the chosen acoustic model using a small set of weight signals with corresponding DOAs close to the estimated one. In fact, solving the inverse problem accounts for decomposing the sound field into different sound waves with specific directions. This effectively creates a *spatial distribution* of reverberation among the weight signals controlling these sound waves, resulting in the weight signals to effectively be *dereverberated signals*. Additionally, this decomposition will occur also in the presence of a noise field; the contribution of the noise field will also be spatially distributed among the weight signals which therefore exhibit a *noise reduction*. It is shown that the WOLA procedure not only allows to split the large-scale optimization problem into smaller subproblems, but also to perform these operation during different time windows, enabling the DOA estimation and dereverberation of a *moving sound source*.

Simulated and real measurement results show that in a sound field generated by a speech source, spatio-spectral and spatial sparsity based regularization have similar performances and outperform spatio-temporal sparsity based regularization both in terms of sound field interpolation and dereverberation. The proposed algorithm is shown to be robust even when localized and diffuse noise are present in the microphone signals; accurate DOA estimation and noise reduction in the dereverberated signal are achieved. Notice that this paper will not focus on the computational complexity of the proposed algorithm which is rather large and could be effectively reduced by employing parallel computing and fast transformations [211,212]. Instead, the aim of the paper is to introduce a novel approach which is substantially different from the traditional dereverberation and localization algorithms and to compare it qualitatively to state-of-the-art algorithms in a variety of scenarios.

This paper is organized as follows: in Section 8.2 the TESM and PWDM are described. Section 8.3 describes the inverse problem that is used to perform the sound field interpolation. In Section 8.4 the ADeLFI algorithm is presented describing the matrix-free optimization algorithm, showing the WOLA procedure and the regularization tuning strategy. Finally, in Section 8.5 the algorithm is validated using simulated and real measurements and in Section 8.6 conclusions are drawn.

Preliminary results have been presented in [213]. The main novelties of this paper are: (i) a novel processing of the weight signals to reduce artifacts in the dereverberated signals, (ii) modifications to the proposed algorithm which allow the possibility of tracking the position of a moving sound source, (iii) a comparison with state-of-the-art dereverberation and DOA estimation algorithms, (iv) the inclusion of more objective perceptual performance measures and (v) the validation of the proposed algorithm using real measurements.

8.2 Acoustic models

8.2.1 Time domain equivalent source method

The time domain Green's function for a point equivalent source in free field is defined as:

$$\phi_{l,m}(t) = \frac{1}{4\pi d_{l,m}} \delta\left(t - \frac{d_{l,m}}{c}\right), \tag{8.1}$$

where c is the speed of sound, δ is the Dirac delta function and $d_{l,m} = \|\mathbf{x}_l - \mathbf{x}_m\|_2$ is the distance between the m-th microphone position $\mathbf{x}_m$ and the l-th equivalent source position $\mathbf{x}_l$. Equation (8.1) represents a particular solution of the free-field wave equation with null initial conditions and describes a *spherical wave*. Equation (8.1) can be discretized over time at a sampling frequency F_s using a *fractional delay* filter with infinite impulse response (IIR) $h_{l,m}$ [192]. The time-domain equivalent source method (TESM) can be described by the following equation [33]:

$$p(\mathbf{x}, n)|_{\mathbf{x}=\mathbf{x}_m} \approx \sum_{l=0}^{N_w-1} \frac{1}{4\pi d_{l,m}} h_{l,m}(n) * w_l(n) \quad \forall \mathbf{x}_m \in \Omega, \tag{8.2}$$

where $*$ represents convolution, $p(\mathbf{x}, n)|_{\mathbf{x}=\mathbf{x}_m}$ is the sound pressure at $\mathbf{x}_m$ at a discrete time n and $\Omega \subset \mathbb{R}^3$ is a source-free volume where any sound field can be well approximated using a collection of N_w equivalent sources [33] controlled by the signals $w_l(n)$ referred to here as *weight signals*. Equation (8.2) can be generalized for N_m discrete positions $\mathbf{x}_m \in \Omega$ and N_t discrete times:

$$\mathbf{P} = \mathsf{D}_t \mathbf{W}, \tag{8.3}$$

where $\mathbf{P} \in \mathbb{R}^{N_t \times N_m}$ is a matrix in which the m-th column is the sound pressure signal $p(\mathbf{x}, n)|_{\mathbf{x}=\mathbf{x}_m}$ for $n = 0, ..., N_t - 1$, and $\mathbf{W} \in \mathbb{R}^{N_t \times N_w}$ is a matrix in which the l-th column is the weight signal $w_l(n)$. The linear mapping $\mathsf{D}_t : \mathbb{R}^{N_t \times N_w} \to \mathbb{R}^{N_t \times N_m}$ maps the weight signals to the sound pressures and represents a *dictionary* of spherical waves. Equation (8.3) should not be confused with a linear matrix equation, *i.e.*, D_t is indeed a mapping rather than a matrix multiplier.

8.2.2 Plane wave decomposition method

A plane wave is defined as

$$\hat{\phi}_{l,m}(f) = e^{-ik_f d_{l,m}}, \tag{8.4}$$

and is the homogeneous solution of the Helmholtz equation. Here the hat notation is used to indicate complex numbers, f is the frequency index and k_f is the wave number defined as $k_f = 2\pi f F_s/(N_f c) = \omega_f/c$, where F_s is the sampling frequency and N_f the number of discrete frequencies utilized. A sound field in a source-free volume Ω can be as well represented by a finite weighted sum of plane waves coming from N_w different directions [34]:

$$\hat{p}(\mathbf{x}, f)|_{\mathbf{x}=\mathbf{x}_m} \approx \sum_{l=0}^{N_w-1} \hat{\phi}_{l,m}(f)\hat{w}_l(f) \;\forall \mathbf{x}_m \in \Omega, \tag{8.5}$$

where the *weight* $\hat{w}_l(f)$ is a complex scalar that weights the l-th plane wave at the frequency index f. Equation (8.5) describes the plane wave decomposition method (PWDM). This equation can be generalized for N_m discrete positions $\mathbf{x}_m \in \Omega$ and N_f discrete frequencies:

$$\hat{\mathbf{P}} = \mathsf{D}_p \hat{\mathbf{W}}, \tag{8.6}$$

where $\hat{\mathbf{P}} \in \mathbb{C}^{N_f \times N_m}$ is a matrix in which the m-th column is the discrete Fourier transform (DFT) of the sound pressure signal $p(\mathbf{x}, n)|_{\mathbf{x}=\mathbf{x}_m}$ and $\hat{\mathbf{W}} \in \mathbb{C}^{N_f \times N_w}$ is a matrix containing the weights $\hat{w}_l(f)$. Similarly to the TESM, the linear mapping $\mathsf{D}_p : \mathbb{C}^{N_f \times N_w} \to \mathbb{C}^{N_f \times N_m}$ represents a dictionary of plane waves.

8.3 The inverse problem

Consider a sound source in the far field and a set of of N_m microphones positioned inside a source-free volume $\Omega \in \mathbb{R}^3$ as depicted in Figure 8.1. The equivalent sources can be positioned in a Fibonacci lattice centered at the center of the microphone array, providing a nearly uniform sampling of the surface of a sphere [193]. In order to achieve accurate interpolation, a large number of equivalent sources must be used.

The aim is to interpolate the sound field inside this volume in order to jointly dereverberate and localize the sound source. What is sought by the inverse problem is to estimate the weight signals that lead to the optimal sound field approximation. This can be achieved by matching the microphone measurements with the acoustic model of choice. The following optimization problem seeks to perform this operation, in the least-squares sense:

$$\mathbf{W}^\star = \underset{\mathbf{W}}{\operatorname{argmin}} \quad q(\mathbf{W}) = \frac{1}{2}\|\mathsf{D}\mathbf{W} - \tilde{\mathbf{P}}\|_F^2, \tag{8.7}$$

where $\|\cdot\|_F$ is the Frobenius norm and the columns of the matrix $\tilde{\mathbf{P}}$ contain the microphone measurements, *i.e.*, the N_t-long measured sound pressure signals.

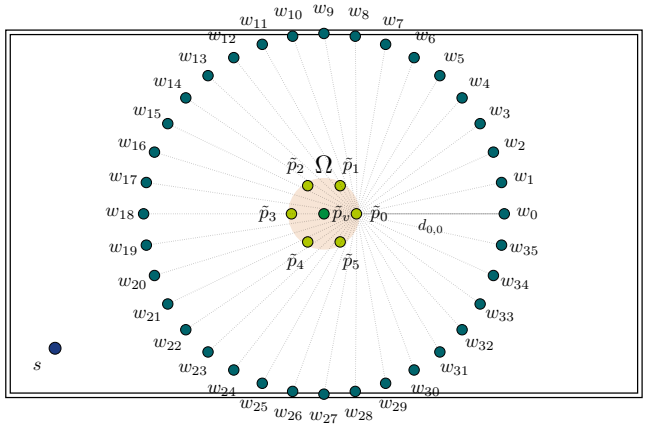


Figure 8.1: Horizontal cross-section of a room. A sound source in the front left corner generates a reverberant sound field. A spherical microphone array (light green dots) measures the sound field of the room. A set of equivalent sources (dark green dots) represent the acoustic model that is used to match the sound pressure measured by the microphones. The sound field is interpolated in the shaded volume Ω. An additional microphone is placed at the center of the array to validate the quality of the interpolation. The distances $d_{l,0}$ between the first microphone and the l-th equivalent source are shown.

In order to simplify the explanation, here and in the following D will be assumed to be a time domain model, *e.g.*, TESM. Only when necessary, the difference between time domain and frequency domain models will be pointed out. This implies that *e.g.*, in $\hat{\mathbf{W}} \in \mathbb{C}^{N_f \times N_w}$, the hat notation indicating complex matrices will be dropped and the symbol $\mathbb{R}$ will be used instead of $\mathbb{C}$ when indicating dimensions. Notice that since $N_f = N_t$ the dimensions are equivalent.

The aim of (8.7) is to minimize the *un-regularized cost function* $q(\mathbf{W})$, which consists of the error between the measured sound pressure and the sound pressure predicted by the acoustic model. Problem (8.7) is heavily *ill-posed*: *i.e.*, if many sound waves are used to construct D, there exist many solutions $\mathbf{W}^\star$ that can minimize the cost function q effectively. This will in general lead to *over-fitting*: the measured sound pressure will coincide with the sound pressure of the acoustic model but only at the microphone positions, leading to a poor sound field interpolation in other positions. To avoid this, it is necessary to regularize problem (8.7) by adding a *regularization term* g to its cost function q, *i.e.*:

$$\mathbf{W}^\star = \underset{\mathbf{W}}{\operatorname{argmin}} \quad q(\mathbf{W}) + \lambda g(\mathbf{W}), \tag{8.8}$$

where λ is a scalar. A possible choice is the *sum of l_2-norms regularization* corresponding to

$$g(\mathbf{W}) = \sum_{l=0}^{N_w-1} \|\mathbf{w}_l\|_2, \tag{8.9}$$

where $\mathbf{w}_l$ indicates the l-th column of $\mathbf{W}$. This regularization promotes only few columns of $\mathbf{W}$ to be non-zero, in practice promoting *spatial sparsity* of the equivalent sources. Another very common regularization is the *l_1-norm regularization* corresponding to

$$g(\mathbf{W}) = \|\mathrm{vec}(\mathbf{W})\|_1 = \sum_{l=0}^{N_w-1} \|\mathbf{w}_l\|_1, \tag{8.10}$$

which promotes *sparsity* in $\mathbf{W}$, *i.e.*, only few elements of the matrix to be non-zero. If a time domain acoustic model (TESM) is used, *spatio-temporal sparsity* is promoted. On the other hand with a frequency domain acoustic model (PWDM), *spatio-spectral sparsity* is promoted.

Once a solution is obtained, the DOA of the sound source can be inferred by finding the weight signal with the *strongest energy*:

$$\|\mathbf{w}_b^\star\|_2^2 = \max\left\{\|\mathbf{w}_0^\star\|_2^2, \ldots, \|\mathbf{w}_{N_w-1}^\star\|_2^2\right\}, \tag{8.11}$$

since the b-th weight signal is associated with an equivalent source position in the Fibonacci lattice and hence to a polar angle $\theta^\star$ and azimuth angle $\varphi^\star$. Figure 8.2 shows the energy of the weight signals as a function of the azimuth and polar angles for the simulation results presented in Section 8.5.1. A clear maximum is visible towards the direction of the sound source shown by the red dot.

A dereverberated signal is also readily available: the weight signal $\mathbf{w}_b^\star$ will have less reverberation compared to the microphone signals. That is because $\mathbf{w}_b^\star$ accounts only for a single equivalent source which contributes to the sound field from a specific direction. If proper sound field interpolation is reached, the effect of reverberation is *spatially distributed* among the equivalent sources of the acoustic model.

However, since a finite number of equivalent sources is used, it is possible that also other equivalent sources in close proximity to the equivalent source corresponding to $\mathbf{w}_b^\star$ contribute significantly to the sound field generated from that particular direction. It is therefore advantageous to employ multiple weight signals to produce the dereverberated signal. This can be achieved by selecting N_b equivalent source signals that are the nearest neighbors of the equivalent source corresponding to $\mathbf{w}_b^\star$ in the Fibonacci lattice. These weight signals,

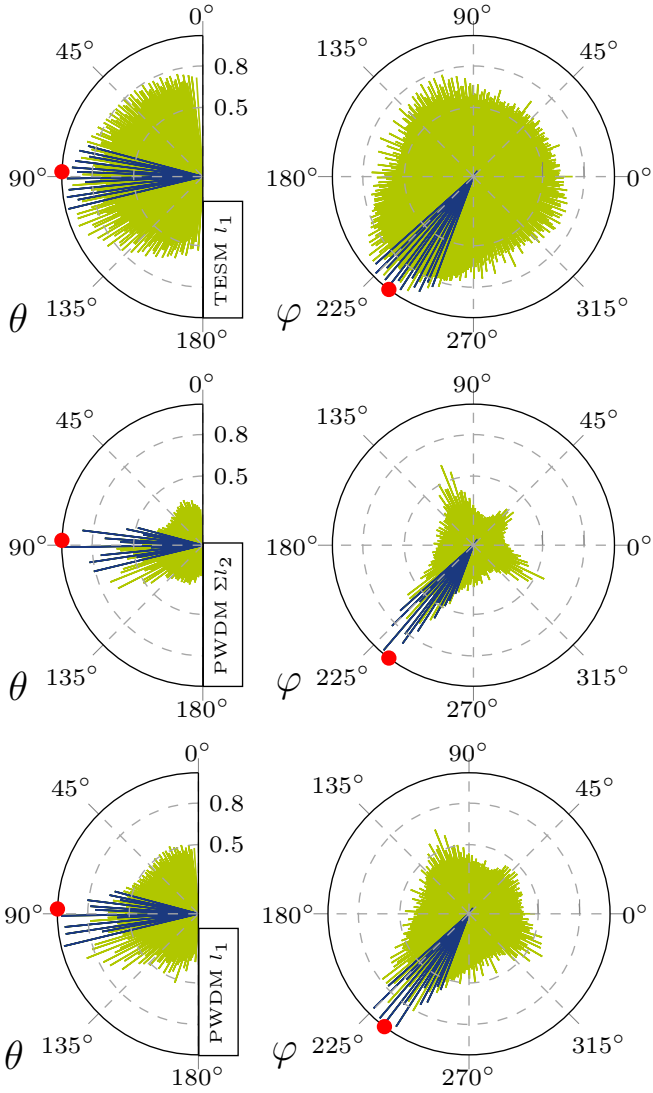


Figure 8.2: Visualization of the normalized energy of the weight signals, *i.e.*, $\|\mathbf{w}_l^\star\|_2^2$, as a function of the azimuth angle φ and polar angle θ. The red dots represent the true source position. Darker lines represent the N_b neighbor weight signals around the estimated DOA. The weight signals are obtained from simulation results using $N_w = 500$ equivalent sources and $N_m = 16$ microphones with a sensor noise of 40 dB SNR. These results will be presented in detail in Section 8.5.1.

together with $\mathbf{w}_b^\star$, can be used as the columns of $\mathbf{W}_b^\star \in \mathbb{R}^{N_t \times (N_b+1)}$. The energy of these weight signals is visualized with darker lines in Figure 8.2. A dereverberated signal can then be obtained by generating a new sound field $\mathbf{P}_b = \mathsf{D}^b \mathbf{W}_b^\star$, using the same acoustic model employed in the inverse problem. Here, $\mathsf{D}^b : \mathbb{R}^{N_t \times (N_b+1)} \to \mathbb{R}^{N_t \times N_m}$ maps a smaller number of weight signals to $\mathbf{P}_b$ which now represents a new sound field created by a limited number of equivalent sources corresponding to the selected directions. Although here D^b utilizes the same microphone positions of the measurements, any position inside Ω can be chosen as well. This is achieved by setting different microphone positions during the construction of D^b. As it will be shown in Section 8.5, the artifacts present in the $\mathbf{P}_b$ signals are often less audible than those in $\mathbf{w}_b^\star$, particularly when spatio-spectral sparsity is promoted. Similarly, it is possible to analyze and process $\mathbf{W}^\star$ to perform joint DOA estimation and dereverberation *e.g.*, using a frame-based approach. As it will be shown in the next session, this is highly preferable not only for computational reasons but also since it allows to track a *moving sound source*.

8.4 The ADeLFI algorithm

In this section the proposed algorithm is presented, referred to as *acoustic dereverberation and localization through field interpolation (ADeLFI)*. The pseudo-code of ADeLFI is given in Algorithm 5 and a detailed explanation is provided in the next subsections.

8.4.1 Optimization algorithm

Problem (8.8) is nonsmooth and can easily become of large scale. A well known algorithm that can address this type of problems is the *proximal gradient (PG) algorithm* which is a first-order optimization algorithm suitable for nonsmooth cost functions and having minimal memory requirements [81, 214]. The PG algorithm generalizes the gradient descent algorithm to a class of nonsmooth problems such as the problem in (8.8) where q is smooth and g is nonsmooth, as the regularization terms of (8.9) and (8.10). The PG algorithm consists of iterating

$$\mathbf{W}^{s+1} = \operatorname{prox}_{\gamma\lambda g}\left(\mathbf{W}^s - \gamma \nabla q(\mathbf{W}^s)\right), \tag{8.12}$$

starting from an initial guess $\mathbf{W}^0$. Here ∇q is the gradient of q, γ is the step-size, and $\operatorname{prox}_{\gamma g}$ is the *proximal mapping* of the function g [81].

For the regularization terms described in (8.9) and (8.10) the proximal mapping consists of a computationally cheap operation. If $g(\mathbf{W}) = \|\operatorname{vec}(\mathbf{W})\|_1$, its

proximal mapping reads:

$$\text{prox}_{\gamma\lambda g}(\mathbf{W}) = \mathcal{P}_+(\mathbf{W} - \lambda\gamma) - \mathcal{P}_+(-\mathbf{W} - \lambda\gamma), \tag{8.13}$$

where $\mathcal{P}_+$ is the element-wise mapping performing $\max\{0, \cdot\}$. On the other hand, if $g(\mathbf{W}) = \sum_{l=0}^{N_w-1} \|\mathbf{w}_l\|_2$, the proximal mapping becomes:

$$\begin{aligned} &\text{prox}_{\gamma\lambda g}(\mathbf{W}) = \\ &[\mathbf{w}_0\mathcal{P}_+(1 - \tfrac{\lambda\gamma}{\|\mathbf{w}_0\|_2}) \dots \mathbf{w}_{N_w}\mathcal{P}_+(1 - \tfrac{\lambda\gamma}{\|\mathbf{w}_{N_w}\|_2})]. \end{aligned} \tag{8.14}$$

In both cases the proximal mapping performs a soft-thresholding of either the elements of $\mathbf{W}$ (l_1-norm) or its columns (sum of l_2-norms) which is a computationally simple operation.

Another fundamental operation needed in the PG algorithm is the evaluation of the gradient of q which is given by

$$\mathbf{J} = \nabla q(\mathbf{W}) = \mathsf{D}^*(\underbrace{\mathsf{D}\mathbf{W} - \tilde{\mathbf{P}}}_{\mathbf{R}}), \tag{8.15}$$

where $\mathbf{J}$ is a matrix with the gradient with respect to $\mathbf{w}_l$ in the l-th column, $\mathbf{R}$ is the residual matrix, *i.e.*, the difference between the sound pressure recorded by the microphones and the sound pressure predicted by the acoustic model, and $\mathsf{D}^* : \mathbb{R}^{N_t \times N_m} \to \mathbb{R}^{N_t \times N_w}$ is the *adjoint mapping* of D. In this context, D is often referred to as the *forward mapping*. The adjoint mapping is a generalization of the transpose of a matrix to linear mappings. In general, when D is a linear mapping between two large finite-dimensional spaces, it is not ideal to use a matrix-vector multiplication based algorithm for its evaluation, as this in fact would require the storage of a very large matrix. Instead, it is possible to compute the mapping using its definition, *i.e.*, by directly applying (8.2) or (8.5) iteratively or by utilizing fast transformations. The same strategy can be adopted for the adjoint mapping whose definition is similar to the definition of the forward mapping. The adjoint mapping of TESM consists of the cross-correlation with the residual $\mathbf{R}$ [33]:

$$j_l(-n) = \sum_{m=0}^{N_m-1} \frac{1}{4\pi d_{l,m}} h_{l,m}(n) * r_m(-n), \tag{8.16}$$

where r_m is the signal appearing in the m-th column of $\mathbf{R}$. Similarly, for the PWDM the adjoint mapping is obtained as the cross-spectrum between $\hat{\phi}_{l,m}$ and $\hat{r}_m$:

$$\hat{j}_l(f) = \sum_{m=0}^{N_m-1} \hat{\phi}^*_{l,m}(f)\hat{r}_m(f), \tag{8.17}$$

where $\hat{r}_m$ is the residual of the m-th microphone in the frequency domain. In both cases j_l indicates the (complex) signal appearing in the l-th column of $\mathbf{J}$. The gradient at the iterate $\mathbf{W}^s$ can be efficiently computed together with the evaluation of $q(\mathbf{W}^s)$: this strategy is also known as back-propagation in machine learning [114] or automatic differentiation [127] and leads to *matrix-free optimization* [104, 214].

Finally, an accelerated variant of the PG algorithm is used employing a limited-memory quasi-Newton method [83]. An implementation of the algorithm is also available online [137].

8.4.2 Weighted overlap-add procedure

Solving the optimization problem in (8.8) using microphone signals with a duration of the order of seconds is not feasible since evaluating the linear mapping D and its adjoint becomes too costly and the optimization problem becomes too large. For example choosing $N_w = 500$ equivalent sources and $N_t = 16000$ (2 s with a sampling frequency $F_s = 8$ kHz) would result in 8×10^6 optimization variables.

To overcome this issue, the optimization problem is split into several smaller subproblems. A weighted overlap-add (WOLA) procedure is used: the microphone signals are split into N_r frames of N_τ samples each. If $N_\tau = 512$ and $N_w = 500$ the subproblem will then have 256×10^3 optimization variables, which is manageable. Analysis and synthesis window functions, here chosen to be both square-rooted Hann windows, are indicated in Algorithm 5 with $\mathbf{u}_a$ and $\mathbf{u}_s$ respectively. The frames are overlapped by N_o samples: here an overlap of 50% is used.

In each frame an acoustic delay resulting from the time it takes for the equivalent source sound waves to reach the microphones is present. When the equivalent sources are far, this acoustic delay can occupy a significant part of each frame. It is therefore necessary to remove the acoustic delay by introducing a *time offset* t_d in the acoustic model. When a spherical microphone array of radius ρ_M is used together with a Fibonacci lattice of radius ρ_W with the same center, this time offset corresponds to $t_d = (\rho_W - \rho_M)/c$. In general, the time offset corresponds to the minimum distance $\min\{d_{l,m} \mid l = 0, \ldots, N_w - 1, m = 0, \ldots, N_m - 1\}$ between the equivalent sources and the microphone positions divided by the speed of sound c. The offset can be incorporated in both acoustic models by modifying (8.1) with:

$$\phi_{l,m}(t) = \tfrac{1}{4\pi d_{l,m}} \delta\left(t - d_{l,m}/c + t_d\right), \tag{8.18}$$

Algorithm 5 Acoustic Dereverberation and Localization through Field Interpolation (ADeLFI) method

Inputs:
$\tilde{\mathbf{P}} \in \mathbb{R}^{N_t \times N_m}$, $\tilde{\mathbf{p}}_v \in \mathbb{R}^{N_t}$
$d_{l,m}$, $d_{l,v}$ for $l = 0, \ldots, N_w - 1$, $m = 0, \ldots, N_m - 1$
g, N_τ, N_b, N_o, $\mathbf{u}_a$, $\mathbf{u}_s$, $\beta \in (0, 1]$, η

Outputs:
$\bar{\mathbf{P}} \in \mathbb{R}^{N_t \times N_m}$, $\bar{\mathbf{w}} \in \mathbb{R}^{N_t}$, $\boldsymbol{\varphi}^\star \in \mathbb{R}^{N_r}$, $\boldsymbol{\theta}^\star \in \mathbb{R}^{N_r}$

Using selected acoustic model construct
$\mathsf{D} : \mathbb{R}^{N_\tau \times N_w} \to \mathbb{R}^{N_\tau \times N_m}$ with $d_{l,m}$ and
$\mathsf{D}^v : \mathbb{R}^{N_\tau \times N_w} \to \mathbb{R}^{N_\tau}$ with $d_{l,v}$.
Compute candidate angles $\breve{\boldsymbol{\varphi}}_w \in \mathbb{R}^{N_w}$, $\breve{\boldsymbol{\theta}}_w \in \mathbb{R}^{N_w}$
Set $k = 0$, $r = 0$, $\epsilon_{v,-1} = +\infty$, $\bar{\mathbf{e}} = \mathbf{0} \in \mathbb{R}^{N_w}$,
$\bar{\mathbf{P}} = \mathbf{0}$, $\bar{\mathbf{w}} = \mathbf{0}$, $\boldsymbol{\varphi}^\star = \mathbf{0}$, $\boldsymbol{\theta}^\star = \mathbf{0}$.
while $k + N_\tau \leq N_t$ **do**
    Select samples relative to the r-th time window
    $\tilde{\mathbf{P}}_\tau \leftarrow \mathsf{S}_k \tilde{\mathbf{P}}$, $\tilde{\mathbf{p}}_{v,\tau} \leftarrow [p_v(k) \ldots p_v(k + N_\tau)]^\intercal$
    (S_k is a selection operator)
    Weight each column of $\tilde{\mathbf{P}}_\tau$ and $\tilde{\mathbf{p}}_{v,\tau}$ using $\mathbf{u}_a$
    $\lambda \leftarrow 10^{-6} \lambda_{\max}$
    for $z = 0, \ldots, N_\lambda - 1$ **do**
        $\mathbf{W}_\tau^{\star,z} \leftarrow \underset{\mathbf{W}_\tau}{\operatorname{argmin}} \frac{1}{2} \|\mathsf{D}\mathbf{W}_\tau - \tilde{\mathbf{P}}_\tau\|_F^2 + \lambda g(\mathbf{W}_\tau)$
        $\epsilon_{v,z} \leftarrow \|\mathsf{D}^v \mathbf{W}_\tau^{\star,z} - \tilde{\mathbf{p}}_{v,\tau}\|_2^2 / \|\tilde{\mathbf{p}}_{v,\tau}\|_2^2$
        if $\epsilon_{v,z} > \epsilon_{v,z-1} + \eta$ **then break**
        else
            $\bar{\mathbf{W}}_\tau^r \leftarrow \mathbf{W}_\tau^{\star,z}$
            increase λ logarithmically
    Weight each column of $\bar{\mathbf{W}}_\tau^r$ using $\mathbf{u}_s$
    $\mathbf{e}^r \leftarrow [\|\bar{\mathbf{w}}_{\tau,1}^r\|_2^2, \ldots, \|\bar{\mathbf{w}}_{\tau,N_w}^r\|_2^2]^\intercal$,
    $\bar{\mathbf{e}} \leftarrow \mathbf{e}^r + \beta \bar{\mathbf{e}}$
    Set b as the index of the maximum element of $\bar{\mathbf{e}}$
    Set $\varphi_r^\star \leftarrow \breve{\varphi}_b$, $\theta_r^\star \leftarrow \breve{\theta}_b$
    $[\bar{w}_k, \ldots, \bar{w}_{k+N_\tau}]^\intercal \leftarrow [\bar{w}_k, \ldots, \bar{w}_{k+N_\tau}]^\intercal + \bar{\mathbf{w}}_{\tau,b}^{r\intercal}$
    Find N_b neighbors equivalent source
    Construct $\mathsf{D}^{b,r}$ and $\bar{\mathbf{W}}_{\tau,b}^r$
    Compute $\bar{\mathbf{P}}_{\tau,b}^r = \mathsf{D}^{b,r} \bar{\mathbf{W}}_{\tau,b}^r$
    Append $\bar{\mathbf{P}}_{\tau,b}^r$ to $\bar{\mathbf{P}}$
    $r \leftarrow r + 1$, $k \leftarrow k + N_\tau - N_o$

and (8.4) with

$$\hat{\phi}_{l,m}(f) = e^{-ik_f(d_{l,m}-ct_d)}. \tag{8.19}$$

Removing the acoustic delay brings also a computational benefit to the TESM since it reduces the order of the IIR fractional delay filters.

The size of the interpolation volume Ω, and hence the microphone array geometry, imposes a constraint on the frame length N_τ. Assuming that the acoustic delay of the equivalent sources is removed, the following inequality must be satisfied:

$$\frac{N_\tau}{cF_s} > \max\left\{\|\mathbf{x}-\mathbf{y}\|_2 \mid \mathbf{x},\mathbf{y} \in \Omega\right\}. \tag{8.20}$$

This means that the length of the frame should at least allow for the sound waves generated by the equivalent sources to reach all of the microphones. In practice it is better to choose a longer frame in order to minimize the duration of the transient which should correspond only to a short initial part of the frame. If this is the case the effect of the transient will then be effectively canceled by the weighting and averaging operations of the WOLA procedure. The above inequality also suggests that the use of a microphone array scattered in a large volume should be avoided when using the ADeLFI algorithm. In the following, a frame length of $N_\tau = 512$ will be used, corresponding to time window of 64 ms at $F_s = 8$ kHz for all the microphone array configurations used in the simulations and real measurements of Section 8.5. This frame length always significantly exceeds the lower bound (8.20).

8.4.3 Tuning of parameter λ

The parameter λ, scaling the regularization term g in (8.8), controls the level of regularization and should be tuned properly. One of the most popular tuning strategies is the K-fold cross validation (KCV): this involves solving the optimization problem multiple times with different values of λ and K *folds* (partitions) of the available data. However, this strategy is not ideal for online audio processing: if N_λ candidate values for λ are used, it is then required to solve KN_λ optimization problems per frame.

Therefore, a novel simplification of the KCV strategy is adopted as follows. An additional microphone, referred to as *validation microphone*, is positioned inside the volume Ω to record the sound pressure $\tilde{\mathbf{p}}_v$ and validate the quality of the interpolation. For each frame, the optimization subproblem is solved multiple times using different values of λ. In Algorithm 5 the best λ is chosen in each frame inside the for-loop with counter z (Line 19). A low level of regularization is initially used, *i.e.*, λ is first chosen to be $10^{-6}\lambda_{\max}$, where $\lambda_{\max}$ is the value

for which $\mathbf{W}_\tau^\star = \mathbf{0}$ [33]. Hence the first optimization problem utilizes a low level of regularization and is initialized with a null initial guess. Once a solution is obtained, the *validation error* $\epsilon_{v,0}$ is computed (Line 21), namely the normalized distance between the validation microphone signal $\tilde{\mathbf{p}}_{\tau,v}$ in the current frame, weighted as well by the square-rooted Hann window, and the validation microphone sound pressure signal predicted by the acoustic model when regularized by λ. In practice $\epsilon_{v,0}$, gives a measure of the quality of the interpolation, which for small values of λ is expected to be large due to overfitting. The subproblem is then solved again after increasing λ logarithmically. This is also warm-started using the previous solution which helps in reducing the number of iterations of the current optimization problem. This procedure is stopped once the validation error stops decreasing, $\epsilon_{v,z} > \epsilon_{v,z-1} + \eta$, namely when the regularization ceases to be beneficial in terms of the quality of the interpolation. Here $\eta = 10^{-4}$ is a small value that prevents the procedure being stopped too early if $\epsilon_{v,z}$ and $\epsilon_{v,z-1}$ are very close. Finally the solution with the best regularization parameter is chosen, that is $\mathbf{W}_\tau^{\star,z-1}$ which was copied to $\bar{\mathbf{W}}_\tau^r$ during the previous iteration. The optimization algorithm solving the subproblems (Line 20) is stopped whenever the number of iterations exceeds 200.

8.4.4 DOA estimation and dereverberation

The last part of Algorithm 5 (starting from Line 27) consists of estimating the DOA and obtaining a dereverberated signal from the weight signals. As described in Section 8.3, the DOA can be inferred from the weight signal with strongest energy. However, the frame-based WOLA procedure offers the possibility of estimating DOAs in different time windows allowing the localization of a *moving sound source*.

After processing the r-th frame of the microphone signals to obtain $\bar{\mathbf{W}}_\tau^r$, the energy of each of its columns can be calculated and stored in a vector $\mathbf{e}^r \in \mathbb{R}^{N_w}$. The index of the maximum element of $\mathbf{e}^r$ will then correspond to the weight signal with strongest energy corresponding to the DOA at the r-th frame. However, it is possible to include the *memory* of the previous DOA estimates by performing a recursive averaging with forgetting factor β in order to give more weight to the latest estimates (Line 28). This will prevent abrupt changes of the DOA estimates. An index b will then be retrieved (Line 29), which can be used to obtain the azimuth and polar angles $\varphi_r^\star$ and $\theta_r^\star$ of the r-th frame DOA (Line 30). These angles are selected out of the candidate angles stored in $\breve{\boldsymbol{\varphi}} = [\breve{\varphi}_0, \ldots, \breve{\varphi}_{N_w-1},]^\intercal$ and $\breve{\boldsymbol{\theta}} = [\breve{\theta}_0, \ldots, \breve{\theta}_{N_w-1},]^\intercal$ which correspond to spherical coordinates with origin at the center of the microphone array, computed using the position of the equivalent sources (Line 10). This enables *tracking* the

DOA of a moving sound source, *i.e.*, creating the vectors $\boldsymbol{\varphi}^\star$ and $\boldsymbol{\theta}^\star$ which contain the estimated azimuth and polar angles of each frame, *i.e.*, $\varphi_r^\star$ and $\theta_r^\star$ for $r = 0, \ldots, N_r - 1$, respectively.

Once the b-th weight signal is chosen, this can be appended to the dereverberated signal $\bar{\mathbf{w}}$ (Line 31). Alternatively, as described in Section 8.3, the weight signals of the N_b nearest neighbors equivalent sources of the one corresponding to $\bar{\mathbf{w}}_{\tau,b}^r$ can also be used to produce dereverberated signals. The neighbor weight signals can be selected together with $\bar{\mathbf{w}}_{\tau,b}^r$ to construct $\bar{\mathbf{W}}_{\tau,b}^r \in \mathbb{R}^{N_\tau \times (N_b+1)}$ (Line 33). Once the acoustic model $\mathsf{D}^{b,r} : \mathbb{R}^{N_\tau \times (N_b+1)} \to \mathbb{R}^{N_\tau \times N_m}$ is constructed, the selected weight signals $\bar{\mathbf{W}}_{\tau,b}^r$ can be used to generate the sound pressure signals $\bar{\mathbf{P}}_{\tau,b}^r$ at the microphone position (or at any other positions inside Ω), corresponding to a new sound field with sound waves arriving only from a limited number of directions. Similarly to what is performed for $\bar{\mathbf{w}}$ in Line 31, dereverberated signals can then be obtained by appending $\bar{\mathbf{P}}_{\tau,b}^r$ to $\bar{\mathbf{P}}$ (Line 35). Once all of the frames are processed, the columns of $\bar{\mathbf{P}}$ will consist of N_t-long dereverberated signals, typically with less pronounced audio artifacts than those in $\bar{\mathbf{w}}$.

8.4.5 Frequency domain acoustic model

In Algorithm 5 it is assumed that a time domain acoustic model is used. For frequency domain acoustic models, minor modifications are needed. Specifically, a DFT must be applied to the columns of $\tilde{\mathbf{P}}_\tau$ after the analysis window function weighting (Line 17). Additionally, once the optimal weight signals $\bar{\mathbf{W}}_\tau^r$ of the r-th frame are obtained, these must be converted back into the time domain using an inverse DFT before the synthesis window function application (Line 26). The Hermitian symmetry of the frequency domain signals appearing in the columns of the matrices $\tilde{\mathbf{P}}_\tau$ and $\bar{\mathbf{W}}_\tau^r$ can readily be exploited to halve the number of optimization variables in the optimization problems. This implies that the number of optimization variables is equivalent to the one obtained when using a time domain model since every complex number consists of two optimization variables, corresponding to the real and imaginary parts.

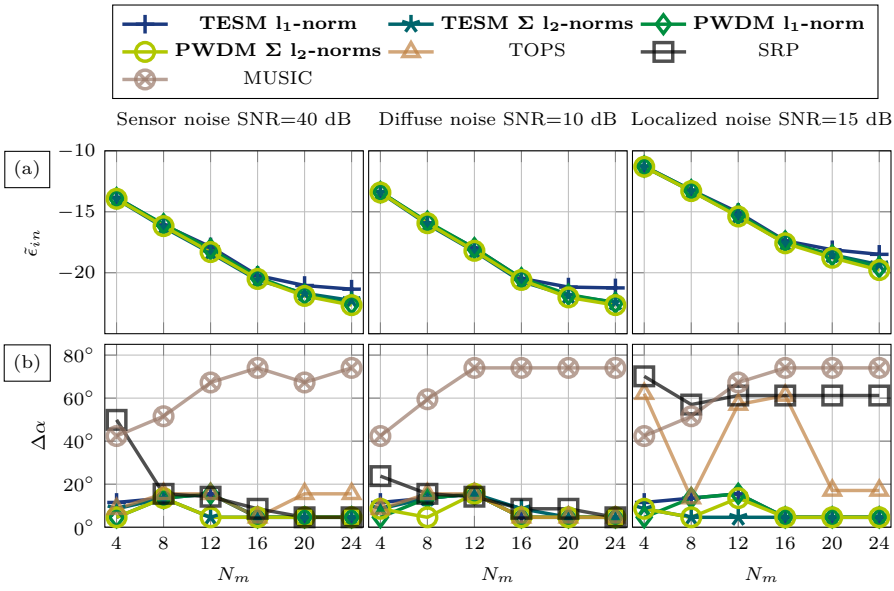


Figure 8.3: Mean of the validation error ϵ_v in dB (a) and angular distance $\Delta\alpha$ in degrees (b) for different types of acoustic models and regularizations of the ADeLFI algorithm (shown in the legend with bold fonts) as a function of the number of microphones (excluding the validation microphone). Notice that for MUSIC, TOPS and SRP-PHAT the validation microphone is included.

8.5 Results

8.5.1 Simulation results

In this section, results of simulations using the ADeLFI algorithm are presented. A common-size reverberant shoebox-shaped room with dimensions $[L_x, L_y, L_z] = [7.34, 8.09, 2.87]$ m and reverberation time of $T_{60} = 1$ s is modeled using the randomized image method (RIM) [196]. The sound source is placed in the front left corner of the room ($\mathbf{x}_s = [L_x/8,\ L_x/8,\ 1.6]$ m), and a sampling frequency of $F_s = 8$ kHz is used to prevent simulation being too long. An anechoic sound sample of 5.3 s of male speech from [215] is convolved with the RIRs to simulate the microphone signals. The microphones are positioned to form a spherical microphone array with a radius of 10 cm, centered at $\mathbf{x}_c = [4.4,\ 5.7,\ 1.4]$ m, position at which the validation microphone is also set. The geometry of the microphone array is chosen for convenience to form Fibonacci lattice with a radius that satisfies the properties discussed in

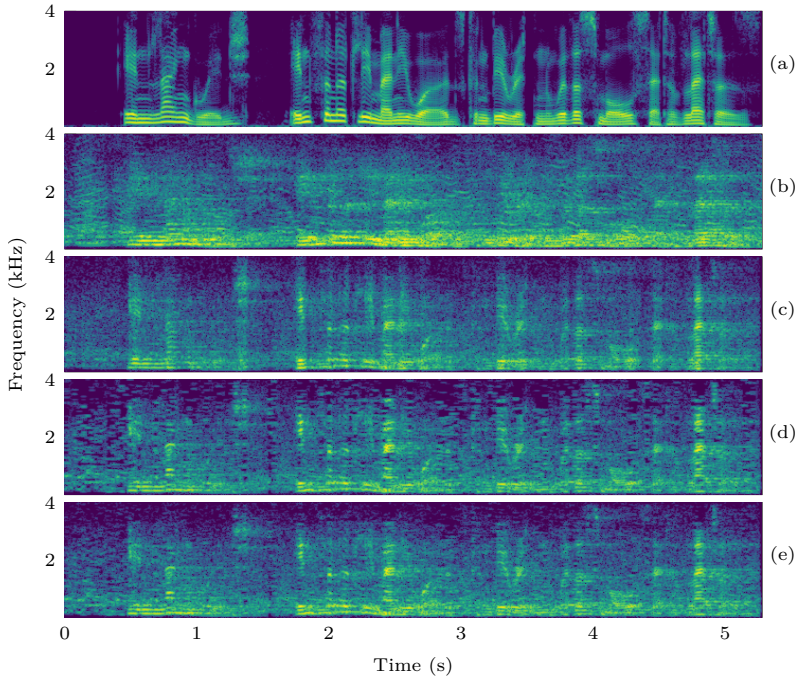


Figure 8.4: Spectrogram of anechoic speech signal (a), reverberant microphone signal (b), dereverberated signal obtained through TESM with sum of l_2-norms regularization (c), and through PWDM with l_1-norm regularization with single component ($\bar{\mathbf{w}}$) (d), and multiple components ($N_b + 1 = 12$) (e) using $N_m = 20$ microphones with diffuse babble noise (SNR = 10dB).

Section 8.4.2. The equivalent sources are also centered at $\mathbf{x}_c$ with a radius of 3 m. The radius of the equivalent sources was chosen in order for the sound waves to have a limited spherical behavior in Ω, in practice resulting very similar to plane waves. Since it is assumed that the sound source is in the far-field, positioning the equivalent sources too close to the microphone array would reduce the spatial sparsity assumption of the weight signals: the resulting optimal weight signals would implicitly apply Huygens-Fresnel principle and many equivalent sources would be simultaneously active to represent a far field sound wave. Here, $N_w = 500$ equivalent sources are used in the acoustic models. The number of equivalent sources is chosen empirically: a lower number leads to a reduction of the performances and a higher number does not particularly change the results while increasing the computational load of the algorithms. Three different scenarios are compared: (i) sensor noise only, (ii) diffuse babble

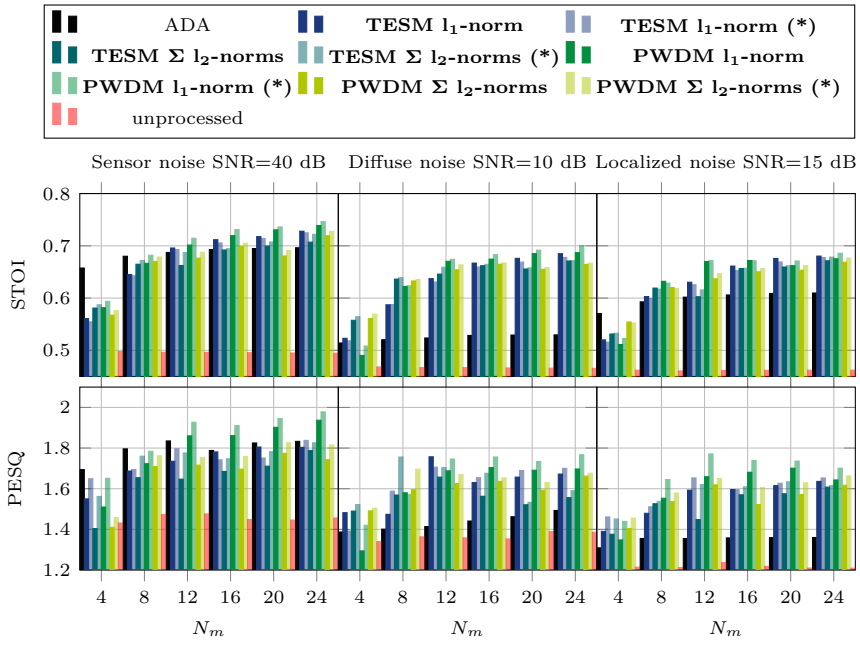


Figure 8.5: STOI and PESQ scores (RAW) for different types of acoustic models and regularizations of ADeLFI (shown in the legend with bold fonts) as a function of the number of microphones (excluding the validation microphone). Results with (*) correspond to dereverberated signals generated using multiple weight signals ($N_b + 1 = 12$).

noise generated using the technique proposed in [216] with a SNR of 10 dB (iii) localized noise from a white source signal placed at $[7/8L_x,\ L_x/8,\ 1.6]$ m with a SNR of 15 dB. Spatially incoherent white noise is added with a SNR of 40 dB to simulate sensor noise in all cases. The validation microphone signal is also corrupted by these types of noise. Here, since only static sound sources are used to simulate the microphone signals, a forgetting factor of $\beta = 1$ is employed.

Figure 8.3(a) shows the mean of the validation errors obtained in each frame:

$$\tilde{\epsilon}_v = \frac{1}{N_r} \sum_{r=0}^{N_r-1} \frac{\|\mathbf{p}_{v,r} - \tilde{\mathbf{p}}_{v,r}\|_2^2}{\|\tilde{\mathbf{p}}_{v,r}\|_2^2}, \tag{8.21}$$

where $\tilde{\mathbf{p}}_{v,r}$ and $\mathbf{p}_{v,r}$ are the r-th frame of the sound pressure signal measured by the validation microphones and predicted by the acoustic model, respectively. Almost identical validation errors are achieved using the PWDM with either

sum of l_2-norms or l_1-norm regularizations and using the TESM with sum of l_2-norms regularization. Slightly worse results are obtained using the TESM with l_1-norm regularization. This algorithm, which has been shown to have good performance for the task of RIR interpolation [33], has poorer performance in this context because the sound field is not generated by a temporally sparse source. The performance is slightly worse in the case of diffuse babble noise and decreases for the case of localized noise. In these scenarios the presence of diffuse and localized noise increases the number of active equivalent sources and makes the sparsity-inducing regularization less effective.

Remarkably, even if the diffuse babble noise has lower SNR than the localized noise, better performance is achieved in the former case. This is due to the different nature of the noise. The diffuse babble noise is highly spatially correlated at low frequencies where most of its energy lies. ADeLFI seems capable of effectively interpolating this diffuse noise field as proven by the low normalized mean squared error (NMSE) shown in Figure 8.3(a). On the other hand, the localized noise is white meaning it generates a full band diffuse sound field in such reverberant environment. At high frequencies the noise is spatially uncorrelated making it more difficult to interpolate this sound field due to the lack of spatial correlation.

As the lower plots of Figure 8.3(b) show, all the ADeLFI configurations achieve good localization even when only 4 microphones are used, with spatio-temporal sparsity giving the lower localization accuracy. Here the minimum angular distance is 4.5°, which is due to the finite number of equivalent sources and, as a consequence, of directions. The localization performance is also compared with well established localization algorithms, namely the MUltiple SIgnal Classification (MUSIC) [38], test of orthogonality of projected subspaces (TOPS) [40] and steered response power-phase transform (SRP-PHAT) [41] the implementation of which is found in [217]. For a fair comparison, these algorithms use the same set of candidate directions given by the Fibonacci lattice used in ADeLFI. While MUSIC fails to retrieve the correct DOA due to the highly reverberant environment, TOPS and SRP-PHAT both achieve good localization in the sensor noise and diffuse babble noise scenario, achieving similar performance as ADeLFI. However, in the presence of localized white noise they are outperformed by all of the different configurations of ADeLFI. In fact in this noise scenario, both TOPS and SRP-PHAT often identify either the DOA of the noise source instead of the DOA of the speech source or something in between. Once more the localized white noise scenario is the most difficult one but ADeLFI proves itself robust achieving almost the same performance as in the other scenarios.

Figure 8.4 compares the spectrogram of the dereverberated signals produced by ADeLFI with the original anechoic speech signal and one of the microphone

signals. In the latter the presence of the babble noise can be seen as well as the speech components being smeared out by reverberation. Both reverberation and noise are effectively reduced by ADeLFI. Figure 8.4 (c) and (d) show the spectrogram of the dereverberated signal produced when using the TESM with spatial sparsity and the PWDM with spatio-spectral sparsity respectively. In the latter figure spectral sparsity is particularly visible at higher frequencies. This effect results in audible artifacts, *i.e.*, the presence of musical noise in the dereverberated signal. This is effectively reduced when multiple weight signals are used as show in Figure 8.4 (e) showing the spectrogram produced at the same microphone position using $N_b + 1 = 12$ weight signals with the PWDM combined with spatio-spectral sparsity. A clear reduction of the spectral sparsity is seen which leads to a substantial reduction of the musical noise.

Figure 8.5 shows speech intelligibility and quality scores obtained using the short-time objective intelligibility (STOI) [218] and perceptual evaluation of speech quality (PESQ) [163] respectively. These are in line with the results of Figure 8.3 with scores increasing with the number of microphones. It can be seen that the PWDM with spatio-spectral sparsity achieves the best performances in particular when multiple weight signals are used. Here a comparison with sound samples dereverberated using the adaptive sparse MCLP-based speech dereverberation (ADA) [219] algorithm are also reported. This is a state-of-the-art dereverberation algorithm based on MCLP that does not require any prior knowledge on the DOA of the desired source and on the acoustic properties of the room. Here, ADA utilizes a forgetting factor of 1. ADA outperforms ADeLFI in the case of the sensor noise scenario reaching high scores with only 4 microphones. However, the performance of ADA deteriorates significantly in the other scenarios, particularly in the case of diffuse babble noise. On the contrary, ADeLFI is more robust and capable of performing *noise reduction* too: since the algorithm aims at interpolating the diffuse noise field as well, it evenly distributes the noise energy among the active equivalent sources and hence selecting the weight signal with the strongest energy generally leads to a SNR increase. This can be seen in Figure 8.4 when comparing the spectrogram of the microphone signal (b) with the dereverberated signal produced by ADeLFI (b-e).

Informal listening tests indicate that dereverberated signals obtained by adopting the spatio-temporal sparsity based regularization have many more artifacts than those obtained with spatial sparsity or spatio-spectral sparsity based regularization, with the latter having less audible distortions. The dereverberation effect increases as more microphone are used. When listening to the dereverberated signals, it is evident that the noise is reduced when compared to the microphone signals. Audio samples can be found in [220].

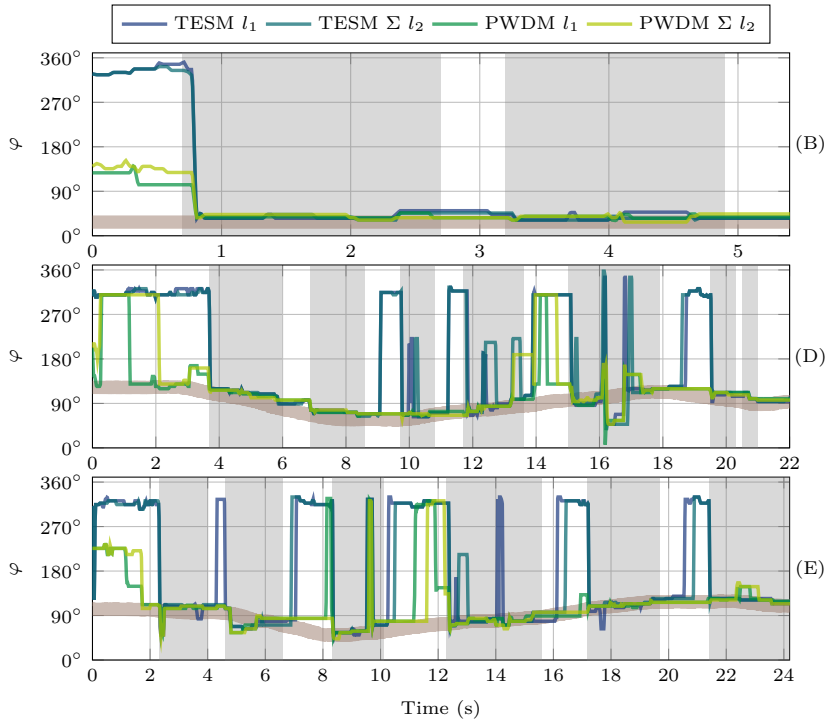


Figure 8.6: Estimated azimuth angle $\varphi^{\star}$ using measurements. Thick line indicates the ground truth. Gray areas visualize the time windows with voice activity.

8.5.2 Results using real measurements

In this section the performance of the ADeLFI algorithm is validated using real measurements. The measurements are taken from the LOCATA challenge development database [221]. This database provides different recordings taken in a reverberant room. Here 5 different scenarios are tested: three scenarios with a static source (denoted in the figures and tables with (A), (B) and (C) and corresponding to recordings 1, 2 and 3 respectively of Task 1 of the LOCATA database) and two scenarios with a moving sound source (denoted (D) and (E) corresponding to recordings 1 and 3 respectively of Task 3 of the LOCATA database). For the static source scenarios a loudspeaker source was used playing speech signals from the CSTR VCTK database [222]. The moving sound sources were created by people talking while walking around the room. The ground-truth positions of the speakers were measured using

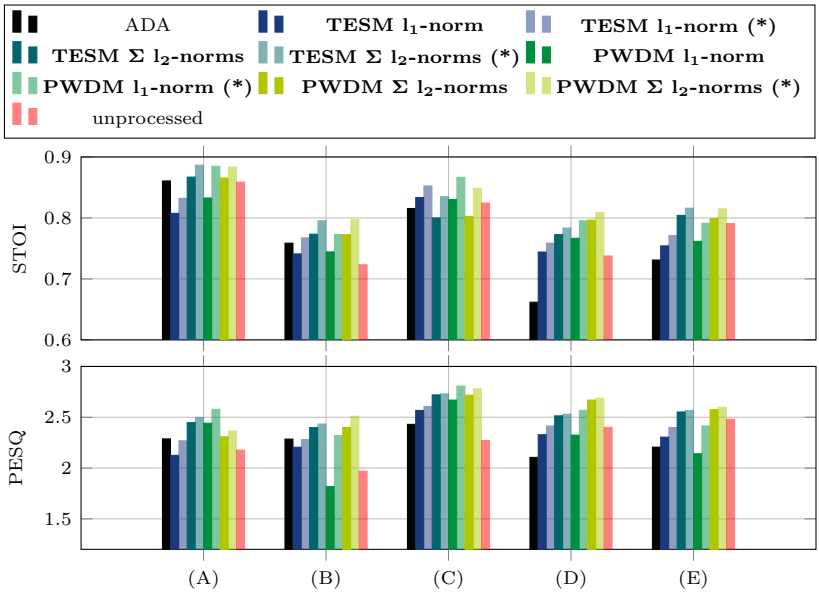


Figure 8.7: STOI and PESQ scores (RAW) for different types of acoustic models and regularizations of ADeLFI (shown in the legend with bold fonts) using measured data with a single static (a-c) and moving sound source (d-e). Results with (*) correspond to dereverberated signals generated using multiple weight signals ($N_b + 1 = 12$).

an optical tracking system. All of the results presented here were obtained using a spherical microphone array of 32 microphones mounted on a rigid sphere with a radius of 4.2 cm (Eigenmike) [223]. Out of these recordings only $N_m = 15$ microphones are used in the ADeLFI algorithm. Two additional microphones are used as validation microphones. Remarkably, although the ADeLFI algorithm assumes omnidirectional microphones, these measurements still lead to satisfactory results as will be shown. A sampling frequency of $F_s = 8$ kHz was used. The microphone recordings contain measurement noise and in some cases traffic noise as well coming from outside of the building (particularly in scenarios (B) and (D)). The forgetting factor is empirically chosen to be $\beta = 0.9$ for all the results presented here, including the static source scenarios. Here, $N_w = 500$ equivalent sources are placed on a Fibonacci lattice with radius of 2 m.

Figure 8.6 shows the estimated azimuth angle $\varphi^\star$ as a function of time. The ground truth is visualized using a thick line. The grey areas in the plots visualize

	(A)	**(B)**	**(C)**	**(D)**	**(E)**
TESM l_1	7.7°	53.2°	10.3°	19.1°	30.9°
TESM Σl_2	7.7°	47.5°	10.3°	16.6°	18.3°
PWDM l_1	**1.5**°	36.6°	10.3°	**14.6**°	**17.2**°
PWDM Σl_2	11.6°	**18.2**°	10.3°	15.4°	18.3°

Table 8.1: Median of the angular distance between the estimated DOA and the ground truth during voice activity time windows for different scenarios of the LOCATA challenge and different types of regularizations and acoustic models.

the time windows where voice activity is present. For the static scenarios, only case (B) is shown for brevity. It can be seen that, as soon as voice activity begins, all of the various configurations of ADeLFI succeed in finding a good estimate of the azimuth angle with similar performance. Figure 8.6 (D-E) shows the case of the moving sound sources where it is seen that the estimated azimuth angle is successfully tracked within the time windows, with few exceptions. Similar results can be observed for the elevation angles and are not reported here for brevity. Instead, Table 8.1 summarizes the median angular distances between the estimated DOAs and the ground truth DOAs in the time windows with voice activity. As it can be seen the PWDM with spatio-spectral sparsity based regularization achieves the most accurate estimates, although it is sometimes almost equaled or surpassed by the PWDM with spatial sparsity based regularization.

Finally, Figure 8.7 shows the STOI and PESQ scores of the dereverberated signals obtained with ADeLFI. The reference signals used to compute these measures are the semi-anechoic speech signals used to drive the loudspeakers for the static sources scenarios while for the moving speaker scenarios the recordings of a microphone near the mouth are used. In most of the cases, ADeLFI improves both the audio quality and the speech intelligibility, with visible improvements when multiple weight signals are used. These results are once more compared with ADA: the scores of the dereverberated signals of ADA are often outperformed by ADeLFI, particularly in the moving source scenarios. Notice that here the forgetting factor of ADA is set to 0.99 and the validation microphones are included in the processing. As for the simulation results of the previous Section, in many cases the best results are achieved using ADeLFI with PWDM in combination with spatio-spectral sparsity based regularization, although PWDM with spatial sparsity based regularization often outperforms this, especially for the moving source scenarios. Sound samples can be found in [220].

8.6 Conclusions

In this paper a novel algorithm for joint source localization and dereverberation has been proposed. This algorithm relies on interpolating the sound field using the measurements of a set of microphones and solving an inverse problem that employs a particular acoustic model. The inverse problem is solved using an accelerated variant of the PG algorithm using matrix-free optimization and a WOLA procedure in order to obtain the weight signals that control the sound waves which effectively interpolate the sound field. The use of two acoustic models has been investigated: the TESM and the PWDM. The inverse problem is regularized using sparsity-promoting regularization and, depending on the choice of the acoustic model, spatial, spatio-temporal and spatio-spectral sparsity can be promoted in the weight signals. The level of regularization is tuned by comparing the interpolated sound field with the sound field measured by an additional microphone. It has been shown that, by finding the weight signal with strongest energy during different time windows, a moving sound source can be localized in terms of DOA. The same weight signal, together with its neighbors can then also be used for a dereverberation task. Simulations have shown that DOA estimation can be achieved using relatively few microphones ($N_m \geq 4$) when a speech source generates the sound field and that spatial and spatio-spectral sparsity based regularizations outperform spatio-temporal sparsity based regularizations in terms of both interpolation quality and dereverberation. The proposed algorithm is shown to be robust against different types of noise using both simulated and real measurements. Compared with state-of-the-art algorithms for both DOA estimation and dereverberation, the algorithm shows competitive performance and additionally provides noise reduction in the dereverberated signals. A main drawback of the proposed algorithm is its computational complexity. For example, using $N_m = 12$ microphones and $N_w = 500$ equivalent sources a real-time factor of 3171 and 968 is reached with PWDM (spatio-spectral sparsity) and TESM (spatial sparsity) respectively using a single core on a Intel Core™ i7-3740QM 2.70 GHz computer. However, many of the numerical operations can be performed in parallel using fast transformations. Future research will focus on the reduction of the computational burden and on the extension of the algorithm under more complex scenarios including the localization of multiple sound sources and the use of moving microphones.

Chapter 9

Conclusion

This thesis explored the use of different physical acoustic models in a variety of audio signal processing tasks such as source localization, sound field acquisition/reproduction and dereverberation. It was demonstrated that, in acoustically challenging environments such as reverberant rooms, the use of acoustic models in audio signal processing methods is not only interesting for the enhanced robustness against noise and reverberation but also for the new perspectives these methods can offer, such as joint formulations of different tasks that are usually addressed separately. In this framework, these applications naturally led to non-trivial ill-posed inverse problems. Sparse regularization was shown to be crucial for the reach of meaningful results, that could be achieved through the solution of challenging large-scale nonsmooth optimization problems.

The novel framework this thesis explored is the result of the multidisciplinary combination of scientific fields such as room acoustic modeling and numerical optimization. While room acoustic modeling, together with sparse regularization, allows for the formulation of the inverse problems, numerical optimization provides a series of tools for the solution of these inverse problems. These scientific fields present different research objectives on their own and for this reason they were initially treated separately in the first two parts of the thesis. In the following two parts of the thesis, audio signal processing methods that employ both acoustic models and optimization algorithms were developed. Here, two types of acoustic models, *i.e.*, the discretized partial differential equation (PDE) models and the wave decomposition models, were used to formulate and solve inverse problems leading to new methods with different features and results.

In the following, the main results obtained in each part of the thesis are briefly summarized. Finally, suggestions for future research conclude the thesis.

Optimization algorithms: Proximal Gradient algorithms

Many of the inverse problems tackled in this thesis consisted of optimization problems of *large scale*, due to the large quantities of parameters that need to be estimated, and with *nonsmooth cost functions*, due to the presence of sparse regularization. Chapter 2 showed how the *proximal gradient (PG) algorithm* and its enhanced variants are suited not only for these types of problems but also for a large variety of optimization problems including constrained and nonconvex ones. It was described how the PG algorithm iterations can be cheap both in terms of memory and computational power, since they avoid matrix factorizations and can instead make use of fast algorithms for the computation of the mappings involved, leading to *matrix-free optimization*. Additionally, a recently proposed variant of the PG algorithm that utilizes quasi-Newton methods to enhance its convergence rate was reviewed. It was shown that these optimization algorithms can be applied to a large set of optimization problems without altering their original structure, thus avoiding any problem reformulation which typically leads to a further dimensionality increase, often necessary *e.g.*, when using interior point methods. An open-source software package that implements these ideas and algorithms was developed. Examples of the simple interface it offers, which follows the typical mathematical formulation of optimization problems, were given. A brief tutorial guide of this software can be found in Appendix A as well. This software was used in Chapters 7 and 8 to efficiently solve the inverse problems under consideration.

Acoustic modeling: sweeping echoes and the image method

Due to their simplicity, rooms with rectangular geometries are often used in room acoustic simulations. However, as it was described in Chapter 3, an acoustic phenomenon called *sweeping echoes* occurs in simulations of perfectly rectangular rooms. The perfect symmetry between the walls, which is typically not present in regular rooms, was proven to be the cause of this acoustic phenomenon which was shown to exist in the image method (IM), in the finite-difference time-domain (FDTD) method and in the analytical solution of the wave equation. Since the most efficient and popular implementation of the IM requires a cuboid geometry of the room, a simple variant, called *the randomized image method (RIM)*, was proposed. A small perturbation of the image source positions was introduced which breaks the symmetry of the perfectly rectangular room causing sweeping echoes to disappear. An open-source software package

was developed to implement the RIM and a quick tutorial can be found in Appendix B. This software was used in Chapters 7 and 8 to generate realistic room impulse responses (RIRs) used in the simulations.

Inverse problems with the FDTD method

The use of the *finite-difference time-domain (FDTD) method*, which aims at numerically solving the wave equation by discretizing it in space and time, was envisaged in a number of audio signal processing tasks involving reverberant rooms. Since the proposed methods addressing these tasks assumed the knowledge of the geometry and of the acoustic impedance modeling the boundary reflections, the problem of *acoustic impedance estimation* was treated in Chapter 4. Here, a novel method for estimation of acoustic impedances across the whole wall surfaces was proposed. This method relied on a inverse problem which matched the FDTD method with measurements taken in a room. Simulations showed that by regularizing the inverse problem with the additional knowledge that the acoustic impedances were *spatially smooth* across the wall surfaces it was possible to estimate them in a regular grid of points on the walls.

Similarly, but with the objective of performing *source localization*, in Chapter 5 the recordings of some microphones scattered inside a reverberant room were matched to the sound pressure predicted by the FDTD method. This was achieved by solving an inverse problem that could lead to the estimation of the source signals, each corresponding to a a particular position of the room, capable of describing the microphone measurements. A sparse regularization promoting *spatio-temporal sparsity* of the sound sources was used to regularize this inverse problem. Although the estimation of the source signals was partially incorrect, it was shown that it could lead to the coordinate positions of the original source. This enabled a more accurate reconstruction of the original source signal, *i.e.*, to perform dereverberation.

A similar inverse problem, but with a different aim, was studied in Chapter 6. Here, the sound field described by the FDTD method was matched to particular sound pressure signals at specific positions in order to control a set of loudspeakers and hence reproduce a sound field inside a room. A *multi-zone sound field control* task was performed, since the sound field was kept silent in some areas of the room during the sound field reproduction task. Spatial sparsity was promoted to regularize the underlying inverse problem. This guaranteed to find the optimal positions of the active loudspeakers while simultaneously returning their control signals. It was shown that in a highly reverberant room, anechoic sounds could be reproduced and that the loudspeakers were capable

of behaving both as "sources" and "sinks" of the sound field while exploiting reverberation of the room.

Inverse problems with the wave decomposition models

The use of *wave decomposition models* was also envisaged in a number of audio signal processing tasks. Contrary to the FDTD method, these acoustic models allow for a description of the acoustic phenomenon only in a limited volume of space but do not require the knowledge of neither the room geometry nor of the acoustic impedances of the walls. Chapter 7 utilized these acoustic models to perform *RIR interpolation*. A RIR can be viewed as the measurement at one particular position of a sound field generated by a single impulsive source. Consequently, wave decomposition models could be used to decompose this particular sound field recorded by a microphone array into a summation of waves arriving from different directions due to the wall reflections. Here two different wave decomposition models were used: the plane wave decomposition method (PWDM) and the time-domain equivalent source method (TESM) which are a frequency domain model consisting of a dictionary of plane waves and a time domain model consisting of dictionary of spherical waves, respectively. These models were both matched to the measured RIRs in order to interpolate them within a volume surrounded by the microphone array. This was achieved by solving an inverse problem. Different types of sparse regularization were compared: spatio-temporal sparsity, spatio-spectral sparsity and spatial sparsity. The level of regularization was tuned using K-fold cross validation (KCV), a popular strategy often used in machine learning. It was demonstrated that since a single sound source producing a single impulse would ideally produce a RIR, spatio-temporal sparsity based regularization, that could be easily imposed using the TESM, gave the best interpolation performance using a limited number of microphones. Additionally, if the impulse response (IR) of the loudspeaker was available, even better results could be achieved. This was verified using both simulated and real measurements.

Chapter 8 extended the ideas of Chapter 7 by interpolating a sound field generated by a speech source in a reverberant room. It was shown that once this interpolation was achieved, joint *source localization* in terms of direction of arrival (DOA) and *dereverberation* could be readily performed. As a matter of fact, the wave decomposition performed by the interpolation gives estimates of signals that *weight* the sound waves coming from specific directions with respect to the microphone array. By inspecting the energy of these weight signals, it was shown that the weight signal with maximum energy corresponded to the DOA of the original sound source. Since reverberation was spatially distributed among the weight signals, the isolation of the weight signal with

strongest energy, or the combination of weight signals corresponding to DOAs close to the estimated DOA, contained lower amounts of reverberation readily leading to dereverberated signals. Processing microphone signals with long duration would have resulted in a computationally intractable optimization problem. A weighted overlap-add (WOLA) strategy was employed that splits the original problem into smaller subproblems and serially processes the microphone signals using frames. Here, at each frame, an additional microphone was used to tune the regularization level. Furthermore, the WOLA procedure offered the possibility to track the DOA of a moving sound source. It was described that since the sound field was generated by speech, *i.e.*, a spectrally sparse sound source, spatio-spectral and spatial sparsity based regularization resulted in the best performance both in terms of localization and audio quality of the dereverberated signals. The proposed method was validated using both simulated and measured microphone signals exhibiting robustness against different types of noise.

9.1 Suggestions for future research

The nature of the acoustic models can have crucial consequences in the performance of the signal processing methods that employ them. In this thesis two fundamentally different types of acoustic models were employed in various audio signal processing applications. Here, the drawbacks and advantages of these acoustic models are discussed together with their industrial and commercial potential. Finally, new ideas for future research are proposed.

Discretized PDE models

Discretized PDE models, such as the FDTD method, are theoretically capable of precisely describing the sound field of an entire room. Therefore, audio signal processing methods that utilize these models naturally employ microphones scattered inside the room. This allows these methods to achieve remarkable results such as the precise localization of sound sources in terms of coordinates even when there is no line of sight between the microphones and the sound sources [44] . The scattered set-up of the microphones leads to question whether these methods could benefit from using wireless acoustic sensor networks (WASNs) which typically require challenging distributed processing with limited communication bandwidth and power consumption. Many of the audio signal processing methods specifically designed for WASN rely on parametric models; different tasks have been addressed such as channel equalization [224], noise reduction [225] and source localization [226], with joint formulations as well [227].

The use of discretized PDE models in the context of WASN is a research path that could be pursued.

However, there are different research objectives that still need to be properly explored, independently from the choice of distributed or centralized processing. The preliminary studies of Part III mainly focused on the non-trivial inverse problem formulations. In fact, the inverse problems solved in Part III committed what is known as *the inverse crime* [228], namely the acoustic model inversion was performed using data generated by the same model. Indeed, there are a number of reasons which makes it very difficult to apply these methods in a real scenario. Firstly, the discretization performed by these methods is limited to a set of positions, meaning that if a microphone or sound source is *off-grid*, *i.e.*, their true positions do not coincide precisely with a grid point imposed by the acoustic model, there might be consequences. Interpolations between neighbor grid points could be performed but these could lead to a further increase of the numerical errors or complications of the inverse problem formulation. The use of numerical methods that allow for *adaptive meshes* could be envisaged [229]. Secondly, the numerical errors of many discretized PDE models *limit the modeling to low frequencies only.* This represents a big issue not only for computational reasons, but also for ill-conditioning of the inverse problems, particularly for time domain methods that inherently model a wide frequency band. Since numerical errors mainly affect high frequencies, this implies that the fitting involved in the inverse problems should take place only at low frequencies where the method is most accurate. As a consequence of this, a large bandwidth of the model should not be used due to its incorrectness. A solution to this would be to employ *model reduction techniques* [230] that could effectively remove the high-frequency erroneous part of the model and consecutively improve the ill-conditioning of the inverse problems as well. Finally, discretized PDE models typically assume *time invariance.* A solution would consist in updating the acoustic models continuously, ideally by achieving real-time estimation of geometry and acoustic impedances which, as described in Chapter 4, is a non-trivial problem in itself. This is what is typically done in model predictive control (MPC) with relatively simpler mechanical models usually described by ordinary differential equations (ODEs) [231]. In the context of discretized PDE models, these estimations would be extremely difficult tasks to accomplish, not only computationally but also in terms of the information required. In fact, performing this solely relying on acoustic measurements would probably require large amounts of microphones distributed in many areas of the room. A possibility to avoid such unpractical set-up could be to exploit other types of physical phenomena in addition to the acoustic one, solving *multi-modal inverse problems* for example employing optics and different sensors such as cameras or lasers.

Wave decomposition models

The use of wave decomposition models leads to novel methods, the application of which in a real scenario is already possible, as proven by the results of Part IV. The main advantages of wave decomposition models over discretized PDE models is their ability of describing the acoustic phenomenon without knowledge of the geometry and acoustic impedances. Additionally, since these models make use of analytical homogeneous or particular solutions of the wave or Helmholtz equations, they are not affected neither by numerical errors, offering the possibility of modeling full-band sound fields, nor by off-grid microphone positions, since they describe the sound field continuously. This comes at the cost of limiting the acoustic modeling only to a relatively small volume of space and not in the entire room. As a consequence of this, the microphones are usually not scattered around the room but are spatially distributed more compactly in the particular area of interest. Still, the use of these models in audio signal processing tasks requires a large number of parameters to be estimated resulting in computationally intensive methods as well.

While there are studies regarding the PWDM and the number of plane waves that one should employ to properly describe the sound field in a volume [34, 187], the positioning and the number of equivalent sources of the TESM was chosen empirically in Part IV. This choice remains an open question: an optimal placement of the equivalent sources could lead to better results and finding the minimum number of equivalent sources needed to describe a specific volume could lead to a smaller scale optimization problem. Additionally, the optimal placement of the microphones should be investigated as well. The methods proposed in Part IV require a precise knowledge of the geometry of the microphone array. It remains to be studied how errors in the relative positions of the microphones affect these methods and whether techniques for the estimation of the microphone positions could be pursued in this framework.

Research valorization perspectives

Industrial and commercial exploitation of the work presented in this thesis is envisaged in a variety of applications. Acoustic modeling plays a fundamental role in virtual/augmented reality, gaming and architectural acoustics. Source localization and dereverberation are crucial in hearing assistance, human-computer interaction, sound scene analysis for smart homes/cities and voice control.

Being computationally intensive, many of the audio signal processing methods described in this thesis are currently out of the reach of a real-time

implementation. However, they are still relevant in many applications that can be performed off-line. For example, impedance estimation and sound field reconstruction methods, could effectively create acoustic models that have more realistic features since their parameters are obtained from real measurements. Similarly, the RIR interpolation results can be readily employed by various well-established methods.

As it was discussed earlier, there is a clear distinction between methods that utilize PDE discretized models (Part III) and wave decomposition models (Part IV). The former are still not applicable in a real scenario while the latter have been already validated using real measurements. In both cases, one of the main computational bottleneck is due to the acoustic models. The parallel implementation of the acoustic models using Graphics Processing Units (GPUs) or dedicated hardware could significantly improve the computational performance of these methods. Additionally, wave decomposition models could be implemented using fast transformations such as the nonequispaced fast Fourier transform (NFFT) [211] or wavelet transforms [232]. Therefore, a real-time implementation of the methods that employ wave decomposition models (Part IV) is not too far-fetched. Dedicated high-performance hardware should be employed to reach such result. However, it is expected that in the future, due to the advances in research and an increase of computational power, the implementation of these methods in large-volume low-cost processing hardware will be feasible.

Finally, two open-source software packages were released. These are briefly described in Appendix A and Appendix B and were written in Julia [108]. Julia is a new open-source high-performance programming language; it was released in 2012 and since then it has a fast growing user community. Unfortunately, Julia code cannot be employed in embedded systems yet, although this might become possible in the near future. Currently, however, Julia and its software packages can be used in single-board computers such as *e.g.*, Raspberry Pi.

New acoustic models and other applications

In the introduction of the thesis it was described how wave decomposition models can be viewed as a trade-off between parametric models and discretized PDE models. Indeed, other trade-offs using different combinations of acoustic models are possible and should be explored. For example, it would be interesting to combine reflection path models, such as ray tracing methods or scattering delay networks (SDNs) with parametric models or with discretized PDE models. Statistical models of room acoustics, which were not treated here at all, could also be included into these combinations.

Although the acoustic phenomenon occurring inside rooms is linear, the use of nonlinear models have demonstrated to produce astonishing results particularly in the context of machine learning [128]. The development of new and more accurate room acoustic models that are nonlinear in their parameters could effectively lead to a reduction of parameters that need to be estimated, at the cost of coping with nonconvex optimization problems instead of convex ones. Here, an interesting parallelism with parametric models can be made. RIRs can be represented using either finite impulse responses (FIRs) or infinite impulse responses (IIRs) filters: the former have more parameters than the latter ones. However, the estimation of IIR parameters is usually more involved. Similarly, developing new acoustic models, perhaps exploiting some physical motivations such as the concept of common acoustic poles [8], is a research path that could be pursued. Recent advances in nonconvex optimization [100] could make it possible to employ these acoustic models in many inverse problems tackling particular applications.

Recently, many audio signal processing tasks are addressed using purely data-driven black-box methods developed in the field of machine learning. One advantage of using acoustic models over this methods consists in avoiding the requirement of large data-sets that are often not available, particularly in the context of room acoustics. A detailed comparison between these two different approaches would be very interesting while their combination is surely a valid research path to investigate.

Finally, as it was described in this thesis, the employment of acoustic models together with sparse regularization can lead to the development of novel, more robust signal processing methods often capable of performing multiple tasks at once. There remains a number of audio signal processing tasks where the use of these and others acoustic models and sparse regularization is still unexplored or is only at an early stage of research such as *e.g.*, source separation, active noise control and acoustic zooming. Optimal placement of microphones and loudspeakers in sound field reconstruction and reproduction, respectively, is still an open problem. The use of moving microphones is also a relatively new idea that should be further explored and where compressed sensing (CS) techniques could bring new results [184].

Appendix A

StructuredOptimization.jl

`StructuredOptimization` : a quick tutorial guide

Niccolò Antonello, Lorenzo Stella

Published online at `https://kul-forbes.github.io/StructuredOptimization.jl/` commit 5e17e43

The candidate's contributions include: software co-development, design of the tutorial guide, text redaction and co-editing.

Abstract

`StructuredOptimization` is a high-level modeling language that utilizes a syntax that is very close to the mathematical formulation of an optimization problem.

This user-friendly interface acts as a parser to utilize three different packages:

- `ProximalOperators` provides proximal mappings of functions that are frequently used in signal processing and optimization.
- `AbstractOperators` provides algorithms for the evaluation and combination of forward and (Jacobian) adjoint of linear and nonlinear mappings.
- `ProximalAlgorithms` is a library of proximal algorithms (aka splitting algorithms) solvers.

`StructuredOptimization` can handle large-scale convex and nonconvex problems with nonsmooth cost functions. It supports complex variables as well.

A.1 Installation

From the Julia command line hit `Pkg.clone("https://github.com/kul-forbes/StructuredOptimization.jl.git")`. Once the package is installed it can be updated along with the others issuing `Pkg.update()` in the command line.

A.2 Standard problem formulation

Currently with `StructuredOptimization` one can solve problems of the form

$$\underset{\mathbf{x}}{\text{minimize}}\ f(\mathbf{x}) + g(\mathbf{x}), \tag{A.1}$$

where f is a smooth function while g is possibly nonsmooth.

A.2.1 Unconstrained optimization

The least absolute shrinkage and selection operator (LASSO) belongs to this class of problems:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \underbrace{\tfrac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2}_{f(\mathbf{x})} + \underbrace{\lambda\|\mathbf{x}\|_1}_{g(\mathbf{x})}. \tag{A.2}$$

Here the squared norm $\frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|^2$ is a *smooth* function f whereas the l_1-norm is a *nonsmooth* function g. This problem can be solved with only few lines of code:

```
julia> using StructuredOptimization

julia> n, m = 100, 10;                  # define problem size

julia> A, y = randn(m,n), randn(m);     # random problem data

julia> x = Variable(n);  # initialize optimization variable

julia> lambda = 1e-2*norm(A'*y,Inf);  # define lambda

julia> @minimize ls( A*x - y ) + lambda*norm(x, 1); # solve problem

julia> ~x                               # inspect solution
100-element Array{Float64,1}:
  0.0
  0.0
  0.0
  0.440254
  0.0
  0.0
  0.0
[...]
```

Note: the function `ls` is a short hand notation for `0.5*norm(...)^2`, namely a least-squares term.

It is possible to access the solution by typing `~x`. By default, variables are initialized by `Array`s of zeros. Different initializations can be set during construction `x = Variable( [1.; 0.; ...] )` or by assignment `~x .= [1.; 0.; ...]`.

A.2.2 Constrained optimization

Constrained optimization is also encompassed by the standard problem formulation (A.1): for a nonempty set $\mathscr{S}$ the constraint of

$$\underset{\mathbf{x}}{\text{minimize}}\ f(\mathbf{x})$$

$$\text{subject to } \mathbf{x} \in \mathscr{S}$$

can be converted into the standard form (A.1) with an *indicator function*

$$g(\mathbf{x}) = \delta_{\mathcal{S}}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \mathcal{S}, \\ +\infty & \text{otherwise.} \end{cases} \tag{A.3}$$

Constraints are treated as *nonsmooth functions*. This conversion is automatically performed by `StructuredOptimization` . For example, the non-negative deconvolution problem:

$$\underset{\mathbf{x}}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{x} * \mathbf{h} - \mathbf{y}\|^2$$

$$\text{subject to } \mathbf{x} \geq 0$$

where $*$ stands for convolution and $\mathbf{h}$ contains the taps of a finite impulse response filter, can be solved using the following lines of code:

```
julia> n = 10;                          # define problem size

julia> x = Variable(n);                 # define variable

julia> h, y = randn(n), randn(2*n-1);
       # random filter taps and output

julia> @minimize ls(conv(x,h)-y) st x >= 0.
```

Note: the convolution mapping was applied to the variable `x` using `conv`. `StructuredOptimization` provides a set of functions that can be used to apply specific operators to variables and create mathematical expressions. In general it is more convenient to use these functions instead of matrices, as these functions apply efficient algorithms for the forward and adjoint mappings leading to *matrix-free optimization*.

A.2.3 Using multiple variables

It is possible to use multiple variables which are allowed to be matrices or even tensors. For example a non-negative matrix factorization problem:

$$\underset{\mathbf{X}_1,\mathbf{X}_2}{\text{minimize}} \ \tfrac{1}{2}\|\mathbf{X}_1\mathbf{X}_2 - \mathbf{Y}\|$$

$$\text{subject to } \mathbf{X}_1 \geq 0, \ \mathbf{X}_2 \geq 0,$$

can be solved using the following code:

```
# matrix variables initialized with random coefficients
julia> X1, X2 = Variable(rand(n,l)), Variable(rand(l,m));

julia> Y = rand(n,m);

julia> @minimize ls(X1*X2-Y) st X1 >= 0., X2 >= 0.
```

A.2.4 Limitations

Currently `StructuredOptimization` supports only *proximal gradient algorithms* (*i.e.*, *forward-backward splitting* base), which require specific properties of the nonsmooth functions and constraints to be applicable. In particular, the nonsmooth functions must have an *efficiently computable proximal mapping*.

If we express the nonsmooth function g as the composition of a function $\tilde{g}$ with a linear operator A:

$$g(\mathbf{x}) = \tilde{g}(\mathsf{A}\mathbf{x}) \tag{A.4}$$

then the proximal mapping of g is efficiently computable if either of the following conditions hold:

1. Operator A is a *tight frame*, namely it satisfies $\mathsf{A}\mathsf{A}^* = \mu\mathsf{Id}$, where $\mu \geq 0$, A^* is the adjoint of A, and Id is the identity operator.
2. Function g is a *separable sum* $g(\mathbf{x}) = \sum_j h_j(\mathsf{B}_j\mathbf{x}_j)$, where $\mathbf{x}_j$ are non-overlapping slices of $\mathbf{x}$, and B_j are tight frames.

Let us analyze these rules with a series of examples. The LASSO example above satisfies the first rule:

```
julia> @minimize ls( A*x - y ) + lambda*norm(x, 1)
```

since the nonsmooth function $\lambda\|\cdot\|_1$ is not composed with any operator (or equivalently is composed with Id which is a tight frame). Also the following problem would be accepted by `StructuredOptimization` :

```
julia> @minimize ls( A*x - y ) + lambda*norm(dct(x), 1)
```

since the discrete cosine transform (DCT) is orthogonal and is therefore a tight frame. On the other hand, the following problem

```
julia> @minimize ls( A*x - y ) + lambda*norm(x, 1)
st x >= 1.0
```

cannot be solved through proximal gradient algorithms, since the second rule would be violated. Here the constraint would be converted into an indicator function and the nonsmooth function g can be written as the sum:

$$g(\mathbf{x}) = \lambda\|\mathbf{x}\|_1 + \delta_{\mathcal{S}}(\mathbf{x}) \tag{A.5}$$

which is not separable. On the other hand this problem would be accepted:

```
julia> @minimize ls( A*x - y ) + lambda*norm(x[1:div(n,2)], 1)
st x[div(n,2)+1:n] >= 1.0
```

as the optimization variables $\mathbf{x}$ are partitioned into non-overlapping groups.

Note: when the problem is not accepted it might be still possible to solve it: see Section 2.5.1.

Appendix B

RIM.jl

Room impulse response generator using the Randomized Image Method (RIM)

Niccolò Antonello

Published online at `https://github.com/nantonel/RIM.jl` commit 3e06363

The candidate's contributions include: software development, design of the tutorial guide, text redaction and editing.

B.1 Installation

From the Julia command line hit

```
Pkg.clone("https://github.com/nantonel/RIM.jl.git")
```

Once the package is installed one can update it along with the others issuing `Pkg.update()` in the command line.

B.2 Usage

Import the package by typing `using RIM`. First an acoustic environment and sampling frequency need to be specified:

```
using RIM
Fs = 8e3          # sampling frequency
env = AcEnv(Fs)   # create new acoustic env with default values
```

By default, the speed of sound is chosen to be 343 m/s. This can be changed using `AcEnv(Fs,c)` where `c` is the speed of sound you want. Create the geometry of the room by typing:

```
Lx,Ly,Lz = 4.,5.,3.;
T60 = 0.7;
geo = CuboidRoom(Lx,Ly,Lz,T60,env);
```

where `Lx`, `Ly`, `Lz` are the room dimensions in meters and `T60` is the desired reverberation time. If the reverberation time is specified, all the walls will have the same reflection coefficient `beta`. Alternatively this can be manually set by specifying 6-element array:

```
beta = [0.9;0.9;0.7;0.7;0.8;0.8];
#[front wall; rear wall; left wall; right wall; floor; ceiling]
geo = CuboidRoom(Lx,Ly,Lz,beta);
```

One can also use frequency-dependent reflection coefficients by means of an infinite impulse response (IIR) filter.

```
b = [0.64;  -0.78;   0.14]
a = [ 1.0;  -1.43;   0.44]
NT = 100;
#number of samples for which the convolution with IIR is truncated
geo = CuboidRoomFD(Lx,Ly,Lz,b,a,NT);
```

Notice that frequency-dependent reflection coefficients lead to simulations that are more computationally expensive.

Once the properties of the room are given, source and microphone positions can be selected:

```
xs = [0.5 0.5 0.5]';                    #src pos (in meters)
xr = [Lx-0.1 Ly-0.3 Lz-0.2; 2. 2. 2.]'; #mic pos
Nt = round(Int64,Fs/2);                 #time samples (1/5 sec)
```

Now type:

```
h = rim(xs,xr,Nt,geo,env);
```

to obtain the room impulse response.

One can also use multiple sound sources, *e.g.*, two sound sources with positions:

```
xs1 = [0.5 0.5 0.5]';      #src 1 pos (in meters)
xs2 = [0.5 0.75 0.5]';     #src 2 pos (in meters)
xr = [Lx-0.1 Ly-0.3 Lz-0.2; 2. 2. 2.]'; #mic pos
```

and signals:

```
Nt = round(Int64,Fs/2);                 #time samples (1/5 sec)
s1 = randn(Nt);                         #src 1 signal
s2 = randn(Nt);                         #src 2 signal
```

Then type:

```
p = rim([s1 s2],[xs1 xs2],xr,Nt,geo,env);
```

to compute the sound pressure generated by these two sound sources.

B.2.1 Changing default parameters with Keyword Arguments

- `CuboidRoom`: The function `CuboidRoom` has the default additional parameters:
  - `Rd = 1e-2`: random displacement of image sources.
  - `Sr`: seed of the randomization which by default is a random integer.

  One can change this by typing:

  ```
  CuboidRoom(Lx,Ly,Lz,T60,env; Rd = myRd, Sr = mySr)
  ```

  or

  ```
  CuboidRoom(Lx,Ly,Lz,beta; Rd = myRd, Sr = mySr)
  ```

- `CuboidRoomFD`: The function `CuboidRoomFD` has the same default additional parameters as `CuboidRoom`.

- `rim`: The function `rim` has the default additional parameters:

- `N = [0;0;0]`: 3-element `Array` representing order of reflection (set to `[0;0;0]` to compute full order).
- `Tw = 20`: taps of fractional delay filter
- `Fc = 0.9`: cut-off frequency of fractional delay filter

One can change this by typing:

```
rim(xs,xr,Nt,geo,env; N = myN, Tw = myTw, Fc = myFc)
```

Other languages implementations

A single-channel MATLAB implementation of the RIM can be found in: `http://www.desena.org/sweep/`

A multi-channel MATLAB version can be found in this repository in `https://github.com/nantonel/RIM.jl/tree/master/src/MATLAB`. Notice that the Julia version outperforms both implementations in terms of speed.

Bibliography

[1] T. van Waterschoot and M. Moonen, "Fifty years of acoustic feedback control: state of the art and future challenges," *Proc. IEEE*, vol. 99, no. 2, pp. 288–327, 2011.

[2] V. Välimäki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "Fifty years of artificial reverberation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 5, pp. 1421–1448, 2012.

[3] V. Välimäki, J. Parker, L. Savioja, J. O. Smith, and J. Abel, "More than 50 years of artificial reverberation," in *Proc. AES 60th Int. Conf. Dereverberation and Reverberation of Audio, Music, and Speech*, 2016.

[4] F. Jacobsen and P. M. Juhl, *Fundamentals of General Linear Acoustics*. John Wiley & Sons, 2013.

[5] H. Kuttruff, *Room Acoustics*. SPON Press, 4th ed., 2000.

[6] G.-B. Stan, J.-J. Embrechts, and D. Archambeau, "Comparison of different impulse response measurement techniques," *J. Audio Eng. Soc.*, vol. 50, no. 4, pp. 249–262, 2002.

[7] A. T. Rosell, "Methods of measuring impulse responses in architectural acoustics," Master's thesis, Technical University of Denmark (DTU), 2009.

[8] Y. Haneda, S. Makino, and Y. Kaneda, "Common acoustical pole and zero modeling of room transfer functions," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 320–328, 1994.

[9] G. Vairetti, E. De Sena, T. van Waterschoot, M. Moonen, M. Catrysse, N. Kaplanis, and S. H. Jensen, "A physically motivated parametric model for compact representation of room impulse responses based on orthonormal basis functions," in *Proc. 10th European Congress Expo. Noise Control Eng. (EURONOISE '15)*, pp. 149–154, 2015.

[10] S. Brenner and R. Scott, *The mathematical theory of finite element methods*. Springer, 2007.

[11] O. von Estorff, *Boundary elements in acoustics: advances and applications*. WIT Press, 2000.

[12] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*. SIAM, 2004.

[13] D. Botteldooren, "Finite-difference time-domain simulation of low-frequency room acoustic problems," *J. Acoust. Soc. Amer.*, vol. 98, no. 6, pp. 3302–3308, 1995.

[14] S. Bilbao and J. O. Smith, III, "Finite difference schemes and digital waveguide networks for the wave equation: Stability, passivity, and numerical dispersion," *IEEE Trans. Speech Audio Process.*, vol. 11, pp. 255–266, May 2003.

[15] K. Kowalczyk and M. van Walstijn, "Wideband and isotropic room acoustics simulation using 2-D interpolated FDTD schemes," *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 1, pp. 78–89, 2010.

[16] K. Kowalczyk and M. van Walstijn, "Room acoustics simulation using 3-D compact explicit FDTD schemes," *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 1, pp. 34–46, 2011.

[17] J. Botts and L. Savioja, "Spectral and pseudospectral properties of finite difference models used in audio and room acoustics," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 9, pp. 1403–1412, 2014.

[18] C. J. Webb and S. Bilbao, "Computing room acoustics with CUDA-3D FDTD schemes with boundary losses and viscosity," in *Proc. 2011 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '11)*, pp. 317–320, 2011.

[19] B. Hamilton and C. J. Webb, "Room acoustics modelling using GPU-accelerated finite difference and finite volume methods on a face-centered cubic grid," *Proc. Digital Audio Effects (DAFx)*, pp. 336–343, 2013.

[20] B. Hamilton, S. Bilbao, and C. J. Webb, "Revisiting implicit finite difference schemes for 3-D room acoustics simulations on GPU," *Proc. Digital Audio Effects (DAFx)*, pp. 41–48, 2014.

[21] S. Bilbao, "Modeling of complex geometries and boundary conditions in finite difference/finite volume time domain room acoustics simulation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 7, pp. 1524–1533, 2013.

[22] S. Bilbao, B. Hamilton, J. Botts, and L. Savioja, "Finite volume time domain room acoustics simulation under general impedance boundary conditions," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 24, no. 1, pp. 161–173, 2016.

[23] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, "Unified analysis of discontinuous Galerkin methods for elliptic problems," *SIAM J. Numer. Anal.*, vol. 39, no. 5, pp. 1749–1779, 2002.

[24] R. Hiptmair, A. Moiola, and I. Perugia, "A survey of Trefftz methods for the Helmholtz equation," in *Building bridges: connections and challenges in modern approaches to numerical partial differential equations*, pp. 237–279, Springer, 2016.

[25] E. Deckers, O. Atak, L. Coox, R. D'Amico, H. Devriendt, S. Jonckheere, K. Koo, B. Pluymers, D. Vandepitte, and W. Desmet, "The wave based method: An

overview of 15 years of research," *Wave Motion*, vol. 51, no. 4, pp. 550–565, 2014.

[26] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods in fluid dynamics*. Springer, 2012.

[27] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *J. Acoust. Soc. Amer.*, vol. 65, no. 4, pp. 943–950, 1979.

[28] J. Borish, "Extension of the image model to arbitrary polyhedra," *J. Acoust. Soc. Amer.*, vol. 75, no. 6, pp. 1827–1836, 1984.

[29] A. Krokstad, S. Strom, and S. Sørsdal, "Calculating the acoustical room response by the use of a ray tracing technique," *J. Acoust. Soc. Amer.*, vol. 8, no. 1, pp. 118–125, 1968.

[30] T. Funkhouser, N. Tsingos, I. Carlbom, G. Elko, M. Sondhi, J. West, G. Pingali, P. Min, and A. Ngan, "A beam tracing method for interactive architectural acoustics," *J. Acoust. Soc. Amer.*, vol. 115, p. 739, 2004.

[31] E. De Sena, H. Hacıhabiboğlu, Z. Cvetković, and J. O. Smith, "Efficient synthesis of room acoustics via scattering delay networks," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 9, pp. 1478–1492, 2015.

[32] A. Sarkissian, "Method of superposition applied to patch near-field acoustic holography," *J. Acoust. Soc. Amer.*, vol. 118, no. 2, pp. 671–678, 2005.

[33] N. Antonello, E. De Sena, M. Moonen, P. A. Naylor, and T. van Waterschoot, "Room impulse response interpolation using a sparse spatio-temporal representation of the sound field," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 10, pp. 1929–1941, 2017.

[34] A. Moiola, R. Hiptmair, and I. Perugia, "Vekua theory for the Helmholtz operator," *Zeitschrift für angewandte Mathematik und Physik*, vol. 62, no. 5, pp. 779–807, 2011.

[35] E. G. Williams, *Fourier acoustics: sound radiation and nearfield acoustical holography*. Academic Press, 1999.

[36] A. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. Winston, 1977.

[37] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[38] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, 1986.

[39] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 984–995, 1989.

[40] Y.-S. Yoon, L. M. Kaplan, and J. H. McClellan, "TOPS: New DOA estimator for wideband signals," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 1977–1989, 2006.

[41] J. H. DiBiase, *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays.* PhD thesis, Brown University, 2000.

[42] O. Oçal, I. Dokmanic, and M. Vetterli, "Source localization and tracking in non-convex rooms," in *Proc. 2014 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '14)*, pp. 1429–1433, 2014.

[43] L. Wang, Y. Liu, L. Zhao, Q. Wang, X. Zeng, and K. Chen, "Acoustic source localization in strong reverberant environment by parametric Bayesian dictionary learning," *Signal Processing*, vol. 143, pp. 232–240, 2018.

[44] I. Dokmanić and M. Vetterli, "Room helps: Acoustic localization with finite elements," in *Proc. 2012 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '12)*, pp. 2617–2620, 2012.

[45] K. Pieper, B. Q. Tang, P. Trautmann, and D. Walter, "Inverse point source location with the Helmholtz equation on a bounded domain," *arXiv preprint arXiv:1805.03310*, 2018.

[46] S. Kitić, L. Albera, N. Bertin, and R. Gribonval, "Physics-driven inverse problems made tractable with cosparse regularization," *IEEE Trans. Signal Process.*, vol. 64, no. 2, pp. 335–348, 2016.

[47] N. Bertin, S. Kitić, and R. Gribonval, "Joint estimation of sound source location and boundary impedance with physics-driven cosparse regularization," in *Proc. 2016 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '16)*, pp. 6340–6344, 2016.

[48] G. Chardon, T. Nowakowski, J. De Rosny, and L. Daudet, "A blind dereverberation method for narrowband source localization," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 5, pp. 815–824, 2015.

[49] T. van Waterschoot and G. Leus, "Distributed estimation of static fields in wireless sensor networks using the finite element method," in *Proc. 2012 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '12)*, pp. 2853–2856, 2012.

[50] J. Murray-Bruce and P. L. Dragotti, "Physics-driven quantized consensus for distributed diffusion source estimation using sensor networks," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, 2016.

[51] P. A. Naylor and N. D. Gaubitch, *Speech dereverberation.* Springer, 2010.

[52] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 2, pp. 145–152, 1988.

[53] E. A. P. Habets and S. Gannot, "Dual-microphone speech dereverberation using a reference signal," in *Proc. 2007 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '07)*, pp. 901–904, 2007.

[54] A. Schwarz and W. Kellermann, "Coherent-to-diffuse power ratio estimation for dereverberation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 6, pp. 1006–1018, 2015.

[55] T. Nakatani, B.-H. Juang, T. Yoshioka, K. Kinoshita, M. Delcroix, and M. Miyoshi, "Speech dereverberation based on maximum-likelihood estimation

with time-varying Gaussian source model," *IEEE Trans. Audio Speech Lang. Process.*, vol. 16, no. 8, pp. 1512–1527, 2008.

[56] A. Jukić, T. van Waterschoot, T. Gerkmann, and S. Doclo, "Multi-channel linear prediction-based speech dereverberation with sparse priors," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 9, pp. 1509–1520, 2015.

[57] T. Ajdler, L. Sbaiz, and M. Vetterli, "The plenacoustic function and its sampling," *IEEE Trans. Signal Process.*, vol. 54, pp. 3790 –3804, Oct. 2006.

[58] R. Mignot, L. Daudet, and F. Ollivier, "Room reverberation reconstruction: Interpolation of the early part using compressed sensing," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 11, pp. 2301–2312, 2013.

[59] R. Mignot, G. Chardon, and L. Daudet, "Low frequency interpolation of room impulse responses using compressed sensing," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 1, pp. 205–216, 2014.

[60] W. Jin and W. B. Kleijn, "Theory and design of multizone soundfield reproduction using sparse methods," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 12, pp. 2343–2355, 2015.

[61] P. Samarasinghe, T. Abhayapala, M. Poletti, and T. Betlehem, "An efficient parameterization of the room transfer function," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 12, pp. 2217–2227, 2015.

[62] B. Bu, T. D. Abhayapala, C. Bao, and W. Zhang, "Parameterization of the three-dimensional room transfer function in horizontal plane," *J. Acoust. Soc. Amer.*, vol. 138, no. 3, pp. EL280–EL286, 2015.

[63] D. M. M. Gómez, J. Astley, and F. M. Fazi, "Low frequency interactive auralization based on a plane wave expansion," *Applied Sciences*, vol. 7, no. 6, 2017.

[64] N. Hahn and S. Spors, "Continuous measurement of spatial room impulse responses on a sphere at discrete elevations," in *Preprints AES 144th Convention*, Audio Engineering Society, 2018.

[65] E. Fernandez-Grande and A. Xenaki, "Compressive sensing with a spherical microphone array," *J. Acoust. Soc. Amer.*, vol. 139, no. 2, pp. EL45–EL49, 2016.

[66] M. A. Poletti, "Three-dimensional surround sound systems based on spherical harmonics," *J. Audio Eng. Soc.*, vol. 53, no. 11, pp. 1004–1025, 2005.

[67] A. J. Berkhout, D. de Vries, and P. Vogel, "Acoustic control by wave field synthesis," *J. Acoust. Soc. Amer.*, vol. 93, no. 5, pp. 2764–2778, 1993.

[68] T. Betlehem, W. Zhang, M. A. Poletti, and T. D. Abhayapala, "Personal sound zones: Delivering interface-free audio to multiple listeners," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 81–91, 2015.

[69] G. N. Lilis, D. Angelosante, and G. B. Giannakis, "Sound field reproduction using the lasso," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 1902–1912, 2010.

[70] N. Radmanesh and I. S. Burnett, "Generation of isolated wideband sound fields using a combined two-stage LASSO-LS algorithm," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 2, pp. 378–387, 2013.

[71] A. H. Moore, M. Brookes, and P. A. Naylor, "Room geometry estimation from a single channel acoustic impulse response," in *Proc. 21st European Signal Process. Conf. (EUSIPCO '13)*, pp. 1–5, 2013.

[72] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, "Acoustic echoes reveal room shape," *Proc. National Academy Sciences*, vol. 110, no. 30, pp. 12186–12191, 2013.

[73] M. Crocco, A. Trucco, and A. Del Bue, "Room reflectors estimation from sound by greedy iterative approach," in *Proc. 2018 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '18)*, 2018.

[74] T. Nowakowski, N. Bertin, J. de Rosny, L. Daudet, *et al.*, "Membrane shape and boundary conditions estimation using eigenmode decomposition," in *Proc. 2016 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '16)*, pp. 3336–3340, 2016.

[75] G. P. Nava, Y. Yasuda, Y. Sato, and S. Sakamoto, "On the in situ estimation of surface acoustic impedance in interiors of arbitrary shape by acoustical inverse methods," *Acoust. Sci. Technol.*, vol. 30, no. 2, pp. 100–109, 2009.

[76] N. Antonello, T. van Waterschoot, M. Moonen, and P. A. Naylor, "Identification of surface acoustic impedances in a reverberant room using the FDTD method," in *Proc. 2014 Int. Workshop Acoustic Signal Enhancement (IWAENC '14)*, pp. 114–118, 2014.

[77] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2nd ed., 2006.

[78] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, 1993.

[79] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, pp. 40–44, 1993.

[80] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011.

[81] N. Parikh and S. P. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[82] L. Stella, A. Themelis, and P. Patrinos, "Forward-backward quasi-Newton methods for nonsmooth optimization problems," *Comput. Optim. Appl.*, vol. 67, no. 3, pp. 443–487, 2017.

[83] A. Themelis, L. Stella, and P. Patrinos, "Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone line-search algorithms," *arXiv:1606.06256*, 2016.

[84] M. Grant, S. Boyd, and Y. Ye, "Disciplined convex programming," in *Global Optimization: From Theory to Implementation*, pp. 155–210, Springer, 2006.

[85] V. Cevher, S. Becker, and M. Schmidt, "Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 32–43, 2014.

[86] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212, Springer, 2011.

[87] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[88] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[89] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[90] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[91] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. on Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[92] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, "A family of variable metric proximal methods," *Math. Programming*, vol. 68, no. 1-3, pp. 15–47, 1995.

[93] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function," *J. Optimiz. Theory Appl.*, vol. 162, no. 1, pp. 107–132, 2014.

[94] P. Frankel, G. Garrigos, and J. Peypouquet, "Splitting methods with variable metric for kurdyka–łojasiewicz functions and general convergence rates," *J. Optimiz. Theory Appl.*, vol. 165, pp. 874–900, Jun 2015.

[95] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "A block coordinate variable metric forward–backward algorithm," *J. Global Optimiz.*, vol. 66, no. 3, pp. 457–485, 2016.

[96] S. Becker and J. Fadili, "A quasi-Newton proximal splitting method," in *Adv. Neural Inf. Process. Syst.*, pp. 2618–2626, 2012.

[97] J. D. Lee, Y. Sun, and M. A. Saunders, "Proximal Newton-type methods for minimizing composite functions," *SIAM J. Optimiz.*, vol. 24, no. 3, pp. 1420–1443, 2014.

[98] N. Komodakis and J.-C. Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *IEEE Signal Process. Mag.*, vol. 32, no. 6, pp. 31–54, 2015.

[99] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods," *Mathematical Programming*, vol. 137, no. 1, pp. 91–129, 2013.

[100] P. Jain and P. Kar, "Non-convex optimization for machine learning," *Found. Trends Machine Learning*, vol. 10, no. 3-4, pp. 142–336, 2017.

[101] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM Review*, vol. 57, no. 2, pp. 225–251, 2015.

[102] N. Boumal, V. Voroninski, and A. Bandeira, "The non-convex burer-monteiro approach works on smooth semidefinite programs," vol. 29, pp. 2757–2765, 2016.

[103] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, 2010.

[104] S. Diamond and S. Boyd, "Matrix-free convex optimization modeling," in *Optimization and its Applications in Control and Data Sciences*, pp. 221–264, Springer, 2016.

[105] J. Folberth and S. Becker, "Efficient adjoint computation for wavelet and convolution operators [lecture notes]," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 135–147, 2016.

[106] M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming," 2008. `http://cvxr.com/cvx/`.

[107] S. Becker, E. Candes, and M. Grant, "TFOCS: Templates for first-order conic solvers," 2012. `http://cvxr.com/tfocs/`.

[108] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.

[109] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, "Application of compressive sensing to sparse channel estimation," *IEEE Commun. Mag.*, vol. 48, no. 11, 2010.

[110] I. Kodrasi and S. Doclo, "Joint dereverberation and noise reduction based on acoustic multi-channel equalization," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 24, no. 4, pp. 680–693, 2016.

[111] S. I. Adalbjörnsson, T. Kronvall, S. Burgess, K. Åström, and A. Jakobsson, "Sparse localization of harmonic audio sources," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 24, no. 1, pp. 117–129, 2016.

[112] B. N. Bhaskar, G. Tang, and B. Recht, "Atomic norm denoising with applications to line spectral estimation," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5987–5999, 2013.

[113] P. Netrapalli, U. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust PCA," *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 1107–1115, 2014.

[114] S. Theodoridis, *Machine learning: a Bayesian and optimization perspective*. Academic Press, 2015.

[115] S. Ling and T. Strohmer, "Self-calibration and biconvex compressive sensing," *Inverse Problems*, vol. 31, no. 11, 2015.

[116] R. T. Rockafellar and R. J. Wets, *Variational analysis*. Springer, 2011.

[117] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *Proc. European Control Conf. (ECC)*, pp. 3071–3076, 2013.

[118] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *Journal of Optimization Theory and Applications*, vol. 169, pp. 1042–1068, June 2016.

[119] J.-J. Moreau, "Proximité et dualité dans un espace Hilbertien," *Bulletin de la Société mathématique de France*, vol. 93, pp. 273–299, 1965.

[120] R. E. Bruck, "An iterative solution of a variational inequality for certain monotone operators in Hilbert space," *Bulletin of the American Mathematical Society*, vol. 81, no. 5, pp. 890–893, 1975.

[121] P.-L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM J. Numer. Anal.*, vol. 16, no. 6, pp. 964–979, 1979.

[122] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," tech. rep., Department of Mathematics, University of Washington, 2008.

[123] Y. Nesterov, "Gradient methods for minimizing composite functions," *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.

[124] P. Patrinos and A. Bemporad, "Proximal Newton methods for convex composite optimization," in *Proc. 52nd IEEE Conf. Decision Control (CDC)*, pp. 2358–2363, 2013.

[125] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos, "A simple and efficient algorithm for nonlinear model predictive control," in *Proc. 56th IEEE Conf. Decision Control (CDC)*, pp. 1939–1944, 2017.

[126] J. F. Claerbout, *Earth soundings analysis: Processing versus inversion*, vol. 6. Blackwell Scientific Publications Cambridge, 1992.

[127] A. Griewank and A. Walther, *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

[128] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[129] L. M. Briceño-Arias and P. L. Combettes, "Convex variational formulation with smooth coupling for multicomponent signal decomposition and recovery," *Numer. Math. Theory Methods Appl.*, vol. 2, pp. 485–508, 2009.

[130] L. M. Briceño-Arias, P. L. Combettes, J.-C. Pesquet, and N. Pustelnik, "Proximal algorithms for multicomponent image recovery problems," *J. Math. Imaging and Vision*, vol. 41, no. 1-2, pp. 3–22, 2011.

[131] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM*, vol. 58, no. 3, p. 11, 2011.

[132] P. L. Combettes, Đ. Dũng, and B. C. Vũ, "Dualization of signal recovery problems," *Set-Valued and Variational Analysis*, vol. 18, no. 3-4, pp. 373–404, 2010.

[133] A. Beck and M. Teboulle, "A fast dual proximal gradient algorithm for convex minimization and applications," *Operations Research Letters*, vol. 42, no. 1, pp. 1–6, 2014.

[134] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM J. Control Optimiz.*, vol. 29, no. 1, pp. 119–138, 1991.

[135] Y. Nesterov, "Smoothing technique and its applications in semidefinite optimization," *Mathematical Programming*, vol. 110, no. 2, pp. 245–259, 2007.

[136] A. Beck and M. Teboulle, "Smoothing and first order methods: A unified framework," *SIAM J. Optimiz.*, vol. 22, no. 2, pp. 557–580, 2012.

[137] L. Stella and N. Antonello, "StructuredOptimization.jl," 2017. `https://github.com/kul-forbes/StructuredOptimization.jl`.

[138] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, "Audio inpainting," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 3, pp. 922–932, 2012.

[139] B. Defraene, N. Mansour, S. D. Hertogh, T. van Waterschoot, M. Diehl, and M. Moonen, "Declipping of audio signals using perceptual compressed sensing," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 12, pp. 2627–2637, 2013.

[140] S. Kitić, N. Bertin, and R. Gribonval, "Sparsity and cosparsity for audio declipping: a flexible non-convex approach," in *Proc. Int. Conf. Latent Variable Analysis and Signal Separation*, pp. 243–250, 2015.

[141] A. Beck and N. Guttmann-Beck, "FOM–a MATLAB toolbox of first order methods for solving convex optimization problems," 2017. `https://sites.google.com/site/fomsolver/`.

[142] L. Stella and P. Patrinos, "ForBES," 2016. `https://github.com/kul-forbes/ForBES`.

[143] E. van den Berg and M. P. Friedlander, "SPOT - A Linear-Operator Toolbox," 2014. `http://www.cs.ubc.ca/labs/scl/spot/`.

[144] F. Heide, S. Diamond, M. Nießner, J. Ragan-Kelley, W. Heidrich, and G. Wetzstein, "ProxImaL: Efficient image optimization using proximal algorithms," *ACM Trans. Graphics*, vol. 35, no. 4, 2016.

[145] K. Kiyohara, K. Furuya, and Y. Kaneda, "Sweeping echoes perceived in a regularly shaped reverberation room," *J. Acoust. Soc. Amer.*, vol. 111, no. 2, pp. 925–930, 2002.

[146] K. Kiyohara, K. Furuya, Y. Haneda, and Y. Kaneda, "Measuring sweeping echoes in rectangular cross-section reverberant fields," *Acta Acustica united with Acustica*, vol. 97, no. 2, pp. 278–283, 2011.

[147] E. De Sena, "On the modeling of rectangular geometries in room acoustic simulations," 2015. `http://www.desena.org/sweep`.

[148] B.-I. Dalenbäck, M. Kleiner, and P. Svensson, "A macroscopic view of diffuse reflection," *J. Audio Eng. Soc.*, vol. 42, no. 10, pp. 793–807, 1994.

[149] C. Borß, "A VST reverberation effect plugin based on synthetic room impulse responses," in *Proc. 12th Int. Conf. Digital Audio Effects (DAFx09)*, 2009.

[150] L. Savioja and V. Välimäki, "Interpolated rectangular 3-D digital waveguide mesh algorithms with frequency warping," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 783–790, 2004.

[151] P. M. Peterson, "Simulating the response of multiple microphones to a single acoustic source in a reverberant room," *J. Acoust. Soc. Amer.*, vol. 80, no. 5, pp. 1527–1529, 1986.

[152] J. Sheaffer, M. van Walstijn, and B. Fazenda, "Physical and numerical constraints in source modeling for finite difference simulation of room acoustics," *J. Acoust. Soc. Amer.*, vol. 135, no. 1, pp. 251–261, 2014.

[153] J. E. Markel and A. H. Gray, *Linear prediction of speech*. Springer, 1982.

[154] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & Systems (2nd edition)*. Prentice-Hall, 1996.

[155] N. F. Declercq, J. Degrieck, R. Briers, and O. Leroy, "A theoretical study of special acoustic effects caused by the staircase of the El Castillo pyramid at the Maya ruins of Chichen-Itza in Mexico," *J. Acoust. Soc. Amer.*, vol. 116, no. 6, pp. 3328–3335, 2004.

[156] F. A. Bilsen, "Repetition pitch glide from the step pyramid at chichen itza," *J. Acoust. Soc. Amer.*, vol. 120, no. 2, pp. 594–596, 2006.

[157] U. P. Svensson, "Edge diffraction toolbox, version B2," 2015. `http://www.iet.ntnu.no/~svensson/software/`.

[158] U. P. Svensson, R. I. Fred, and J. Vanderkooy, "An analytic secondary source model of edge diffraction impulse responses," *J. Acoust. Soc. Amer.*, vol. 106, no. 5, pp. 2331–2344, 1999.

[159] Vistorian Building Commission, "Guide to standards and tolerances," 2007.

[160] B. Yegnanarayana and P. S. Murthy, "Enhancement of reverberant speech using LP residual signal," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 3, pp. 267–281, 2000.

[161] N. D. Gaubitch, P. A. Naylor, and D. B. Ward, "On the use of linear prediction for dereverberation of speech," in *Proc. 2003 Int. Workshop Acoustic Echo Noise Control (IWAENC '03)*, pp. 99–102, Sept. 2003.

[162] J. Wouters, W. Damman, and A. J. Bosman, "Vlaamse opname van woordenlijsten voor spraakaudiometrie." NKO – K.U.Leuven/U.Z.Leuven, 1994.

[163] ITU-T, "Perceptual evaluation of of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," in *ITU-T Recpmmendation P.862, Int. Telecommun. Union*, 2001.

[164] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, A. Sehr, W. Kellermann, and R. Maas, "The REVERB challenge: a common evaluation framework for dereverberation and recognition of reverberant speech," in *Proc. 2013 IEEE Workshop Appls. Signal Process. Audio Acoust. (WASPAA '13)*, pp. 1–4, 2013.

[165] P. C. Loizou, *Speech Enhancement: Theory and Practice*. CRC Press, 2007.

[166] European Broadcasting Union (EBU), "Sound quality assessment material recordings for subjective tests," 2008. `https://tech.ebu.ch/publications/sqamcd`.

[167] M. G. Christensen and A. Jakobsson, *Multi-pitch estimation*. Morgan & Claypool Publishers, 2009.

[168] F. Opolko and J. Wapnick, *McGill University Master Samples*. McGill University, 2006. DVD edition.

[169] T. Cox and P. D'Antonio, *Acoustic absorbers and diffusers: theory, design and application*. Spon Press, 2004.

[170] J. Virieux and S. Operto, "An overview of full-waveform inversion in exploration geophysics," *Geophysics*, vol. 74, no. 6, pp. WCC1–WCC26, 2009.

[171] D. A. Cook, M. F. Mueller, F. Fedele, and A. J. Yezzi, "Adjoint active surfaces for localization and imaging," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 316–331, 2015.

[172] P. C. Hansen and T. K. Jensen, "Smoothing-norm preconditioning for regularizing minimum-residual methods," *SIAM J. Matrix Analysis Appl.*, vol. 29, no. 1, pp. 1–14, 2006.

[173] N. Huleihel and B. Rafaely, "Spherical array processing for acoustic analysis using room impulse responses and time-domain smoothing," *J. Acoust. Soc. Amer.*, vol. 133, no. 6, pp. 3995–4007, 2013.

[174] S. Nam and R. Gribonval, "Physics-driven structured cosparse modeling for source localization," in *Proc. 2012 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '12)*, pp. 5397–5400, 2012.

[175] M. A. Gerzon, "Ambisonics in multichannel broadcasting and video," *J. Acoust. Soc. Amer.*, vol. 33, no. 11, pp. 859–871, 1985.

[176] E. De Sena, H. Hacıhabiboğlu, and Z. Cvetković, "Analysis and design of multichannel systems for perceptual sound field reconstruction," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 8, pp. 1653–1665, 2013.

[177] M. Olik, J. Francombe, P. Coleman, P. J. Jackson, M. Olsen, M. Møller, R. Mason, and S. Bech, "A comparative performance study of sound zoning methods in a reflective environment," in *Proc. AES 52nd Int. Conf. Sound Field Control-Engineering and Perception*, 2013.

[178] P.-A. Gauthier, A. Berry, and W. Woszczyk, "Sound-field reproduction in-room using optimal control techniques: Simulations in the frequency domain," *J. Acoust. Soc. Amer.*, vol. 117, no. 2, pp. 662–678, 2005.

[179] M. Kolundzija, C. Faller, and M. Vetterli, "Reproducing sound fields using MIMO acoustic channel inversion," *J. Audio Eng. Soc.*, vol. 59, no. 10, pp. 721–734, 2011.

[180] N. Antonello, T. van Waterschoot, M. Moonen, and P. A. Naylor, "Evaluation of a numerical method for identifying surface acoustic impedances in a reverberant room," in *Proc. 10th European Congress Expo. Noise Control Eng. (EURONOISE '15)*, pp. 1–6, 2015.

[181] I. W. Selesnick and I. Bayram, "Sparse signal estimation by maximally sparse convex optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1078–1092, 2014.

[182] P.-Y. Chen and I. W. Selesnick, "Group-sparse signal denoising: non-convex regularization, convex optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 13, pp. 3464–3478, 2014.

[183] N. Antonello, "Sound field control in a reverberant room using the finite-difference time-domain (FDTD) method videos," 2016. `ftp://ftp.esat.kuleuven.be/Directory:SISTA/nantonel/Videos/AES60Leuven/`.

[184] T. Ajdler, L. Sbaiz, and M. Vetterli, "Dynamic measurement of room impulse responses using a moving microphone," *J. Acoust. Soc. Amer.*, vol. 122, no. 3, pp. 1636–1645, 2007.

[185] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.

[186] T. Betlehem and T. D. Abhayapala, "Theory and design of sound field reproduction in reverberant rooms," *J. Acoust. Soc. Amer.*, vol. 117, no. 4, pp. 2100–2111, 2005.

[187] R. A. Kennedy, P. Sadeghi, T. D. Abhayapala, and H. M. Jones, "Intrinsic limits of dimensionality and richness in random multipath fields," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2542–2556, 2007.

[188] E. Fernandez-Grande, "Sound field reconstruction using a spherical microphone array," *J. Acoust. Soc. Amer.*, vol. 139, no. 3, pp. 1168–1178, 2016.

[189] S. Spors, R. Rabenstein, and J. Ahrens, "The theory of wave field synthesis revisited," in *Preprints AES 124rd Convention*, pp. 17–20, 2008.

[190] S. Tervo, J. Pätynen, A. Kuusinen, and T. Lokki, "Spatial decomposition method for room impulse responses," *J. Audio Eng. Soc*, vol. 61, no. 1/2, pp. 17–28, 2013.

[191] M. Costabel, "Time-dependent problems with the boundary integral equation method," *Encyclopedia of Computational Mechanics*, 2004.

[192] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay," *IEEE Signal Process. Mag.*, vol. 13, no. 1, pp. 30–60, 1996.

[193] Á. González, "Measurement of areas on a sphere using Fibonacci and latitude–longitude lattices," *Mathematical Geosciences*, vol. 42, no. 1, pp. 49–64, 2010.

[194] R. Marques, C. Bouville, M. Ribardière, L. P. Santos, and K. Bouatouch, "Spherical Fibonacci point sets for illumination integrals," in *Computer Graphics Forum*, vol. 32, pp. 134–143, Wiley Online Library, 2013.

[195] L. Stella and N. Antonello, "ProximalOperators.jl," 2016. `http://github.com/kul-forbes/ProximalOperators.jl` `https://lirias.kuleuven.be/handle/123456789/587243`.

[196] E. De Sena, N. Antonello, M. Moonen, and T. van Waterschoot, "On the modeling of rectangular geometries in room acoustic simulations," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 4, pp. 774–786, 2015.

[197] M. Vorlander, *Auralization - Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality*. Springer, 2008.

[198] N. Antonello, "Randomized image method (RIM)," 2016. `https://github.com/nantonel/RIM.jl` `https://lirias.kuleuven.be/handle/123456789/488661`.

[199] N. Antonello, "Room impulse response interpolation using a sparse spatio-temporal representation of the sound field software," 2017. `https://lirias.kuleuven.be/handle/123456789/587256`.

[200] T. Kronvall, F. Elvander, S. I. Adalbjörnsson, and A. Jakobsson, "Multi-pitch estimation via fast group sparse learning," in *Proc. 23th European Signal Process. Conf. (EUSIPCO '16)*, pp. 1093–1097, 2016.

[201] N. Antonello, "RIRs interpolation videos," 2016. `ftp://ftp.esat.kuleuven.be/stadius/nantonel/Videos/RIR_Intp/`.

[202] J. Pätynen, S. Tervo, and T. Lokki, "Analysis of concert hall acoustics via visualizations of time-frequency and spatiotemporal responses," *J. Acoust. Soc. Amer.*, vol. 133, no. 2, pp. 842–857, 2013.

[203] J. K. Nielsen, J. R. Jensen, S. H. Jensen, and M. G. Christensen, "The single- and multichannel audio recordings database (SMARD)," in *Proc. 2014 Int. Workshop Acoustic Signal Enhancement (IWAENC '14)*, 2014. `http://www.smard.es.aau.dk/`.

[204] M. Brandstein and D. Ward, *Microphone arrays: signal processing techniques and applications*. Springer, 2001.

[205] S. Argentieri, P. Danes, and P. Souères, "A survey on sound source localization in robotics: From binaural to array processing methods," *Computer Speech & Language*, vol. 34, no. 1, pp. 87–112, 2015.

[206] I. Kodrasi, S. Goetze, and S. Doclo, "Regularization for partial multichannel equalization for speech dereverberation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 9, pp. 1879–1890, 2013.

[207] N. Antonello, T. van Waterschoot, M. Moonen, and P. A. Naylor, "Source localization and signal reconstruction in a reverberant field using the FDTD method," in *Proc. 22nd European Signal Process. Conf. (EUSIPCO '14)*, pp. 301–305, 2014.

[208] T. Nowakowski, J. de Rosny, and L. Daudet, "Robust source localization from wavefield separation including prior information," *J. Acoust. Soc. Amer.*, vol. 141, no. 4, pp. 2375–2386, 2017.

[209] P. K. T. Wu, N. Epain, and C. Jin, "A dereverberation algorithm for spherical microphone arrays using compressed sensing techniques," in *Proc. 2012 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '12)*, pp. 4053–4056, 2012.

[210] N. Epain, T. Noohi, and C. Jin, "Sparse recovery method for dereverberation," in *Proc. REVERB Workshop*, 2014.

[211] A. Duijndam and M. Schonewille, "Nonuniform fast Fourier transform," *Geophysics*, vol. 64, no. 2, pp. 539–551, 1999.

[212] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Elad, and M. Israeli, "Fast and accurate polar Fourier transform," *Applied and Computational Harmonic analysis*, vol. 21, no. 2, pp. 145–167, 2006.

[213] N. Antonello, E. De Sena, M. Moonen, P. A. Naylor, and T. van Waterschoot, "Joint source localization and dereverberation by sound field interpolation using sparse regularization," in *Proc. 2018 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '18)*, pp. 6892–6896, 2018.

[214] N. Antonello, L. Stella, P. Patrinos, and T. van Waterschoot, "Proximal gradient algorithms: Applications in signal processing," *arXiv preprint arXiv:1803.01621*, 2018.

[215] Bang and Olufsen, "Music for Archimedes." CD B&O 101, 1992.

[216] E. A. Habets, I. Cohen, and S. Gannot, "Generating nonstationary multisensor signals under a spatial coherence constraint," *J. Acoust. Soc. Amer.*, vol. 124, no. 5, pp. 2911–2917, 2008.

[217] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A Python package for audio room simulations and array processing algorithms," in *Proc. 2018 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP '18)*, pp. 351–355, 2018.

[218] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 19, no. 7, pp. 2125–2136, 2011.

[219] A. Jukić, T. van Waterschoot, and S. Doclo, "Adaptive speech dereverberation using constrained sparse multichannel linear prediction," *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 101–105, 2017.

[220] N. Antonello, "ADelFI audio samples," 2018. `https://nantonel.github.io/adelfi/`.

[221] H. W. Löllmann, C. Evers, A. Schmidt, H. Mellmann, H. Barfuss, P. A. Naylor, and W. Kellermann, "The LOCATA challenge data corpus for acoustic source localization and tracking," in *IEEE Sensor Array Multichannel Signal Process. Workshop (SAM)*, 2018. `www.locata-challenge.org`.

[222] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," 2016. `http://homepages.inf.ed.ac.uk/jyamagis/page3/page58/page58.html`.

[223] mh acoustics, "EM32 eigenmike microphone array release notes (v17.0)," 2013. `http://www.mhacoustics.com/sites/default/files/ReleaseNotes.pdf`.

[224] T. van Waterschoot and M. Moonen, "Distributed estimation and equalization of room acoustics in a wireless acoustic sensor network," in *Proc. 20th European Signal Process. Conf. (EUSIPCO '12)*, pp. 2709–2713, 2012.

[225] J. Szurley, A. Bertrand, P. Ruckebusch, I. Moerman, and M. Moonen, "Greedy distributed node selection for node-specific signal estimation in wireless sensor networks," *Signal Processing*, vol. 94, pp. 57–73, 2014.

[226] A. Griffin, A. Alexandridis, D. Pavlidi, Y. Mastorakis, and A. Mouchtaris, "Localizing multiple audio sources in a wireless acoustic sensor network," *Signal Processing*, vol. 107, pp. 54–67, 2015.

[227] A. Hassani, J. Plata-Chaves, M. H. Bahari, M. Moonen, and A. Bertrand, "Multi-task wireless sensor network for joint distributed node-specific signal enhancement, LCMV beamforming and DOA estimation," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 3, pp. 518–533, 2017.

[228] J. Kaipio and E. Somersalo, *Statistical and computational inverse problems*. Springer, 2006.

[229] W. Bangerth and A. Joshi, "Adaptive finite element methods for the solution of inverse problems in optical tomography," *Inverse Problems*, vol. 24, no. 3, 2008.

[230] Z. Bai and Y. Su, "Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method," *SIAM J. Sci. Comput.*, vol. 26, no. 5, pp. 1692–1709, 2005.

[231] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[232] L. Demanet and L. Ying, "Wave atoms and sparsity of oscillatory patterns," *Applied and Computational Harmonic Analysis*, vol. 23, no. 3, pp. 368–387, 2007.

Curriculum Vitae

Niccolò Antonello received his B.Sc. in Electronic Engineering at the Università degli Studi di Padova (Italy) and his M.Sc. in Acoustic Engineering at the Technical University of Denmark (DTU) in 2010 and 2012, respectively.

Since 2014 he has been a Ph.D. candidate at KU Leuven (Belgium) as an Early Stage Researcher in the Marie Curie Initial Training Network "Dereverberation and Reverberation of Audio, Music, and Speech (DREAMS)". His research interests include room acoustics, inverse problems and numerical optimization.

Website: `https://nantonel.github.io`

Publication List

International Journal Papers

1. **N. Antonello**, L. Stella, P. Patrinos, and T. van Waterschoot, "Proximal gradient algorithms: applications in signal processing", *to be submitted for publication*, 2018.

2. **N. Antonello**, E. De Sena, M. Moonen, P. A. Naylor, and T. van Waterschoot, "Joint acoustic localization and dereverberation by sound field interpolation," submitted for publication, June 2018.

3. **N. Antonello**, E. De Sena, M. Moonen, P. A. Naylor, and T. van Waterschoot, "Room impulse response interpolation using a sparse spatio-temporal representation of the sound field," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 25, no. 10, pp. 1929-1941, Oct. 2017.

4. E. De Sena, **N. Antonello**, M. Moonen, and T. van Waterschoot, "On the modeling of rectangular geometries in room acoustic simulations," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 23, no.6, pp. 774-768, Apr. 2015.

International Conference Papers

1. **N. Antonello**, E. De Sena, M. Moonen, P. Naylor, and T. van Waterschoot, "Joint source localization and dereverberation by sound field interpolation using sparse regularization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP-18)*, Calgary, Canada, Apr. 2018.

2. **N. Antonello**, E. De Sena, M. Moonen, P. Naylor and T. van Waterschoot, "Sound field control in a reverberant room using the Finite Difference Time Domain method," in *Proc. AES 60th Int. Conf.*, Leuven, Belgium, Feb. 2016.

3. **N. Antonello**, T. van Waterschoot, M. Moonen, and P. Naylor, "Evaluation of a numerical method for identifying surface acoustic impedances in a reverberant room," in *Proc. 10th European Congress Expo. Noise Control Eng. (EURONOISE '15)*, Maastricht, the Netherlands, June 2015.

4. **N. Antonello**, T. van Waterschoot, M. Moonen, and P. Naylor, "Identification of surface acoustic impedances in a reverberant room using the FDTD method", in *Proc. 2014 Int. Workshop Acoustic Signal Enhancement (IWAENC 2014)*, Antibes, France, Sep. 2014.

5. **N. Antonello**, T. van Waterschoot, M. Moonen, and P. Naylor, "Source localization and signal reconstruction in a reverberant field using the FDTD method", in *Proc. 22nd European Signal Process. Conf. (EUSIPCO 2014)*, Lisbon, Portugal, Sep. 2014.



FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING
Stadius Center for Dynamical Systems,
Signal Processing and Data Analytics
Kasteelpark Arenberg 10, B-3001 Leuven
http://www.esat.kuleuven.be/stadius/