

# Low-rank Modeling in Room Acoustics

**Martin Jälmbý**

Supervisor:  
Prof. dr. ir. Toon van Waterschoot  
Co-supervisor:  
Prof. dr. ir. Filip Elvander  
(Aalto University)

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

March 2024



# Low-rank Modeling in Room Acoustics

**Martin JÄLMBY**

Examination committee:

Prof. dr. ing. Karel Kellens, chair

Prof. dr. ir. Toon van Waterschoot, supervisor

Prof. dr. ir. Filip Elvander, co-supervisor

(Aalto University)

Prof. dr. Mariya Ishteva

Prof. dr. ir. Youri Meuret

Prof. dr. ir. Panagiotis Patrinos

Prof. dr. Enzo De Sena

(University of Surrey)

Dr. ir. Karl Janssens

(Siemens)

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

March 2024

© 2024 KU Leuven – Faculty of Engineering Technology  
Uitgegeven in eigen beheer, Martin Jälmy, 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

In completing this PhD, I owe much to many. None more, however, than my supervisor Toon van Waterschoot. An extraordinary person, whose great mind is matched only by his great heart. For granting me this opportunity and for your unwavering belief in me, thank you, I could not have asked for a better supervisor.

I am equally grateful to my co-supervisor Filip Elvander, an excellent researcher and a fantastic teacher. Thank you for your continuous encouragement and for always taking the time to answer my questions.

Thank you to Karel Kellens, Mariya Ishteva, Youri Meuret, Panos Patrinos, Enzo De Sena, and Karl Janssens on the examination committee, for your insightful comments and the interesting discussions during the preliminary defense.

To my colleagues, thank you for making sure that this time was not only very educational, but also a lot of fun.

Thanks to the administrative and technical staff of KU Leuven in general, and ESAT-STADIUS in particular, for help in all matters practical.

On a personal level I would like to thank 50k, die Gurke, Faolan, Muita areia, Smallworld, The High Priest, and Verbindingsbro for bringing meaning to life outside of work.

Finally, to my mother Margaretha, my brother Henrik, and my sister Maria. For your ever-present love and support, thank you.

Martin Jälmbly  
Brussels, 2024



# Abstract

When considering the acoustics of a room as a linear time-invariant system, the room impulse response (RIR) describes the response of the room to an audio signal from a given source position to a given receiver position. If the RIR is perfectly known, the response of the room to any acoustic signal can be predicted. In practice, however, there are several difficulties. For a reverberant room, a finite impulse response (FIR) representation of the RIR can be on the order of  $10^5$  samples. Considering the position dependence of the RIR, the complete description of a room requires the RIRs for a multitude of closely spaced source and receiver positions. This is problematic from both a storage and a processing point of view, as the convolution with long FIR filters, e.g., for the purpose of auralization, is computationally demanding, particularly under the strict latency constraints often found in acoustic signal processing.

In this thesis we consider the concept of *low rank*, for the purpose of compact modeling of, and fast low-latency convolution with, RIRs. The idea to employ low-rank modeling in room acoustics developed from the observation that the RIR can be well modeled as a sum of decaying sinusoids. It has previously been shown that if such a signal is reshaped into a matrix or tensor, it will have a rank corresponding to twice the number of components of the sum. Due to modeling approximations and measurement noise, the reshaped RIRs considered in this thesis will not be low rank in a strict sense, but rather that when considering a low-rank approximation of the reshaped RIR, the error will be comparatively small.

The thesis begins with a thorough derivation of the sum-of-decaying-sinusoids model and its low-rank properties, along with a confirmation that the model is relevant for real-life RIRs. Next, the low-rank ideas are employed for the purpose of estimation of RIRs from noisy input-output signals. Subsequently, it is shown how the low-rank structure can be utilized for fast low-latency convolution of RIRs with audio signals. Next, it is demonstrated how the compression of RIRs by low-rank approximation preserves many important

qualities of the RIR, such as reverberation time and center time, and how these methods perform well also with respect to objective signal-based measures. An algorithm for low-latency convolution of RIRs reshaped into tensors is proposed. Finally, it is shown how low rank can be exploited for the joint compression of multiple RIRs in a room. An algorithm for the simultaneous multi-channel convolution of these compressed RIRs is proposed.

The main contribution of this thesis is providing a novel view on, and modeling tools for, RIRs. It is demonstrated how the low-rank framework is useful for RIR approximation, RIR estimation, fast low-latency convolution with audio, the joint compression of all the RIRs of a room, and how low-rank modeling preserves many important RIR qualities.

In this thesis it is shown that the estimation of RIRs from noisy input-output signals could be improved by considering a prior of the matricization of the RIR being low rank. It is also shown that the joint compression of multiple matricized RIRs was a significant improvement as compared to compressing the matricized RIRs one by one. However, it is also demonstrated that the reshaping of the RIR to a higher-order tensor, as opposed to a 2-D matrix, yielded superior results in every aspect, given a fixed compression rate. In light of this, it is believed that of all the exciting areas of possible future research, the estimation of RIRs with a prior of the tensorization of the RIR being low rank, and the joint compression of multiple tensorized RIRs, hold the most potential.

# Korte Inhoud

Wanneer we de akoestiek van een ruimte beschouwen als een lineair tijdsinvariant systeem, beschrijft de impulsrespons van de ruimte (RIR) de respons van de ruimte op een geluidssignaal van een bepaalde bronpositie naar een bepaalde ontvangerpositie. Als de RIR perfect gekend is, kan de respons van de ruimte op een willekeurig akoestisch signaal voorspeld worden. In de praktijk zijn er echter verschillende problemen. Voor een ruimte met nagalm kan de lengte van de eindige-impulsrespons (FIR) voorstelling van de RIR in de orde van  $10^5$  coëfficiënten liggen. Gezien de positie-afhankelijkheid van de RIR zijn voor een volledige beschrijving van een ruimte de RIR's voor een groot aantal dicht bij elkaar gelegen bron- en ontvangerposities nodig. Dit is problematisch vanuit het oogpunt van zowel opslag als verwerking, omdat de convolutie met lange FIR-filters, bijvoorbeeld voor auralisatie, rekenkundig veeleisend is, vooral onder de strikte beperkingen m.b.t. vertraging die vaak voorkomen bij akoestische signaalverwerking.

In dit proefschrift beschouwen we het concept van *lage rang*, met als doel de compacte modellering van, en snelle convolutie aan beperkte vertraging met, RIR's. Het idee om lagerangmodellering toe te passen in de ruimte-akoestiek is ontstaan vanuit de observatie dat de RIR goed gemodelleerd kan worden als een som van gedempte trillingen. Eerder is aangetoond dat als een dergelijk signaal wordt omgevormd tot een matrix of tensor, deze een rang heeft die overeenkomt met tweemaal het aantal componenten van de som. Als gevolg van modelbenaderingen en meetruis zullen de omgevormde RIR's die in dit proefschrift beschouwd worden niet laag in rang zijn in strikte zin, maar eerder dat wanneer een lagerangbenadering van de hervormde RIR beschouwd wordt, de fout relatief klein zal zijn.

Het proefschrift begint met een grondige afleiding van het model dat bestaat uit een som van gedempte trillingen en de lagerangeigenschappen daarvan, samen met een bevestiging dat het model relevant is voor echte RIR's. Vervolgens worden de lagerangideeën gebruikt voor het schatten van RIR's uit ruizige

input-outputsignalen. Vervolgens wordt aangetoond hoe de lagerangstructuur gebruikt kan worden voor snelle convolutie met lage vertraging van RIR's met audiosignalen. Vervolgens wordt aangetoond hoe de compressie van RIR's door middel van lagerangbenadering veel belangrijke kwaliteiten van de RIR behoudt, zoals nagalmtijd en centertijd, en hoe deze methoden ook goed presteren met betrekking tot objectieve signaalgebaseerde maten. Er wordt een algoritme voorgesteld voor convolutie met lage vertraging van RIR's die zijn omgevormd tot tensoren. Tot slot wordt getoond hoe een lage rang kan worden gebruikt voor de gezamenlijke compressie van meerdere RIR's in een ruimte. Er wordt een algoritme voorgesteld voor de gelijktijdige meerkanaalsconvolutie van deze gecomprimeerde RIR's.

De belangrijkste bijdrage van dit proefschrift is het bieden van een nieuwe kijk op en modelleringshulpmiddelen voor RIR's. Er wordt aangetoond hoe het lagerangraamwerk nuttig is voor RIR-benadering, RIR-schatting, snelle convolutie aan beperkte vertraging met audio, de gezamenlijke compressie van alle RIRs van een ruimte en hoe lagerangmodellering veel belangrijke RIR-kwaliteiten behoudt.

In dit proefschrift wordt aangetoond dat de schatting van RIR's van ruizige input-outputsignalen verbeterd kan worden door voorkennis te gebruiken over de lage rang van de matricisatie van de RIR. Er is ook aangetoond dat het gezamenlijk comprimeren van meerdere gematriciseerde RIR's een significante verbetering oplevert vergeleken met het individueel comprimeren van de gematriciseerde RIR's. Er wordt echter ook aangetoond dat het omvormen van de RIR tot een hogere-orde tensor, in tegenstelling tot een 2D matrix, in elk aspect superieure resultaten opleverde, gegeven een vaste compressieverhouding. In het licht hiervan wordt aangenomen dat van alle beloftevolle gebieden van mogelijk toekomstig onderzoek, de schatting van RIR's met gebruik van de voorkennis dat de tensorisatie van de RIR een lage rang heeft, en de gezamenlijke compressie van meerdere getensoriseerde RIR's, het meeste potentieel hebben.

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Korte Inhoud</b>	<b>v</b>
<b>Contents</b>	<b>vii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Room Acoustics . . . . .	1
1.1.1 Linear Time-invariant Systems . . . . .	3
1.1.2 Recording of Room Impulse Responses . . . . .	4
1.1.3 Room Impulse Response Simulations . . . . .	5
1.1.4 Applications . . . . .	6
1.2 Low-rank Modeling . . . . .	8
1.3 Challenges . . . . .	11
1.3.1 Storage . . . . .	12
1.3.2 Speed vs. Latency . . . . .	12
1.4 Overview of the Thesis . . . . .	13

<b>2</b>	<b>Low-rank Tensor Modeling of Room Impulse Responses</b>	<b>17</b>
2.1	Introduction . . . . .	19
2.2	Signal Model . . . . .	20
2.2.1	Physical Motivation of Signal Model . . . . .	21
2.2.2	Low-rank Properties . . . . .	23
2.3	Numerical Results . . . . .	25
2.4	Conclusions and Future Work . . . . .	28
<b>3</b>	<b>Low-rank Room Impulse Response Estimation</b>	<b>31</b>
3.1	Introduction . . . . .	33
3.1.1	Notation . . . . .	36
3.2	Signal Model . . . . .	36
3.3	Estimation with Low-rank Heuristics . . . . .	37
3.3.1	Nuclear Norm Regularization . . . . .	40
3.3.2	Quadratic Envelope . . . . .	41
3.4	Algorithm . . . . .	43
3.4.1	Preliminaries . . . . .	43
3.4.2	Proposed Algorithm . . . . .	44
3.4.3	Computational Complexity . . . . .	46
3.5	Numerical Results . . . . .	47
3.5.1	Impact of $T_{60}$ . . . . .	47
3.5.2	SMARD . . . . .	48
3.5.3	Hyperparameter Tuning . . . . .	49
3.5.4	SNR . . . . .	50
3.5.5	Convergence Analysis . . . . .	50
3.5.6	Hyperparameter Sensitivity . . . . .	52
3.5.7	Frequency Content of Input Signal . . . . .	53

3.5.8	Speech Signal Input . . . . .	53
3.5.9	Input-output Relations . . . . .	54
3.5.10	Cramér–Rao Lower Bound . . . . .	55
3.6	Conclusion . . . . .	57
3.7	Appendix: Quadratic Envelope of $\lambda\ x\ _0$ . . . . .	58
3.8	Appendix: Proximal Operators . . . . .	59
<b>4</b>	<b>Fast Low-latency Convolution by Low-rank Tensor Approximation</b>	<b>63</b>
4.1	Introduction . . . . .	65
4.1.1	Notation . . . . .	66
4.2	Signal Model . . . . .	66
4.3	Low-rank Convolution Algorithm . . . . .	67
4.3.1	Partitioned Truncated SVD Filter . . . . .	67
4.3.2	Fast Convolution by Tensor Approximation . . . . .	68
4.3.3	Complexity . . . . .	70
4.4	Results . . . . .	71
4.5	Conclusions . . . . .	73
<b>5</b>	<b>Compression of Room Impulse Responses for Compact Storage and Fast Low-latency Convolution</b>	<b>75</b>
5.1	Introduction . . . . .	77
5.1.1	Notation and Signal Model . . . . .	79
5.2	Room Impulse Response Compression . . . . .	80
5.2.1	Compression by Truncation . . . . .	80
5.2.2	Compression by Thresholding . . . . .	81
5.2.3	Compression by Low-rank Approximation . . . . .	81
5.2.4	Compression Benchmark: Opus . . . . .	82
5.3	Convolution by Low-rank Approximation . . . . .	82

5.3.1	Partitioned Truncated SVD Filter . . . . .	84
5.3.2	Fast Time-domain Convolution by Tensor Approximation . . . . .	85
5.3.3	Complexity . . . . .	87
5.4	Objective Quality Measures . . . . .	88
5.4.1	Channel-based Objective Quality Measures . . . . .	89
5.4.2	Signal-based Objective Measures . . . . .	92
5.5	Numerical Results . . . . .	93
5.5.1	Normalized Misalignment . . . . .	95
5.5.2	Reverberation Time $T_{60}$ . . . . .	95
5.5.3	Echo Density . . . . .	96
5.5.4	Early Decay Time . . . . .	98
5.5.5	TOA of Direct Component . . . . .	99
5.5.6	Center Time . . . . .	99
5.5.7	Signal Distortion . . . . .	100
5.5.8	ViSQOLAudio . . . . .	101
5.6	Conclusions . . . . .	102
<b>6</b>	<b>Multi-channel Low-rank Convolution of Jointly Compressed Room Impulse Responses</b>	<b>105</b>
6.1	Introduction . . . . .	107
6.1.1	Notation . . . . .	109
6.2	Signal Model, Algorithm, and Motivation . . . . .	109
6.2.1	Signal Model . . . . .	109
6.2.2	GLRAM . . . . .	110
6.2.3	Proposed Room Compression Methods . . . . .	111
6.2.4	Computational Complexity and Compression . . . . .	113
6.2.5	Fast Low-latency Convolution . . . . .	114
6.3	Numerical Results . . . . .	116

6.3.1	Compact-array RIR Measurements . . . . .	116
6.3.2	Spatial Variation of $\mathbf{U}$ and $\mathbf{V}$ . . . . .	117
6.3.3	Synthetic Room Impulse Responses . . . . .	119
6.3.4	Distributed-array RIR Measurements . . . . .	120
6.3.5	Changes in Source Position . . . . .	123
6.3.6	The Relationship Between $\ell_1$ and $\ell_2$ . . . . .	124
6.4	Conclusions . . . . .	125
<b>7</b>	<b>Valorization</b>	<b>127</b>
7.1	VR/AR . . . . .	127
7.2	Sound Zone Control . . . . .	129
7.3	Cultural Heritage Preservation of Historically Important Buildings	130
<b>8</b>	<b>Conclusions and Future Research</b>	<b>133</b>
8.1	Conclusions . . . . .	133
8.2	Future Research . . . . .	136
	<b>Bibliography</b>	<b>141</b>
	<b>List of Publications</b>	<b>159</b>



# List of Figures

1.1	Example of a room impulse response . . . . .	3
2.1	Normalized misalignments as a function of RIR length $N$ . . .	27
2.2	Normalized misalignments as a function tensor dimension $D$ . .	27
2.3	Normalized misalignment as a function of space saving $C(\hat{\mathbf{h}})$ . .	28
3.1	Nuclear norm and quadratic envelope, varying parameter values	43
3.2	Normalized misalignment as a function of $T_{60}$ . . . . .	48
3.3	Linear regression for $\lambda_{QE}$ (top) and $\lambda_{NN}$ . . . . .	50
3.4	Normalized misalignment as a function of $\text{SNR}_{\text{dB}}$ . . . . .	51
3.5	Normalized misalignment as a function of number of iterations .	51
3.6	Normalized misalignment with perturbed parameters . . . . .	52
3.7	Normalized misalignment as a function of $\alpha$ . . . . .	53
3.8	Average normalized misalignment, using speech signals . . . . .	55
3.9	Normalized output misalignment as a function of $\alpha$ , different evaluation signal . . . . .	56
3.10	Comparison with CRLB . . . . .	58
4.1	Normalized output misalignment, short RIRs with speech . . .	72
4.2	Normalized output misalignment, long RIRs with music . . . . .	73

5.1	Examples of RIR compression methods . . . . .	83
5.2	Computational complexity of proposed convolution . . . . .	89
5.3	Normalized misalignment . . . . .	96
5.4	RMSE for $T_{60}$ . . . . .	97
5.5	Differences in $T_{60}$ estimate . . . . .	97
5.6	RMSE for echo density . . . . .	98
5.7	RMSE for EDT . . . . .	99
5.8	RMSE for TOA of direct component . . . . .	100
5.9	RMSE for center time . . . . .	101
5.10	Mean signal distortion . . . . .	102
5.11	Maximum signal distortion . . . . .	103
5.12	ViSQOLAudio . . . . .	104
6.1	Similarity of SVD for closely spaced RIRs . . . . .	113
6.2	Normalized misalignment, $\mathcal{H}_{\text{Model}} = \mathcal{H}_{\text{Comp}}$ . . . . .	117
6.3	Spatial variation of $\mathbf{U}$ and $\mathbf{V}$ and principal angles . . . . .	118
6.4	Difference between LoCo-LoCo-PIñaTa and SVD . . . . .	120
6.5	Normalized misalignment for distributed-array RIRs . . . . .	121
6.6	Example RIR, LoCo-LoCo-PIñaTa, and SVD . . . . .	122
6.7	Example RIRs, measured and synthetic . . . . .	122
6.8	Normalized misalignment for varying $\ell_1$ and $\ell_2$ . . . . .	124

# List of Tables

3.1	Parameter values used when varying $\alpha$ . . . . .	54
5.1	Measures . . . . .	90
5.2	RIR lengths . . . . .	94
6.1	Damping coefficients and reverberation times . . . . .	119
6.2	Changes in source position . . . . .	123



# Chapter 1

## Introduction

### 1.1 Room Acoustics

Room acoustics is the study of the behavior of sound in enclosed spaces. The acoustics of a room impacts how we perceive speech uttered in the room, how we hear music played in the room, and it helps us determine the room's size, shape, and materials. The interaction between an audio signal and the room is highly complicated. For a listener, the part of the sound that travels directly from the source to the receiver is important, but just as important are the delayed and attenuated versions that have bounced off from the walls, the floor, and the ceiling, known as reverberation or reverb. Reverb can be desirable, as it gives an attractive fullness to music, but it can also be undesirable, as it can cause decreased intelligibility of speech.

The complicated relationship between a sound source, the room, and the observed sound pressure, is governed by the acoustic wave equation, a partial differential equation, representing how the change of the acoustic pressure, both with respect to time and space, is dependent on the current acoustic pressure in the point considered. Together with an account for the geometry of the room and its boundaries, so-called boundary conditions, and the initial state of the sound field, so-called initial conditions, this gives a complete description of the sound field in the room. The frequency-domain counterpart of the wave equation is called the Helmholtz equation. The solution of the Helmholtz equation yields the so called steady-state solution, i.e., the response of the system when its transient response has sufficiently decayed.

For the general case, these equations do not have any analytical solutions. For

exceptional cases, however, such as a shoe-box shaped room with rigid walls, explicit expressions exist. In other cases, approximated versions of the wave equation and Helmholtz equation have to be solved numerically. In lightly damped rooms, sound will resonate at certain frequencies, so-called resonance frequencies. The sound field at these resonance frequencies are called modes. The resonance frequencies mainly depend on the dimensions of the room whereas the damping of the modes mainly depends on the materials of the walls. If a sound is played at a frequency close to the resonance frequency of one of these modes, the sound will be amplified due to the creation of standing waves. At lower frequencies the modes tend to be well separated, but for higher frequencies the modes tend to overlap, due to them being more closely spaced in frequency and their larger bandwidth due to stronger damping.

A room is characterized by the room impulse response (RIR). Simply put, it is what is heard at a certain position if a short impulse-like sound is played at another position. This pertains to the theory of linear time-invariant systems, which will be further explained in Chapter 1.1.1, but for now it is sufficient that the RIR can be well modeled as a sum of decaying sinusoids (which will be properly shown in Chapter 2),

$$h(\mathbf{r}_r, \mathbf{r}_s, t) \approx \sum_{m=1}^{m_s} \mu_m(\mathbf{r}_r, \mathbf{r}_s) e^{-\beta_m t} \cos(2\pi f_m t + \phi_m), \quad (1.1)$$

for  $t \geq 0$ . Here,  $\mu_m$  denotes the initial amplitude,  $\mathbf{r}_r \in \mathbb{R}^3$  the position of the receiver,  $\mathbf{r}_s \in \mathbb{R}^3$  the position of the source,  $\beta_m \in \mathbb{R}_+$  the exponential decay constant,  $f_m$  the frequency, and  $\phi_m$  the phase. For ease of notation we will from hereon drop the dependence on  $\mathbf{r}_r$  and  $\mathbf{r}_s$  and simply refer to  $h(\mathbf{r}_r, \mathbf{r}_s, t)$  as  $h(t)$ . The attentive reader will notice the approximate equality in (1.1). Had the number of terms in the sum been infinite, an exact equality would be upheld. In this thesis, however, we will mainly consider digital audio. The RIR therefore needs to be band-limited in order to be digitally sampled, hence the truncation of the sum at  $m_s$  terms. The frequencies of (1.1),  $f_m$ , are the previously mentioned resonance frequencies of the room. The damping coefficients,  $\beta_m$ , depend on frequency, with higher frequencies typically being more strongly damped. The initial amplitudes of the respective modes,  $\mu_m$ , depend on the position of the source and the receiver, but the frequencies and damping coefficients are position-independent properties of the room. An example of an RIR, taken from [1], is displayed in Fig. 1.1. Therein we can see the three different parts the RIR is typically divided into. First, the *direct component*, the sound traveling in a direct path from the source to the receiver. Next, *early reflections*, sound arriving at the receiver shortly after the direct component, after having bounced off from one or a few of the surfaces of the room. Finally, *late reverberation*, that

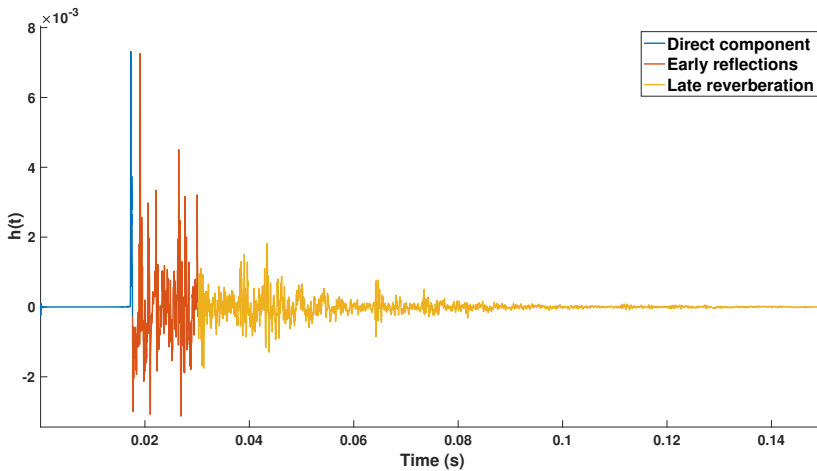


Figure 1.1: Example of a room impulse response (author's coloring).

arrives at the listener after multiple reflections, where the sound is perceived as coming from all directions simultaneously.

The RIR can tell us a lot about the room. One of the most well-known acoustic qualities of a room is the reverberation time,  $T_{60}$ , the time it takes for the sound level in a room to drop 60 dB after a stationary sound source has been switched off. While it can be estimated from formulas accounting for the room's volume, surface area, and absorption coefficients, it is more accurately derived from the RIR. Reverberation times vary widely between rooms. From a tenth of a second for a small office, to well over five seconds for a cathedral.

### 1.1.1 Linear Time-invariant Systems

Linear time-invariant (LTI) systems are a widely used model in mathematics and engineering. An LTI system  $\mathcal{H}$  is a system that takes in an input sequence  $x(n), n = 1, 2, \dots$ , and returns an output sequence  $y(n), n = 1, 2, \dots$ , where  $y(n) = \mathcal{H}(x(n))$ , according to two rules. Firstly, *linearity*, i.e., for any two sequences  $x_1(n)$  and  $x_2(n)$  and  $a_1, a_2 \in \mathbb{R}$ ,

$$\mathcal{H}(x_1(n)) = y_1(n), \mathcal{H}(x_2(n)) = y_2(n) \Rightarrow$$

$$\mathcal{H}(a_1x_1(n) + a_2x_2(n)) = a_1\mathcal{H}(x_1(n)) + a_2\mathcal{H}(x_2(n)) \quad (1.2)$$

i.e., that inputting the linear combination of two signals to the system yields the same result as inputting them separately to the system and then taking the

linear combination of the responses. Secondly, *time invariance*,

$$\mathcal{H}(x(n)) = y(n) \Rightarrow \mathcal{H}(x(n - N)) = y(n - N), \quad (1.3)$$

for  $N \in \mathbb{Z}$ , i.e., that inputting a signal to the system with a delay will not change the output of the system, only add the same delay. An LTI system is characterized by its impulse response  $h(n)$ , i.e., the output of the system when the input is an *impulse*  $\delta(n)$ ,

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}. \quad (1.4)$$

If the impulse response of an LTI system is known, it allows us to predict the output of the system for any input, as any input signal can be written as a sum of scaled and delayed impulses. As previously mentioned, in the context of modeling the acoustics of a room as an LTI system, the impulse response, from a source position to a receiver position, is known as the RIR. The output of the LTI system with the impulse response  $h(n)$ , when the input signal is  $x(n)$ , is found via the *convolution* between  $h(n)$  and  $x(n)$ ,

$$y = h * x, \quad (1.5)$$

where

$$y(n) = \sum_k h(k)x(n - k). \quad (1.6)$$

## 1.1.2 Recording of Room Impulse Responses

In theory, the RIR could be found by emitting an impulse from a loudspeaker and recording the response of the room with a microphone. In practice, however, this has proved to be difficult. Playing a unit impulse from a loudspeaker does not create enough energy to yield a good enough signal-to-noise ratio (SNR). Other methods such as the clapping of hands, popping of a balloon, or shooting a starter's pistol, fail to produce enough energy at low frequencies [2]. Therefore, typically one emits a known signal and the RIR is found through the means of deconvolution. Maximum length sequences (MLS) [3, 4] used to be the most common signals to emit, but since their introduction, exponential sine sweeps (ESS) [5] are the standard. However, MLS sequences are still used under noisy conditions. Further, due to the fact that recorded RIRs commonly contain noise, appropriate post-processing is often necessary [2].

### 1.1.3 Room Impulse Response Simulations

As the recording of RIRs is tedious, time consuming, and in many cases not feasible, considerable efforts have been put into trying to accurately simulate the acoustics of a room. RIR simulation methods can generally be divided into two approaches, wave-based methods and geometrical acoustic (GA) methods. The wave-based methods are, in principle, able to yield more accurate results, particularly at lower frequencies, but are computationally expensive. The GA methods are less accurate, but computationally more manageable [6].

The image method (IM) is one of the most well-known GA methods for creating simulated RIRs [7]. It replaces the contributions to the RIR that would come from reflections off the walls, the ceiling, and the floor with virtual image sources, outside of the simulated room. These image sources are placed in such a fashion that the distance and angle of arrival of the original path (source - wall(s)/ceiling/floor - receiver) and the simulated path (image source - receiver) correspond. The image source method efficiently computes early reflections, but for higher reflection orders it quickly becomes computationally unmanageable, as the number of image sources grows exponentially with the reflection order [6]. The original (IM) considered only shoe-box shaped rooms, but has since been extended to arbitrary polyhedral geometries [8]. For certain regular room dimensions, the original IM can struggle with an artifact phenomenon called a sweeping echo [9]. It has been shown that this can be alleviated by introducing small geometric irregularities in the modeling [10].

Another GA based method is that of ray tracing. It works by letting the source emit sound in the form of particles that are shot out in discrete rays. Each particle carries a certain amount of energy that is progressively lost when it travels through the air and bounces off from the walls of the simulated room. A particle is tracked until a predetermined simulation length is reached, or it carries less energy than some given threshold. A receiver is represented as a surface or a volume. When a particle passes through the volume or surface, the energy of particle and the time is logged. From this information, the simulated RIR can be created [11]. A fundamental advantage of the ray tracing approach, as opposed to the image source approach, is the ability to consider not only specular reflections, but any kind of reflection, including diffuse ones [6]. As an extension of the ray tracing there is, among others, the beam tracing method [12]. In beam tracing, instead of tracing discrete rays, the sound energy takes on the form of volumetric objects that can be detected by point receivers [6].

The aim of the wave-based methods is to numerically solve a discretized version of the wave equation or the frequency domain counterpart, the Helmholtz equation. They are computationally expensive, especially when considering

high frequencies [13]. There are many different methods, among them the finite-difference time-domain (FDTD) method [14, 15], the finite element method (FEM) [13], and the boundary element method (BEM) [16].

Due to distinct strengths and weaknesses of the wave-based and GA methods, it is also possible to use them both in hybrid methods, where a wave-based model handles the modeling up until a certain cross-over frequency, where a GA model takes over. The results of these simulations are then merged [17, 18, 19].

### 1.1.4 Applications

Some of the effects that a room might have on an audio signal are considered desirable, whereas others are considered detrimental. That the RIR can be used both for adding effects to a sound signal, such as a nice reverb to a piece of music, and removing unwanted effects, such as strong echo-like reflections, is reflected in the potential applications considered here. This list is by no means exhaustive. Common to all of them, however, is that they depend on accurate knowledge of the RIR.

#### Artificial reverberation

Reverberation has a long history as a sound effect in music, with composers like Bach composing music to exploit the acoustics of certain buildings [20]. Digital artificial reverberation traces its roots back to the early 1960s, with works like [21, 22]. It is used in e.g., music production [23], video games [24], and movie production [25]. Algorithms for artificial reverberation can be divided into three main categories: delay networks, physical room models, and convolution-based algorithms [2]. Delay network methods include comb filters [21], all-pass filters [21], feedback delay networks (FDN) [26], and scattering delay networks (SDN)[27]. The physically-based reverberation includes the previously mentioned wave-based methods, with the FDTD method being the favored one, and GA-based methods. Finally, convolution-based algorithms, which achieve reverberation by convolving an audio signal with an FIR representation of the RIR, either measured in a room or the result of a simulation [2].

#### Auralization

Auralization is *the process of rendering audible, by physical or mathematical modeling, the sound field of a source in space, in such a way as to simulate the binaural listening experience at a given position in the modeled space* [28]. The

aim is the recreation of the aural impression of the acoustic characteristics of a space, more so than the recreation of the sensation of the music or speech [28]. It is when acoustic effects are processed into an audible result and can be said to be to hearing what visualization is to sight [17].

For successful auralization, the sound source needs to be placed in 3-D space. We need to "hear the room". This can be done by convolving the audio signal with an appropriate RIR prior to playing it. We also need to get a sense of the directivity of the source. In the context of binaural synthesis, intended for playback on a pair of headphones, this can be done in two ways. One can add spatial cues to the different channels (left ear and right ear) such as interaural time difference (ITD) and interaural level difference (ILD) cues, or one can convolve the audio signal with head-related transfer functions (HRTFs), which accounts for the sound bouncing off from the head, torso, pinna, etc. [17].

### **Virtual reality**

Virtual reality (VR) is a computer-generated environment for interaction in real time. Thanks to the arrival of head-mounted displays (HMD) at the consumer market it is quickly becoming a popular and widespread technology [17]. Products include Meta Quest, HTC Vive, and PlayStation VR. The use of VR is, however, not limited to entertainment, as VR has proven useful in areas like decision making in building design, skill training, and therapy in mental health [29, 30, 31, 32]. As multi-modality is crucial to immersion, the visuals need to be paired with acoustic and haptic dimensions as well [17]. Although the visual aspect is probably the thing that first comes to mind when considering VR, realistic reverberation has been shown to be just as important for the perceived immersion [33]. Recent developments of graphic processors and VR subsystems, such as motion trackers, have allowed for the simulation of complex scenes with a high degree of realism [17].

### **Room response equalization**

Room response equalization (RRE) (also known as room correction, room inversion, and room compensation) is, simply put, measuring the RIR and equalizing its effect through inversion of the filter [34], with the term "equalization" referring to the goal of the system's frequency response having an equal magnitude response to all frequencies [35]. It should be guided by knowledge about the human auditory system, perception, and psychoacoustics. In practice, however, it comes with a lot of difficulties. Imperfect RIR measurements, exact equalization of the RIR requiring very long filters, boosting

of the notches of the room transfer function (RTF), the frequency-domain counterpart of the RIR, causing audible effects, and the RIR possibly being non-minimum-phase, to name a few [34]. Equalization can be done by filtering the audio signal before transmission, which is more common for applications like RRE [36]. In dereverberation, a closely related concept referring to the reduction of reverberation at some frequencies [37], it is more common to filter after the transmission [36], often for the purpose of enhanced speech intelligibility [38].

## Sound zone control

Sound zone control refers to the concept of creating personal sound zones. The objective is to render audio within a restricted area of a room, while leaving the rest of the room unaffected. This is useful in a home environment, as well as in shared spaces, such as offices. It is done through an elaborate setup of loudspeakers, and by filtering the audio through control filters, based on RIRs. In sound zone control, one typically considers bright zones, in which one would like high acoustic energy, and dark zones, where the acoustic energy is to be minimized. Sound zone control techniques can be classified into two categories, acoustic contrast control (ACC) and pressure matching (PM). In ACC, the goal is to maximize the contrast between the acoustic energy in the bright zones and in the dark zones. In PM, the aim is still to keep the acoustic energy in the dark zones as small as possible, but in the bright zones, a certain sound field is to be reproduced [39]. It is also possible to consider a weighted combination of ACC and PM [40, 41].

## 1.2 Low-rank Modeling

Low-rank modeling pertains to the mathematical concept of rank. To understand this, one needs to understand what a vector space is, and to understand that one needs to understand what a vector is. The answer to the question of what a vector is depends on whom you ask. A physics student might tell you that it is an arrow pointing in space, defined by its length and direction. A computer science student might say that it is an ordered list of numbers. For the correct answer<sup>1</sup>, one would have to ask a mathematician, who would say that a vector can be anything, as long as there is a sensible notion of adding two vectors, and multiplying the vector with a scalar [42]. The most common example of a vector is that of a vector in the  $n$ -dimensional Euclidean space. For scalars

---

<sup>1</sup>No bias here, of course

$x_1, x_2, \dots, x_n \in \mathbb{R}$ , the vector  $\mathbf{x}$  is defined as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n. \quad (1.7)$$

A vector space, which we will denote  $X$ , is then a non-empty set of vectors, together with the two previously mentioned operations, vector addition and scalar multiplication, i.e.

$$\begin{aligned} \mathbf{w} &= \mathbf{x} + \mathbf{y}, \quad \mathbf{w}, \mathbf{x}, \mathbf{y} \in X \\ \mathbf{z} &= \lambda \mathbf{x}, \quad \lambda \in \mathbb{R}, \mathbf{z}, \mathbf{x} \in X \end{aligned} \quad (1.8)$$

The operations in (1.8) need to satisfy a set of axioms. For the examples considered in this thesis, these axioms will be satisfied, but the curious reader can find more on this in any linear algebra textbook.

Each vector space has a dimension, which can be thought of as the number of independent directions of the vector space. To navigate a vector space one needs a basis for the vector space. A basis is a set of vectors with two properties: that the vectors are linearly independent, and that they span the vector space. A set of vectors is said to be linearly independent if one vector can not be expressed as a linear combination of the other vectors. Formally, the vectors  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  are linearly independent if and only if

$$\lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 + \dots + \lambda_n \mathbf{x}_n = 0 \iff \lambda_1 = \lambda_2 = \dots = \lambda_n = 0. \quad (1.9)$$

A set of vectors  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  is said to span a vector space  $X$  if any vector  $\mathbf{x} \in X$  can be written as a linear combination of  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ . Formally,  $\forall \mathbf{x} \in X, \exists \lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{R}$  s.t.

$$\mathbf{x} = \lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2 + \dots + \lambda_n \mathbf{x}_n. \quad (1.10)$$

The dimension of the vector space is then defined as the number of elements in a basis for the vector space.

The concept of a vector is a natural extension of scalars, from 0-dimensional objects to 1-dimensional. The next step in that generalization, is the 2-dimensional matrix  $\mathbf{X}$ , defined as

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad (1.11)$$

for  $x_{ij} \in \mathbb{R}$ ,  $i = 1, 2, \dots, m$ ,  $j = 1, 2, \dots, n$ . The vertical slices of a matrix are known as columns. These columns are themselves vectors and, as such, they span a vector space. The dimension of this vector space is called the rank of the matrix. The rank can not exceed the number of rows or columns of the matrix, i.e., for  $\mathbf{X} \in \mathbb{R}^{m \times n}$ ,  $\text{rank}(\mathbf{X}) \leq \min(m, n)$ . If equality is upheld, the matrix is said to have full rank, if not, it is said to be rank-deficient. There is no strict definition for when a matrix is *low rank*, but as with many other concepts in mathematics and engineering, a matrix can be said to be low rank if it becomes meaningful to exploit that structure in some application. Important to note is that for applications considered in this thesis, matrices will not be low rank in a strict sense. The matrices in this thesis come from reshaped vectors of real-life recorded RIRs, and are therefore subject to both model approximations and measurement noise. Therefore, low-rank is here meant in a looser sense, i.e., a matrix  $\mathbf{X}$  is said to be low-rank if  $\exists \tilde{\mathbf{X}}$ , s.t.  $\mathbf{X} \approx \tilde{\mathbf{X}}$  and  $\text{rank}(\tilde{\mathbf{X}}) \ll \text{rank}(\mathbf{X})$ .

A low-rank approximation of a matrix can be achieved with the singular value decomposition (SVD). For a real matrix  $\mathbf{X} \in \mathbb{R}^{m \times n}$ , the SVD

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \quad (1.12)$$

$\mathbf{U} \in \mathbb{R}^{m \times m}$ ,  $\mathbf{S} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{V} \in \mathbb{R}^{n \times n}$ , decomposes the linear transformation that  $\mathbf{X}$  represents, into three distinct steps. First,  $\mathbf{V}^T$  performs a rotation and/or reflection of the vector in the domain of  $\mathbf{X}$ ,  $\mathbb{R}^m$ . Secondly,  $\mathbf{S}$  stretches the vector and possibly adds or removes dimensions. Finally,  $\mathbf{U}$ , performs a rotation and/or reflection in the codomain of  $\mathbf{X}$ ,  $\mathbb{R}^n$ . The matrix  $\mathbf{S}$  is diagonal, and the diagonal elements can be interpreted as the relative importance of the respective dimensions. By convention, these are ordered in a non-increasing fashion. In this thesis, and in low-rank modeling in general, it is often found that some singular values are very small in comparison to others. In such cases, it can be useful to compute a low-rank approximation  $\tilde{\mathbf{X}}$  of the matrix  $\mathbf{X}$ , by doing a truncated SVD, where the first  $p$  singular values, for some  $p \in \mathbb{N}$ , are kept, but the ones after that set to zero,  $\tilde{\mathbf{X}} = \mathbf{U}[:, 1:p]\mathbf{S}[1:p, 1:p]\mathbf{V}[:, 1:p]^T$ . This comes with a potentially big reduction of the number of coefficients that has to be stored. Storing the full matrix  $\mathbf{X} \in \mathbb{R}^{m \times n}$  requires the storage of  $nm$  variables, whereas the storage of the  $\tilde{\mathbf{X}}$  requires the storage of only  $p(n+m)$  coefficients. For large matrices of approximately low rank, the difference can be substantial.

The 2-dimensional matrix can, in turn, be further generalized to a so called tensor,  $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_D}$ , a multi-dimensional array which can be of arbitrary dimension  $D$ , and is defined as

$$\mathcal{X} = \sum_{r=1}^R \mathbf{X}_1[:, r] \circ \mathbf{X}_2[:, r] \circ \dots \circ \mathbf{X}_D[:, r], \quad (1.13)$$

for  $\mathbf{X}_j \in \mathbb{R}^{d_j \times R}$ ,  $j = 1, 2, \dots, D$ , and where  $\circ$  denotes outer product of two vectors. The rank of the tensor,  $R$ , is defined as the minimal number of rank-1 tensors needed to generate the tensor as their sum. An element in the tensor  $\mathcal{X}$  is given by

$$\mathcal{X}[d_1, d_2, \dots, d_D] = \sum_{r=1}^R \mathbf{X}_1[d_1, r] \mathbf{X}_2[d_2, r] \dots \mathbf{X}_D[d_D, r]. \quad (1.14)$$

Vectors, matrices, and tensors occur naturally in science and engineering when observing phenomena with one, two, or three or more modes of variations, respectively. Matrices and tensors can, however, also be created by reshaping lower-dimensional data into a higher-dimensional structure. As will be shown throughout this thesis, the benefits of low-rank modeling is often even more pronounced when leveraging the structures of higher-order tensors, as compared to 2-D matrices.

The previously discussed SVD for 2D matrices, (1.12), is optimal in the sense that the  $r$ -truncated SVD is the best rank- $r$  matrix approximation of the matrix  $\mathbf{X}$ , i.e. for  $\mathbf{X}_k = \mathbf{U}[:, 1:k] \mathbf{S}[1:k, 1:k] \mathbf{V}[:, 1:k]^T$

$$\arg \min_{\text{rank}(\mathbf{A})=k} \|\mathbf{X} - \mathbf{A}\|_2 = \arg \min_{\text{rank}(\mathbf{A})=k} \|\mathbf{X} - \mathbf{A}\|_F = \mathbf{X}_k. \quad (1.15)$$

Finding low-rank approximations for higher-order tensors is a much more difficult problem. There are several different tensor decomposition methods, such as multilinear SVD (MLSVD), Tucker decomposition, and the (canonical) polyadic decomposition (CPD). In this thesis we will work exclusively with the CPD. The nice results of (1.15) for 2-D matrices do unfortunately not translate to higher dimensions.

Low-rank modeling in signal processing has seen an increase of popularity as of late. It has been proven useful in areas such as data science [43], blind source separation [44], blind system identification [45], echo cancellation [46], machine learning [47], video background subtraction [48], and matrix completion [49]. In works like [50] and [44] it is demonstrated how low rank can be exploited for several different kinds of signals, including polynomials, sinusoids, and exponentially decaying sinusoids, like in (1.1).

### 1.3 Challenges

The modeling of RIRs comes with a number of challenges. Among them are the computational storage and the need for low-latency processing, that are here further described, and subsequently tackled in this thesis.

### 1.3.1 Storage

Human hearing ranges from 20 Hz to 20 kHz. Given the Nyquist-Shannon sampling theorem stating that the sampling rate needs to be at least twice the maximum frequency to be captured, 48 kHz is a common sampling rate in many applications. As previously mentioned, the reverberation time varies a lot between rooms, from a tenth of a second to several seconds. If one wants to store the RIR as a finite impulse response (FIR) filter with a length corresponding to the reverberation time, this means storing hundreds of thousands of filter coefficients. Further, the RIR is the acoustic channel between one specific source position and one specific receiver position in a room. If the source or the receiver moves, one has to consider a new RIR. In order to represent the acoustics of an entire room, for all possible source-receiver position configurations one has to consider RIRs defined on a large grid. The spacing of the grid,  $\Delta x$ , depends on the frequency one wants to be able to accurately restore [51],

$$\Delta x < \frac{c}{2f_u}, \quad (1.16)$$

where  $c$  denotes the speed of sound and  $f_u$  the cut-off frequency. Even with a moderate ambition of  $f_u = 4$  kHz (which would leave out important frequencies of a lot of music), this would mean that one would have to sample RIRs every 5 cm. A recently released database of RIR measurements, considering a  $1 \text{ m} \times 1 \text{ m}$  planar grid of microphones with 5 cm spacing, measured at 32 different source positions, came out to 3.5 GB of data, despite each RIR being truncated at 0.68 seconds [52]. Taken together, the possibly long filters and the necessity of closely spaced RIRs, storing the acoustics of an entire space quickly becomes a big undertaking.

### 1.3.2 Speed vs. Latency

In the mentioned applications, fast processing is paramount to an immersive experience. When considering convolution for real-time processing, a direct implementation of (1.6) is impractical for longer RIRs due to its heavy computational load,  $n_h$  multiply-add instructions per output sample, where  $n_h$  denotes the length of the filter  $\mathbf{h}$ . Starting with Cooley and Tukey [53] and Stockham [54], attempts have been made to carry out the convolution in a more efficient way by use of the convolution theorem, that states that convolution in the time domain corresponds to element-wise multiplication in the frequency domain. While efficient, at  $\mathcal{O}(n_h \log(n_h))$  multiplications per output sample, a direct implementation of this would, however, introduce a latency of  $2n_h$ . This quickly becomes prohibitive in real-time applications common in acoustic

signal processing, such as virtual video conferencing [55], augmented/mixed reality [56] and virtual reality [57, 58], sound zone control [59, 60], network music performance [61], and artificial reverberation [2].

Trying to find a balance between computational load and input/output latency is therefore something that has attracted considerable attention, with works like [62, 63, 64, 65]. The basic idea is to divide the signal and possibly also the filter into smaller blocks, uniformly or non-uniformly, perform the convolution for each block individually, and then add up the contributions. The use of GPU's and optimization with respect to the hardware can enhance the performance even more [2]. These works have shown that achieving low latency at reasonable computational cost is possible, but all come with non-trivial implementation. Further, partitioned frequency domain filters are subjected to restrictions with regard to assembling them into a network of filters [66].

## 1.4 Overview of the Thesis

**Chapter 2** treats the subject of low-rank approximations of reshaped RIRs. A physical motivation for the use of low-rank models in acoustics is provided and the usefulness of said models when applied to real-life RIRs is demonstrated. It is shown how low-rank tensor approximations outperform competing methods such as truncation and thresholding, with respect to normalized misalignment. A considerable improvement is seen when considering approximations of tensors of orders 3, 4, or 5, as opposed to 2-D matrices.

Chapter 2 has been published as

- M. Jälmy, F. Elvander, and T. van Waterschoot, “Low-rank tensor modeling of room impulse responses,” in *Proc. 29th European Signal Process. Conf. (EUSIPCO)*, pp. 111–115, 2021.

**Chapter 3** concerns estimation of RIRs from input-output data. It is shown how one can enforce a low-rank structure on the solution to the problem of estimating the RIR from noisy observations by the use of two different regularization functions, nuclear norm and quadratic envelope, respectively. This is shown to improve estimation quality, as compared to using no regularization at all or using Tikhonov regularization, with respect to normalized misalignment of either the estimated channel, or output, when the RIRs are convolved with a signal. The two different low-rank regularization methods perform similarly for the most part, but the quadratic envelope is shown to be less sensitive to

a non-optimally set regularization parameter, which could occur when oracle knowledge of the SNR can not be assumed.

Chapter 3 has been published as

- M. Jälmy, F. Elvander, and T. van Waterschoot, "Low-rank room impulse response estimation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 957–969, 2023.

**Chapter 4** is about how the low-rank properties of RIRs can be used for fast, low-latency time-domain convolution. An algorithm for low-latency convolution, using a 3-D tensor approximation of the RIR, is introduced. The RIR does not need to be decompressed for the purpose of using this convolution algorithm. It is shown to outperform a previously existing algorithm using truncated SVD of the matricized RIR, with respect to normalized misalignment of the output, when the approximated RIRs have been convolved with speech or music. This agrees with the results from Chapter 2, where a significant improvement was shown when considering 3-D tensor approximation of the RIR, as opposed to a 2-D matrix approximation.

Chapter 4 has been published as

- M. Jälmy, F. Elvander, and T. van Waterschoot, "Fast low-latency convolution by low-rank tensor approximation," in *Proc. 2023 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 1–5, 2023.

**Chapter 5** treats the comparison of different methods for compressing RIRs. Compression through low-rank approximation of 2-D matrices, 3-D and 5-D tensors is considered, along with truncation and thresholding. These methods are compared to the state-of-the-art audio compression standard Opus. These compressions are compared with respect to how they preserve several objective qualities of the RIR, such as reverberation time, echo density, early decay time, center time, and the time of arrival of the direct component, as well as with respect to the signal-based objective measures frequency-weighted log-spectral signal distortion and ViSQOLAudio. This chapter also introduces an extension of the low-latency convolution algorithm from Chapter 4, where we here consider tensors of arbitrary dimension. Much like for the algorithm in Chapter 4, the RIR does not need to be decompressed for the purpose of using the introduced algorithm. Thresholding performs well with respect to some measures, but in general, compression through low-rank approximation of 3-D and 5-D tensors performs the best.

Chapter 5 has been submitted for publication as

- M. Jälmy, F. Elvander, and T. van Waterschoot, "Compression of Room Impulse Responses for Compact Storage and Fast Low-latency Convolution," Submitted for publication, Dec. 2023.

**Chapter 6** is about the exploitation of low-rank in the modeling of RIRs in the multi-channel case. Here we exploit not only the low-rank structure of a single RIR, but also the similarity of RIRs in a room, for the purpose of compact storage and low-latency convolution. Two methods are introduced for the simultaneous compression of multiple RIRs and in addition to that, an algorithm for multi-channel low-rank convolution is introduced. It is shown how one of the proposed methods outperforms the benchmark methods of SVD approximation and the state-of-the-art Opus compression when the considered RIRs stem from closely spaced microphones, whereas the other method is proven to be more useful when the microphones are farther apart, even covering an entire room.

Chapter 6 has been submitted for publication as

- M. Jälmy, F. Elvander, and T. van Waterschoot, "Multi-channel Low-rank Convolution of Jointly Compressed Room Impulse Responses," Submitted for publication, Dec. 2023.

**Chapter 7** treats the valorization of the research conducted in this thesis. We consider three possible areas of valorization. Two of them, virtual reality and sound zone control, are for industrial purposes, whereas the third one, the preservation of culturally important acoustic spaces, is for the purpose of societal valorization.

**Chapter 8** wraps up the thesis. Herein, conclusions are presented and areas for potential future research are identified.



## Chapter 2

# Low-rank Tensor Modeling of Room Impulse Responses

Low-Rank Tensor Modeling of Room Impulse Responses

Martin Jälmbly, Filip Elvander, and Toon van Waterschoot

Published in *Proc. 29th European Signal Process. Conf. (EUSIPCO)*, 2021

© 2021 The Authors. Reprinted, with permission, from:

M. Jälmy, F. Elvander, and T. van Waterschoot, “Low-rank tensor modeling of room impulse responses,” in *Proc. 29th European Signal Process. Conf. (EUSIPCO)*, pp. 111–115, 2021.

Changes include layout, representation, and minor editing aspects.

The candidate’s contributions as first author include all performed research and writing of the manuscript under guidance from the co-authors.

## Abstract

In this paper, we showcase the benefit of low-rank modeling in the context of acoustic room impulse responses (RIRs). The RIR is modeled as a sum of exponentially decaying sinusoids, and by leveraging the inherent low-rank structure in tensorizations of such signals, we are able to achieve a very compact representation. The contribution of this paper is to provide a clear motivation for the use of low-rank models for the RIR, directly following from physical descriptions of room acoustics. Further, new light is shed on details of the derivation of the RIR model, in relation to solving the inhomogeneous Helmholtz equation. We demonstrate the very compact representations that can be achieved. Finally, the suggested model also displays good robustness properties with respect to the positions of the source and the receiver. Simulations performed on real-life room impulses responses support the theoretical findings.

## 2.1 Introduction

Room impulse responses (RIRs) are essential to algorithms within a wide variety of acoustic signal processing tasks, such as speech dereverberation [67], source separation [68], source localization [1], echo cancellation [69], and artificial reverberation [70]. There are several different ways of modeling the RIR, including as an infinite impulse response (IIR) filter (see e.g. [71]), and as a finite impulse response (FIR) filter (see e.g. [72]). The advantage of an IIR filter model is that it offers a compact representation, but with the drawback of possible difficulties in estimating the filter parameters [73]. The FIR filter model, on the other hand, is appreciated for its simplicity, but has the downside that a large number of parameters is needed to fully represent the characteristics of the room [73]; for a regular office-sized room, such an FIR filter can be several thousands of taps long [38]. This may be prohibitive from a computational and memory requirement point of view, highlighting the need for compact representations of RIRs.

Low-rank models have received a lot of attention in signal processing literature as of late, and their usefulness has been demonstrated in areas such as data science [43], blind source separation [44], blind system identification [45], and echo cancellation [46]. In works like [50] and [44] the authors demonstrate that several kinds of signals can be considered *low-rank*, in the sense that if the signal vector is reshaped into a matrix or tensor, that matrix or tensor will be of low-rank. These signals include polynomials, sinusoids and what we will use as model for the RIR, a sum of exponentially decaying sinusoids [74]. As we

will see, by exploiting this inherent structure, one is able to dramatically reduce the number of parameters needed to represent the RIR.

The main contribution of this paper is to provide a clear, physical motivation for the use of low-rank matrix or tensor models of RIRs. We also shed new light on the approximations that are made throughout the derivation of the RIR model, all the way from the solution to the inhomogeneous Helmholtz equation. The suitability of these low-rank models is demonstrated with simulations on real-life recorded RIRs.

This paper is organized as follows: in Section 2.2 we introduce the signal model, along with the physical justification of the use of a sum of exponentially decaying sinusoids as RIR model, as well as the motivation for using a low-rank approximation. In Section 2.3 we present simulations that support the theoretical findings, and finally, Section 2.4 summarizes and concludes this work and points out areas for possible future research.

## 2.2 Signal Model

As previously indicated, in this work we consider modeling the RIR as a sum of exponentially decaying sinusoids. Assuming that the RIR is band-limited, and thereby can be sampled, the discrete time representation  $h$  is given by

$$h(\mathbf{r}_r, \mathbf{r}_s, n) = \sum_{m \in \mathbb{N}} \mu_m(\mathbf{r}_r, \mathbf{r}_s) e^{-\beta_m n} \cos(\omega_m n + \phi_m), \quad (2.1)$$

for  $n = 1, 2, \dots, N$ . Here,  $\mu_m$  denotes the initial amplitude,  $\mathbf{r}_r \in \mathbb{R}^3$  the position of the receiver,  $\mathbf{r}_s \in \mathbb{R}^3$  the position of the source,  $\beta_m \in \mathbb{R}_+$  the exponential decay constant,  $\omega_m \in [0, \pi]$  the angular frequency, and  $\phi_m$  the phase. For ease of notation we will from hereon drop the dependence on  $\mathbf{r}_r$  and  $\mathbf{r}_s$  and simply refer to  $h(\mathbf{r}_r, \mathbf{r}_s, n)$  as  $h(n)$ . As we will see, this model follows directly from simple physical descriptions of room acoustics, and directly allows for low-rank approximations.

Assuming that  $q = \sqrt{N} \in \mathbb{N}$ , the RIR vector  $\mathbf{h} = [h(1), h(2), \dots, h(N)]^T$  can be reshaped into a square matrix  $\mathbf{H} \in \mathbb{R}^{q \times q}$ ,

$$\mathbf{H} = \begin{bmatrix} h(1) & h(q+1) & \dots & h(q(q-1)+1) \\ \vdots & \vdots & & \vdots \\ h(q) & h(2q) & \dots & h(N) \end{bmatrix}. \quad (2.2)$$

As will be shown, it follows from (2.1), together with some physical considerations of room acoustics, that  $\mathbf{H}$  may be well approximated by a low-rank matrix  $\hat{\mathbf{H}}$ .

The idea of modeling an RIR as a sum of decaying sinusoids is well established in the literature. In works such as [75, 76, 77] this has been used for lower frequencies. The idea of low-rank modeling of RIRs is also anchored in previous work. This is done in e.g. [46], with support from the theoretical discussions in [44]. In this paper, however, we provide a previously missing link between said model, the actual physics of the problem and the use of low-rank tensor-based modeling. As is demonstrated, this allows for compact representations of RIRs.

## 2.2.1 Physical Motivation of Signal Model

The sound field in a source-free room is governed by the wave equation,

$$\nabla^2 p(\mathbf{r}_r, t) = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2}, \quad (2.3)$$

subject to initial and boundary conditions, where  $p$  denotes sound pressure,  $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$  is the Laplace operator,  $c$  the speed of sound, and  $t$  denotes time. Considering the frequency domain counterpart, we get the time-independent form of the wave equation, also known as the Helmholtz equation,

$$(\nabla^2 + k^2) P(\mathbf{r}_r, \omega) = 0, \quad (2.4)$$

where the wave number  $k$  and the angular frequency  $\omega$  are related via  $k = \frac{\omega}{c}$ . We denote the eigenfunctions to the Laplace operator as  $\psi_m$ , with the corresponding eigenvalue  $k_m^2$ . For arbitrary room geometries, there are no analytical expressions for these eigenfunctions, or room modes,  $\psi_m$ . However, for simple geometries, such as shoe-box type rooms with rigid walls, there are closed-form expressions [74]. The eigenfunctions form a complete and orthogonal set [74] and any sound field can be expressed in terms of them [78]. This can be used when we look at a slightly modified version of (2.4), where we introduce a point source with volume velocity  $Q$ , at the source position  $\mathbf{r}_s$ , with a driving frequency  $\omega$ . Further, denoting the value of the gas density by  $\rho_0$ , and the imaginary unit  $i \triangleq \sqrt{-1}$ , we get [74]

$$(\nabla^2 + k^2) G(\mathbf{r}_r, \mathbf{r}_s, \omega) = -i\omega\rho_0 Q \delta(\mathbf{r}_r - \mathbf{r}_s), \quad (2.5)$$

where  $G(\mathbf{r}_r, \mathbf{r}_s, \omega)$  is the so called Green's function, or room transfer function. We can express it in terms of the aforementioned eigenfunctions  $\psi_m$  [74], yielding

$$G(\mathbf{r}_r, \mathbf{r}_s, \omega) = i\omega Q \rho_0 \sum_{m \in \mathbb{N}} \frac{\psi_m(\mathbf{r}_s) \psi_m(\mathbf{r}_r)}{K_m (k_m^2 - k^2)}, \quad (2.6)$$

where  $K_m$  is a normalizing constant. This is also known as the modal decomposition. In general, the eigenfunctions and eigenvalues are complex-valued, although for the special case of perfectly reflecting room boundaries,

they are real-valued [74]. The eigenfunctions and eigenvalues depend only on the dimensions of the room, and not on the source and receiver positions [79]. Furthermore, the eigenvalue  $k_m^2$ , corresponding to mode  $m$ , satisfies  $k_m = \frac{\tilde{\omega}_m}{c} + i\frac{\beta_m}{c}$ , where  $\tilde{\omega}_m$  is the resonance frequency of the  $m$ th mode, and  $\beta_m$  is its decay constant [74]. Inserting this in (2.6), along with  $k = \frac{\omega}{c}$ , and expanding the denominator yields

$$G(\mathbf{r}_r, \mathbf{r}_s, \omega) = \sum_{m \in \mathbb{N}} \frac{A_m i \omega}{-\omega^2 + \tilde{\omega}_m^2 + 2i\tilde{\omega}_m \beta_m - \beta_m^2}, \quad (2.7)$$

where, for the sake of compactness, we collected all the, for our purposes, less interesting variables in  $A_m \triangleq A_m(\mathbf{r}_s, \mathbf{r}_r) = Q\rho_0 c^2 \psi_m(\mathbf{r}_s) \psi_m(\mathbf{r}_r) / K_m$ .

Further, we are going to assume that the decay constants are very small compared to the eigenfrequencies, i.e.  $\beta_m \ll \tilde{\omega}_m$  [74]. This means that each mode in the sum of (2.7) is impactful only when the driving frequency  $\omega$  is very close to the resonance frequency  $\tilde{\omega}_m$ . This allows us to, for each term in the sum, replace  $\tilde{\omega}_m$  with  $\omega$  in the third term of the denominator. With this,

$$\begin{aligned} G(\mathbf{r}_r, \mathbf{r}_s, \omega) &\approx \sum_{m \in \mathbb{N}} \frac{A_m i \omega}{-\omega^2 + \tilde{\omega}_m^2 + 2i\omega\beta_m - \beta_m^2} \\ &\approx \sum_{m \in \mathbb{N}} \frac{A_m i \omega}{-\omega^2 + \beta_m^2 + 2i\beta_m\omega + \omega_m^2} \\ &= \sum_{m \in \mathbb{N}} A_m \frac{i\omega + \beta_m}{-\omega^2 + \beta_m^2 + 2i\beta_m\omega + \omega_m^2} \\ &\quad - \sum_{m \in \mathbb{N}} \frac{A_m \beta_m}{\omega_m} \frac{\omega_m}{-\omega^2 + \beta_m^2 + 2i\beta_m\omega + \omega_m^2} \\ &= \sum_{m \in \mathbb{N}} A_m \mathcal{F}(\theta(t) e^{-\beta_m t} \cos(\omega_m t)) \\ &\quad - \sum_{m \in \mathbb{N}} \frac{A_m \beta_m}{\omega_m} \mathcal{F}(\theta(t) e^{-\beta_m t} \sin(\omega_m t)) \end{aligned} \quad (2.8)$$

where  $\theta(t)$  denotes the Heaviside step function,  $\mathcal{F}$  the Fourier transform and  $\omega_m \triangleq \sqrt{\tilde{\omega}_m^2 - 2\beta_m^2} > 0$  (see also [80] for a time-domain analog). This manoeuvre is something that has been done implicitly before [74, 71], but to the best of the authors' knowledge, this is the first time the necessity of it is made explicit. We see that the adjusted resonance frequency  $\omega_m$  is a slight shift towards zero, compared to the modal frequency  $\tilde{\omega}_m$ . With this, the impulse response  $h(t)$  is

obtained by taking the inverse Fourier transform of (2.8), yielding

$$\begin{aligned} h(t) &= \sum_{m \in \mathbb{N}} \theta(t) e^{-\beta_m t} \left( A_m \cos(\omega_m t) - \frac{A_m \beta_m}{\omega_m} \sin(\omega_m t) \right) \\ &= \sum_{m \in \mathbb{N}} \theta(t) \mu_m e^{-\beta_m t} \cos(\omega_m t + \phi_m) \end{aligned} \quad (2.9)$$

where  $\mu_m = A_m \sqrt{1 + \left(\frac{\beta_m}{\omega_m}\right)^2}$  and  $\phi_m = \arctan\left(\frac{\beta_m}{\omega_m}\right)$ . This is the continuous-time counterpart of (2.1).

Considering that the number of modes,  $N_f$ , below a certain physical frequency  $f$ , satisfies  $N_f \approx \frac{4\pi}{3c^3} f^3$  [78], one would expect to have to use a very large number of terms in (2.9) in order to get a good approximation of the RIR. However, as pointed out in [81], because of the large number of resonant frequencies with overlapping modal curves, the number of distinguishable peaks in the frequency response is in practice considerably smaller than  $N_f$ . That is, (2.9) may be well approximated by the truncated sum

$$h(t) \approx \sum_{m=1}^M \mu_m e^{-\beta_m t} \cos(\omega_m t + \phi_m) \theta(t), \quad (2.10)$$

where  $M$  is on the order of  $10^2$  [81]. Given the finite number of terms, the approximated RIR in (2.10) is band-limited<sup>1</sup> and can be sampled, yielding the discrete-time representation in (2.1), after truncation to  $N$  discrete-time samples.

## 2.2.2 Low-rank Properties

Here we will see how truncation of the modal decomposition (2.9), at some finite number of terms  $M$ , corresponds to finding a rank- $2M$  approximation of the reshaped signal vector.

It is well established, see e.g. [44, 50] and references therein, that when a signal vector consisting of a sum of  $M$  causal, exponentially decaying sinusoids, is reshaped into a matrix, that matrix will have rank  $2M$ . This becomes clear if we decompose  $\mathbf{H}$  in the following fashion. Let  $\mathbf{H} = \mathbf{H}_1 + \mathbf{H}_2 + \dots + \mathbf{H}_M$ , where  $\mathbf{H}_m$  corresponds to the  $m$ th decaying sinusoid,

<sup>1</sup>In practice, given that the human hearing is limited to the range of approximately 20 Hz - 20 kHz [78], the impulse response may, using appropriate filtering, be considered band-limited to this interval. This also serves as justification for removing the cavity mode [78], corresponding to  $m = 0$ , with  $\omega_m = 0$ .

$$\begin{aligned}
\mathbf{H}_m &= \begin{bmatrix} h_m(1) & h_m(q+1) & \dots & h_m(q(q-1)+1) \\ \vdots & \vdots & & \vdots \\ h_m(q) & h_m(2q) & \dots & h_m(N) \end{bmatrix} \\
&= \frac{\mu_m e^{i\phi_m}}{2} \begin{bmatrix} z_m \\ z_m^2 \\ \vdots \\ z_m^q \end{bmatrix} \begin{bmatrix} 1 & z_m^q & \dots & z_m^{(q-1)q} \end{bmatrix} \\
&\quad + \frac{\mu_m e^{-i\phi_m}}{2} \begin{bmatrix} z_m^* \\ z_m^{*2} \\ \vdots \\ z_m^{*q} \end{bmatrix} \begin{bmatrix} 1 & z_m^{*q} & \dots & z_m^{*(q-1)q} \end{bmatrix}.
\end{aligned} \tag{2.11}$$

Here,  $z_m = e^{i\omega_m - \beta_m}$  and  $*$  denotes complex conjugation. We see that each of the  $M$  terms composing  $\mathbf{H}$  can be written as a sum of two rank-1 matrices, i.e. as a rank-2 matrix. As long as  $z_m$  are distinct, the matrix  $\mathbf{H}$  will have rank  $2M$  [50]. For this type of signal, this low-rank structure persists also for higher-order tensorization. The rank of a  $D$ -dimensional tensor  $\mathcal{G} \in \mathbb{R}^{S_1 \times S_2 \times \dots \times S_D}$ , where  $S_d$  denotes the size of the  $d$ th dimension, is defined as the smallest number of rank-1 tensors that generate the tensor as their sum. Furthermore, a  $D$ -dimensional tensor is said to be rank-1 if it can be written as the outer product of  $D$  vectors [82]. Consequently, a rank- $R$  tensor  $\mathcal{G}$  can be written as

$$\mathcal{G} = \sum_{r=1}^R \mathbf{h}_r^{(1)} \circ \mathbf{h}_r^{(2)} \circ \dots \circ \mathbf{h}_r^{(D)}, \tag{2.12}$$

where  $\circ$  denotes outer product, and  $\mathbf{h}_r^{(d)} \in \mathbb{R}^{S_d}$ . Assuming<sup>2</sup> that  $\sqrt[D]{N} \in \mathbb{N}$ , where  $N$  is the length of  $\mathbf{h}$ , the RIR may be reshaped into a  $D$ -dimensional tensor  $\mathcal{H} \in \mathbb{R}^{S \times S \times \dots \times S}$ . Throughout this paper we are going to assume that  $S_1 = S_2 = \dots = S_D = S = \sqrt[D]{N}$  and that  $\sqrt[D]{N} \in \mathbb{N}$ . Then we can reshape the RIR  $\mathbf{h}$  into a  $D$ -dimensional tensor, that will be of rank  $2M$  as well. The upside of this higher-order tensorization is the possibility of further decreasing the number of coefficients needed to represent the RIR, turning what is commonly called the *curse of dimensionality*, into a blessing instead.

A tensor can be decomposed in several different ways. The purpose of this paper is not to showcase the strengths and weaknesses of different tensor decompositions,

<sup>2</sup>In general, arbitrary tensor dimensions may be utilized, as they all give rise to a rank- $2M$  tensor. The square case is considered here, as to simplify the exposition.

for that the reader is referred to works such as [82] and [83]. The aim is rather to demonstrate that the low-rank structure obtained in (2.2), with  $D = 2$ , persists for  $D > 2$ , and the possible compactness of representation that follows. Therefore, we have chosen to use the polyadic decomposition, where the tensor is written as a sum of rank-one tensors [83]. For the numerical computation of these tensor decompositions we use the high-level function `cpd` of the Matlab toolbox Tensorlab [84].

Finally, it should be noted that the physical motivation given here serves only as a justification of the low-rank modeling employed in this paper. The method described is purely data-driven.

## 2.3 Numerical Results

In order to illustrate the usefulness of the considered low-rank framework, and to demonstrate its applicability to actual measured RIRs, we apply it to the *single- and multichannel audio recordings database* (SMARD) [1]. These recordings are sampled at 48 kHz, for 12 seconds, yielding impulse responses of  $576 \cdot 10^3$  taps. These are recorded at various source and receiver positions, and with varying equipment, in a room of size 7.34 m  $\times$  8.09 m  $\times$  2.87 m, with a reverberation time of approximately 0.15 s. In total, the dataset contains 1008 RIRs. It should be noted that in all numerical simulations performed here, the RIRs are truncated at some discrete-time index  $N \ll 576 \cdot 10^3$ .

In order to evaluate the accuracy of the proposed low-rank model, we consider three different scenarios. Specifically we consider the accuracy of the model as a function of the length  $N$  of the RIR, tensorization dimension  $D$ , and memory saving. Here, the memory saving  $C(\hat{\mathbf{h}})$  corresponding to an approximation  $\hat{\mathbf{h}}$  is defined as

$$C(\hat{\mathbf{h}}) = 1 - \frac{\Upsilon(\hat{\mathbf{h}})}{N}, \quad (2.13)$$

where  $\Upsilon(\hat{\mathbf{h}})$  is the number of coefficients of  $\hat{\mathbf{h}}$  and  $C(\hat{\mathbf{h}}) \in [0, 1)$ . For  $C(\hat{\mathbf{h}}) = 0$  there is no memory saving, whereas for  $C(\hat{\mathbf{h}})$  closer to 1, the degree of memory saving is larger. For the proposed method,  $\Upsilon(\hat{\mathbf{h}}) = RD \sqrt[D]{N}$ , where  $R$  is the rank of the approximation.

In the first scenario, where  $N$  is varied,  $C(\hat{\mathbf{h}})$  and  $D$  are kept fixed at 0.8 and 3, respectively. In the second scenario, the tensor dimension  $D$  is varied, with  $N = 7000$  and  $C(\hat{\mathbf{h}}) = 0.8$  are being fixed.<sup>3</sup> In the third scenario,  $C(\hat{\mathbf{h}})$  is varied

---

<sup>3</sup>In order for  $\sqrt[D]{N} \in \mathbb{N}$ ,  $N$  must be adjusted as  $D$  is changed

and  $N = 8000$  and  $D = 3$  are kept fixed. It may here be noted that in all three scenarios, the tensor rank  $R$  varies with  $N$ ,  $D$ , and  $C(\hat{\mathbf{h}})$ , respectively.

In all three scenarios, accuracy is measured by the normalized misalignment, defined as

$$\mathcal{M}(\hat{\mathbf{h}}) = \frac{\|\hat{\mathbf{h}} - \mathbf{h}\|_2}{\|\mathbf{h}\|_2}, \quad (2.14)$$

where  $\mathbf{h}$  is the RIR we are aiming to approximate. Furthermore, we consider two different starting points for the RIR; the first one being when the impulse is emitted from the source (i.e. the RIR contains an initial delay), the second one when the direct component arrives at the microphone (i.e. the RIR does not contain an initial delay). The results presented here are averaged over all the 1008 RIRs of the dataset.

For comparison, we consider two alternative approaches to RIR approximation. In the first alternative approach, we consider simply truncating the RIR some finite discrete-time index  $K$ , setting the last  $N - K$  values to 0. In the second alternative approach,  $\mathbf{h}$  is approximated by keeping only its  $K$  largest entries (in absolute value), and setting the rest to 0. We refer to these approximations as *Truncation* and *KMax* respectively. As to be comparable to the proposed method,  $K$  is set equal to the number of coefficients used by the proposed low-rank model, varying with the different scenarios, i.e., all these methods under comparison yields the same value for the memory saving  $C(\hat{\mathbf{h}})$ .

Fig. 2.1 shows the average normalized misalignment and standard deviation for the discussed approximations, as  $N$  is varied. As may be noted, the proposed method outperforms both comparison methods in terms of normalized misalignment when the initial delay is removed, and on par with *KMax* when the initial delay is kept. Furthermore, the standard deviation is lower, indicating a robustness to factors such as distance to walls, and distance between source and receiver.

Fig. 2.2 displays the accuracy of the proposed model as a function of the dimension of the tensorization. The clear drop in normalized misalignment from  $D = 2$  to  $D = 3$  displays the clear benefit of performing a higher-order tensorization, as opposed to only a matricization. The low standard deviation once again points to robustness to source and receiver placement.

Fig. 2.3 depicts the proposed method as a function of memory saving. Below a certain compression rate the suggested method outperforms *KMax* and, yet again, the standard deviation is consistently lower.

The systematically lower standard deviation is an encouraging discovery. In light of the derivation of the time-domain signal model from the modal decomposition

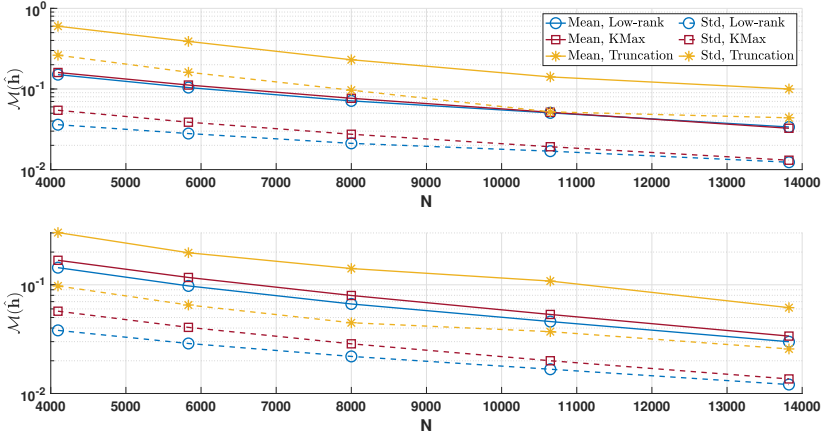


Figure 2.1: Averaged normalized misalignments, and corresponding standard deviations, initial delay kept (top), and removed (bottom) respectively, as a function of  $N$ .

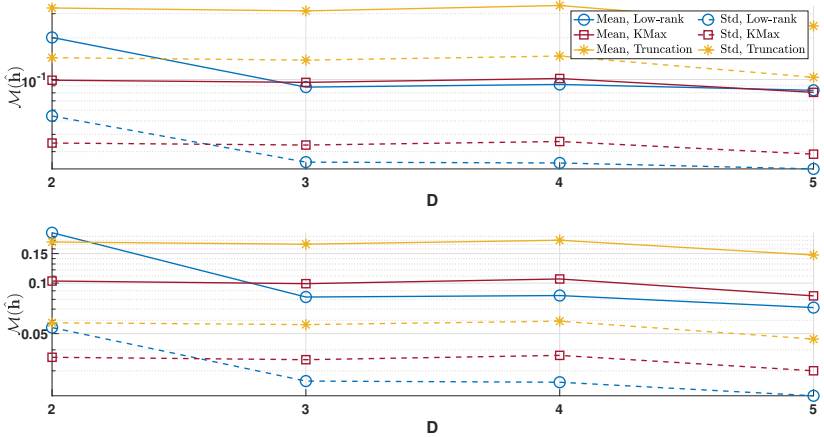


Figure 2.2: Averaged normalized misalignments, and corresponding standard deviations, initial delay kept (top), and removed (bottom) respectively, as a function tensorization dimension  $D$ .

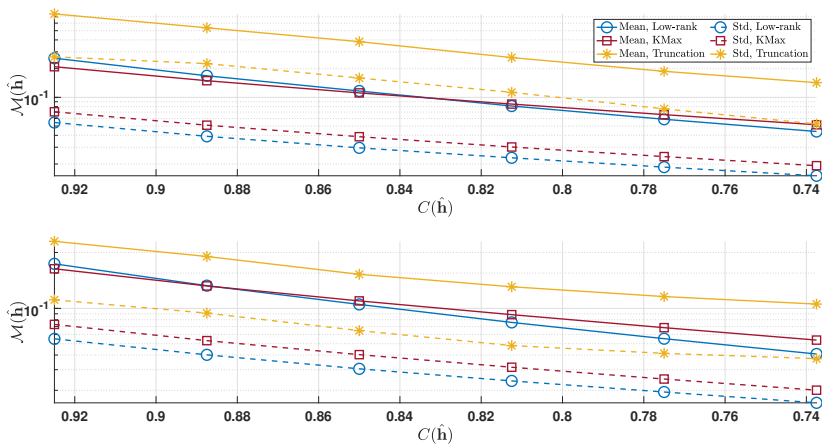


Figure 2.3: Averaged normalized misalignments, and corresponding standard deviations, initial delay kept (top), and removed (bottom) respectively, as a function of space saving  $C(\hat{\mathbf{h}})$ .

in the frequency domain, this is to be expected, since the eigenfunctions do not depend on source and receiver position.

It can also be observed that the suggested model performs better when the initial delay of the RIR is removed, particularly in comparison to the performance of *KMax*. This is also reasonable; the delay is ignored by *KMax*, whereas it has to be explicitly handled by the proposed model, preventing it from more accurately modeling the more information-dense parts of the RIR.

## 2.4 Conclusions and Future Work

In this paper we have given a glimpse of the potential of low-rank modeling of room impulse responses, as well as provided a physical justification for the use of low-rank approximations. We have also clarified the importance of certain approximations made throughout the derivation of the RIR model, which in previous works have been vague. We notice that the low-rank modeling clearly outperforms simple truncation and performs on par with using the largest entries in the RIR, in terms of normalized misalignment. The standard deviation, however, is consistently lower, suggesting that the proposed method is robust to the placement of source and receiver throughout the room.

In this paper many design choices had to be made, for the sake of the brevity of the presentation. Future research will focus on how the choice of these parameters, as well as properties of the room, impacts the success of the low-rank model described here. We also intend to further investigate the error introduced by the suggested approximation. If this approximation distorts the output of a convolution in a systematic way, with respect to more perceptually oriented evaluation, that could be an indication of what applications this method would be particularly useful in.



## Chapter 3

# Low-rank Room Impulse Response Estimation

Low-Rank Room Impulse Response Estimation

Martin Jälmyby, Filip Elvander, and Toon van Waterschoot

Published in *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 957–969, 2023.

© 2023 IEEE. Reprinted, with permission, from:

M. Jälmy, F. Elvander, and T. van Waterschoot, “Low-rank room impulse response estimation,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 957–969, 2023.

Changes include layout, representation, and minor editing aspects.

The candidate’s contributions as first author include all performed research and writing of the manuscript under guidance from the co-authors.

## Abstract

In this paper we consider low-rank estimation of room impulse responses (RIRs). Inspired by a physics-driven room-acoustical model, we propose an estimator of RIRs that promotes a low-rank structure for a matricization, or reshaping, of the estimated RIR. This low-rank prior acts as a regularizer for the inverse problem of estimating an RIR from input-output observations, preventing overfitting and improving estimation accuracy. As directly enforcing a low rank of the estimate results is an NP-hard problem, we consider two different relaxations, one using the nuclear norm, and one using the recently introduced concept of quadratic envelopes. Both relaxations allow for implementing the proposed estimator using a first-order algorithm with convergence guarantees. When evaluated on both synthetic and recorded RIRs, it is shown that under noisy output conditions, or when the spectral excitation of the input signal is poor, the proposed estimator outperforms comparable existing methods. The performance of the two low-rank relaxations methods is similar, but the quadratic envelope has the benefit of superior robustness to the choice of regularization hyperparameter in the case when the signal-to-noise ratio is unknown. The performance of the proposed method is compared to that of ordinary least squares, Tikhonov least squares, as well as the Cramér-Rao lower bound (CRLB).

## 3.1 Introduction

Considering the acoustics of a room as a linear time-invariant (LTI) system, room impulse responses (RIRs) describe the impact of the room on a sound signal, between a point source and a point receiver. They play a vital part in algorithms within a multitude of acoustic signal processing tasks, such as source localization [85], speech dereverberation [38], auralization [58], source separation [86], and echo cancellation [69].

There are several ways of modeling the RIR. Among the most popular ones are the infinite impulse response (IIR) (see e.g., [87, 71]) and finite impulse response (FIR) models (see e.g., [87, 72]). The IIR model offers the possibility of a more compact representation, however with the downside of possible difficulties estimating the filter parameters [73], and potential issues with instability [81]. The FIR model is simple and straightforward, but with the disadvantage that comparatively many coefficients are needed to accurately represent the RIR [73]. For a regular office-sized room, the FIR model can be several thousands of taps long [38], which can be prohibitive from a computational and memory requirement point of view [88, 89].

If the RIR is estimated from an output signal originating from an input signal with poor spectral excitation, i.e., lacking power in certain frequency bands, the estimation problem can become ill-posed, and the resulting estimation could suffer from large variance. To counteract this, it is common to use regularization, see e.g., [90]. The standard approach in RIR estimation is to use Tikhonov regularization [91], a special case of the types of regularization presented in [90]. However, we will demonstrate in this work that by exploiting an (approximate) low-rank structure of RIRs, estimation performance may be improved as compared to state-of-the-art estimators with and without regularization. As of late, low-rank modeling, and low-rank regularization, has become increasingly popular in signal processing, data science and related fields, such as machine learning [47], video background subtraction [48], and matrix completion [49].

In these applications, low-rank structures of matrices or tensors parametrizing the signals under consideration are exploited as to, e.g., obtain compact representations or to regularize otherwise ill-posed estimation problems. The class of signals amenable to this type of treatment includes polynomials, rational functions, smooth and periodic functions, and what will be used in this paper as a model for motivating low-rank structure of RIRs, sums of decaying sinusoids [92].

It should here be noted that in the system identification literature, low-rank structures are most commonly utilized for describing the Toeplitz matrix representing the discrete convolution operation, i.e., the LTI system acting on an input. This has been discussed by, e.g., Marconato *et al.* in [93] and [94]. In these works, empirical Bayes methods are primarily being used, and it is shown that low-rank regularization acts as a powerful method for improving the accuracy of the estimation. In particular, promoting Toeplitz matrices with low-rank is shown to add robustness in the face of input signals with poor spectral excitation.

In contrast, the low-rank structure considered herein does not refer to aforementioned Toeplitz matrix, but rather to a matricization of the impulse response itself. Specifically, we here aim to exploit that RIR vectors when reshaped into a matrix allow for low-rank approximations, something we have taken advantage of in our recent work [95] (see also [96, 92] for relevant work outside the domain of acoustic signal processing). System identification with this type of low-rank structure has earlier been explored by Paleologu *et al.* in works like [46, 97, 98] using iterative Wiener filter, recursive least-squares, and Kalman filter respectively, but there under the name *nearest Kronecker product*. The main focus there are shorter impulse responses, particularly for network echo cancellation. In contrast, the scenario considered in this paper concerns longer impulse responses for applications in acoustic signal processing.

Furthermore, the estimation algorithms in the above mentioned papers require the user to exactly specify the number of low-rank components to be estimated, thereby relying on oracle knowledge of the system complexity. In the work presented herein, we instead propose an estimator where the rank of the estimate is implicitly controlled by means of hyperparameters. In particular, the proposed estimator poses RIR estimation as an inverse problem, with low-rank estimates being promoted by the use of a regularization term.

It may here be noted that directly penalizing the rank of an RIR matrix yields an NP-hard problem [99]. A popular remedy to this in general low-rank estimation is nuclear norm regularization, i.e., penalizing the sum of the singular values, since it yields a convex optimization problem that can be solved by standard methods. This approach has however been shown to introduce a shrinkage, or bias towards zero, of the singular values (see e.g., [47]).

In this work, we propose to counter this bias by utilizing the so-called quadratic envelope of the rank penalty. Quadratic envelopes, introduced in [100], constitute a class of minorizers constructed from point-wise best function approximations from below by quadratic functions, and have recently been used for addressing bias issues in sparsity problems within the realm of image processing and computer vision by Carlsson *et al.* in works like [100, 101, 102, 103]. A drawback of this relaxation is that the resulting optimization problem is not necessarily convex, but in many cases this is outweighed by the very attractive property that, for appropriate hyperparameter values, the relaxation does not move any local minima of the problem being approximated. Its merit is, however, to the best of the authors' knowledge, yet to be explored in the context of acoustic signal processing. In addition, the problem considered herein falls into a class of problems for which convergence of the proximal gradient method, or forward-backward splitting (FBS), to a local minimum can be guaranteed.

The contribution of this paper is therefore to provide a framework for low-rank estimation of room impulse responses. We present an easy-to-use algorithm and show that low-rank regularization outperforms benchmark methods, that is, ordinary least squares and Tikhonov regularized least squares. This is demonstrated using synthetic as well as real-life RIRs, driven by synthetic as well as real-life signals and it is shown that this works well on real life-recorded RIRs. The outperformance is consistent, but particularly noticeable when the SNR is low, or when the spectral excitation is poor. Further, the quadratic envelope performs exceptionally well when prior knowledge about the SNR can not be assumed. Finally, it should be noted that the use of the proposed algorithm is not restricted to acoustics and RIRs, but rather it is applicable to any physical system of which the impulse response is well modeled by a sum of decaying sinusoids.

This paper is organized as follows: first, Section 3.1 is concluded with an introduction of the notation used throughout the paper. In Section 3.2, the signal model is introduced. In Section 3.3, different possible relaxations of the low-rank penalty term are discussed, and in Section 3.4, the proposed algorithm is introduced. Numerical results are presented in Section 3.5. Finally, in Section 3.6, conclusions of the work presented here are discussed, and possible avenues for future research are being pointed out.

### 3.1.1 Notation

Vectors are denoted by lower case, bold letters, such as  $\mathbf{h}$ , and matrices are denoted by upper case, bold letters, such as  $\mathbf{H}$ . Subscript on an upper case bold letter indicates matrix column, i.e.,  $\mathbf{H}_j$  denotes the  $j$ th column of  $\mathbf{H}$ , and on a lower case bold letter it indicates vector element, i.e.,  $\mathbf{h}_j$  denotes the  $j$ th element of  $\mathbf{h}$ . A bar,  $\bar{\cdot}$ , denotes complex conjugation of a complex number and the hat symbol,  $\hat{\cdot}$ , denotes an estimated quantity. The symbol  $\odot$  denotes the Hadamard product, i.e., element-wise multiplication of vectors or matrices, the nabla operator,  $\nabla$ , refers to the gradient of a function, and  $\mathbb{E}[\cdot]$  denotes statistical expectation. Linear operators are denoted by upper case calligraphic letters, such as  $\mathcal{A}$ , and an asterisk,  $*$ , denotes adjoint. Sets are also denoted by upper case calligraphic letters, but it will be obvious from context what is considered. The norm of a matrix or an operator is denoted  $\|\cdot\|$  and refers to  $\|\cdot\|_2$ ,  $\langle \mathbf{u}, \mathbf{v} \rangle_n = \sum_{j=1}^n \mathbf{u}_j \mathbf{v}_j$  denotes the inner product between two vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ , and  $i = \sqrt{-1}$  denotes the imaginary unit.

## 3.2 Signal Model

We assume an observed acoustic pressure signal in discrete time  $y(n), n = 1, 2, \dots, n_y$ , with corresponding signal vector  $\mathbf{y} \in \mathbb{R}^{n_y}$ . This observed signal is the result of a known signal  $x(n), n = 1, 2, \dots, n_x$ , with corresponding vector  $\mathbf{x} \in \mathbb{R}^{n_x}$ , being generated by a point source in a room, and can therefore be modeled as the convolution of  $\mathbf{x}$  with the room impulse response  $h(n), n = 1, 2, \dots, n_h$ , with corresponding vector  $\mathbf{h} \in \mathbb{R}^{n_h}$ ,  $\mathbf{x} * \mathbf{h}$ , where

$$(\mathbf{x} * \mathbf{h})_n = \sum_{k=1}^{n_h} \mathbf{h}_k \mathbf{x}_{n+n_h-k}, \quad (3.1)$$

for  $n = 1, 2, \dots, n_y$ , i.e., we consider only the part of the RIR where the system is fully excited. We will assume that  $n_x \geq n_h$ , to not have to consider initial conditions. We further assume measurement noise  $\mathbf{e} \in \mathbb{R}^{n_y}$ , that we

assume to be white Gaussian with variance  $\sigma^2$ , i.e.,  $\mathbf{e} \in \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{n_y})$ , where  $\mathbf{I}_{n_y} \in \mathbb{R}^{n_y \times n_y}$  denotes the identity matrix. In the more general case of non-white, non-stationary noise, a combination of prefiltering and regularization can be used, as elaborated on in [91]. Taken together, we have that

$$\mathbf{y} = \mathbf{x} * \mathbf{h} + \mathbf{e}, \quad (3.2)$$

which can also be written as

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{e}, \quad (3.3)$$

where  $\mathbf{X} \in \mathbb{R}^{n_y \times n_h}$  is the Toeplitz matrix corresponding to the convolution operation in (3.2).

We will have that  $n_y = n_x - n_h + 1$ . We define the linear operator  $\mathcal{A} : \mathbb{R}^{n_h} \rightarrow \mathbb{R}^{n_y}$ , such that  $\mathcal{A}(\mathbf{h}) = \mathbf{x} * \mathbf{h}$ . Here we also define the adjoint operator  $\mathcal{A}^* : \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_h}$ , as the unique operator for which

$$\langle \mathbf{h}, \mathcal{A}^*(\mathbf{y}) \rangle_{n_h} = \langle \mathcal{A}(\mathbf{h}), \mathbf{y} \rangle_{n_y}, \quad \forall \mathbf{h} \in \mathbb{R}^{n_h}, \mathbf{y} \in \mathbb{R}^{n_y}. \quad (3.4)$$

### 3.3 Estimation with Low-rank Heuristics

The room impulse response  $\mathbf{h}$  can be estimated from the input and observed signal vectors with the least squares method,

$$\underset{\mathbf{h}}{\text{minimize}} \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{h})\|_2^2, \quad (3.5)$$

and under the assumption that the noise  $\mathbf{e}$  is white Gaussian, this yields the maximum likelihood estimator. Inverse problems like these are, however, often ill-posed [104, 105], particularly in acoustic signal processing, where poor excitation is a common occurrence [91]. The inverse mapping from  $\mathbf{y}$  to  $\mathbf{h}$  might therefore be unstable due to ill-conditioning. The fact that the output is corrupted by noise could yield an over-fitted solution, and finally, the solution could be non-unique [105]. In order to have a well-posed problem, a possible approach is to use regularization [90]. This is done by appending a penalty term to the optimization problem (3.5). With that, the problem can be written as

$$\underset{\mathbf{h}}{\text{minimize}} f(\mathbf{h}) + g(\mathbf{h}). \quad (3.6)$$

We consider the function  $f : \mathbb{R}^{n_h} \rightarrow \mathbb{R}$  to be the data-fit term  $\frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{h})\|_2^2$ , i.e., a function penalizing the distance between the model output and the observation. The function  $g : \mathbb{R}^{n_h} \rightarrow \mathbb{R}$  is the regularization term. Note that

this seemingly simple form includes constrained optimization problems, since we can let  $g$  be an indicator function for some non-empty set  $\mathcal{S}$ ,

$$g(\mathbf{h}) = \mathcal{I}_{\mathcal{S}}(\mathbf{h}) = \begin{cases} 0 & , \mathbf{h} \in \mathcal{S} \\ \infty & , \mathbf{h} \notin \mathcal{S} \end{cases}. \quad (3.7)$$

This form also includes the aforementioned Tikhonov regularization, where  $g(\mathbf{h}) = \lambda_{\text{T}} \|\mathbf{h}\|_2^2$ , yielding

$$\underset{\mathbf{h}}{\text{minimize}} \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{h})\|_2^2 + \lambda_{\text{T}} \|\mathbf{h}\|_2^2, \quad (3.8)$$

where  $\lambda_{\text{T}} \geq 0$  is a parameter controlling the degree of regularization.

The regularization should, however, preferably be designed using prior knowledge about the problem [91]. We will therefore demonstrate the low-rank properties of RIRs, and argue how these can be exploited in regularization. An RIR can be well modelled as an infinite sum of decaying sinusoids, see e.g., [74]. The frequencies of these decaying sinusoids correspond to the modes of the room, i.e., frequencies at which standing waves would occur in the absence of wall absorption. The number of room modes,  $N_f$ , below a certain frequency  $f$  satisfies  $N_f \approx \frac{4\pi}{3c^3} f^3$  [78], where  $c$  is the speed of sound. One might therefore expect to have to use a large number of terms to approximate this sum. However, in [81] it is concluded that the large overlap between the modal curves, which occurs particularly at higher frequencies [73], makes the number of distinguishable peaks in the frequency magnitude response much smaller, and it is justified to approximate the infinite sum with a finite one. For a longer exposition, see e.g., [81] and [95]. With this, the room impulse response can be modeled by

$$h(\mathbf{r}_r, \mathbf{r}_s, n) = \sum_{m=1}^{m_s} \mu_m(\mathbf{r}_r, \mathbf{r}_s) e^{-\beta_m n} \cos(\omega_m n + \phi_m), \quad (3.9)$$

for  $n = 1, 2, \dots, n_h$ . Here,  $\mu_m$  denotes the initial amplitude,  $\mathbf{r}_r, \mathbf{r}_s \in \mathbb{R}^3$  the position of the receiver and the source, respectively,  $\beta_m \in \mathbb{R}_+$  the exponential decay constant,  $\omega_m \in [0, \pi]$  the angular frequency  $\phi_m \in [0, 2\pi)$ , the phase, and  $m_s \in \mathbb{N}$  is the number of decaying sinusoids used in the model. For ease of notation we will drop the dependence on  $\mathbf{r}_r$  and  $\mathbf{r}_s$  and refer to  $h(\mathbf{r}_r, \mathbf{r}_s, n)$  simply as  $h(n)$ , the same  $h(n)$  as presented in Section 3.2.

When a vector consisting of the sum of  $m_s$  discrete time decaying sinusoids is reshaped into a matrix, that matrix will have rank  $2m_s$ , see e.g., [44]. Further, as previously mentioned, there is a large spectral overlap of some of the modal curves of the decaying sinusoids that make up the RIR, particularly at higher frequencies. These two facts taken together makes for good conditions for

enforcing a low-rank structure on the solution, i.e., that when the solution vector is reshaped into a matrix, that matrix should have low rank. In previous work we have shown the connection between the physics driven modeling of room acoustics and the use of the sum of decaying sinusoids model, along with the connection to low rank modeling [95]. For more on the physical motivation we refer to [95], but for the convenience of the reader, we here reiterate the low-rank motivation. It should be noted that a measured RIR will, when matricized, not be low rank in the strict sense, because of measurement noise and model errors. *Low rank* is here meant in a less strict sense, i.e., that a low-rank approximation of the RIR will render a small approximation error.

Let us define the reshaping operator  $\mathcal{R}_{pq} : \mathbb{R}^{n_h} \rightarrow \mathbb{R}^{p \times q}$ , that reshapes a vector  $\mathbf{h} \in \mathbb{R}^{n_h}$  into a matrix  $\mathbf{H} \in \mathbb{R}^{p \times q}$ , for which  $n_h = pq$ . Then, if we assume that  $q = \sqrt{n_h} \in \mathbb{N}$ , the RIR vector  $\mathbf{h}$  can be reshaped into a square matrix  $\mathbf{H} = \mathcal{R}_{qq}(\mathbf{h}) \in \mathbb{R}^{q \times q}$ ,

$$\mathbf{H} = \begin{bmatrix} h(1) & h(q+1) & \dots & h(q(q-1)+1) \\ \vdots & \vdots & & \vdots \\ h(q) & h(2q) & \dots & h(n_h) \end{bmatrix}. \quad (3.10)$$

Then,  $\mathbf{H} = \mathbf{H}_1 + \mathbf{H}_2 + \dots + \mathbf{H}_{m_s}$ , where  $\mathbf{H}_m$  corresponds to the  $m$ th decaying sinusoid of (3.9),

$$\begin{aligned} \mathbf{H}_m &= \begin{bmatrix} h_m(1) & h_m(q+1) & \dots & h_m(q(q-1)+1) \\ \vdots & \vdots & & \vdots \\ h_m(q) & h_m(2q) & \dots & h_m(n_h) \end{bmatrix} \\ &= \frac{\mu_m e^{i\phi_m}}{2} \begin{bmatrix} z_m \\ z_m^2 \\ \vdots \\ z_m^q \end{bmatrix} \begin{bmatrix} 1 & z_m^q & \dots & z_m^{(q-1)q} \end{bmatrix} \\ &\quad + \frac{\mu_m e^{-i\phi_m}}{2} \begin{bmatrix} \bar{z}_m \\ \bar{z}_m^2 \\ \vdots \\ \bar{z}_m^q \end{bmatrix} \begin{bmatrix} 1 & \bar{z}_m^q & \dots & \bar{z}_m^{(q-1)q} \end{bmatrix}, \end{aligned} \quad (3.11)$$

where  $z_m = e^{i\omega_m - \beta_m}$ . We see that each of the  $m_s$  terms that make up  $\mathbf{H}$ , can be written as a sum of two rank-1 matrices, i.e., as a rank-2 matrix. As long as  $z_m$  are distinct, the matrix  $\mathbf{H}$  will have rank  $2m_s$  [50]. In general,  $\mathbf{H}$  could be a non-square matrix, but here we consider the square case, as to simplify the exposition.

In light of this discussion, the problem to be solved would ideally be

$$\begin{aligned} & \underset{\mathbf{h}}{\text{minimize}} && \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{h})\|_2^2 \\ & \text{s.t.} && \text{rank}(\mathcal{R}_{qq}(\mathbf{h})) \leq 2m_s, \end{aligned} \quad (3.12)$$

or, equivalently, using the form of (3.6),

$$\underset{\mathbf{h}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{h})\|_2^2 + \mathcal{I}_{2m_s}(\mathcal{R}_{qq}(\mathbf{h})), \quad (3.13)$$

where the penalty function,  $\mathcal{I}_{2m_s}$ , is the indicator function for the set of matrices of rank at most  $2m_s$ . This problem, however, is non-convex and NP-hard [99]. Furthermore, the exact rank,  $2m_s$ , is typically not known. Instead of the constrained problem of (3.12) and (3.13), one could consider the penalized version,

$$\underset{\mathbf{h}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{h})\|_2^2 + \lambda \text{rank}(\mathcal{R}_{qq}(\mathbf{h})). \quad (3.14)$$

Here,  $\lambda \geq 0$  is a parameter controlling the degree of penalization of the rank of the solution, in analogy to  $\lambda_T$  in (3.8). This transforms the problem of not knowing the rank  $2m_s$ , into the problem of choosing  $\lambda$ , but the problem remains NP-hard [47]. Considering that the rank of a matrix is equal to the number of non-zero singular values, (3.14) is equivalent to

$$\underset{\mathbf{h}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{h})\|_2^2 + \lambda \|\sigma(\mathcal{R}_{qq}(\mathbf{h}))\|_0. \quad (3.15)$$

where  $\|\cdot\|_0$  is the  $\ell_0$ -pseudo-norm and  $\sigma(\mathbf{H}) : \mathbb{R}^{q \times q} \rightarrow \mathbb{R}^q$  is the function that takes an  $\mathbb{R}^{q \times q}$ -matrix and returns a vector of its singular values in non-increasing order, given its singular value decomposition (SVD)  $\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ , for  $\mathbf{U} \in \mathbb{R}^{q \times q}$ ,  $\mathbf{S} \in \mathbb{R}^{q \times q}$ ,  $\mathbf{V}^T \in \mathbb{R}^{q \times q}$ . The NP-hardness persists, however, so in order to be able to find an approximate solution to (3.15), the problem needs to be relaxed. This can be done in different ways, which will now be further explored.

### 3.3.1 Nuclear Norm Regularization

In order to achieve low-rank solutions, throughout this paper, two different kinds of penalty functions will be used and compared. Firstly, the popular approach of using the  $\ell_1$ -norm instead of the  $\ell_0$ -norm in the penalization of  $\sigma(\mathbf{H})$  [106],

$$\underset{\mathbf{h}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{h})\|_2^2 + \lambda_{\text{NN}} \|\mathcal{R}_{qq}(\mathbf{h})\|_*, \quad (3.16)$$

where  $\|\mathbf{H}\|_*$  denotes the nuclear norm of  $\mathbf{H}$ , defined as  $\|\mathbf{H}\|_* = \text{trace}((\mathbf{H}^*\mathbf{H})^{1/2}) = \sum_{n=1}^q \sigma_n(\mathbf{H}) = \|\sigma(\mathbf{H})\|_1$ , and  $\lambda_{\text{NN}} \geq 0$  is analogous to  $\lambda$  in (3.14). This is the first of the two optimization problems we will aim to solve in Section 3.4. The advantage of problem (3.16) is that it is convex and can be solved efficiently [107]. The disadvantage is that this solution will be biased, in that the singular values will be shrunk [47].

Several different methods have been proposed in an attempt to remedy the issue of the shrinking of large singular values. Among them, the weighted nuclear norm [108], which has been generalized to so called *generalized singular value thresholding* in [109].

### 3.3.2 Quadratic Envelope

The second approach to achieving a low-rank solution that will be used in this paper, is the use of a function created in attempt to closer emulate the  $\ell_0$ -pseudonorm, the so called *quadratic envelope* of the  $\ell_0$ -norm. The quadratic envelope of a function consists of pointwise, quadratic approximations, that nowhere overestimate the original function. For the function  $g : \mathbb{R}^n \rightarrow \mathbb{R}$ , the quadratic envelope, at the point  $\mathbf{u} \in \mathbb{R}^n$ , is defined as

$$\mathcal{Q}_\gamma(g)(\mathbf{u}) = \sup_{\alpha \in \mathbb{R}, \mathbf{v} \in \mathbb{R}^n} \left\{ \alpha - \frac{\gamma}{2} \|\mathbf{u} - \mathbf{v}\|^2 : \alpha - \frac{\gamma}{2} \|\mathbf{z} - \mathbf{v}\|^2 \leq g(\mathbf{z}), \forall \mathbf{z} \right\}, \quad (3.17)$$

i.e., the supremum of all functions of the form  $\alpha - \frac{\gamma}{2} \|\mathbf{u} - \mathbf{v}\|^2$ , that nowhere overestimates the original function  $g(\cdot)$ . The parameter  $\gamma > 0$  controls how close this envelope will be to the original function. A smaller value of  $\gamma$  means a larger degree of smoothing of the function, whereas  $\mathcal{Q}_\gamma(g)$  converges point-wise to  $g$  as  $\gamma \rightarrow \infty$  [100].

Firstly, it is readily verified that

$$\lambda \|\mathbf{u}\|_0 = \lambda \sum_{n=1}^{n_u} \|u_n\|_0. \quad (3.18)$$

In other words, the problem decouples in all dimensions, and we can, without loss of generality, look at a one-dimensional version of the aforementioned function, i.e.,  $\mathbf{u} \equiv u$ . This has been considered before, in works like [110] and [111], with support of the theory in [112] and [113], but we restate the results here. Further, since this is to be applied to the singular values, we will, to simplify the exposition, only consider  $u \geq 0$ .

### Proposition

The quadratic envelope of  $\lambda_{\text{QE}}\|u\|_0$  is given by

$$\mathcal{Q}_\gamma(\lambda_{\text{QE}}\|\cdot\|_0)(u) = \begin{cases} \sqrt{2\lambda_{\text{QE}}\gamma}u - \frac{\gamma}{2}u^2 & , u < \mu \\ \lambda_{\text{QE}} & , \mu \leq u \end{cases} \quad (3.19)$$

or, equivalently,

$$\mathcal{Q}_\gamma(\lambda_{\text{QE}}\|\cdot\|_0)(u) = \lambda_{\text{QE}} - \frac{\gamma}{2}\max(\mu - u, 0)^2, \quad (3.20)$$

where  $\mu = \sqrt{2\lambda_{\text{QE}}/\gamma}$ , and  $\lambda_{\text{QE}} \geq 0$  is analogous to  $\lambda_{\text{NN}}$  in (3.16).

### Proof

The proof is based on [113], but for the convenience of the reader we iterate it here. See Appendix 3.7.

The major benefit of the quadratic envelope, as compared to the nuclear norm, manifests itself in (3.19). For values  $u$  larger than  $\mu$ , the quadratic envelope is constant. The differences between the nuclear norm and the quadratic envelope in approximating  $\|u\|_0$  are further illustrated in Fig. 3.1. The effect of the parameter  $\gamma$  is noticeable, in that a larger  $\gamma$  yields an envelope closer to  $\|u\|_0$ . Further, it can be seen that the main effect of increasing  $\lambda_{\text{QE}}$  is raising the level at which the function remains constant.

A particular case of the penalty (3.20), when  $\gamma = 2$ , has been considered in [102]. Adding (3.20) to (3.5) yields the final form of the second optimization problem we are aiming to solve in Section 3.4,

$$\underset{\mathbf{h}}{\text{minimize}} \frac{1}{2}\|\mathbf{y} - \mathcal{A}(\mathbf{h})\|_2^2 + \mathcal{Q}(\mathcal{R}_{qq}(\mathbf{h})), \quad (3.21)$$

where we, for ease of notation, for a matrix  $\mathbf{X}$ , define

$$\mathcal{Q}(\mathbf{X}) = \mathcal{Q}_\gamma(\lambda_{\text{QE}}\|\cdot\|_0)(\sigma(\mathbf{X})). \quad (3.22)$$

As previously mentioned,  $\gamma$  is a parameter that controls the fidelity of  $\mathcal{Q}_\gamma(g)$  to the original function  $g$ . If  $\gamma \leq \|\mathcal{A}\|$ , then the objective function in (3.21) is convex. This is of course a very desirable property as these problems are generally considered comparatively easy to solve. However, for  $\gamma > \|\mathcal{A}\|^{21}$  we have the following upside: if  $\hat{\mathbf{h}}$  is a (strict) local minimizer of (3.21), then  $\hat{\mathbf{h}}$  is also a (strict) local minimizer of (3.15). In addition to this, their global minima coincide [100, Theorem 5.2].

<sup>1</sup>In [114] this was erroneously given as  $\gamma > \|\mathcal{A}\|$

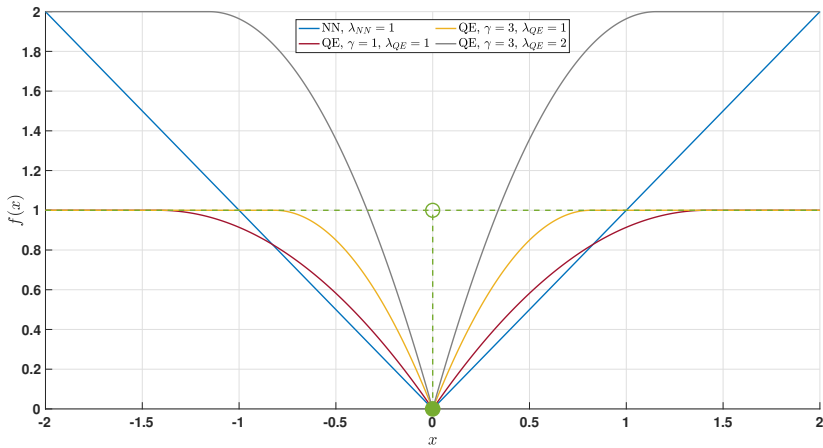


Figure 3.1: Nuclear norm and quadratic envelope, for varying parameter values. For reference, the  $\ell_0$ -pseudo-norm is indicated by the dashed green line.

## 3.4 Algorithm

### 3.4.1 Preliminaries

As previously indicated, the optimization problem will be solved using the proximal gradient method, also known as forward-backward splitting (FBS), which can be seen in Algorithm 1. It has become very popular recently due to its ability to handle non-smooth functions. For more on the proximal gradient method, see e.g., [115, 116, 105]. The general idea is to alternate between taking a step in the negative direction of the gradient of the smooth data-fit term  $f$ , and trying to minimize the (possibly non-smooth) penalty function  $g$ , using the so-called *proximal operator*. For a function  $g(\mathbf{V}) : \mathbb{R}^{q \times p} \rightarrow \mathbb{R}$ , with the step length parameter  $\rho > 0$ , it is defined as

$$\text{prox}_{\rho g}(\mathbf{V}) = \arg \min_{\mathbf{Z}} \left( g(\mathbf{Z}) + \frac{1}{2\rho} \|\mathbf{Z} - \mathbf{V}\|_F^2 \right). \quad (3.23)$$

In Algorithm 1, the gradient step,  $\mathbf{u}^k - \rho \nabla f(\mathbf{u}^k)$ , is to be interpreted as the forward step, and the proximal mapping,  $\text{prox}_{\rho g}(\cdot)$ , as the backward step. Here,  $L$  denotes the Lipschitz constant of  $\nabla f$ . In order to guarantee that Algorithm 1 will converge, a few additional technical assumptions have to be made about  $f$  and  $g$  respectively. These are all met for the problems considered in this paper, more details can be found in [117].

---

**Algorithm 1:** Proximal Gradient Method (PG)
 

---

 Set  $\mathbf{u}^0 \in \mathbb{R}^n$  and  $\rho \in (0, 2L^{-1}]$ 
**for**  $k = 0, 1, \dots$  **do**
 $\quad \mathbf{u}^{k+1} = \text{prox}_{\rho g}(\mathbf{u}^k - \rho \nabla f(\mathbf{u}^k))$ 


---

In order to facilitate the exposition we will introduce a couple of linear operators. The operator  $\mathcal{H} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  denotes the reversion of the order of the elements in a vector, i.e.  $\mathcal{H}(\mathbf{x}) = [\mathbf{x}_{n_x}, \mathbf{x}_{n_x-1}, \dots, \mathbf{x}_1]^T$ . The operator  $\mathcal{F} : \mathbb{R}^n \rightarrow \mathbb{C}^n$  denotes the discrete Fourier transform (DFT), and  $\mathcal{F}^{-1} : \mathbb{C}^n \rightarrow \mathbb{R}^n$  the inverse DFT.<sup>2</sup> The operator  $\mathcal{P}^k : \mathbb{R}^n \rightarrow \mathbb{R}^{n+k}$  denotes padding of a vector with  $k$  zeroes at the end. Finally, the operator  $\mathcal{C}_k^n : \mathbb{R}^m \rightarrow \mathbb{R}^{n-k+1}$ , where  $n \geq k$  and  $m \geq n - k + 1$ , is the linear operator that picks out the  $k$ th till the  $n$ th element of a vector, i.e.,  $\mathcal{C}_k^n(\mathbf{y}) = [\mathbf{y}_k, \mathbf{y}_{k+1}, \dots, \mathbf{y}_n]$ .

### 3.4.2 Proposed Algorithm

The previously introduced operators  $\mathcal{A}$  and  $\mathcal{A}^*$  can now be expressed in terms of these operators,

$$\mathcal{A}(\mathbf{h}) = \mathcal{C}_{n_h}^{n_x} \left( \mathcal{F}^{-1} \left( \mathcal{F}(\mathcal{P}^{n_x-1}(\mathbf{h})) \odot \mathcal{F}(\mathcal{P}^{n_h-1}(\mathbf{x})) \right) \right), \quad (3.24)$$

and

$$\mathcal{A}^*(\mathbf{y}) = \mathcal{C}_{n_y}^{n_x} \left( \mathcal{F}^{-1} \left( \mathcal{F}(\mathcal{P}^{n_x-n_h}(\mathcal{H}(\mathbf{x}))) \odot \mathcal{F}(\mathcal{P}^{2n_x-n_h-n_y}(\mathbf{y})) \right) \right), \quad (3.25)$$

respectively.

The first step in iteratively solving (3.16) and (3.21), respectively, is the gradient step, with respect to the data-fit term  $f$  in the cost function. The gradient of (3.5) is given by

$$\nabla f = \mathcal{A}^*(\mathcal{A}(\mathbf{h})) - \mathcal{A}^*(\mathbf{y}). \quad (3.26)$$

For the backward step, there is a seemingly complicating fact in that this is to be taken with respect to a matrix and its singular values. This, however, turns out not to be a problem. After the gradient step, the vector is reshaped into a matrix, and an SVD is performed on that matrix.

---

<sup>2</sup>In the general case,  $\mathcal{F}^{-1} : \mathbb{C}^n \rightarrow \mathbb{C}^n$ , but because of the conjugate symmetry properties of the signals considered here, we will have  $\mathcal{F}^{-1} : \mathbb{C}^n \rightarrow \mathbb{R}^n$

### Proposition

The proximal operator of  $\mathcal{Q}(\mathbf{H})$  is given by

$$\text{prox}_{\rho\mathcal{Q}(\cdot)}(\mathbf{H}) = \mathbf{U} \mathcal{D} \left( \text{prox}_{\rho\mathcal{Q}_\gamma(\lambda_{\text{QE}}\|\cdot\|_0)}(\sigma(\mathbf{H})) \right) \mathbf{V}^T, \quad (3.27)$$

and the proximal operator of  $\lambda_{\text{NN}}\|\sigma(\mathbf{H})\|_1$  is given by

$$\text{prox}_{\rho\lambda_{\text{NN}}\|\cdot\|_*}(\mathbf{H}) = \mathbf{U} \mathcal{D} \left( \text{prox}_{\rho\lambda_{\text{NN}}\|\cdot\|_1}(\sigma(\mathbf{H})) \right) \mathbf{V}^T, \quad (3.28)$$

where  $\mathcal{D} : \mathbb{R}^q \rightarrow \mathbb{R}^{q \times q}$  is the operation of creating a diagonal  $q \times q$ -matrix, with the argument of the operator as the diagonal. The explicit expressions for  $\text{prox}_{\rho\mathcal{Q}(\cdot)}(\mathbf{H})$  and  $\text{prox}_{\rho\lambda_{\text{NN}}\|\cdot\|_*}(\mathbf{H})$  are given in Appendix 3.8.

### Proof

See Appendix 3.8. It may be noted that this is due to the orthogonal invariance of  $\mathcal{Q}(\mathbf{H})$  and  $\lambda_{\text{NN}}\|\sigma(\mathbf{H})\|_1$ , respectively [116].

After either of the proximal operators has acted on the vector of singular values, all that remains is to reassemble the matrix, with the adjusted singular values, and then vectorize, in order to get the current iterate of the estimated RIR. Finally, a check is made to see if the algorithm is making enough progress, i.e., if the new iterate is sufficiently different from the previous one, or if the iterative procedure should be terminated. These steps are summarized in Algorithm 2. Solving (3.16) or (3.21) differs only in the use of proximal operator. That is reflected in step 3 of Algorithm 2, in that 3a) corresponds to solving (3.16) and 3b) corresponds to solving (3.21).

The parameter  $\rho$  in (3.23) controls the balance between minimizing the penalty function, and staying close the current point  $\mathbf{V}$ , and serves as step-size for the proximal gradient algorithm. Small values of  $\rho$  will yield a  $\mathbf{Z}$  close to  $\mathbf{V}$ , whereas larger values of  $\rho$  will result in  $\mathbf{Z}$  being closer to the minimum of  $g(\mathbf{Z})$ . In order to assure that the sequence generated by Algorithm 1 is bounded, we must chose  $\rho \in (0, 2/L)$  [118].

We have that  $\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{z})\| \leq \|\mathcal{A}^*\mathcal{A}\|\|\mathbf{w} - \mathbf{z}\|$  i.e.,  $L \leq \|\mathcal{A}^*\mathcal{A}\|$ . For a linear operator, the norm of the operator is the same as for the corresponding matrix, i.e.,  $\|\mathcal{A}^*\mathcal{A}\| = \|\mathbf{X}^T\mathbf{X}\|$ , where  $\mathbf{X} \in \mathbb{R}^{n_y \times n_h}$  is the Toeplitz matrix s.t.  $\mathcal{A}(\mathbf{h}) = \mathbf{X}\mathbf{h}$ . Computing  $\|\mathbf{X}^T\mathbf{X}\|$  is, however, computationally demanding and something that should be avoided if possible. In this instance, it is possible to use a bound on  $\|\mathbf{X}\|$ . It is shown in [119] that  $\|\mathbf{X}\| \leq 2M$ , where  $M$  denotes the essential supremum of the absolute value of the Fourier series of  $\mathbf{x}$ . Taken

---

**Algorithm 2:** Low-Rank RIR Estimation Algorithm
 

---

Input:  $\mathbf{h}^0$ ,  $\rho$ ,  $\gamma$ ,  $\lambda$ ,  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\text{tol}$ ,  $\text{maxIter}$   
**for**  $k = 0, 1, \dots, \text{maxIter}$  **do**  
   1:  $\mathbf{w}^k = \mathbf{h}^k - \rho(\mathcal{A}^*(\mathcal{A}(\mathbf{h}^k)) - \mathcal{A}^*(\mathbf{y}))$  {Forward step}  
   2:  $\mathbf{W}^k = \mathcal{R}_{qq}(\mathbf{w}^k)$   
   3 a):  $\mathbf{H}^{k+1} = \text{prox}_{\rho\lambda_{\text{NN}}\|\cdot\|_*}(\mathbf{W}^k)$  {Backward step}  
   3 b):  $\mathbf{H}^{k+1} = \text{prox}_{\rho\mathcal{Q}(\cdot)}(\mathbf{W}^k)$  {Backward step}  
   4:  $\mathbf{h}^{k+1} = \mathcal{R}_{qq}^{-1}(\mathbf{H}^{k+1})$   
   **if**  $\|\mathbf{h}^{k+1} - \mathbf{h}^k\| / (\|\mathbf{h}^k\| + \delta) \leq \text{tol}$  **then**  
     **break**  
   **end if**  
**end for**

---

together with the well-established fact that, for any two linear operators  $\mathcal{A}$  and  $\mathcal{B}$ ,  $\|\mathcal{A}\mathcal{B}\| \leq \|\mathcal{A}\|\|\mathcal{B}\|$ , we have that  $L \leq 4M^2$ . In order to ensure that the strict inequality is upheld, we let

$$\rho = 0.99 (2M^2)^{-1}. \quad (3.29)$$

Finally, three more parameters need to be set by the user. A maximum number of iterations,  $\text{maxIter}$ , that the user is willing to run, a tolerance level,  $\text{tol}$ , to determine when the algorithm is not making sufficient progress anymore and should be terminated, and a small number  $\delta$  to avoid division by zero in the normalization of the norm of the difference between the new and the old iterate. Preliminary simulations showed that  $\text{maxIter} = 10^4$ ,  $\text{tol} = 10^{-4}$ , and  $\delta = 10^{-6}$  were suitable choices, and these values will be used throughout this paper.

### 3.4.3 Computational Complexity

In this Section we study the computational complexity of Algorithm 2, as a function of the length of the RIR,  $n_h$ . The difference between using the nuclear norm or quadratic envelope penalty is negligible, in terms of complexity per iteration. Asymptotically, as  $n_h \rightarrow \infty$ , the most demanding steps of the algorithm is step 3. In step 3, we perform a singular value decomposition of the  $\sqrt{n_h} \times \sqrt{n_h}$ -matrix  $\mathbf{W}^k$ , which requires  $\mathcal{O}(n_h^{3/2})$  multiplications. Then, two  $\sqrt{n_h} \times \sqrt{n_h}$ -matrix multiplications are carried out, which, if naively executed, requires  $\mathcal{O}(n_h^{3/2})$  multiplications.

In terms of wall time, for problems of the size studied in this paper, step 1 of the algorithm is the most demanding. Although the fast Fourier transform

(FFT) require only  $\mathcal{O}(n_h \log(n_h))$  multiplications, the fact that we in step 1 have to carry out 4 of these (including the IFFT's) each iteration, makes this the most time-consuming part of the algorithm.

## 3.5 Numerical Results

In order to visualize the dependence of the proposed and benchmark algorithms' performance on room related parameters, we will first provide numerical results using synthetically generated RIRs. Then, to demonstrate the practical applicability of the proposed algorithm, we proceed by presenting results of simulations using real-life RIRs. The accuracy of an RIR estimate,  $\hat{\mathbf{h}}$ , is measured by the *normalized misalignment*, defined as

$$\mathcal{M}_{\text{dB}}(\hat{\mathbf{h}}) = 20 \log_{10} \left( \frac{\|\hat{\mathbf{h}} - \mathbf{h}\|_2}{\|\mathbf{h}\|_2} \right), \quad (3.30)$$

where  $\mathbf{h}$  is the RIR we are aiming to estimate. In Sections 3.5.9 and 3.5.10 we will use slightly modified evaluation measures, which will be further explained there. Throughout this Section we will compare the performance of the two proposed methods to that of two benchmark methods. Firstly, ordinary least squares, i.e., without regularization, corresponding to solving (3.5). Secondly, least squares, with Tikhonov regularization, i.e., solving

$$\underset{\mathbf{h}}{\text{minimize}} \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{h})\|_2^2 + \lambda_{\text{T}} \|\mathbf{h}\|_2^2, \quad (3.31)$$

where  $\lambda_{\text{T}}$  controls the degree of regularization.

### 3.5.1 Impact of $T_{60}$

First we present simulations performed on synthetically generated RIRs, in order to show how the algorithms under comparison perform, as a function of the reverberation time,  $T_{60}$ , of the room. The  $T_{60}$  is defined as the time required for the sound level to drop 60 dB after switching off a stationary source. The RIRs are generated by letting each tap independently be drawn from a normal distribution, with an exponentially decaying envelope, corresponding to the given  $T_{60}$ . This is a stochastic reverberation model [120], and can be attributed to [121] and [122]. In this simulation example, for each value of  $T_{60}$ , taken from a grid of 9 values in the range [33, 300] ms, 10 synthetic RIRs were generated. The system was driven by a white Gaussian noise. Finally, independent white

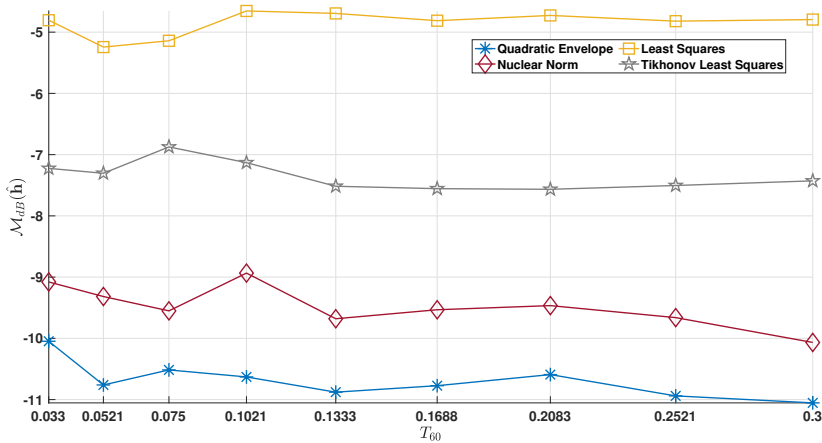


Figure 3.2: Normalized misalignment as a function of  $T_{60}$ .

Gaussian measurement noise was added, with signal-to-noise ratio ( $\text{SNR}_{\text{dB}}$ ) = 10, defined as

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left( \frac{P_S}{P_N} \right), \quad (3.32)$$

where  $P_S$  and  $P_N$  denote the power of the output signal without the noise, and the power of the noise, respectively. The RIR length,  $n_h$ , was set to correspond to the reverberation time, and  $n_y$  was set to  $1.2n_h$ , in order to have comparable overdetermined systems, for the various values of  $T_{60}$ . The hyperparameters  $\lambda_T$ ,  $\lambda_{\text{NN}}$ ,  $\lambda_{\text{QE}}$ , and  $\gamma$  were tuned for each individual setting, by finding the optimal values using another set of randomly generated RIRs. The results are shown in Fig. 3.2. There it can be seen that the performance is close to constant as a function of  $T_{60}$  for all the analyzed methods, and that the low-rank models outperform ordinary least squares, as well as Tikhonov-regularized least squares, with a slight preference for the quadratic envelope.

### 3.5.2 SMARD

In order to demonstrate the proposed methods' applicability to actual measured RIRs, we apply it to the *single- and multichannel audio recordings database* (SMARD) [1]. These recordings are sampled at 48 kHz, for 12 seconds, yielding impulse responses of  $576 \cdot 10^3$  taps. These are recorded at various source and receiver positions, and with varying equipment, in a room of size 7.34 m  $\times$  8.09 m  $\times$  2.87 m, with a reverberation time of approximately 0.15 s. In total,

the dataset contains 1008 RIRs. Two things should be noted regarding the numerical simulations performed here. Firstly, the RIRs are truncated at some discrete-time index  $N \ll 576 \cdot 10^3$ . Secondly, the truncated RIRs are set to start at the, in absolute value, largest value of the original RIR recording, i.e., at the point where the direct component hits the microphone.

### 3.5.3 Hyperparameter Tuning

As with any algorithm requiring hyperparameters, the parameters of the proposed algorithm need to be rigorously set, in order for the algorithm to be effective. In this Section we expand on the tuning of the hyperparameters for the compared methods. The dependence on the variables of the problem largely decouples between the hyperparameters, with the optimal choice of  $\gamma$  being impacted by the input signal  $\mathbf{x}$ , and  $\lambda_{QE}$ ,  $\lambda_{NN}$ , and  $\lambda_T$  depending on the SNR, much like in sparse estimation. Using a random subset of 20 of the RIRs of SMARD, we here vary the  $\text{SNR}_{\text{dB}}$  for white Gaussian measurement noise from 0 to 20 in steps of 2, and find the optimal choice of  $\lambda_{QE}$ ,  $\lambda_{NN}$ , and  $\lambda_T$  respectively, for each value of the SNR. We then perform linear regression to find models for  $\lambda_{QE}$ ,  $\lambda_{NN}$ , and  $\lambda_T$ , as a function of the SNR. This is done with  $n_h = 3600$ ,  $n_y = 4320$ ,  $\gamma = 10^2$  and with white Gaussian noise of unit power as input. In accordance with the results of this, we let  $\lambda_{QE} = 10^{0.4-0.14 \cdot \text{SNR}_{\text{dB}}}$ ,  $\lambda_{NN} = 10^{-3.7-0.08 \cdot \text{SNR}_{\text{dB}}}$ , and  $\lambda_T = 10^{3.6-0.1 \cdot \text{SNR}_{\text{dB}}}$ . For  $\lambda_{QE}$  and  $\lambda_{NN}$ , this can be illustrated in Fig. 3.3. Different choices of  $n_h$  and  $n_y$  would yield different values for the model parameters, hence simulations from hereon throughout this paper will be done with  $n_h = 3600$  and  $n_y = 4320$ , respectively. The performance of the proposed algorithms, as a function of SNR, will be discussed in Section 3.5.4.

As for  $\gamma$ , the main dependency is on the input  $\mathbf{x}$ , but the dependency is more involved. The optimal choice is impacted by the power of the input signal, as well as the length of the RIR,  $n_h$  (and thereby the length of the input signal,  $n_x$ ), and finally also the frequency content of the input signal. It is therefore very difficult to find a model that covers all possible choices of these variables. For a vast majority of the experiments performed, preliminary simulations showed that  $\gamma = 10^2$  was a good enough choice. In Sections 3.5.7 and 3.5.9, where the frequency content of the input signal is varied, each scenario requires its own optimized  $\gamma$ , which can be found in Table 3.1. In Section 3.5.8, where snippets of speech are used as input,  $\gamma = 10^{-2}$  proved to be a good choice.

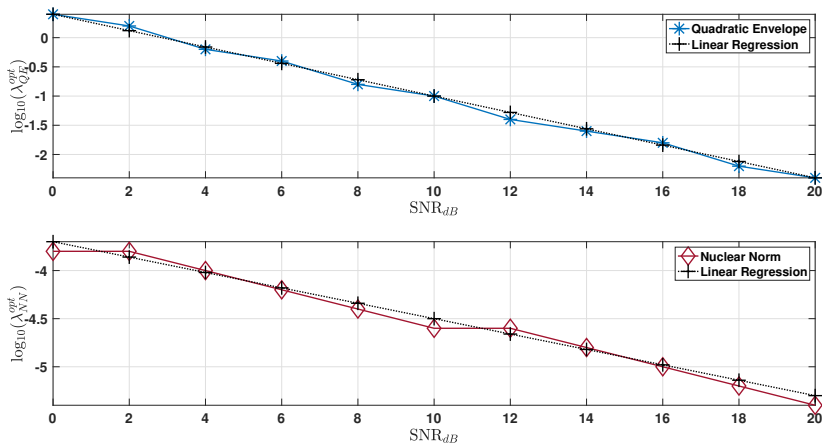


Figure 3.3: Linear regression for  $\lambda_{QE}$  (top) and  $\lambda_{NN}$  (bottom), respectively, as a function of  $SNR_{dB}$ .

### 3.5.4 SNR

The benefit of low-rank modeling of the RIR becomes particularly evident when the signal-to-noise ratio is low. With white Gaussian noise as input and values of  $\lambda_{QE}$ ,  $\lambda_{NN}$ ,  $\lambda_T$ , and  $\gamma$  as described in Section 3.5.3, and using all of the 1008 RIRs from SMARD, we showcase the adequacy of the proposed modeling framework. The results can be seen in Fig. 3.4. The low-rank enforcing methods are clearly preferable to ordinary least squares, as well as Tikhonov regularized least squares. As expected, the difference compared to ordinary least squares is particularly apparent when the SNR is low.

### 3.5.5 Convergence Analysis

For the least squares methods, ordinary and with Tikhonov regularization, we have closed-form expressions, and no iterative procedure is needed. For the nuclear norm and quadratic envelope, however, we here showcase the difference in speed of convergence, using part of the simulations done in Section 3.5.4. In Fig. 3.5, we see the averaged normalized misalignment for the two iterative methods, as a function of the iteration index, for the case where  $SNR_{dB} = 10$ . The difference in average number of iterations needed for convergence for the quadratic envelope and nuclear norm, for this particular value of SNR, was negligible. We see in Fig. 3.5 that the normalized misalignment, for a given number of iterations, is lesser for the quadratic envelope, compared to the

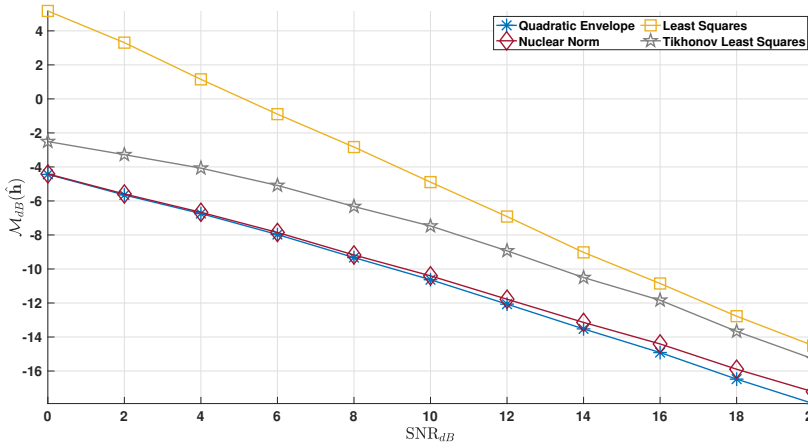


Figure 3.4: Normalized misalignment as a function of SNR<sub>dB</sub>.

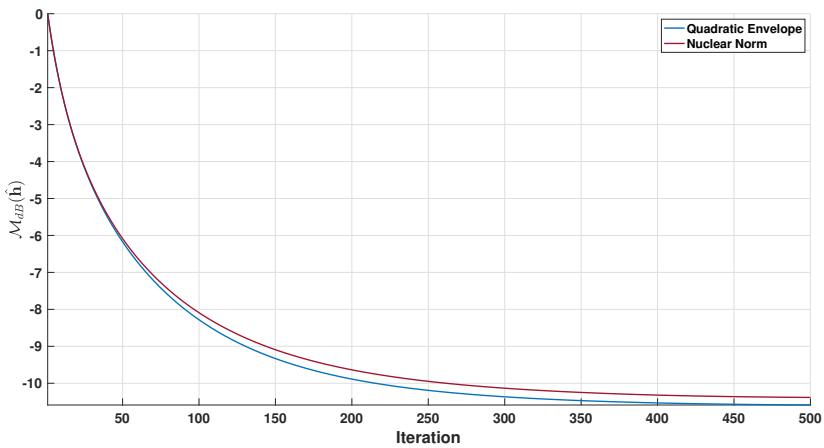


Figure 3.5: Normalized misalignment as a function of number of iterations.

nuclear norm. Preliminary simulations showed similar behaviour for other SNR values, both in terms of the relation between the number of iterations needed for convergence of the respective methods, and the decay of the normalized misalignment, as a function of the iteration index.

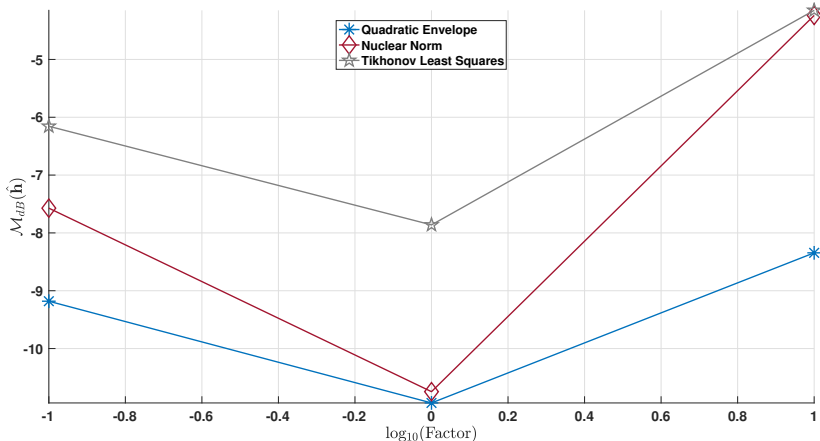


Figure 3.6: Normalized misalignment with perturbed parameters.

### 3.5.6 Hyperparameter Sensitivity

In Section 3.5.4 we saw that the quadratic envelope and nuclear norm regularizations performed similarly as a function of the SNR, when oracle knowledge of the SNR was assumed. Using all the 1008 RIRs of SMARD, with  $\text{SNR}_{\text{dB}} = 10$ , and white Gaussian noise as input, we here consider what happens if the algorithms are provided over- or underestimated values of their respective regularization parameters. According to the formulas provided in Section 3.5.3, we set  $\lambda_{\text{QE}} = 10^{0.4-0.14 \cdot 10}$ ,  $\lambda_{\text{NN}} = 10^{-3.7-0.08 \cdot 10}$ , and  $\lambda_{\text{T}} = 10^{3.6-0.1 \cdot 10}$  respectively. We then look at what happens if these parameters are multiplied by a factor of  $10^{-1}$  or  $10^1$ , so as to simulate an underestimation and an overestimation respectively. The averaged results are found in Fig. 3.6. There it can be seen that the quadratic envelope clearly is the method losing the least in accuracy, for the perturbed parameter values. This suggests that the quadratic envelope would be particularly useful in scenarios where the SNR is not known exactly. For reference, the average misalignment using ordinary least squares is on the order of  $-5$  dB, indicating that when the knowledge of the SNR is poor and the regularization parameter is overestimated, some types of regularization do more harm than good.

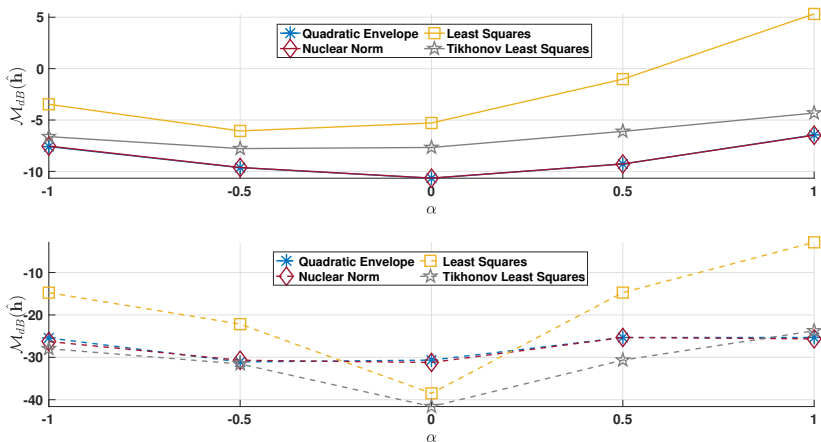


Figure 3.7: Normalized misalignment as a function of  $\alpha$ . Average (top) and standard deviation (bottom), respectively.

### 3.5.7 Frequency Content of Input Signal

In this Section, we investigate how the proposed method performs as a function of inverse frequency power of the input, i.e., colored noise with the power spectral density  $1/|f|^\alpha$ . As previously mentioned, the hyperparameter values' dependency on the variables of the problem is complicated. Therefore, when displaying the aptitude of the proposed method as a function of  $\alpha$ , the hyperparameters were tuned on 20 randomly selected RIRs. These parameter values are shown in Table 3.1. Then, with these hyperparameter values, simulations were done on all of the RIRs of SMARD, and with  $SNR_{dB} = 10$ . The results are displayed in Fig. 3.7. There it can be seen that the low-rank estimation methods, nuclear norm regularization as well as the quadratic envelope of the  $\ell_0$ -norm, are particularly successful, in comparison to ordinary least squares, when the color of the noise is not white. As for the performance of Tikhonov least squares, we note that it is fairly constant as a function of  $\alpha$ . Thus, the difference in performance between the low-rank methods and Tikhonov least squares is largest for  $\alpha$  close to zero.

### 3.5.8 Speech Signal Input

As indicated in Section 3.5.7, the proposed method provides the most improvement to the RIR estimation when the spectral excitation by the input signal is poor. We study this input by letting the driving signal be a speech

Table 3.1: Parameter values used when varying  $\alpha$ .

$\alpha$	$\gamma$	$\lambda_{\text{QE}}$	$\lambda_{\text{NN}}$	$\lambda_{\text{T}}$
-1	$10^{-4.2}$	$10^4$	$10^{-5.4}$	$10^2$
-0.5	$10^{-3.8}$	$10^4$	$10^{-5}$	$10^{2.2}$
0	$10^{-3.2}$	$10^4$	$10^{-4.6}$	$10^{2.6}$
0.5	$10^{-2.4}$	$10^4$	$10^{-4.8}$	$10^3$
1	$10^{-1.8}$	$10^4$	$10^{-5.6}$	$10^{3.6}$

signal. Yet again, we set all the parameter values in accordance with the discussion in Section 3.5.3 (note that here  $\gamma = 10^{-2}$ ), and use 100 randomly chosen RIRs from SMARD, all driven by 10 different snippets of anechoic male and female speech, chosen at random from the TSP Speech Database [123]. In order to make sure that the speech snippets contained enough power, they were all chosen with the criterion that  $\|\mathbf{x}\| \geq 2$ . The results can be seen in Fig. 3.8. Here, we have opted to not plot the normalized misalignment of the ordinary least squares estimation. It was consistently at least two orders of magnitude larger than the other methods, and the inclusion of it in Fig. 3.8 would cloud the difference between the other methods. It can be seen that, the low-rank methods outperform Tikhonov regularized least squares. This further strengthens the hypothesis that the low-rank modeling is most useful when the input spectral excitation is poor.

### 3.5.9 Input-output Relations

In order to further investigate the merit of the low-rank estimation, we will in this Section, as previously indicated, use a slightly modified evaluation measure. Instead of, as previously, looking at the distance between the estimated RIR and the measured RIR, we will here look at the following metric,

$$\mathcal{M}_{\text{dB}}^O(\hat{\mathbf{h}}) = 20 \log_{10} \left( \mathbb{E} \left[ \frac{\|\hat{\mathbf{h}} * \mathbf{x} - \mathbf{h} * \mathbf{x}\|_2}{\|\mathbf{h} * \mathbf{x}\|_2} \right] \right), \quad (3.33)$$

where  $\mathbf{x}$  is a previously unseen signal, with the same properties as the input signal generating the output  $\mathbf{y}$ , from which  $\hat{\mathbf{h}}$  is estimated, and the expectation is with respect to  $\mathbf{x}$ . We will refer to this as *normalized output misalignment*. Under these conditions, we replicate the experiment in Section 3.5.7. The same parameter values as in Section 3.5.7 are used, i.e., the ones found in Table 3.1. Here, a subset of 100 randomly chosen RIRs from SMARD were used,

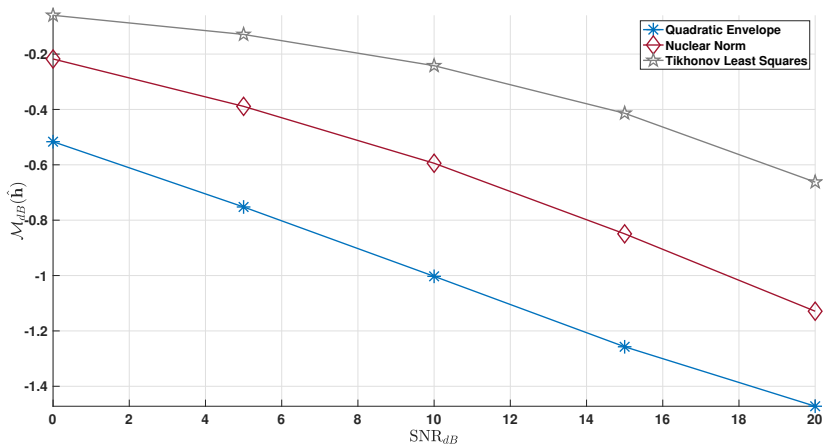


Figure 3.8: Average normalized misalignment, using speech signals, as a function of SNR<sub>dB</sub>.

and (3.33) was estimated using 50 different instances of  $\mathbf{x}$ , drawn from the same distribution as the input signal. The results can be seen in Fig. 3.9. It can be noticed that, for all but ordinary least squares, the normalized output misalignment is decreasing as a function of  $\alpha$ . One possible explanation for this is the discrepancy in spectral excitation. For  $\alpha = -1$ , poor estimation at a certain frequency is more likely to be masked by poor spectral excitation by the input signal, at that particular frequency. The benefit of the low-rank regularization is, however, evident all across the board.

### 3.5.10 Cramér–Rao Lower Bound

In this Section we study the mean squared error (MSE),

$$\text{MSE}(\hat{\mathbf{h}}) = \mathbb{E} \left[ \|\hat{\mathbf{h}} - \mathbf{h}\|_2^2 \right], \tag{3.34}$$

of the suggested estimators, in relation to theoretical bounds given by the Cramér–Rao Lower Bound (CRLB), the smallest variance that can be achieved by an unbiased estimator. Let

$$\Sigma_{\mathbf{h}} = \mathbb{E} \left\{ \left( \hat{\mathbf{h}} - \mathbf{h} \right) \left( \hat{\mathbf{h}} - \mathbf{h} \right)^T \right\} \tag{3.35}$$

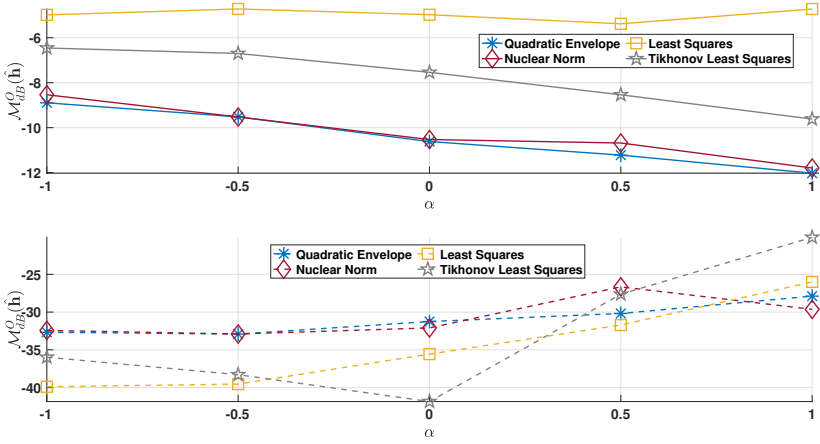


Figure 3.9: Normalized output misalignment as a function of  $\alpha$ , different evaluation signal. Average (top) and standard deviation (bottom), respectively.

denote the error covariance matrix of an unbiased estimator  $\hat{\mathbf{h}}$ . We have that  $\text{MSE}(\hat{\mathbf{h}}) = \text{tr}(\boldsymbol{\Sigma}_{\mathbf{h}})$ , and it can also be noticed that

$$\text{MSE}(\hat{\mathbf{h}}) = \text{Var}(\hat{\mathbf{h}}) + \text{Bias}(\hat{\mathbf{h}}, \mathbf{h})^2. \quad (3.36)$$

We then define

$$\text{MSE}_{\text{dB}}(\hat{\mathbf{h}}) = 20 \log_{10}(\text{MSE}(\hat{\mathbf{h}})). \quad (3.37)$$

The CRLB of the variance of low-rank estimation algorithms has been explored by Tang *et. al* in [124]. For the convenience of the reader, we here reiterate the principal content of their results. Let  $\boldsymbol{\Sigma}$  denote the measurement noise covariance matrix. We then have that

$$\boldsymbol{\Sigma}_{\mathbf{h}} \geq \mathbf{P}[\mathbf{P}^T (\mathbf{X}^T \boldsymbol{\Sigma}^{-1} \mathbf{X}) \mathbf{P}]^{-1} \mathbf{P}^T, \quad (3.38)$$

with

$$\mathbf{P} = [\bar{\mathbf{V}} \otimes \tilde{\mathbf{U}} \quad \tilde{\mathbf{V}} \otimes \tilde{\mathbf{U}} \quad \tilde{\mathbf{V}} \otimes \bar{\mathbf{U}}] \in \mathbb{R}^{q^2 \times 2qr - r^2}, \quad (3.39)$$

$\mathbf{U} = [\tilde{\mathbf{U}} \quad \bar{\mathbf{U}}] = [\mathbf{U}_1 \quad \mathbf{U}_2 \quad \dots \quad \mathbf{U}_r \quad \mathbf{U}_{r+1} \quad \dots \quad \mathbf{U}_n]$ ,  $\mathbf{V} = [\tilde{\mathbf{V}} \quad \bar{\mathbf{V}}] = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \dots \quad \mathbf{V}_r \quad \mathbf{V}_{r+1} \quad \dots \quad \mathbf{V}_p]$ ,  $\mathbf{U}_j$  and  $\mathbf{V}_j$  denoting the  $j$ th column of  $\mathbf{U}$  and  $\mathbf{V}$  respectively,  $r$  is the rank, and we remind the reader that  $\mathbf{U}$  and  $\mathbf{V}$  come from the SVD of  $\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ , and that  $\mathbf{X}$  is the Toeplitz matrix from (3.3). Further,  $\mathbf{A} \geq \mathbf{B}$  means that  $\mathbf{A} - \mathbf{B}$  is positive semidefinite. It follows from (3.38) that, by taking the trace of both sides of the inequality, using the

cyclic property of the trace operation, and the fact that  $\mathbf{P}^T \mathbf{P} = \mathbf{I}_{(2q-r)r}$ ,

$$\text{MSE} \geq \text{tr} \left( [\mathbf{P}^T (\mathbf{X}^T \boldsymbol{\Sigma}^{-1} \mathbf{X}) \mathbf{P}]^{-1} \right), \quad (3.40)$$

i.e., the right-hand side of (3.40) is the CRLB.

As previously touched upon, an RIR from SMARD will, when matricized, not be low-rank in a strict sense. In order to avoid complicated error bounds depending on the misspecification of the model, we will therefore, in this Section, make low-rank approximations of the matricized RIRs, in order to be able to study the properties of the estimation, in relation to the CRLB. Hence,  $\mathbf{h}$  will in this Section be a rank-20 approximation of the recorded and truncated RIR. This is a meritable approximation, as per the discussion in [95]. A rigorous analysis of performance bounds for a not strictly low-rank RIR that has (purposefully) been misspecified as low-rank, is beyond the scope of this paper, and will be left for future work.

Comparison is made with two different theoretical bounds. The first one is under the assumption that the low-rank property is unknown. For the second bound, the low-rank property is known. Yet again, the parameter values from Section 3.5.3 are used. Experiments were performed using 50 randomly chosen RIRs from SMARD, and for each RIR running 20 Monte Carlo-simulations. The results, averaged over the Monte Carlo-simulations and the different RIRs, can be seen in Fig. 3.10. As expected, the first bound is greater than the second one. The low-rank methods constantly outperform the first bound, which is to be expected, since they are using information not available to that theoretical bound. In relation to the second bound, the low-rank methods outperform the bound for lower values of SNR. This is possible since the estimator is biased. Also this outperformance is appealing to intuition. For a lower SNR, the gain that comes from the regularization in terms of improved conditioning, can reduce variance so that it outweighs the added bias, something that is generally not possible with an increasing SNR.

## 3.6 Conclusion

In this paper we have presented a method for exploiting low-rank, or close to low-rank, properties of RIRs. The low-rank penalty serves as a regularizer and prevents an overfitted solution, similar to the idea of sparse estimation. The low-rank penalty is enforced on a matricization of the coefficient vector, and consists of either the nuclear norm, or the quadratic envelope of the  $\ell_0$ -pseudo-norm of the vector of singular values of the matrix. The results of the proposed methods are compared to those of ordinary least squares, as well as Tikhonov

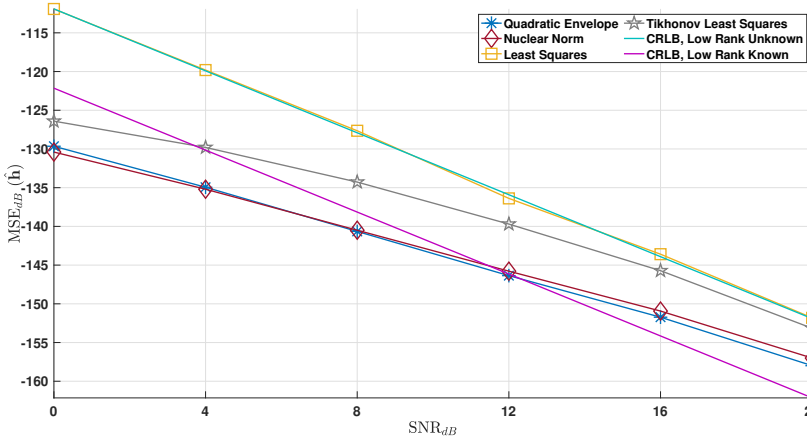


Figure 3.10: Comparison with CRLB.

regularized least squares. It is shown that the low-rank methods work very well in general, but excel particularly when either the SNR is low, or when the input spectral excitation is poor, which can stem from the input signal being either colored noise or a speech signal. The difference between the two investigated low-rank methods, i.e., nuclear norm and quadratic envelope, is comparatively small. The main benefit of the nuclear norm is the convexity, and that only one hyper parameter has to be tuned. The quadratic envelope, on the other hand, displays superior robustness to a suboptimally chosen regularization parameter, in the case where oracle knowledge about the SNR cannot be assumed.

Future work will focus on investigating the properties of the error introduced by the low-rank estimation. A structure to this error could indicate possible improvements, as well as potential applications, for the methods proposed here. The interplay between the bias introduced by a misspecification, and the possible reduced estimation variance, is another research area we aim to explore.

### 3.7 Appendix: Quadratic Envelope of $\lambda \|x\|_0$

When trying to find  $\mathcal{Q}_\gamma(\lambda_{QE} \|\cdot\|_0)(u)$ , we notice two things. Firstly,  $\mathcal{Q}_\gamma(\lambda_{QE} \|\cdot\|_0)(0) = 0$  and secondly,  $\forall u$  s.t.  $|u| \geq C$ , for some constant  $C > 0$ ,  $\mathcal{Q}_\gamma(\lambda_{QE} \|\cdot\|_0)(u) = \lambda_{QE}$ . The question is then what the value of  $C$  is, and what values  $\mathcal{Q}_\gamma(\lambda_{QE} \|\cdot\|_0)(u)$  takes on for  $u \in (-C, C)$ . The limiting factor

for the envelope is that  $\lambda_{\text{QE}}\|0\|_0 = 0$ , i.e., the negative quadratic function must pass through, or below, the origin. This yields the constraint that  $\alpha - \frac{\gamma}{2}v^2 \leq 0 \Leftrightarrow \alpha \leq \frac{\gamma}{2}v^2$ . Looking for the supremum, we want to chose the largest possible  $\alpha$ , hence  $\alpha = \frac{\gamma}{2}v^2$ . Inserting this into (3.17) and simplifying yields

$$\sup_{v \in \mathbb{R}} \left\{ \gamma uv - \frac{\gamma u^2}{2}; \gamma zv - \frac{\gamma z^2}{2} \leq \lambda_{\text{QE}}\|z\|_0, \forall z \in \mathbb{R} \right\}. \quad (3.41)$$

We know this is valid in  $z = 0$  so we can write

$$\sup_{v \in \mathbb{R}} \left\{ \gamma uv - \frac{\gamma u^2}{2} : \gamma zv - \frac{\gamma z^2}{2} \leq \lambda_{\text{QE}}, \forall z \in \mathbb{R} \setminus \{0\} \right\}. \quad (3.42)$$

If the peak of the parabola is at or below  $v = \lambda_{\text{QE}}$  then, naturally, all other points on the parabola will be as well. Therefore we insert  $z = v$  and simplify

$$\sup_{v \in \mathbb{R}} \left\{ \gamma uv - \frac{\gamma u^2}{2} : \frac{\gamma v^2}{2} \leq \lambda_{\text{QE}} \right\} = \sup_{v \in \mathbb{R}} \left\{ \gamma uv - \frac{\gamma u^2}{2} : |v| \leq \mu \right\}, \quad (3.43)$$

where  $\mu = \sqrt{2\lambda_{\text{QE}}/\gamma}$ . The expression we aim to maximize,  $\gamma uv - \frac{\gamma u^2}{2}$ , is growing in  $v$  if  $v > 0$ , and in  $-v$  if  $v < 0$ . In other words, we want to choose  $v$  as big as possible, in modulo, without ever overestimating  $g(z)$ , yielding  $y = \text{sign}(u)\mu$ . Inserting this value for  $v$  into  $\mathcal{Q}_\gamma(\lambda_{\text{QE}}\|\cdot\|_0)(u)$  and setting this expression equal to  $\lambda_{\text{QE}}$  yields that this occurs at  $u = \pm\mu$ , thus, the sought after point  $C = \mu$ . Taken together we have

$$\mathcal{Q}_\gamma(\lambda_{\text{QE}}\|\cdot\|_0)(u) = \begin{cases} \sqrt{2\lambda_{\text{QE}}}\gamma u - \frac{\gamma}{2}u^2 & , u < \mu \\ \lambda_{\text{QE}} & , \mu \leq u \end{cases} \quad (3.44)$$

or, equivalently,

$$\mathcal{Q}_\gamma(\lambda_{\text{QE}}\|\cdot\|_0)(u) = \lambda_{\text{QE}} - \frac{\gamma}{2}\max(\mu - u, 0)^2. \quad (3.45)$$

### 3.8 Appendix: Proximal Operators

The regularization terms of (3.16) and (3.21) are both *orthogonally invariant* [116]. A function  $g : \mathbb{R}^{q \times q} \rightarrow \mathbb{R}$  is said to be orthogonally invariant if

$$g(\mathbf{S}) = g(\mathbf{U}\mathbf{S}\mathbf{V}^T) \quad (3.46)$$

for all  $\mathbf{S} \in \mathbb{R}^{q \times q}$ ,  $\mathbf{U} \in \mathbb{R}^{q \times q}$ , and  $\mathbf{V}^T \in \mathbb{R}^{q \times q}$ , where  $\mathbf{U}$  and  $\mathbf{V}^T$  are orthogonal matrices. More specifically, this means that

$$g(\mathbf{H}) = g(\mathcal{D}(\sigma(\mathbf{H}))). \quad (3.47)$$

A function  $g$  is orthogonally invariant if and only if  $g = \tilde{g} \circ \sigma$ , where  $\tilde{g} : \mathbb{R}^q \rightarrow \mathbb{R}$  is absolutely symmetric. More details can be found in [116]. This implies that, for the proximal operator of an orthogonally invariant function  $g$ , it holds that

$$\text{prox}_{\rho g}(\mathbf{H}) = \mathbf{U} \mathcal{D}(\text{prox}_{\rho \tilde{g}}(\sigma(\mathbf{H}))) \mathbf{V}^T, \quad (3.48)$$

i.e., the proximal step is performed on the vector of singular values [116]. For the nuclear norm regularizer, the corresponding proximal operator  $\text{prox}_{\lambda_{\text{NN}} \|\cdot\|_1}(\sigma(\mathbf{H}))$  in (3.28) is given by element-wise applying the so-called soft-thresholding operator  $\mathcal{S}_{\lambda_{\text{NN}}} : \mathbb{R}^q \rightarrow \mathbb{R}^q$  [125],

$$\mathcal{S}_{\lambda_{\text{NN}}}(\sigma(\mathbf{H})_n) = \max(\sigma(\mathbf{H})_n - \lambda_{\text{NN}}, 0), \quad (3.49)$$

$n = 1, 2, \dots, q$  where  $\sigma(\mathbf{H})_n$ , denotes the  $n$ th element of  $\sigma(\mathbf{H})$ .

For the penalty term in (3.21), we make use of the fact that the function decouples and that we therefore, without loss of generality, can consider the one-dimensional case. Further, since this is to be used for singular values, we consider only  $u \geq 0$  as argument and  $z \geq 0$  as function value. Finally, in order for the proximal operator to be single-valued, we must have that  $\gamma\rho < 1$ , since this guarantees that the curvature is strictly positive everywhere.

The proximal operator, (3.23) returns the argument that minimizes the sum of two terms; the original function  $\mathcal{Q}_\gamma(\lambda_{\text{QE}} \|\cdot\|_0)(z)$ , and the proximity term,  $\frac{1}{2\rho}(u - z)^2$ . What is sought after is the argument  $z$  that minimizes the sum of these, for a given  $u$ . First, it can be concluded that for  $u = 0$ , where the proximal operator is not differentiable, the optimal choice of  $z$  is 0. This renders a cost of 0, so no choice can be better. Next, the question is if there is a  $u > 0$ , for which choosing a  $z > 0$  yields a lower cost than  $z = 0$  and, if so, what  $u$ ? For  $z > 0$  the proximal operator is differentiable. We can set that derivative equal to zero to find the optimal choice of  $z$  there,  $z_{\text{opt}}$ . This yields that

$$z_{\text{opt}} = \begin{cases} \frac{u - \mu\gamma\rho}{1 - \gamma\rho} & , 0 < u < \mu \\ u & , \mu \leq u, \end{cases} \quad (3.50)$$

where  $\mu = \sqrt{2\lambda_{\text{QE}}/\gamma}$ . Comparing the costs and determining when it is better to choose  $z > 0$ , as opposed to  $z = 0$ , is equivalent of finding for which  $x$

$$\lambda_{\text{QE}} - \frac{\gamma}{2} \max(\mu - z_{\text{opt}}, 0)^2 + \frac{1}{2\rho}(u - z_{\text{opt}})^2 < \frac{1}{2\rho}u^2. \quad (3.51)$$

Inserting  $z_{\text{opt}}$  from (3.50) yields that  $x > \mu\gamma\rho$ . Taken together,

$$\text{prox}_{\rho \mathcal{Q}_\gamma(\lambda_{\text{QE}} \|\cdot\|_0)}(u) = \begin{cases} 0 & , 0 \leq u \leq \mu\gamma\rho \\ \frac{u - \mu\gamma\rho}{1 - \gamma\rho} & , \mu\gamma\rho < u < \mu \\ u & , \mu \leq u \end{cases}. \quad (3.52)$$

To the best of the authors' knowledge, the explicit expression has not been published before.



## Chapter 4

# Fast Low-latency Convolution by Low-rank Tensor Approximation

Fast Low-latency Convolution by Low-rank Tensor  
Approximation

Martin Jälmbly, Filip Elvander, and Toon van Waterschoot

Published in *Proc. 2023 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2023

© 2023 IEEE. Reprinted, with permission, from:

M. Jälmy, F. Elvander, T. van Waterschoot, "Fast low-latency convolution by low-rank tensor approximation," *Proc. 2023 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 1–5, 2023.

Changes include layout, representation, and minor editing aspects.

The candidate's contributions as first author include all performed research and writing of the manuscript under guidance from the co-authors.

## Abstract

In this paper we consider fast time-domain convolution, exploiting low-rank properties of an impulse response (IR). This reduces the computational complexity, speeding up the convolution, without introducing latency. Previous work has considered a truncated singular value decomposition (SVD) of a two-dimensional matricization, or reshaping, of the IR. We here build upon this idea, by providing an algorithm for convolution with a three-dimensional tensorization of the IR. We provide simulations using real-life acoustic room impulse responses (RIRs) of various lengths, convolving them with music, as well as speech signals. The proposed algorithm is shown to outperform the comparable existing algorithm in terms of signal quality degradation, for all considered scenarios, without increasing the computational complexity, or the memory usage.

## 4.1 Introduction

Ever since the seminal works by Cooley and Tukey, [53], and Stokham, [54], a popular approach to efficiently compute the convolution of two time-domain signals has been to perform it in the discrete frequency domain. The convolution theorem states that convolution in the time-domain equals point-wise multiplication in the frequency domain, allowing for considerable reduction of the computational complexity in most cases, owing to the computational efficiency of the fast Fourier transform (FFT) algorithm. The frequency-domain convolution has been further improved by methods such as overlap-add (OLA) and overlap-save (OLS), and partitioned convolution (see e.g. [64]). Frequency-domain convolution is, however, block-based and inevitably introduces input-output latency [126]. Perceptual convolution, introduced by Lee *et al.* in [127], is another possible way to speed up the convolution. Here, a perceptual criterion is used in order to reduce the computational complexity, without considerable signal quality degradation. Yet another approach to accelerate computations is optimizing the implementation with respect to the processor architecture, and the use of graphics processing units (see e.g. [128] and the references therein).

In this paper we look to exploit the (approximate) low-rank structure of an impulse response (IR) in order to carry out fast time-domain convolution. Low-rank modeling is used in a variety of applications, such as image analysis [129], matrix completion [125], and recently also acoustic signal processing [130, 95, 114], and has been shown to yield possibly more compact models. Utilizing low rank in time-domain convolution has been considered before by Atkins *et al.* in [126]. Therein, the authors consider a low-rank approximation

of a two-dimensional matricization of the IR. In this work, we extend upon this idea and show how the concept of low-rank convolution can be generalized, and we provide a detailed algorithm for the three-dimensional case. This is motivated by the findings in [95], where it was shown that higher-order tensorization yields a lower IR approximation error, given a fixed compression rate. By performing the convolution in the time domain, low latency can be achieved. In [126], to further promote a compact representation of the IR, the resulting finite impulse response (FIR) filter is approximated by an infinite impulse response (IIR) filter. Herein, the proposed method is concerned with exploiting low-rank structure of IRs on FIR form, with possible extensions of the IIR methodology being a topic of future research. Accordingly, as to allow for a fair comparison between the algorithm in [126] and the herein proposed method, the IIR approximation is not considered in the numerical section.

This paper is organized as follows: first Section 4.1 is concluded with an introduction of the notation used in this paper. Then, in Section 4.2, we introduce the signal model. Low-rank convolution is explained in Section 4.3, where we first give an account of the algorithm from [126], then introduce the proposed algorithm, and lastly expand on its complexity. Simulation results are presented in Section 4.4, and in Section 4.5 we present our conclusions.

### 4.1.1 Notation

We denote scalars, vectors, matrices, and tensors by lower case (e.g.,  $h$ ), bold lowercase (e.g.,  $\mathbf{h}$ ), bold uppercase (e.g.,  $\mathbf{H}$ ), and calligraphic letters (e.g.,  $\mathcal{H}$ ), respectively. Linear operators are also denoted by calligraphic letters, but it will be clear from context what is considered. The selection of one or several elements from a vector, matrix, or tensor will be denoted by square brackets, e.g.  $\mathbf{H}[m : n, j]$  is a vector containing the  $m$ th till the  $n$ th element of the  $j$ th column of  $\mathbf{H}$ . The symbol  $\circ$  denotes the outer product, i.e.,  $(\mathbf{x}_1 \circ \mathbf{x}_2 \circ \dots \circ \mathbf{x}_D)[j_1, j_2, \dots, j_D] = \mathbf{x}_1[j_1]\mathbf{x}_2[j_2] \dots \mathbf{x}_D[j_D]$ .

## 4.2 Signal Model

In this work we consider a discrete-time impulse response (IR)  $h(k)$ , for  $k = 0, 1, \dots, n_h - 1$ , arranged in the vector  $\mathbf{h} \in \mathbb{R}^{n_h}$ , as well as a discrete-time signal  $x(k)$ , for  $k = 1, 2, \dots, n_x$ , arranged in the vector  $\mathbf{x} \in \mathbb{R}^{n_x}$ . The

convolution of these vectors yield the discrete-time output

$$y(k) = \sum_{n=0}^{n_h-1} h(n)x(k-n), \quad (4.1)$$

for  $k = 1, 2, \dots, n_y$ , with corresponding vector  $\mathbf{y} \in \mathbb{R}^{n_y}$ , where  $n_y = n_h + n_x - 1$ . Generally, throughout this paper, an element is considered to be 0, if the index is out of its defined range, equivalent to appropriate zero-padding.

Many signals can be considered *low-rank*, in the sense that if the signal vector is reshaped into a matrix, the matrix will have low rank (see e.g., [45]). This persists for higher-order tensorization, where the rank of a tensor  $\mathcal{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2} \times \dots \times n_{s_D}}$ , with  $n_{s_d}$  denoting the size of the  $d$ th dimension,  $d = 1, 2, \dots, D$ , is defined as the smallest number of rank-1 tensors needed to generate the tensor as their sum. As IRs used in real-world applications are often estimated from noisy measurements, a recorded (estimated) IR will rarely be low-rank in a strict sense. Therefore, in this paper, *low-rank* is used in a less strict sense, i.e., a low-rank approximation will render a small approximation error. An example of this are acoustic room impulse responses (RIRs), which we have explored in previous work [95].

## 4.3 Low-rank Convolution Algorithm

### 4.3.1 Partitioned Truncated SVD Filter

As to give background to the herein proposed method for fast convolution, we first give a brief description of the method in [126]. Assuming  $n_h = n_{s_1}n_{s_2}$ , for  $n_{s_1}, n_{s_2} \in \mathbb{N}$ , an output sample  $y(k)$  of the convolution in (4.1) can instead be written as

$$y(k) = \sum_{j=1}^{n_{s_2}} \mathbf{x}_k^{(j)T} \mathbf{h}^{(j)}, \quad (4.2)$$

where  $\mathbf{h}^{(j)} \triangleq [h((j-1)n_{s_1}) \dots h(jn_{s_1}-1)]^T$  and  $\mathbf{x}_k^{(j)} \triangleq [x(k-(j-1)n_{s_1}) \dots x(k-jn_{s_1}+1)]^T$ , for  $j = 1, 2, \dots, n_{s_2}$ . This is merely a rearrangement, as in (4.1),  $y(k)$  is written as an inner product computed from vectors of length  $n_h = n_{s_1}n_{s_2}$ , whereas in (4.2), it is written as the sum of  $n_{s_2}$  inner products, where the corresponding vectors are of length  $n_{s_1}$ . Furthermore, the IR  $\mathbf{h}$  can be reshaped into a matrix  $\mathbf{H} = [\mathbf{h}^{(1)} \dots \mathbf{h}^{(n_{s_2})}] \in \mathbb{R}^{n_{s_1} \times n_{s_2}}$ . Assuming that this matrix is rank-1, i.e., it can be written as the outer product  $\mathbf{H} = \mathbf{s}_1 \circ \mathbf{s}_2$ , for two vectors  $\mathbf{s}_1 \in \mathbb{R}^{n_{s_1}}$  and  $\mathbf{s}_2 \in \mathbb{R}^{n_{s_2}}$ , we have that

$$\mathbf{H} = [\mathbf{s}_1\mathbf{s}_2[1] \quad \mathbf{s}_1\mathbf{s}_2[2] \quad \dots \quad \mathbf{s}_1\mathbf{s}_2[n_{s_2}]], \quad (4.3)$$

i.e., the  $j$ th column of  $\mathbf{H}$ , corresponding to  $\mathbf{h}^{(j)}$ , is the vector  $\mathbf{s}_1$  scaled by  $\mathbf{s}_2[j]$ ,  $j = 1, 2, \dots, n_{s_2}$ . The following property is readily verified,

$$\mathbf{x}_k^{(j)} = \mathbf{x}_{k+an_{s_1}}^{(j+a)}, a \in \mathbb{Z}. \quad (4.4)$$

Because of (4.3) and (4.4), only the first inner product of the sum in (4.2) has to be computed per output sample  $k$ , as

$$y(k) = \left( \mathbf{x}_k^{(1)T} \mathbf{s}_1 \right) \mathbf{s}_2[1] + \sum_{j=2}^{n_{s_2}} \left( \mathbf{x}_k^{(j)T} \mathbf{s}_1 \right) \mathbf{s}_2[j]. \quad (4.5)$$

The other inner products of the sum, i.e.,  $\mathbf{x}_k^{(j)T} \mathbf{s}_1$ , for  $j = 2, \dots, n_{s_2}$ , have already been computed for a previous time sample, and can therefore be fetched from memory and multiplied with the appropriate entry from  $\mathbf{s}_2$ . This reduces the number of multiplications per sample to be carried out, from  $n_h = n_{s_1} n_{s_2}$  to  $n_{s_1} + n_{s_2}$ . These ideas can be extended to a matrix  $\mathbf{H}$  of arbitrary rank  $R$ . Instead of  $\mathbf{H}$  being just the outer product of two vectors, it is now a sum of  $R$  outer products,

$$\mathbf{H} = \mathbf{S}_1 \mathbf{S}_2^T = \sum_{r=1}^R \mathbf{S}_1[:, r] \circ \mathbf{S}_2[:, r] = \sum_{r=1}^R \mathbf{S}_1[:, r] \mathbf{S}_2[:, r]^T, \quad (4.6)$$

for  $\mathbf{S}_1 \in \mathbb{R}^{n_{s_1} \times R}$ , and  $\mathbf{S}_2 \in \mathbb{R}^{n_{s_2} \times R}$ . We have a similar pattern as in (4.3) in that

$$\mathbf{h}^{(j)} = \mathbf{H}[:, j] = \sum_{r=1}^R \mathbf{S}_1[:, r] \mathbf{S}_2[j, r], \quad (4.7)$$

which enables us to extend (4.5) to

$$y(k) = \sum_{r=1}^R \left( \mathbf{x}_k^{(1)T} \mathbf{S}_1[:, r] \mathbf{S}_2[1, r] + \sum_{j=2}^{n_{s_2}} \mathbf{x}_k^{(j)T} \mathbf{S}_1[:, r] \mathbf{S}_2[j, r] \right), \quad (4.8)$$

where only  $R$  inner products have to be computed for each time sample. Similar to (4.5), this reduces the number of multiplications to  $R(n_{s_1} + n_{s_2})$ .

### 4.3.2 Fast Convolution by Tensor Approximation

Next, we show how the ideas from [126] can be extended, and propose a detailed algorithm for the three-dimensional case. Let the IR  $\mathbf{h}$  be reshaped into a tensor

$\mathcal{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2} \times \dots \times n_{s_D}}$ , and assume that  $\mathcal{H}$  is of rank  $R$ . Then, analogously to (4.6),

$$\mathcal{H} = \sum_{r=1}^R \mathbf{S}_1[:, r] \circ \mathbf{S}_2[:, r] \circ \dots \circ \mathbf{S}_D[:, r], \quad (4.9)$$

where  $\mathbf{S}_d \in \mathbb{R}^{n_{s_d} \times R}$ ,  $d = 1, 2, \dots, D$ , and in analog to (4.7) we have that

$$\mathcal{H}[:, j_2, j_3, \dots, j_D] = \sum_{r=1}^R \mathbf{S}_1[:, r] \mathbf{S}_2[j_2, r] \dots \mathbf{S}_D[j_D, r]. \quad (4.10)$$

The equality of (4.4) can be generalized according to

$$\mathbf{x}_k^{(j_2, j_3, \dots, j_D)} = \mathbf{x}_{k + \sum_{m=2}^D a_m \prod_{d=1}^{m-1} n_{s_d}}^{(j_2+a_2, j_3+a_3, \dots, j_D+a_D)}, \quad (4.11)$$

where  $\mathbf{x}_k^{(j_2, j_3, \dots, j_D)} \in \mathbb{R}^{n_{s_1}}$  is a vector containing the  $n_{s_1}$  latest samples of  $\mathbf{x}$ , in reversed order, starting at  $x(k - \sum_{m=2}^D (j_m - 1) \prod_{d=1}^{m-1} n_{s_d})$ , and  $a_2, a_3, \dots, a_D \in \mathbb{Z}$ . The pattern from (4.2) extends to

$$y(k) = \sum_{j_2=1}^{n_{s_2}} \dots \sum_{j_D=1}^{n_{s_D}} \mathbf{x}_k^{(j_2, j_3, \dots, j_D)T} \mathbf{h}^{(j_2, j_3, \dots, j_D)}, \quad (4.12)$$

where  $\mathbf{h}^{(j_2, j_3, \dots, j_D)} = \mathcal{H}[:, j_2, j_3, \dots, j_D]$  is a vector containing  $n_{s_1}$  consecutive elements of  $\mathbf{h}$ , starting at  $h(\sum_{m=2}^D (j_m - 1) \prod_{d=1}^{m-1} n_{s_d})$ . Subsequently, the property of (4.8) is generalized to

$$y(k) = \sum_{r=1}^R \sum_{j_2=1}^{n_{s_2}} \dots \sum_{j_D=1}^{n_{s_D}} \mathbf{x}_k^{(j_2, \dots, j_D)T} \mathbf{S}_1[:, r] \mathbf{S}_2[j_2, r] \dots \mathbf{S}_D[j_D, r]. \quad (4.13)$$

Here, it may be noted that only  $R$  inner products of length  $n_{s_1}$  have to be computed for each  $k$ , reducing the number of multiplications to  $R \sum_{d=1}^D n_{s_d}$ . However, if naively implemented, the sum in (4.13) will yield many superfluous operations, computing inner products where one of the vectors contains only zeroes. In order to fully exploit the IR structure and to maximize efficiency, it is therefore important to keep track of which operations actually need to be carried out. We here propose an explicit algorithm for the case  $D = 3$ .

Let  $\mathcal{H} = \sum_{r=1}^R \mathbf{S}_1[:, r] \circ \mathbf{S}_2[:, r] \circ \mathbf{S}_3[:, r]$ , where  $\mathcal{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2} \times n_{s_3}}$  and  $\mathbf{S}_d \in \mathbb{R}^{n_{s_d} \times R}$ , for  $d = 1, 2, 3$ . The operator  $\mathcal{I} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  denotes the reversion of the order of the elements in a vector, i.e.,  $\mathcal{I}(\mathbf{x}) = [x(n_x) \ x(n_x - 1) \ \dots \ x(1)]^T$ , and  $\mathbf{0}_n \in \mathbb{R}^n$  is a vector of zeros. We introduce the matrix  $\mathbf{C} \in \mathbb{R}^{n_h \times R}$  to store the already computed inner products. The

---

**Algorithm 3:** Fast Low-latency Convolution by Low-rank 3-D Tensor Approximation
 

---

 Input:  $\mathcal{H} = \sum_{r=1}^R \mathbf{S}_1[:, r] \circ \mathbf{S}_2[:, r] \circ \mathbf{S}_3[:, r]$ ,  $\mathbf{x}$ 

 Output:  $\mathbf{y}$ 
**for**  $k = 1, 2, \dots, n_y$  **do**

   **for**  $r = 1, 2, \dots, R$  **do**

      **if**  $k \leq n_{s_1} + n_x - 1$  **then**

         $z_b = \max(k - n_x, 0)$ ;
 
         $z_a = \max(n_{s_1} - k, 0)$ ;
 
         $x_b = \max(k - n_{s_1} + 1, 1)$ ;
 
         $x_e = \min(k, n_x)$ ;
 
         $\mathbf{x}_k = [\mathbf{0}_{z_b}^T \quad \mathcal{I}(\mathbf{x}[x_b : x_e])^T \quad \mathbf{0}_{z_a}^T]^T$ ;
 
         $\mathbf{C}[\text{mod}(k - 1, n_h) + 1, r] = \mathbf{x}_k^T \mathbf{S}_1[:, r]$ ;
 
         $y(k) = \mathbf{S}_2[1, r] \mathbf{S}_3[1, r] \mathbf{C}[\text{mod}(k - 1, n_h) + 1, r]$ ;
 
       $l = \max(\lfloor (k - n_x) / n_{s_1} \rfloor + 1, 2)$ ;
 
       $u = \min(\lfloor (k - 1) / n_{s_1} \rfloor + 1, n_{s_2} n_{s_3})$ ;
 
      **for**  $c = l : 1 : u$  **do**

         $j_2 = \text{mod}(c - 1, n_{s_2}) + 1$ ;
 
         $j_3 = \lfloor (c - 1) / n_{s_2} \rfloor + 1$ ;
 
         $\tilde{c} = k - ((j_2 - 1)n_{s_1} + (j_3 - 1)n_{s_1}n_{s_2}) - 1$ ;
 
         $y(k) = y(k) + \mathbf{S}_2[j_2, r] \mathbf{S}_3[j_3, r] \mathbf{C}[\text{mod}(\tilde{c}, n_h) + 1, r]$ ;
 

rationale of the algorithm is to, for each  $k$ , first compute the  $R$  necessary inner products, with appropriate zero-padding, store to memory, and add to  $y(k)$  with appropriate scaling with corresponding element from  $\mathcal{H}$ . Next, the remaining non-zero inner products of the sum of (4.13) are fetched from memory, scaled with the corresponding entry of  $\mathcal{H}$  and added to  $y(k)$ . The proposed algorithm is summarized in Algorithm 3.

### 4.3.3 Complexity

As noted by the authors of [126], an output sample  $y(k)$  requires  $R(n_{s_1} + n_{s_2})$  multiply-add instructions in the two-dimensional case, whereas a conventional time-domain FIR filter requires  $n_h$  operations, where we remind the reader that  $n_h = \prod_{d=1}^D n_{s_d}$ . The computational complexity for a general,  $D$ -dimensional tensorization, is  $R \sum_{d=1}^D n_{s_d}$ , meaning that the computational complexity of the proposed algorithm generalizes the one of [126]. Furthermore, the contribution

to the end result from the different entries in the sums of (4.13) are independent from each other. It is therefore possible to parallelize the computations.

The two-dimensional algorithm from [126] requires a memory of size  $R(n_{s_1} + n_{s_2} + n_h) + n_{s_1}$  variables, compared to  $2n_h$  for a conventional FIR filter. For the proposed method, a memory of size  $R(\sum_{d=1}^D n_{s_d} + n_h) + n_{s_1}$  is required, i.e., the memory complexity of the proposed algorithm directly generalizes the one of [126].

## 4.4 Results

We denote by  $\Upsilon(\mathcal{H}) = R \sum_{d=1}^D n_{s_d}$ , with  $D = 2, 3$ , for the two algorithms compared here ( $D = 2$  for [126], and  $D = 3$  for Algorithm 3), the number of multiply-add instructions needed for a low-rank convolution of rank  $R$ . Further, by

$$C(\mathcal{H}) = 1 - \frac{\Upsilon(\mathcal{H})}{n_h}, \quad (4.14)$$

where  $C(\mathcal{H}) \in [0, 1)$ , we denote the *complexity reduction*, relative to time-domain convolution. For  $C(\mathcal{H}) = 0$  there is no complexity reduction, whereas for  $C(\mathcal{H})$  closer to 1, the degree of complexity reduction is larger. The accuracy is measured by the *normalized output mean-squared error*

$$\mathcal{M}_{\text{dB}}^O(\mathcal{H}) = 20 \log_{10} \left( \mathbb{E} \left[ \frac{\|\mathcal{H} * \mathbf{x} - \mathbf{h} * \mathbf{x}\|_2}{\|\mathbf{h} * \mathbf{x}\|_2} \right] \right), \quad (4.15)$$

where  $\mathbf{h}$  denotes the original IR, and  $\mathcal{H} * \mathbf{x}$  is the low-rank convolution, as defined in (4.13). The low-rank approximations of 2D-matrices are performed using a truncated SVD. For the numerical computation of the tensor decompositions we use the high-level function *cpd* of the Matlab toolbox Tensorlab [84]. To simplify the exposition, we will only consider the square case, i.e.,  $n_{s_1} = n_{s_2} = n_{s_3}$ .

To demonstrate the performance of the proposed algorithm, we first apply it to RIRs from the *single- and multichannel audio recordings database* (SMARD) [1], and speech signals from the TSP speech database [123]. The RIRs from SMARD are from a regular office-sized room, with a reverberation time<sup>1</sup> of 0.15 seconds. The speech signals from TSP are simple utterances by both male and female speakers, in an anechoic environment. Both SMARD and TSP are sampled at 48 kHz. From the databases we randomly choose 100 RIRs and 100 speech signals. The RIRs are set to start right before the arrival of the direct

<sup>1</sup>The time required for the sound level to drop 60 dB after switching off a stationary source.

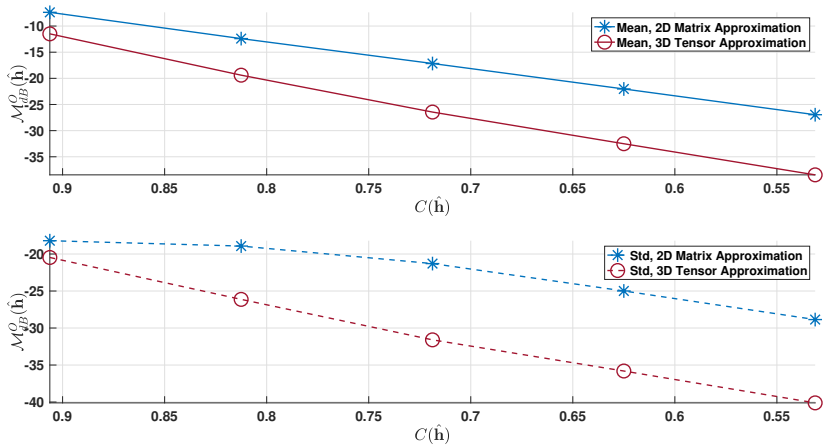


Figure 4.1: Normalized output mean-squared error, as a function of compression rate, for speech signals and short RIRs: mean (top) and standard deviation (bottom).

component and set to be of length  $n_h = 4096$  samples. The speech signals are truncated at one second, i.e.,  $n_x = 48000$ . The averaged results are shown in Fig. 4.1. There it can be seen that the proposed algorithm outperforms the algorithm from [126] for all considered values of the compression rate. Further it can be noted that the standard deviation for the proposed method is lower throughout.

Next we showcase that the proposed algorithm works very well also for longer RIRs, which we take from [131], a binaural RIR database recorded at RWTH Aachen University. We use 24 different RIRs, recorded at 48 kHz in a lecture room, with a reverberation time of 0.78 seconds, and we let  $n_h = 46656$ . As input signals we use music from the EBU-SQAM database [132], which contains snippets of reverberant music from various genres, sampled at 44.1 kHz. We use 30 seconds of pieces by ABBA, Verdi, and Händel, respectively. To have a sampling frequency matching that of the RIRs, we upsample the music to 48 kHz using Matlab’s *resample*, yielding  $n_x = 1.44 \cdot 10^6$ . The averaged results are shown in Fig. 4.2. The proposed algorithm is superior to the algorithm from [126] also under these circumstances. The standard deviation is lower for the proposed algorithm in this case as well.

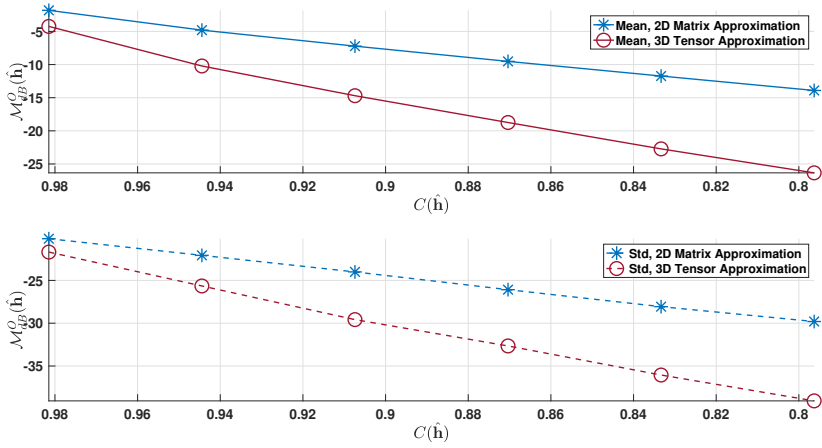


Figure 4.2: Normalized output mean-squared error, as a function of compression rate, for music signals and long RIRs: mean (top) and standard deviation (bottom).

## 4.5 Conclusions

In this paper we have shown that the ideas of low-rank convolution, previously presented for two dimensions, are extendable to higher dimensions, to obtain fast low-latency convolution algorithms. We have outlined the ideas and provided an algorithm for a three-dimensional tensorization of the IR, and shown that this outperforms the previously presented method we extend upon, in terms of signal quality degradation, without increasing the computational complexity, or the memory usage. This is shown with simulations using real-life RIRs of various lengths, convolved with both music and speech signals. Future work will focus on the development of an algorithm for tensors of arbitrary dimensions, and how that algorithms performs with respect to perceptual measures.



## Chapter 5

# Compression of Room Impulse Responses for Compact Storage and Fast Low-latency Convolution

Compression of Room Impulse Responses for Compact Storage and Fast Low-latency Convolution

Martin Jälmbly, Filip Elvander, and Toon van Waterschoot

Submitted for publication, Dec. 2023.

The candidate's contributions as first author include all performed research and writing of the manuscript under guidance from the co-authors.

## Abstract

Room impulse responses (RIRs) are used in several applications, such as augmented reality and virtual reality. These applications require a large number of RIRs to be convolved with audio, under strict latency constraints. In this paper we consider the compression of RIRs, in conjunction with fast time-domain convolution. We consider three different methods of RIR approximation for the purpose of RIR compression, and compare them to state-of-the-art compression. The methods are evaluated using several standard objective quality measures, both channel-based and signal-based. We also propose a novel low-rank-based algorithm for fast time-domain convolution, and show how the convolution can be carried out without the need to decompress the RIR. Numerical simulations are performed using RIRs of different lengths, recorded in three different rooms. It is shown that compression using low-rank approximation is a very compelling option to the state-of-the-art Opus compression, as it performs as well or better than on all but one considered measure, with the added benefit of being amenable to fast time-domain convolution.

## 5.1 Introduction

Modeling the acoustics of a room as a linear time-invariant system, the room impulse response (RIR) describes the impact of the room on an acoustic excitation signal, from a certain source position to a certain receiver position. The availability of the RIR, or an accurate estimate thereof, is imperative to a multitude of acoustic signal processing tasks, such as source localization [85], speech dereverberation [38], auralization [57, 133], source separation [86], listening room compensation [134], and echo cancellation [69]. There are several ways of modeling the RIR. Among the more popular ones are the infinite impulse response (IIR) (see e.g., [87, 71]) and finite impulse response (FIR) (see e.g., [87, 72]) models. The IIR model offers the possibility of a more compact representation, however with the downside of possible difficulties estimating the filter parameters [73], and potential issues with instability [81]. The FIR model is simple and straightforward, but with the disadvantage that comparatively many coefficients are needed to accurately represent the RIR [73]. For example, for an office-sized room, the FIR model can be several thousands of taps long [38]. A concert hall, on the other hand, can have a reverberation time of a couple of seconds [135], which at a sampling rate of 48 kHz yields an RIR with a length on the order of  $10^5$  samples. This can be prohibitive from both a memory requirement and computational complexity point of view, when using the RIR for convolution [88, 89, 136].

In recent years, archaeoacoustics and the cultural heritage preservation of acoustic scenes has gained increased interest from the research community, see e.g., [137] and references therein. However, in order to faithfully reconstruct the sound field in a room, the spatial resolution of the grid of measurements needs to be on the order of 10 cm [51]. Considering that the RIR depends on both the source and receiver position, even for a small room, the number of required source/receiver configurations for which the RIR has to be measured and stored will be in the millions, hence amounting to hundreds of gigabytes of data for the acoustic representation of a single room, indicating a significant need for compact representations of RIRs.

The computational challenges posed by long RIRs are particularly apparent in acoustic signal processing applications requiring low input-output latency, such as virtual video conferencing [55], augmented/mixed reality [56] and virtual reality [57, 58], sound zone control [59, 60], network music performance [61], and artificial reverberation [2]. In this paper, we consider rendering techniques based on convolution, highlighting the need for fast, low-latency convolution with long RIRs.

Ever since the works of Cooley and Tukey [53], and Stockham[54], a popular approach has been to carry out convolution in the frequency domain. With the help of the convolution theorem, which states that (circular) convolution in the (discrete) time domain is equivalent to point-wise multiplication in the (discrete) frequency domain, one is able to significantly reduce the computational burden of convolution in most cases, owing to the computational efficiency of the fast Fourier transform (FFT) algorithm. Frequency-domain convolution has since been further improved by methods such as overlap-add (OLA) and overlap-save (OLS), and partitioned convolution. For an overview of these methods see e.g., [64, 138]. A drawback of frequency-domain convolution is, however, that it is block-based, and therefore inevitably introduces latency. Further, partitioned frequency-domain filters are subject to restrictions with regard to assembling them into networks of filters (in parallel or serial structure), which is not the case for time-domain filters [66]. Another possible way to attempt to speed up the computations is by perceptual convolution [127]. There, the convolution is simplified, based on a perceptual criterion. The number of frequency-domain multiplications, and the memory storage, are reduced by up to 60%, without considerable quality degradation. Another path is optimization with respect to processor architecture, and the use of graphics processing units (see e.g., [128] and references therein). Yet another approach is to effectively shorten the RIR by treating the different parts of the RIR separately. For example, in [139], convolution is carried out for the first parts of the RIR, corresponding to the direct component and early reflections. The late reverberation, however, is modeled as a velvet noise sequence, yielding a very sparse FIR filter. Instead of

being convolved with the sparse FIR, the input signal is propagated in the delay line of the filter, and only the samples coinciding with a non-zero component of the sparse FIR are added together to yield the output.

In this paper we consider RIR compression and fast low-latency time-domain convolution based on three different methods; truncation, (hard) thresholding, and low-rank approximation. The exploitation of the (approximate) low-rank structure of reshaped RIRs is something we have considered in previous work. The physical motivation for it, and its applicability to real-life RIRs, was demonstrated in [95]. How the low-rank structure can be exploited when estimating RIRs from noisy input-output relations was shown in [114] and the simultaneous compression of multiple RIRs was considered in [140]. Atkins *et al.* showed in [126] how this low-rank structure can be exploited in time-domain convolution, an idea we extended upon in recent work [141]. Jaderberg *et al.* showed in [142] how speeding up convolutional neural networks can be done by leveraging low rank, but the authors consider dimensions no higher than 3.

The contribution of this paper is threefold. Firstly, we provide an extensive comparison of the aforementioned compression methods, with respect to several objective quality measures, both channel-based and signal-based. Secondly, we propose an approximate fast time-domain convolution method based on  $N$ -D low-rank tensor approximation of an RIR. This yields lower computational complexity than traditional time-domain convolution, and lower latency than FFT-based fast convolution. Thirdly, we show how the problem of compression and fast time-domain convolution can be handled within the same framework. This comes with the major advantage that the compressed RIR does not need to be decompressed before it can be used for convolution.

This paper is organized as follows: first, Section 5.1 is concluded with an introduction of the notation used throughout the paper, as well as the introduction of the signal model. In Section 5.2, the different RIR approximations considered for RIR compression are introduced. In Section 5.3, convolution by low-rank approximation is introduced. Section 5.4 introduces the different objective quality measures that will be used for evaluation. Numerical results are presented in Section 5.5, and finally, conclusions are presented in Section 5.6.

### 5.1.1 Notation and Signal Model

We denote scalars, vectors, matrices, and tensors by lowercase (e.g.,  $h$ ), bold lowercase (e.g.,  $\mathbf{h}$ ), bold uppercase (e.g.,  $\mathbf{H}$ ), and calligraphic letters (e.g.,  $\mathcal{H}$ ), respectively. Sets are also denoted by calligraphic letters, but it will be clear from context what is considered. The selection of one or several elements from a

vector, matrix, or tensor will be denoted by square brackets, e.g.  $\mathbf{H}[m : n, j]$  is a vector containing the  $m$ th till the  $n$ th element of the  $j$ th column of  $\mathbf{H}$ , and the hat symbol,  $\hat{\cdot}$ , indicates an approximated quantity. The symbol  $\circ$  denotes the outer product, i.e.,  $(\mathbf{x}_1 \circ \mathbf{x}_2 \circ \dots \circ \mathbf{x}_D)[j_1, j_2, \dots, j_D] = \mathbf{x}_1[j_1]\mathbf{x}_2[j_2] \dots \mathbf{x}_D[j_D]$ ,  $(\cdot)$  denotes vectorization of a matrix or a tensor, and  $\lfloor \cdot \rfloor$  denotes the flooring operation.

We consider a discrete-time RIR  $h(k)$ , for  $k = 0, 1, \dots, n_h - 1$ , arranged in a vector  $\mathbf{h} \in \mathbb{R}^{n_h}$ , as well as a discrete-time signal  $x(k)$ , for  $k = 1, 2, \dots, n_x$ , arranged in the vector  $\mathbf{x} \in \mathbb{R}^{n_x}$ . The convolution of these vectors yields the discrete-time output

$$y(k) = \sum_{n=0}^{n_h-1} h(n)x(k-n), \quad (5.1)$$

for  $k = 1, 2, \dots, n_y$ , with corresponding vector  $\mathbf{y} \in \mathbb{R}^{n_y}$ , where  $n_y = n_h + n_x - 1$ . Generally, throughout this paper, an element is considered to be 0, if the index is out of its defined range, equivalent to appropriate zero-padding.

## 5.2 Room Impulse Response Compression

We will consider three different RIR approximations for RIR compression, and compare them to a state-of-the-art compression benchmark.

### 5.2.1 Compression by Truncation

Firstly, we consider an RIR compressed by *truncation*,  $\hat{\mathbf{h}}_{\text{T}}$ , where

$$\hat{\mathbf{h}}_{\text{T}}(n) = \begin{cases} \mathbf{h}(n), & n \leq n_{\text{T}} \\ 0, & n > n_{\text{T}} \end{cases} \quad (5.2)$$

for some  $n_{\text{T}} \in \mathbb{N}$ ,  $n_{\text{T}} \leq n_h$ . This method is amenable to accelerated convolution, as the length of the impulse response is shortened, decreasing the number of multiply-add instructions per output sample from  $n_h$  to  $n_{\text{T}}$ .

## 5.2.2 Compression by Thresholding

Secondly, we consider an RIR compressed by *thresholding*<sup>1</sup>,  $\hat{\mathbf{h}}_K$ , defined as

$$\hat{\mathbf{h}}_K(n) = \begin{cases} \mathbf{h}(n), n \in \mathcal{K}_{n_k} \\ 0, n \notin \mathcal{K}_{n_k} \end{cases} \quad (5.3)$$

where  $\mathcal{K}_{n_k}$  is the set of indices of the  $n_k$ , in absolute value, largest elements of  $\mathbf{h}$ . Also this RIR approximation method yields a possibly faster convolution. As many of the elements of  $\hat{\mathbf{h}}_K$  are zero, these do not have to be considered in the convolution. For a sparse impulse response  $\hat{\mathbf{h}}_K$  we can define the convolution between  $\hat{\mathbf{h}}_K \in \mathbb{R}^{n_h}$  and  $\mathbf{x} \in \mathbb{R}^{n_x}$  as

$$y(k) = \sum_{n \in \mathcal{K}_{n_k}} \hat{\mathbf{h}}_K(n)x(k-n). \quad (5.4)$$

This reduces the number of multiply-add instructions per output sample from  $n_h$  to  $n_k$ . The argument could be made that the positions of the non-zero components need to be stored, and that that is something that needs to be taken into account as well. However, whereas the coefficients themselves are floating numbers, the positions are integers, taking up significantly less space. Therefore, the impact of having to store the positions was ignored when considering the compression of thresholding.

## 5.2.3 Compression by Low-rank Approximation

Lastly, we consider an RIR compressed by *low-rank* approximation,  $\hat{\mathbf{h}}_{LR}$ . Assuming  $n_h = n_{s_1}n_{s_2}$ , with  $n_{s_1}, n_{s_2} \in \mathbb{N}$ , the RIR  $\mathbf{h} \in \mathbb{R}^{n_h}$  can be reshaped into a matrix  $\mathbf{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2}}$ ,

$$\mathbf{H} = \begin{bmatrix} h(1) & h(n_{s_1} + 1) & \dots & h(n_{s_1}(n_{s_2} - 1) + 1) \\ \vdots & \vdots & & \vdots \\ h(n_{s_1}) & h(2n_{s_1}) & \dots & h(n_h) \end{bmatrix}. \quad (5.5)$$

With the use of the singular value decomposition (SVD)  $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$  and assuming the singular values in  $\mathbf{\Sigma}$  are arranged in non-increasing order, we can then make a rank- $R$  approximation of  $\mathbf{H}$ ,

$$\hat{\mathbf{H}}_{2D} = \mathbf{U}[:, 1 : R]\mathbf{\Sigma}[1 : R, 1 : R]\mathbf{V}[:, 1 : R]^T. \quad (5.6)$$

Finally,  $\hat{\mathbf{h}}_{2D} = \hat{\mathbf{H}}_{2D}(\cdot)$ . Similarly, assuming  $n_h = \prod_{d=1}^D n_{s_d}$ ,  $n_{s_d} \in \mathbb{N}$ , the vector  $\mathbf{h}$  can be reshaped into a tensor  $\mathcal{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2} \times \dots \times n_{s_D}}$ , of arbitrary dimension

<sup>1</sup>In previous work, [95], we have referred to this as *KMax*.

$D$ , where  $n_{s_d}$  denotes the size of the  $d$ th dimension and the rank of a tensor is defined as the smallest number of rank-1 tensors needed to generate the tensor  $\mathcal{H}$  as their sum. In a similar fashion as to the matrix, we can then make a rank- $R$  approximation  $\hat{\mathcal{H}}_{\text{LR}}$  of  $\mathcal{H}$ . For this we will be using a (canonic) polyadic decomposition (see e.g., [143] and references therein). This is done using the high-level function *cpd* of the Matlab toolbox Tensorlab [84]. Subsequently,  $\hat{\mathbf{h}}_{\text{LR}} = \hat{\mathcal{H}}_{\text{LR}}(\cdot)$ . We will, in addition to aforementioned  $\hat{\mathbf{h}}_{2\text{D}}$ , consider low-rank approximation of 3-D and 5-D tensors, denoted  $\hat{\mathbf{h}}_{3\text{D}} = \hat{\mathcal{H}}_{3\text{D}}(\cdot)$  and  $\hat{\mathbf{h}}_{5\text{D}} = \hat{\mathcal{H}}_{5\text{D}}(\cdot)$ , respectively. The absence of a 4-D tensor approximation is explained in Section 5.5. Also the low-rank approximation method allows for fast time-domain convolution, which we have explored in recent work for up to three dimensions [141]. Here we will extend this idea to tensors of arbitrary dimensions. This will be further explained in Section 5.3.

### 5.2.4 Compression Benchmark: Opus

The three methods proposed above, truncation, thresholding, and low-rank approximation, will be compared to the state-of-the-art Opus interactive speech and audio codec [144, 145]. The Opus codes is created from two core technologies: Skype’s SILK codec [146], based on Linear Prediction (LP), and Xiph.Org’s CELT codec [147, 148], based on the Modified Discrete Cosine Transform (MDCT). The idea behind this construction is that LP is considered to code low frequencies more efficiently, whereas for music and higher speech frequencies, MDCT is superior. The double layers allow Opus to achieve higher quality for a wide range of audio. The Opus codec was created for, and has previously mainly been considered for, speech and music, but it has recently also gained attention as a possible way to compress RIRs [149]. In this work, the Opus encoding was done using Matlab’s *audiowrite*. It should be noted that although Opus shrinks the file size of the stored RIR, the number of coefficients remains the same. The RIR compressed by Opus, that will be denoted  $\hat{\mathbf{h}}_{\text{O}}$ , is therefore, to the best of the authors’ knowledge, not amenable to fast time-domain convolution. In order to give the reader a feel for the different approximations, an example RIR, taken from [1], and a selection of the compressed RIRs obtained with the different compression methods, at a compression rate (see (5.32)) of 0.8, are displayed in Fig. 5.1.

## 5.3 Convolution by Low-rank Approximation

Accelerating convolution by exploiting low-rank approximations was first considered by Atkins *et. al* in [126]. The authors there considered a low-

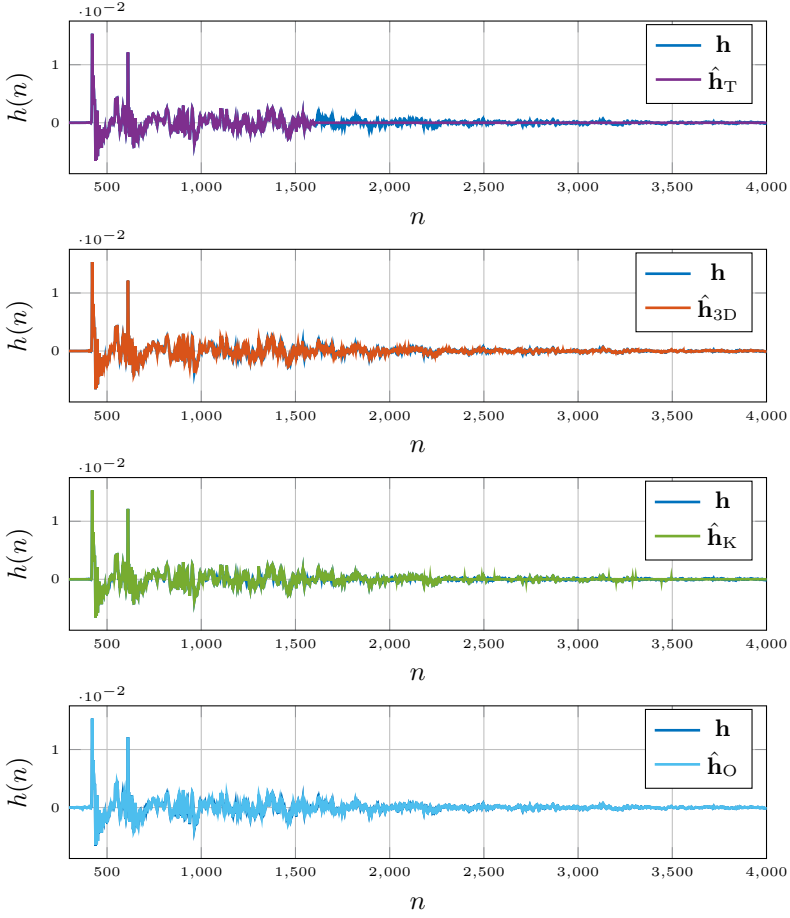


Figure 5.1: Example results of RIR compression methods.

rank approximation of a matricization of the RIR, using the SVD. In recent work, [141], we have extended this idea to a three-dimensional tensorization of the RIR. We will here show how this idea can be further extended to a tensorization of arbitrary dimension. We will first demonstrate the 2-D case presented in [114], and then explain the extension to a tensor of arbitrary dimension.

### 5.3.1 Partitioned Truncated SVD Filter

Assuming  $n_h = n_{s_1} n_{s_2}$ , for  $n_{s_1}, n_{s_2} \in \mathbb{N}$ , an output sample  $y(k)$  of the convolution in (5.1) can be written as

$$y(k) = \sum_{j=1}^{n_{s_2}} \mathbf{x}_k^{(j)T} \mathbf{h}^{(j)}, \quad (5.7)$$

where

$$\mathbf{h}^{(j)} \triangleq [h((j-1)n_{s_1}) \quad \dots \quad h(jn_{s_1} - 1)] \in \mathbb{R}^{n_{s_1}} \quad (5.8)$$

and

$$\mathbf{x}_k^{(j)} \triangleq [x(k-(j-1)n_{s_1}) \quad \dots \quad x(k-jn_{s_1} + 1)] \in \mathbb{R}^{n_{s_1}}, \quad (5.9)$$

for  $j = 1, 2, \dots, n_{s_2}$ . Instead of as in (5.1) writing  $y(k)$  as the inner product of two vectors of length  $n_h = n_{s_1} n_{s_2}$ , it is in (5.7) written as the sum of  $n_{s_2}$  inner products of vectors of length  $n_{s_1}$ . Further, the RIR  $\mathbf{h}$  can be reshaped into a matrix  $\mathbf{H} = [\mathbf{h}^{(1)} \quad \dots \quad \mathbf{h}^{(n_{s_2})}] \in \mathbb{R}^{n_{s_1} \times n_{s_2}}$ . For now we are going to assume that this matrix is rank-1, i.e., it can be written as the outer product  $\mathbf{H} = \mathbf{s}_1 \circ \mathbf{s}_2$ , for two vectors  $\mathbf{s}_1 \in \mathbb{R}^{n_{s_1}}$  and  $\mathbf{s}_2 \in \mathbb{R}^{n_{s_2}}$ . Under this assumption, we have that

$$\mathbf{H} = [\mathbf{s}_1 \mathbf{s}_2[1] \quad \mathbf{s}_1 \mathbf{s}_2[2] \quad \dots \quad \mathbf{s}_1 \mathbf{s}_2[n_{s_2}]], \quad (5.10)$$

i.e., the  $j$ th column of  $\mathbf{H}$ , corresponding to  $\mathbf{h}^{(j)}$ , is the vector  $\mathbf{s}_1$  scaled by  $\mathbf{s}_2[j]$ ,  $j = 1, 2, \dots, n_{s_2}$ . Further, the following property is readily verified,

$$\mathbf{x}_k^{(j)} = \mathbf{x}_{k+an_{s_1}}^{(j+a)}, \quad a \in \mathbb{Z}. \quad (5.11)$$

Because of (5.10) and (5.11), only the first inner product of the sum in (5.7) has to be computed per output sample  $k$ , the other inner products of the sum, i.e.,  $\mathbf{x}_k^{(j)T} \mathbf{s}_1 = \mathbf{x}_{k-n_{s_1}}^{(j-1)T} \mathbf{s}_1$ , for  $j = 2, \dots, n_{s_2}$ , have already been computed for a previous time sample, and can therefore be fetched from memory and multiplied with the appropriate entry from  $\mathbf{s}_2$ ,

$$y(k) = \left( \mathbf{x}_k^{(1)T} \mathbf{s}_1 \right) \mathbf{s}_2[1] + \underbrace{\sum_{j=2}^{n_{s_2}} \left( \mathbf{x}_k^{(j)T} \mathbf{s}_1 \right)}_{\text{Fetch from memory}} \mathbf{s}_2[j]. \quad (5.12)$$

This reduces the number of multiplications per sample to be carried out, from  $n_h = n_{s_1} n_{s_2}$  to  $n_{s_1} + n_{s_2}$ . These ideas can be extended to a matrix  $\mathbf{H}$  of arbitrary rank  $R$ . Instead of  $\mathbf{H}$  being just the outer product of two vectors, it is now a sum of  $R$  outer products,

$$\mathbf{H} = \mathbf{S}_1 \mathbf{S}_2^T = \sum_{r=1}^R \mathbf{S}_1[:, r] \circ \mathbf{S}_2[:, r] = \sum_{r=1}^R \mathbf{S}_1[:, r] \mathbf{S}_2[:, r]^T, \quad (5.13)$$

for  $\mathbf{S}_1 \in \mathbb{R}^{n_{s_1} \times R}$ , and  $\mathbf{S}_2 \in \mathbb{R}^{n_{s_2} \times R}$ . Equation (5.12) can now be extended to

$$y(k) = \sum_{r=1}^R \left( \left( \mathbf{x}_k^{(1)T} \mathbf{S}_1[:, r] \right) \mathbf{S}_2[1, r] + \underbrace{\sum_{j=2}^{n_{s_2}} \left( \mathbf{x}_k^{(j)T} \mathbf{S}_1[:, r] \right) \mathbf{S}_2[j, r]}_{\text{Fetch from memory}} \right) \quad (5.14)$$

where only  $R$  inner products have to be computed for each time sample. Similar to (5.12), this reduces the number of multiplications to  $R(n_{s_1} + n_{s_2})$ .

### 5.3.2 Fast Time-domain Convolution by Tensor Approximation

We are now ready to extend these ideas to a tensor of arbitrary dimension. Assuming  $\mathbf{h} \in \mathbb{R}^{n_h}$ , with  $n_h = \prod_{d=1}^D n_{s_d}$ , for  $n_{s_1}, n_{s_2}, \dots, n_{s_D} \in \mathbb{N}$ , let  $\mathbf{h}$  be reshaped into a tensor  $\mathcal{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2} \times \dots \times n_{s_D}}$ , and assume that  $\mathcal{H}$  is of rank  $R$ . Then, analogously to (5.13),

$$\mathcal{H} = \sum_{r=1}^R \mathbf{S}_1[:, r] \circ \mathbf{S}_2[:, r] \circ \dots \circ \mathbf{S}_D[:, r], \quad (5.15)$$

where  $\mathbf{S}_d \in \mathbb{R}^{n_{s_d} \times R}$ ,  $d = 1, 2, \dots, D$ , and in analog to (5.10), but with arbitrary dimension and rank, we have that

$$\mathcal{H}[:, j_2, j_3, \dots, j_D] = \sum_{r=1}^R \mathbf{S}_1[:, r] \mathbf{S}_2[j_2, r] \dots \mathbf{S}_D[j_D, r]. \quad (5.16)$$

The equality of (5.11) can be generalized according to

$$\mathbf{x}_k^{(j_2, j_3, \dots, j_D)} = \mathbf{x}_{k + \sum_{d=2}^D a_d \prod_{p=1}^{d-1} n_{s_p}}^{(j_2 + a_2, j_3 + a_3, \dots, j_D + a_D)}, \quad (5.17)$$

where  $\mathbf{x}_k^{(j_2, j_3, \dots, j_D)} \in \mathbb{R}^{n_{s_1}}$  is a vector containing the  $n_{s_1}$  latest samples of  $\mathbf{x}$ , in reversed order, starting at  $x(k - \sum_{d=2}^D (j_d - 1) \prod_{p=1}^{d-1} n_{s_p})$ , and  $a_2, a_3, \dots, a_D \in \mathbb{Z}$ . While verifying (5.17) can seem like a daunting task, it becomes clearer when considering the indices of the first entry of the vectors on the left and right-hand side of (5.17), respectively,

$$\begin{aligned}
 k - \sum_{d=2}^D (j_d - 1) \prod_{p=1}^{d-1} n_{s_p} &= \\
 k + \sum_{d=2}^D a_d \prod_{p=1}^{d-1} n_{s_p} - \sum_{d=2}^D (j_d + a_d - 1) \prod_{p=1}^{d-1} n_{s_p} &. \tag{5.18}
 \end{aligned}$$

The pattern from (5.7) extends to

$$y(k) = \sum_{j_2=1}^{n_{s_2}} \dots \sum_{j_D=1}^{n_{s_D}} \mathbf{x}_k^{(j_2, j_3, \dots, j_D)T} \mathbf{h}^{(j_2, j_3, \dots, j_D)}, \tag{5.19}$$

where  $\mathbf{h}^{(j_2, j_3, \dots, j_D)} = \mathcal{H}[:, j_2, j_3, \dots, j_D]$  is a vector containing  $n_{s_1}$  consecutive elements of  $\mathbf{h}$ , starting at  $h(\sum_{d=2}^D (j_d - 1) \prod_{p=1}^{d-1} n_{s_p})$ . Subsequently, the property of (5.14) is generalized to

$$\begin{aligned}
 y(k) = \sum_{r=1}^R &\left( \mathbf{x}_k^{(1, \dots, 1)T} \mathbf{S}_1[:, r] \mathbf{S}_2[1, r] \dots \mathbf{S}_D[1, r] + \right. \\
 &\left. \sum_{j_2=2}^{n_{s_2}} \dots \sum_{j_D=2}^{n_{s_D}} \underbrace{\left( \mathbf{x}_k^{(j_2, \dots, j_D)T} \mathbf{S}_1[:, r] \right)}_{\text{Fetch from memory}} \mathbf{S}_2[j_2, r] \dots \mathbf{S}_D[j_D, r] \right) \tag{5.20}
 \end{aligned}$$

with a corresponding structure of what has to be computed and what can be fetched from memory. Similarly to the previous case, we have a reduction in complexity. Only  $R$  inner products of length  $n_{s_1}$  have to be computed for each time index  $k$ , reducing the number of multiplications to  $R \sum_{d=1}^D n_{s_d}$ . When naively implemented, the sum in (5.20) will yield many superfluous operations, where one of the vectors contains only zeros. To fully exploit the structure of the RIR, and to maximize efficiency, it is therefore important to keep track of which operations actually need to be carried out and keep the number of multiplications with zeros to a minimum. We here propose an explicit algorithm.

Let  $\mathcal{H} = \sum_{r=1}^R \mathbf{S}_1[:, r] \circ \mathbf{S}_2[:, r] \circ \dots \circ \mathbf{S}_D[:, r]$ , where  $\mathcal{H} \in \mathbb{R}^{n_{s_1} \times n_{s_2} \times \dots \times n_{s_D}}$ , and  $\mathbf{S}_d \in \mathbb{R}^{n_{s_d} \times R}$ , for  $d = 1, 2, \dots, D$ . The operator  $\mathcal{I} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  denotes the reversion of the order of the elements in a vector, i.e.,

$\mathcal{I}(\mathbf{x}) = [x(n_x) \ x(n_x - 1) \ \dots \ x(1)]^T$ , and  $\mathbf{0}_n \in \mathbb{R}^n$  is a vector of zeros. The foundation of the algorithm is that, for each  $k$ , we compute the  $R$  necessary inner products, store the resulting values to memory and add these to  $y(k)$  with appropriate scaling by the corresponding elements of  $\mathcal{H}$ . Next, the remaining non-zero inner products in the sum of (5.20) are fetched from memory, scaled by the corresponding entry of  $\mathcal{H}$  and added to  $y(k)$ . The fast low-latency convolution algorithm by low-rank tensor approximation is summarized in Algorithm 4. A few remarks regarding Algorithm 4, for providing intuition as well as clarity, are in order:

- New inner products need to be computed and stored to memory as long as  $k \leq n_{s_1} + n_x - 1$ , this is done within the if-statement starting at line 5.
- Within the for-statement starting at line 14 the old inner products are fetched from memory and added to the output.
- On line 15, for  $d = 2$ , the upper limit of  $\prod_{p=2}^{d-1} n_{s_p}$  is lower than the lower limit, in which case, by convention,  $\prod_{p=2}^1 n_{s_p} = 1$ .

### 5.3.3 Complexity

By the authors of [126], it was noted that an output sample  $y(k)$  requires  $R(n_{s_1} + n_{s_2})$  multiply-add instructions, in the two-dimensional case, compared to the  $n_h = n_{s_1} n_{s_2}$  multiply-add instructions of conventional FIR filter convolution. The computational complexity for a general,  $D$ -dimensional tensorization is a generalization of the one in [126], and amounts to  $R \sum_{d=1}^D n_{s_d}$  multiply-add instructions, as compared to  $n_h = \prod_{d=1}^D n_{s_d}$  multiply-add instructions of conventional FIR filter convolution. Further, as the contribution to the end result of the entries in the sum of (5.20) are independent from each other, it is possible to perform these computations in parallel. To provide some intuition, an example is shown in Fig. 5.2. Here the complexity of traditional time-domain convolution is, for varying values of  $n_h$ , compared to that of the proposed algorithm for the case of square 2-D matricization and 3-D tensorizations of rank 4 and 12.

The two-dimensional algorithm from [126] requires a memory of size  $R(n_{s_1} + n_{s_2} + n_h) + n_{s_1}$  variables, compared to  $2n_h$  for a conventional FIR filter. For the proposed method, it is  $R(\sum_{d=1}^D n_{s_d} + n_h) + n_{s_1}$ , i.e., also the memory requirement for the proposed method is a generalization of the one in [126].

---

**Algorithm 4:** Fast Low-latency Convolution by Low-rank Tensor Approximation

---

Input:  $\mathcal{H} = \sum_{r=1}^R \mathbf{S}_1[:, r] \circ \mathbf{S}_2[:, r] \circ \dots \circ \mathbf{S}_D[:, r]$ ,  $\mathbf{x}$   
Output:  $\mathbf{y}$   
**for**  $k = 1, 2, \dots, n_y$  **do**  
  **for**  $r = 1, 2, \dots, R$  **do**  
    **if**  $k \leq n_{s_1} + n_x - 1$  **then**  
       $u_b = \max(k - n_x + 1, 0)$ ;  
       $u_e = \min(k, n_{s_1})$ ;  
       $x_b = \max(k - n_{s_1} + 1, 1)$ ;  
       $x_e = \min(k, n_x)$ ;  
       $\mathbf{C}[\text{mod}(k - 1, n_h) + 1, r] = [\mathcal{I}(\mathbf{x}[x_b : x_e])]^T \mathbf{S}_1[u_b : u_e, r]$ ;  
       $y(k) = \prod_{d=2}^D \mathbf{S}_d[1, r] \mathbf{C}[\text{mod}(k - 1, n_h) + 1, r]$ ;  
    **end if**  
     $l = \max(\lfloor (k - n_x) / n_{s_1} \rfloor + 1, 2)$ ;  
     $u = \min(\lfloor (k - 1) / n_{s_1} \rfloor + 1, \prod_{d=2}^D n_{s_d})$ ;  
    **for**  $c = l, l + 1, \dots, u$  **do**  
       $j_d = \left\lfloor \text{mod}\left((c - 1), \prod_{p=2}^d n_{s_p}\right) / \prod_{p=2}^{d-1} n_{s_p} \right\rfloor + 1$ ;  $d = 2, \dots, D$   
       $\tilde{c} = k - \sum_{d=2}^D (j_d - 1) \prod_{p=1}^{d-1} n_{s_p}$   
       $y(k) = y(k) + \prod_{d=2}^D \mathbf{S}_d[j_d, r] \mathbf{C}[\text{mod}(\tilde{c} - 1, n_h), r]$ ;  
    **end for**  
  **end for**  
**end for**

---

## 5.4 Objective Quality Measures

Audio technology can generally be designed to be either physically motivated or perceptually motivated. Physically motivated techniques are typically computationally intensive, in the attempt to physically accurately represent the sound field. Perceptually motivated systems are in general less computationally demanding, as they aim only to be accurate enough for human perception [136]. The physical accuracy of low-rank approximations of RIRs was evaluated in [95], in this work we aim to investigate the perceptual accuracy of compression by low-rank approximation and the other aforementioned compression methods. In this section we describe a variety of parameters regarding the perception of room acoustics and corresponding objective measures. These measures can be divided into two categories, channel-based objective measures and signal-based objective measures [38]. The channel-based measures concern only how well the

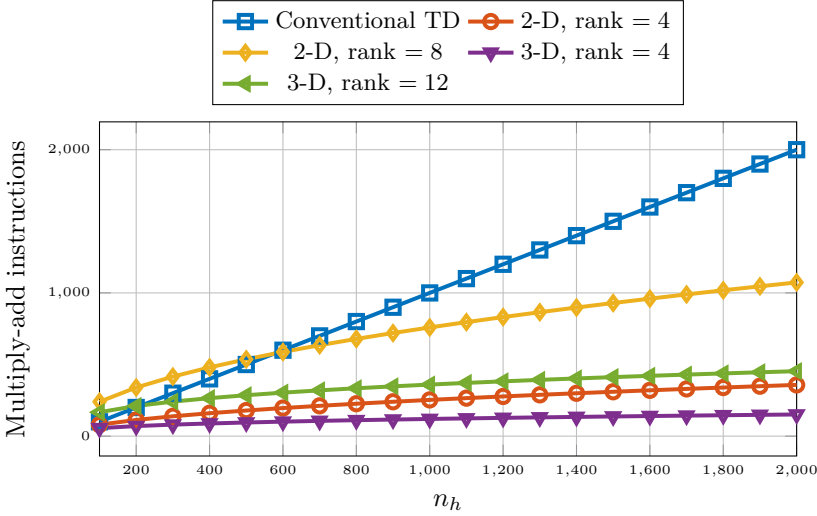


Figure 5.2: Computational complexity of proposed convolution.

approximation of the channel, i.e., the compressed RIR, relates to the measured channel, i.e., the RIR. Signal-based measures, on the other hand, pertain to how the approximated channel distorts the signal output, after the compressed RIR has been convolved with e.g., music or speech.

The objective of the different measures considered here differs slightly. For some of them a high value is desirable, for others a lower value is better. For most of them, however, invariance is what is sought after, i.e. that the value of the measured quantity for a compressed RIR is as close as possible to the measured quantity for the original RIR. For an easy overview for the reader, the measures considered in this paper, their definitions, whether they are channel- or signal-based, and their objectives, are recapped in Table 5.1.

### 5.4.1 Channel-based Objective Quality Measures

The perhaps most obvious way to measure the quality of a compressed RIR is by the *normalized misalignment*, defined as

$$\mathcal{M}_{\text{dB}}(\hat{\mathbf{h}}) = 20 \log_{10} \left( \frac{\|\hat{\mathbf{h}} - \mathbf{h}\|_2}{\|\mathbf{h}\|_2} \right). \quad (5.21)$$

Table 5.1: Measures

Measure	Definition	Channel/Signal	Objective
Normalized Misalignment	(5.21)	Channel	Low
Reverberation Time $T_{60}$	(5.23)	Channel	Invariance
Echo Density	(5.24)	Channel	Invariance
Early Decay Time (EDT)	(5.26)	Channel	Invariance
Center Time	(5.27)	Channel	Invariance
TOA of Direct Component	(5.28)	Channel	Invariance
Frequency-weighted Log-spectral Signal Distortion (SD)	(5.31)	Signal	Low
ViSQOLAudio	[150, 151, 152]	Signal	High

The problem with this measure is, however, that it is not necessarily a good indicator of whether the compressed RIR will yield an auditory perception faithful to the original RIR.

*Reverberation time* is a well-known objective measure for room acoustics. This is the time it takes for the sound level to drop 60 dB, after a stationary sound source has been switched off, and is denoted  $T_{60}$ . In practice, this measure is typically estimated as double the time it takes for the sound level to drop from  $-5$  dB to  $-35$  dB [135]. Finding the time it takes for the sound level to drop a certain amount is done via the energy decay curve which, since the work by Schroeder [153], is most commonly found using backwards integration. As we consider discrete-time signals in this paper, the energy decay curve  $D(n)$  is found using backwards summation,

$$D(n) = \sum_{k=n}^{n_h} h^2(k) = \sum_{k=0}^{n_h} h^2(k) - \sum_{k=0}^n h^2(k). \quad (5.22)$$

Letting  $n_{-x\text{dB}}$  denote the time sample when the energy decay curve  $D(n)$  has decreased to  $x$  dB below its starting value,  $T_{60}$  is found as

$$T_{60} = 2(n_{-35\text{dB}} - n_{-5\text{dB}}) f_s, \quad (5.23)$$

where  $f_s$  denotes sampling frequency. Reverberation can cause degraded speech intelligibility, but it is also what gives music fullness, by blending the sounds of different instruments and voices [135]. It further provides, together with the energy ratio between direct and reverberant sound and the time of arrival of the early reflections, information about the size of a space and the distance to the boundaries [154].

The *echo density* profile of an RIR is the fraction of impulse response coefficients which lie outside the standard deviation of the coefficient amplitudes, for a particular time window. A simple and robust measure for echo density was introduced by Abel *et. al* in [155],

$$\eta(n) = \frac{1/\operatorname{erfc}(1/\sqrt{2})}{2\delta + 1} \sum_{k=n-\delta}^{n+\delta} w(k) \mathbf{1}_{\{|h(k)| > \sigma\}}, \quad (5.24)$$

where  $\operatorname{erfc}(1/\sqrt{2}) = 0.3173$ ,  $2\delta + 1$  is the window length in samples,  $\mathbf{1}_{\{\cdot\}}$  is an indicator function,  $w(k)$  is a window function, for which  $\sum_k w(k) = 1$ , and

$$\sigma = \left[ \sum_{k=n-\delta}^{n+\delta} w(k) h^2(k) \right]^{1/2}. \quad (5.25)$$

Throughout this paper we will use a Hanning window with  $\delta = 550$ , when  $f_s = 44.1$  kHz and  $\delta = 600$  when  $f_s = 48$  kHz, corresponding to a window length of 25 ms, as per the discussion in [155]. Further, we will only consider the part of the echo density profile where the entire window fits.

In reverberant music or speech, later parts of the reverberation tend to be masked by the direct and early components of the next note or syllable. Therefore, the alternative measure *early decay time* (EDT), has proved to be better correlated with reverberance, a perceptual attribute of reverberation, than reverberation time, in the aforementioned scenarios [135]. The EDT is defined as

$$\text{EDT} = 6(n_{-10\text{dB}})f_s. \quad (5.26)$$

The parameter *center time*, denoted  $t_s$ , describes the balance between early and late energy in the RIR [135], defined as

$$t_s = \frac{\sum_{k=0}^{n_h} k h^2(k)}{\sum_{k=0}^{n_h} h^2(k)}, \quad (5.27)$$

i.e., the center of gravity of the RIR. Two other measures that are commonly mentioned in this context are *mode density* [27, 156] and *reflections density* [74, 136]. These are, however, better suited to characterize synthetically generated RIRs. As we here consider only real-life RIRs, these measures will not be considered in this paper.

The time of arrival (TOA) of the direct component, defined as

$$\text{TOA} = \left( \arg \max_n |h(n)| \right) / f_s, \quad (5.28)$$

is crucial in tasks such as room geometry estimation [157] and acoustic source localization [158]. How the TOA of the direct component is preserved by a compression method is not well captured by the normalized misalignment and will therefore be considered as a separate measure in Section 5.5.

For all the channel-based measures introduced above, expect normalized misalignment, we aim for a minimal deviation between the compressed and original RIR measure. We will therefore, in Section 5.5, present the root-mean-square error (RMSE) for these quantities,

$$\text{RMSE}_g(\hat{\mathbf{h}}) = \sqrt{\frac{\sum_{j=1}^{n_{\text{RIR}}} |g(\mathbf{h}_j) - g(\hat{\mathbf{h}}_j)|^2}{n_{\text{RIR}}}}, \quad (5.29)$$

where  $g$  is the considered measure, and  $n_{\text{RIR}}$  denotes the number of RIRs used in the evaluation. We alert the reader that we in Section 5.5 will consider RMSE in linear scale for certain measures and in logarithmic scale for other measures, depending on what best highlights the difference in performance between the considered compression methods. All considered quantities except echo density are scalar, making the computation of the RMSE straightforward. Echo density, however, is a discrete-time sequence. There the RMSE will be computed as

$$\text{RMSE}_{\text{ED}}(\hat{\mathbf{h}}) = \sqrt{\frac{\sum_{j=1}^{n_{\text{RIR}}} \|\mathbf{h}_{\text{ED}}^j - \hat{\mathbf{h}}_{\text{ED}}^j\|_2^2}{n_{\text{RIR}} n_{\text{ED}}}}, \quad (5.30)$$

where  $\mathbf{h}_{\text{ED}}^j = [\eta(1), \eta(2), \dots, \eta(n_{\text{ED}})]^T$  denotes the echo density profile of the  $j$ th RIR, and  $n_{\text{ED}}$  the length of the echo density profile.

### 5.4.2 Signal-based Objective Measures

Next, we present measures of output signal degradation. The ultimate goal of any acoustic signal enhancement or reproduction task is to achieve good signal quality. One way to measure this is by using subjective listening test. These tests are, however, expensive, tedious, and time consuming [159, 149]. Therefore, several objective measures have been developed to predict the outcome of subjective listening tests. The frequency-weighted log-spectral signal distortion (SD) [160] is a perceptually weighted objective measure of distortion of a sound signal, w.r.t. a reference signal

$$\text{SD}(t) = \sqrt{\int_{f_l}^{f_u} w_{\text{ERB}}(f) \left( 10 \log_{10} \frac{P_{\hat{\mathbf{y}}}(f, t)}{P_{\mathbf{y}}(f, t)} \right)^2 df}, \quad (5.31)$$

where  $P_{\hat{\mathbf{y}}}$  and  $P_{\mathbf{y}}$  are the short-term power spectra of  $\hat{\mathbf{y}} = \mathbf{x} * \hat{\mathbf{h}}$  and  $\mathbf{y} = \mathbf{x} * \mathbf{h}$ , for a sound signal  $\mathbf{x}$ , respectively, and  $w_{\text{ERB}}$  is a frequency-weighting function, that gives equal weight to each auditory critical band between  $f_l = 300$  Hz and  $f_u = 6500$  Hz. In Section 5.5, we will present both the mean and maximum SD values for the respective scenarios.

Hines *et al.* introduced the Virtual Speech Quality Objective Listener (ViSQOL) [150, 151], an objective measure for predicting the subjective assessment of perceived speech quality, based on the Neurogram Similarity Index Measure (NSIM) [161]. ViSQOL was subsequently extended to ViSQOLAudio [152], to comprise not only speech, but also audio and music signals, and has shown high correlation with the subjective listening test MUSHRA [162]. Narbutt *et al.* have extended ViSQOL and ViSQOLAudio to AMBIQUAL [163, 164], that aims to predict not only listening quality, but also localization accuracy, for spatial audio. We don't consider spatial audio in this work, and will therefore not use AMBIQUAL. In addition to the aforementioned acoustic qualities and measures, there are several other measures concerning perceived speech quality, such as PESQ [165] and POLQA [166]. These are intended to predict the perceived quality of speech, rather than audio or music, and will not be considered here.

## 5.5 Numerical Results

To compare the performance of the here investigated methods, we apply them to three different datasets of RIRs, with varying reverberation time. First we apply it to the single- and multichannel audio recordings database (SMARD) [1], which contains RIRs from a listening room with a reverberation time of approximately 0.15 s, sampled at 48 kHz. Next, we apply the methods to the two different datasets from the MYRiAD database [167]. The first one is from the Alamire Interactive Laboratory (AIL), which has a reverberation time of 0.5 s, and the second one is from the SONORA Audio Laboratory (SAL), with a reverberation time of 2.1 s. These are sampled at 44.1 kHz.

For the low-rank methods, the matricization or tensorization of the RIRs brings about the question of the size of the dimensions. For a  $D$ -dimensional tensorization, it is required that  $\prod_{d=1}^D n_{s_d} = n_h$ , but this can be achieved in several different ways. The impact of the size of the dimensions is beyond the scope of this paper, and we will here present only square matricizations and tensorizations, i.e.,  $n_{s_1} = n_{s_2} = \dots = n_{s_D}$ . As a consequence of this, we must have that  $n_{s_d} = \sqrt[D]{n_h} \in \mathbb{N}$ . For this reason, the length of the RIRs for the different compression methods will vary slightly. We will here present the results for low-rank approximations of different dimensions, thresholding, truncation,

Table 5.2: RIR lengths used for the different data sets

<b>Name</b>	$n_{\text{RIR}}$	$n_{h_2}$	$n_{h_3}$	$n_{h_5}$
SMARD	100	$88^2 = 7744$	$20^3 = 8000$	$6^5 = 7776$
AIL	40	$181^2 = 32761$	$32^3 = 32768$	$8^5 = 32768$
SAL	20	$316^2 = 99856$	$47^3 = 103823$	$10^5 = 100000$

and, as a benchmark, Opus. In order to be able to have RIR lengths in as close proximity as possible, we present low-rank approximations for  $D = 2, 3$ , and  $5$ , neglecting  $D = 4$ , as the length of the RIR for that dimension of tensorization would differ too much from the others. The RIR lengths used for the 2-D, 3-D, and 5-D approximations are denoted  $n_{h_2}$ ,  $n_{h_3}$ , and  $n_{h_5}$ , respectively. The RIR length used for thresholding, truncation, and Opus is denoted  $n_h$ , and will be equal to the largest of  $n_{h_2}$ ,  $n_{h_3}$ , and  $n_{h_5}$ , for the respective scenarios. The different RIR lengths used in the simulations are found in Table 5.2. We alert the reader that these lengths apply to both the approximation and their respective reference RIR, as some of the objective measures introduced in Section 5.4 require that the approximated RIR and the reference RIR are of equal length. For the generation of the output signals, the compressed RIRs are convolved with 5 different, randomly selected, 15 s snippets of music from EBU-SQAM [132]. When convolving these snippets of music with the RIRs from SMARD, the music was upsampled to 48 kHz using Matlab’s *resample*, in order to have matching sampling frequencies.

We denote by  $\Upsilon(\hat{\mathbf{h}})$  the number of coefficients needed to be stored for a certain compressed RIR  $\hat{\mathbf{h}}$ , and remind the reader that for the low-rank approximations,  $\Upsilon(\hat{\mathbf{h}}) = R \sum_{d=1}^D n_{s_d}$ . For all the compression methods except Opus, the number of coefficients stored coincides with the number of multiply-add instructions needed to carry out time-domain convolution with the approximated RIR. For the original RIR, this number is  $n_h$ . Therefore, by

$$C(\hat{\mathbf{h}}) = 1 - \frac{\Upsilon(\hat{\mathbf{h}})}{n_h}, \quad (5.32)$$

where  $C(\hat{\mathbf{h}}) \in [0, 1)$ , we denote both *compression rate* and *complexity reduction*. For  $C(\hat{\mathbf{h}}) = 0$  there is no compression or complexity reduction, whereas for  $C(\hat{\mathbf{h}})$  closer to 1, the degree of complexity reduction is larger. We provide simulations in the range from  $C(\hat{\mathbf{h}}) = 0.7$  to  $C(\hat{\mathbf{h}}) = 0.95$ , as these are the minimum and maximum values of compression supported by Opus, for all the sets of RIRs considered here, when using Matlab’s built-in function *audiowrite*.

RIRs should ideally be estimated from noiseless measurements, but this condition is often not met in practice [168, 169, 170]. As the RIRs used

in this paper are taken from databases of real-life recorded RIRs, they will contain some measurement noise. However, to simulate a realistic environment, white Gaussian noise was added to each recorded and truncated RIR before compression and convolution. The power of the noise was adjusted to yield a signal-to-noise ratio (SNR) of 20 dB, where

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left( \frac{P_R}{P_N} \right), \quad (5.33)$$

where  $P_R$  and  $P_N$  denote the power of the RIR without the noise, and the power of the noise, respectively. The ground-truth values of the quantities considered in this section is computed with respect to truncated RIR, before the noise is added.

On a couple of occasions, the performance of one, or several, compression methods was significantly worse than the other methods. In those cases, these approximations have been left out of the figures, as including them would significantly impact the scaling of the figure, and prevent the reader from noticing the differences between the more competitive methods. When this has been done, remarks have been made in the corresponding subsection to alert the reader.

### 5.5.1 Normalized Misalignment

As can be seen in Fig. 5.3, in terms of normalized misalignment for the RIR compression, truncation and 2-D matricization falls short. However, 3-D tensorization, 5-D tensorization, and thresholding are all outperforming Opus.

### 5.5.2 Reverberation Time $T_{60}$

Compression based on low-rank approximation or thresholding also performs very well when it comes to the preservation of the reverberation time  $T_{60}$ . This is displayed in Fig. 5.4, where we observe a consistent outperformance of Opus. The unexpected performance deterioration for the low-rank approximation and thresholding is due to the added noise. Overestimation of  $T_{60}$  for noisy RIRs is a well-known phenomenon [171, 172]. This is due to a slower drop-off of the decay curve (5.22). The approximations serve as denoising but for lower values of compression there is still a systematic overestimation of the reverberation time. This is illustrated in Fig. 5.5, where histogram of the differences between the  $T_{60}$  estimates for the 3-D tensor approximation and that of the measured RIR, for the RIRs of SMARD, at the compression rate of 0.7, is displayed.

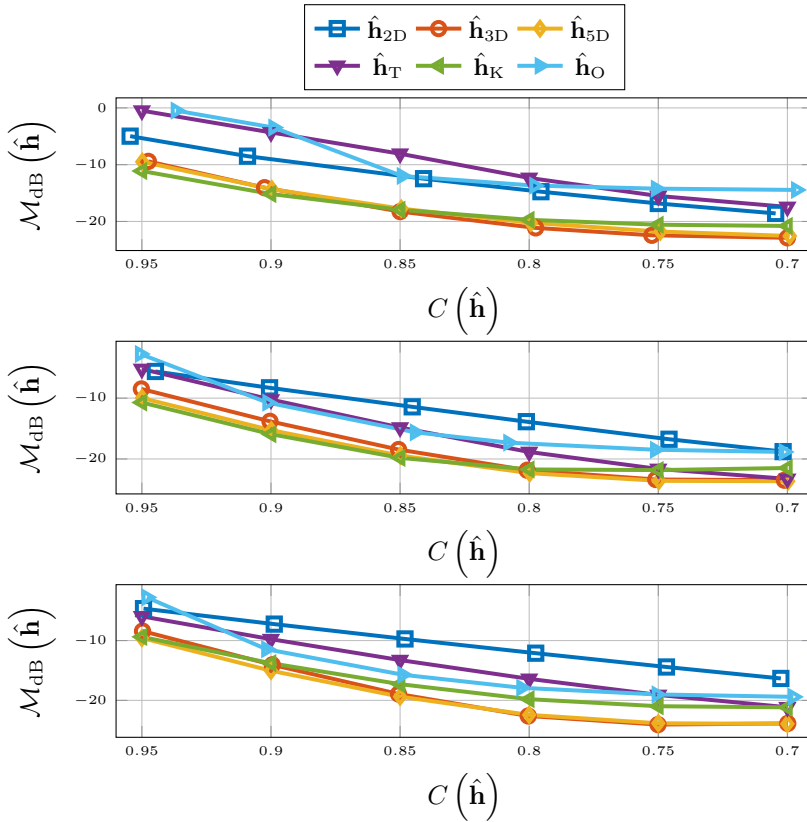


Figure 5.3: Normalized misalignment for SMARD (top), AIL (middle), and SAL (bottom).

We alert the reader that these are differences and not absolute differences, i.e. the fact that all numbers are positive shows the consistent overestimation. Preliminary simulations showed that this systematic overestimation could partly be alleviated by estimating the  $T_{60}$  a shorter time interval, i.e., corresponding to the decay from  $-5$  dB to  $-25$  dB, but not entirely.

### 5.5.3 Echo Density

When it comes to preserving echo density, displayed in Fig. 5.6, Opus is the best of the compared compression methods for longer RIRs. For short RIRs, 2-D matrix approximation and 3-D tensor approximation outperforms Opus,

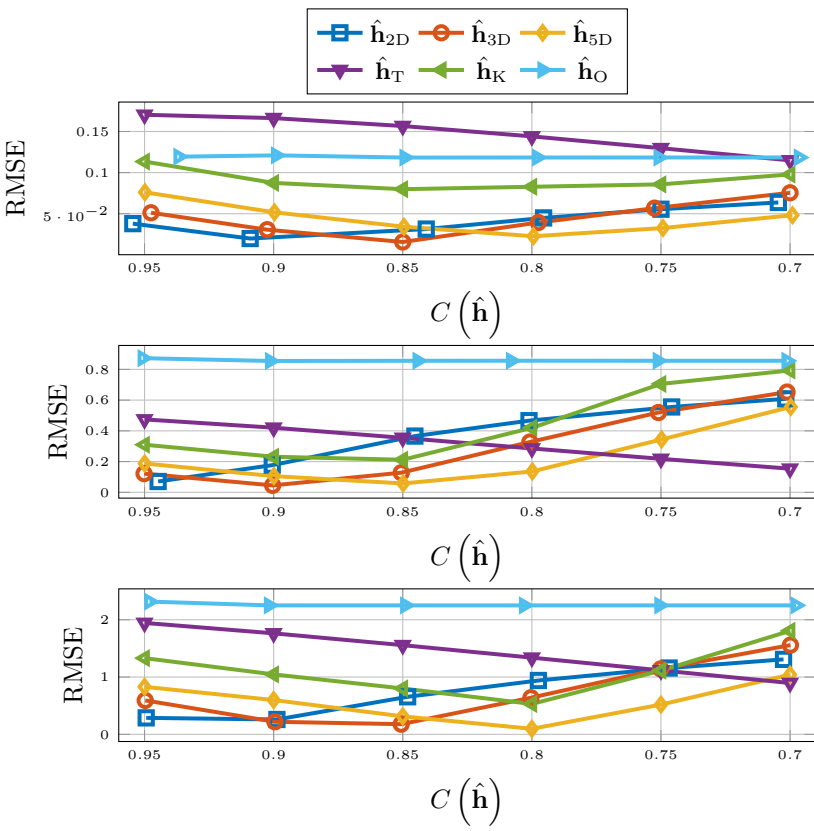


Figure 5.4: RMSE for  $T_{60}$  for SMARD (top), AIL (middle), and SAL (bottom).

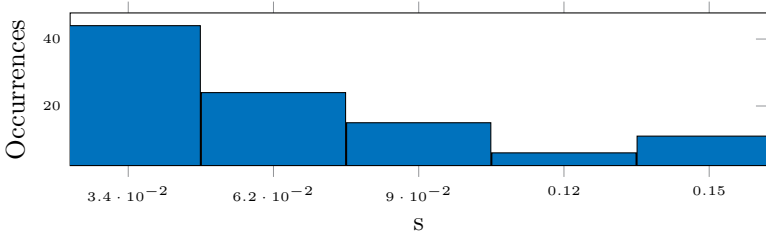


Figure 5.5: Histogram of differences in  $T_{60}$  estimate,  $T_{60}(\hat{\mathbf{h}}_{3D}) - T_{60}(\mathbf{h})$ , for 3-D tensor approximation and recorded RIR.

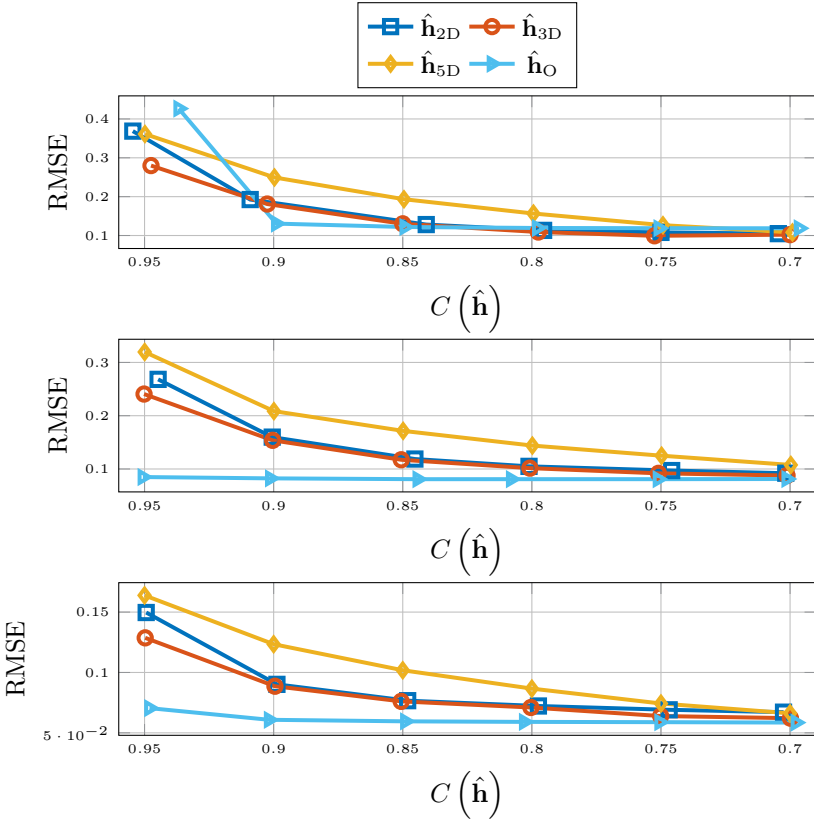


Figure 5.6: RMSE for echo density for SMARD (top), AIL (middle), and SAL (bottom).

but 5-D tensor approximation does not. Truncation and thresholding are not included in Fig. 5.6 due to poor performance.

### 5.5.4 Early Decay Time

The performance of the different compression methods with respect to preserving EDT is shown in Fig. 5.7. For this measure, truncation and 2-D matricization performs worst for all considered cases. Opus works better for longer RIRs and for higher compression rates, but for shorter RIRs, and all but the highest compression rates, thresholding, and 3-D and 5-D tensorization are better options.

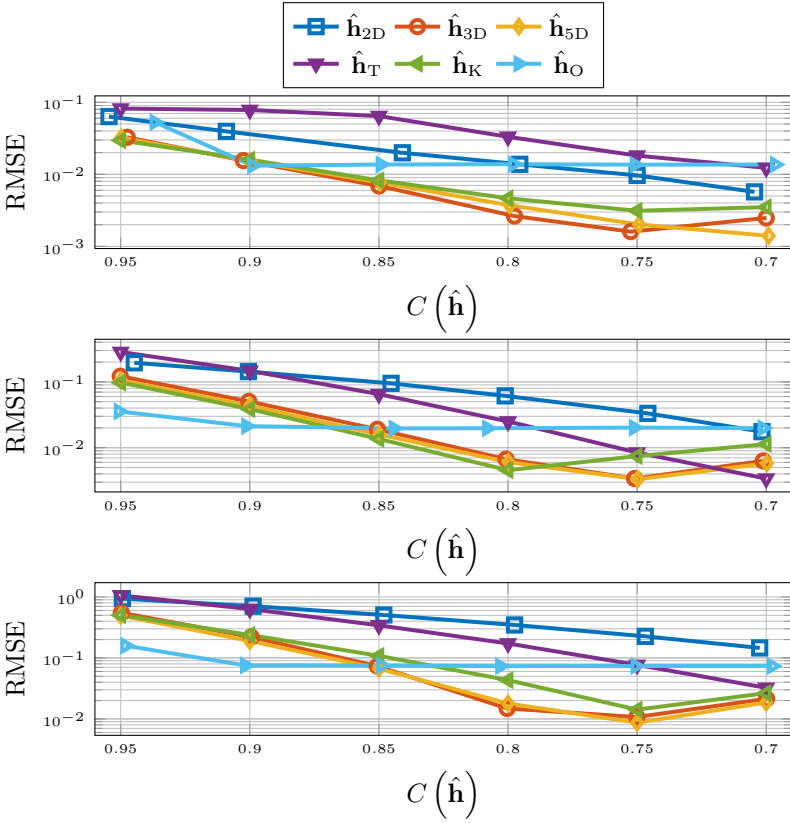


Figure 5.7: RMSE for EDT for SMARD (top), AIL (middle), and SAL (bottom).

### 5.5.5 TOA of Direct Component

For the preservation of the TOA of the direct component, there is a clear discrepancy between the compression methods based on low-rank approximation and the other methods. This is evident from Fig. 5.8, where the results are displayed.

### 5.5.6 Center Time

In Fig. 5.9 we see the RMSE for the center time. There it can be observed that the 2-D matrix approximation does not perform on the level of Opus, but thresholding, and the higher-order tensor approximations do, for all but the

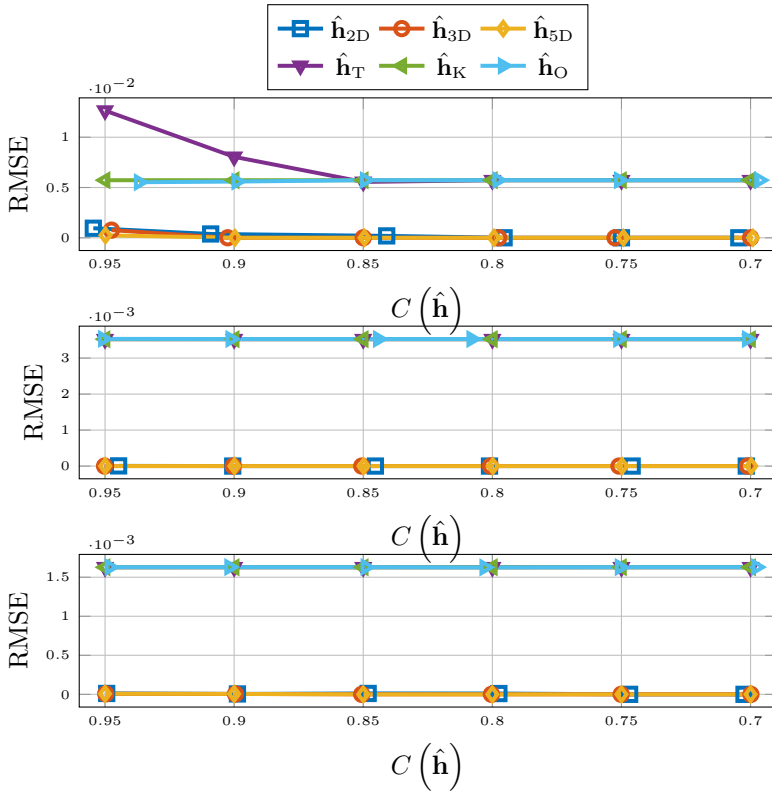


Figure 5.8: RMSE for TOA of direct component for SMARD (top), AIL (middle), and SAL (bottom).

highest compression rates. The performance of compression by truncation has been left out of the figure.

### 5.5.7 Signal Distortion

The results for the mean SD are better for the higher-order low-rank methods and thresholding, compared to Opus, except for the highest compression rates for the longest RIRs. This can be seen in Fig. 5.10. The results for 2-D matricization and truncation was yet again worse and left out of the plot to better show the difference between the other compression methods.

As for the maximum SD, displayed in Fig. 5.11, truncation had to be left out

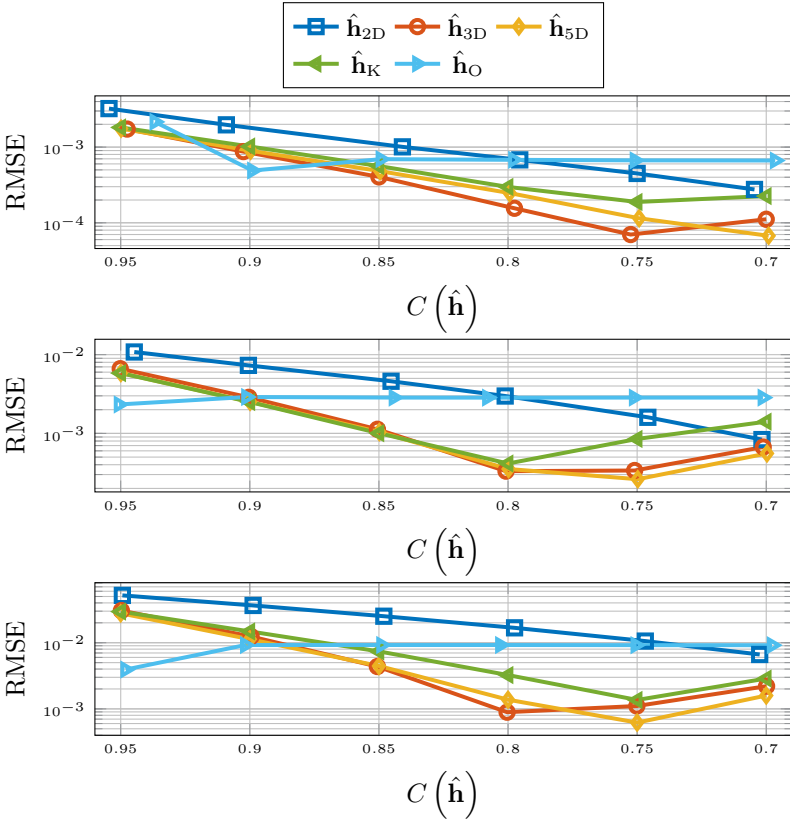


Figure 5.9: RMSE for center time for SMARD (top), AIL (middle), and SAL (bottom).

of the plots and for the AIL dataset, thresholding had to be left out too, due to their poor performance for higher compression rates. The higher-order low-rank approximation perform better than Opus for all the considered values of the compression rate.

### 5.5.8 ViSQOLAudio

In Fig. 5.12 the ViSQOLAudio scores for varying compression rate are displayed. It is only for high compression rates of very long RIRs where Opus is a better option than 3-D tensorization, 5-D tensorization, and thresholding. For

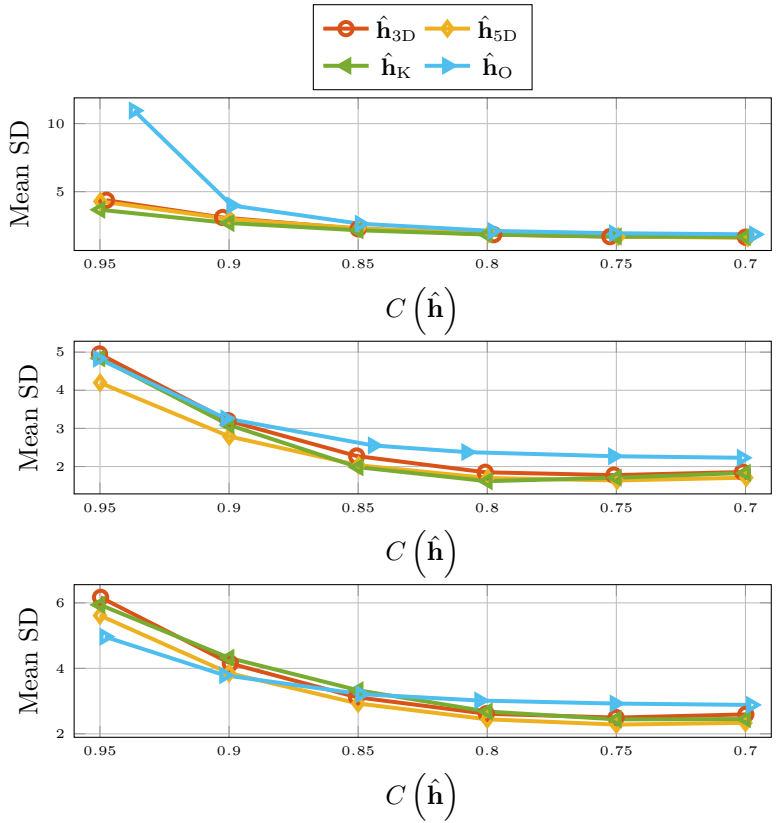


Figure 5.10: Mean signal distortion for SMARD (top), AIL (middle), and SAL (bottom).

ViSQOLAudio, the results for 2-D matricization and truncation were left out of Fig. 5.12 due to poor performance.

## 5.6 Conclusions

In this work we have considered different RIR approximation methods for the purpose of RIR compression, aiming to save data storage and accelerate time-domain convolution. It was found that RIR truncation performs worst in almost all scenarios considered and can therefore not be recommended. With the exception of echo density, the RIR compression by thresholding generally

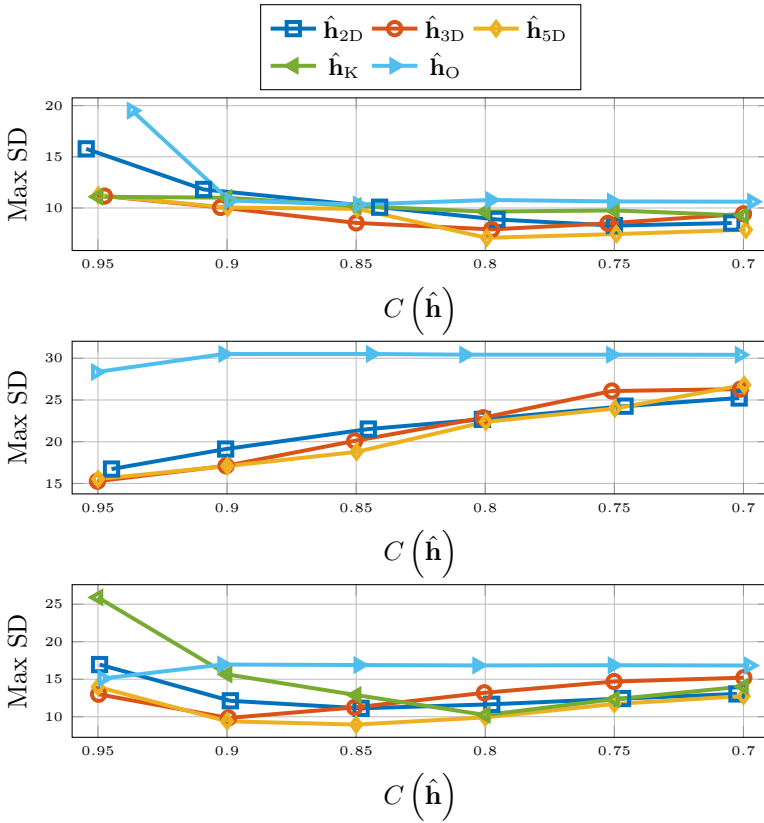


Figure 5.11: Maximum signal distortion for SMARD (top), AIL (middle), and SAL (bottom).

preserves well the RIR qualities considered here, compared to the state-of-the-art Opus. For the low-rank approximation methods, 2-D matricization falls short on certain measures, such as mean signal distortion, and ViSQOLAudio. The 3-D and 5-D tensor approximations generally outperform thresholding and they are more robust, as there was no considered scenario or measure where they performed significantly worse than the other methods, and they perform better than thresholding with respect to the signal-based measures. Much like thresholding, 3-D and 5-D tensor approximations can't compete with Opus when it comes to preserving echo density, and for the highest level of compression rate, Opus is also better when it comes to preserving EDT and center time. For all other considered measures and scenarios, 3-D and 5-D tensor approximations are as good, or better, than Opus. Add to this the fact that the low-rank tensor

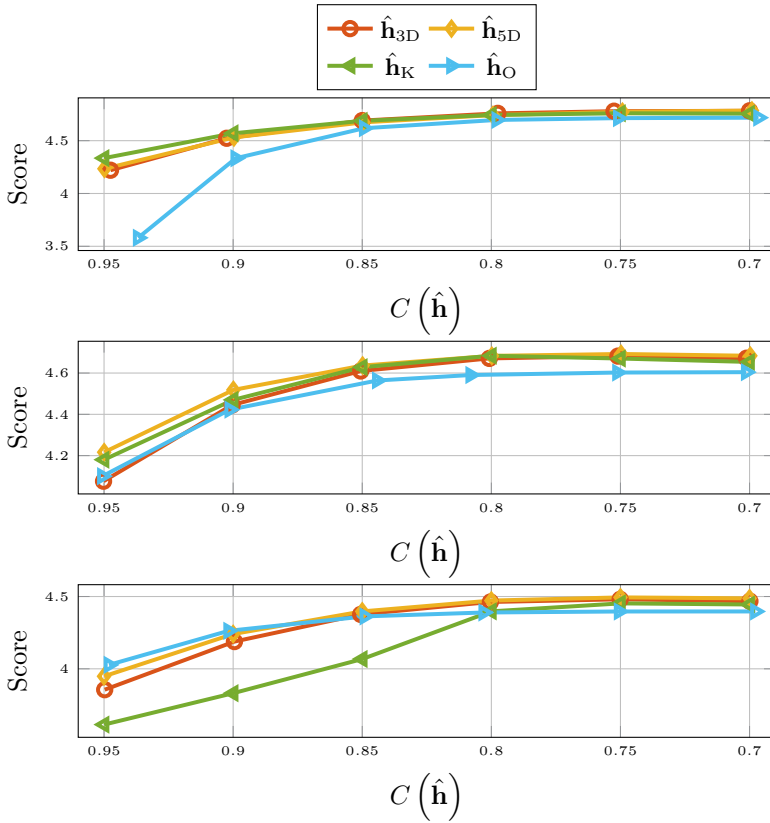


Figure 5.12: ViSQOLAudio for SMARD (top), AIL (middle), and SAL (bottom).

approximations are amenable to fast time-domain convolution, and they stand out as the superior choice compared to Opus.

Future research should mainly focus on three open questions. Firstly, investigating whether the promising results for the objective measures considered here will translate into superior performance also in subjective listening tests. Secondly, the fact that the low-rank approximations preserve the TOA of the direct component almost flawlessly indicates that these approximations could be very useful also in the context of spatial RIRs, which needs to be further explored. Finally, the occasional discrepancy in performance between the 3-D and 5-D tensorization methods is not yet well enough understood, and needs to be further investigated.

## Chapter 6

# Multi-channel Low-rank Convolution of Jointly Compressed Room Impulse Responses

Multi-channel Low-rank Convolution of Jointly Compressed Room Impulse Responses

Martin Jälmy, Filip Elvander, and Toon van Waterschoot

Submitted for publication, Dec. 2023.

The candidate's contributions as first author include all performed research and writing of the manuscript under guidance from the co-authors.

## Abstract

The room impulse response (RIR) describes the response of a room to an acoustic excitation signal and models the acoustic channel between a point source and receiver. RIRs are used in a wide range of applications, e.g., virtual reality. In such an application, the availability of closely spaced RIRs and the capability to achieve low latency are imperative to provide an immersive experience. However, representing a complete acoustic environment using a fine grid of RIRs is prohibitive from a storage point of view and without exploiting spatial proximity, acoustic rendering becomes computationally expensive. We therefore propose two methods for the joint compression of multiple RIRs, *Low-rank Compression of Room Impulse Responses for Low-latency Convolution* and *Low-rank Compression of Room Impulse Responses for Low-latency Convolution with Partially Invariant Transformation*, based on the generalized low-rank approximation of matrices (GLRAM), for the purpose of efficiently storing RIRs and allowing for low-latency convolution, for which we propose an algorithm. We show how one of the components of the GLRAM decomposition is virtually invariant to the change of position of the source throughout the room and how this can be exploited in the modeling and convolution. In simulations we show how this offers high compression, with less quality degradation than for comparable benchmark methods.

## 6.1 Introduction

The room impulse response (RIR) describes the impact of a room on an acoustic excitation signal played within the room and is used in a wide variety of applications [58, 85, 86]. Typically, the RIR is modeled either as an infinite impulse response (IIR) (see, e.g., [87, 71]), or as a finite impulse response (FIR) model (see, e.g., [87, 72]). The IIR model is generally more compact, but the filter parameters can be difficult to estimate, due to issues of instability [73, 81]. The FIR model is simple and straightforward, but the number of parameters needed is large, on the order of  $10^3$  for an office-sized room, and  $10^4$  or even  $10^5$  for a reverberant room such as concert hall, at a sampling rate of 48 kHz [38, 135]. This can be prohibitive from both a memory requirement and computational complexity point of view, when using the RIR for convolution [88, 89, 136]. The RIR is position-dependent, meaning that if the source or the receiver moves, the corresponding RIR changes. As a consequence, in order to faithfully reconstruct the sound field in a room, the spatial resolution of the grid of measurements needs to be on the order of 10 cm [51]. Even for small rooms, the number of source/receiver configurations for which the RIR has to

be stored will be in the millions, amounting to hundreds of gigabytes of data for a single room. Consequently, there is a need for compact representations of the RIRs of a room.

An application of RIRs, with a market that has seen a surge in recent years, and is expected to continue to grow, is that of simulated experiences, such as virtual reality (VR), with purposes ranging from pure entertainment to decision making in building design, skill training, and therapy in mental health [29, 30, 31, 32]. In VR it is desirable to allow the user to move around in the simulated space, thereby allowing for a more immersive experience. It is necessary to have closely spaced RIRs to be able to represent small sound sources in order for the experience to be immersive [30]. The storage of said RIRs can, however, be burdensome, due to the limited storage of real-world products [173]. Furthermore, the processing needs to be as light as possible, as most of the computational resources are used for the visual rendering [30]. This highlights the need for compact storage of, and fast low-latency convolution with, RIRs.

In order to provide alleviation in terms of storage and processing, we have in previous work considered low-rank models of matricizations and tensorizations of RIR vectors [95], how RIRs can be estimated on a low-rank form [114], how this low-rank structure can be leveraged in fast, low-latency time-domain convolution [141], how low-rank models preserve objective RIR qualities and perform with respect to objective signal-based measures [174], and for multi-channel active noise control [60]. In this paper we extend upon these ideas and propose the joint compression of multiple RIRs, for the purpose of saving storage space, as well as making them amenable to fast low-latency multi-channel convolution. This compression will be done using two different methods building upon the generalized low-rank approximation of matrices (GLRAM), introduced in [175]. We propose that the set of RIRs used to find the components of GLRAM does not need to be the same as the set of RIRs one aims to compress. This allows for scenarios where the compression is learnt on one set of RIRs and then later used on another set of RIRs, possibly unknown at the time when the compression was learnt. Huang *et al.* has in [130] considered system identification from input-output data, of RIRs corresponding to adjacent source positions, on the form of a tensor decomposition.

The contribution of this paper is fourfold. Firstly, we show how multiple RIRs of a room can be compressed jointly, with less quality degradation than comparable state-of-the-art methods, using joint low-rank representations. Secondly, we show how the components of this compression vary throughout the room, and how this insight can be leveraged in the modeling. Thirdly, we demonstrate how the compression can be learnt using a set of RIRs, and then used to compress a different set of RIRs. Finally, we propose an algorithm for multi-channel

low-rank convolution with the jointly compressed RIRs, without the need to decompress these.

This paper is organized as follows: In Section 6.2 the signal model and the proposed algorithms are presented. Numerical results are presented in Section 6.3. Finally, in Section 6.4, conclusions are presented and possible areas for future research are pointed out.

### 6.1.1 Notation

We denote scalars, vectors, and matrices, by lowercase (e.g.,  $h$ ), bold lowercase (e.g.,  $\mathbf{h}$ ), and bold uppercase letters (e.g.,  $\mathbf{H}$ ), respectively. Sets are denoted by calligraphic letters (e.g.,  $\mathcal{H}$ ), and the cardinality of a set is denoted  $|\mathcal{H}|$ . Linear operators are also denoted by uppercase calligraphic letters, but it will be obvious from context what is considered.  $\mathbf{I}_n$  is an  $n \times n$  identity matrix. The selection of one or several elements from a vector or matrix will be denoted by square brackets, e.g.,  $\mathbf{H}[m : p : n, j]$  is a vector containing every  $p$ th element from the  $m$ th till the  $n$ th row of the  $j$ th column of  $\mathbf{H}$  (the omission of  $p$  indicates that every element between  $m$  and  $n$  is considered). The hat symbol,  $\hat{\cdot}$ , indicates an approximated quantity. The operator  $\mathcal{I} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  denotes the reversion of the order of the elements in a vector, i.e.  $\mathcal{I}(\mathbf{x}) = [x(n_x), x(n_x - 1), \dots, x(1)]^T$ . Finally, eigenvalues and singular values are ordered in a non-increasing fashion, with respect to the magnitude.

## 6.2 Signal Model, Algorithm, and Motivation

### 6.2.1 Signal Model

We consider a discrete-time RIR  $h(k)$ ,  $k = 1, 2, \dots, n_h$ , arranged in the vector  $\mathbf{h} \in \mathbb{R}^{n_h}$ . The RIR can be modeled as a sum of decaying sinusoids (see [95] and references therein),

$$h(\mathbf{r}_r, \mathbf{r}_s, k) = \sum_{m=1}^{m_s} \mu_m(\mathbf{r}_r, \mathbf{r}_s) e^{-\beta_m n} \cos(\omega_m n + \phi_m), \quad (6.1)$$

for  $k = 1, 2, \dots, n_h$ . Here,  $\mu_m$  denotes the initial amplitude,  $\mathbf{r}_r, \mathbf{r}_s \in \mathbb{R}^3$  the position of the receiver and the source, respectively,  $\beta_m \in \mathbb{R}_+$  the exponential decay constant,  $\omega_m \in [0, \pi]$  the angular frequency,  $\phi_m \in [0, 2\pi)$  the phase and  $m_s \in \mathbb{N}$  is the number of decaying sinusoids used in the model. For the ease of notation, we will drop the dependence on  $\mathbf{r}_r$  and  $\mathbf{r}_s$  and

refer to  $h(\mathbf{r}_r, \mathbf{r}_s, k)$  as  $h(k)$ . Consider the matricization, or reshaping, of  $\mathbf{h} = [h(1) \ h(2) \ \dots \ h(n_h)]^T$  into a matrix  $\mathbf{H} \in \mathbb{R}^{r \times c}$ ,

$$\mathbf{H} = \begin{bmatrix} h(1) & h(r+1) & \dots & h(r(c-1)+1) \\ \vdots & \vdots & & \vdots \\ h(r) & h(2r) & \dots & h(cr) \end{bmatrix}, \quad (6.2)$$

where it is assumed that  $n_h = rc$ . When a vector consisting of the sum of  $m_s$  discrete-time decaying sinusoids is reshaped into a matrix, that matrix will have rank  $2m_s$  (see, e.g., [44]). The low-rank structure of the matricized RIR is something that can be exploited for purpose of compact storage [95], as well as fast, low-latency time-domain convolution [141]. We have in previous work focused on the low-rank structure of single RIRs. Here we look also to exploit the similarity of RIRs from closely spaced source or receiver positions, by considering joint low-rank approximation of multiple matricized RIRs. We will see that the original GLRAM decomposition is too restrictive for the purpose of joint compression of RIRs when they correspond to source or receiver positions too far apart. However, by considering a modified form, good compression can be achieved and the approximated RIRs are amenable to fast low-latency time-domain convolution.

### 6.2.2 GLRAM

Consider a set of RIRs, which with (6.2) can be represented as a set of matrices  $\mathbf{H}_j \in \mathbb{R}^{r \times c}$ ,  $j = 1, 2, \dots, n$ . GLRAM constructs joint low-rank approximations  $\mathbf{H}_j \approx \hat{\mathbf{H}}_j = \mathbf{L}\mathbf{D}_j\mathbf{R}^T$ ,  $j = 1, 2, \dots, n$ , by minimizing the criterion

$$\underset{\mathbf{L}, \mathbf{R}, \mathbf{D}_j}{\text{minimize}} \quad \sum_{j=1}^n \|\mathbf{H}_j - \mathbf{L}\mathbf{D}_j\mathbf{R}^T\|_F^2, \quad (6.3)$$

$$\text{s.t.} \quad \mathbf{L}^T\mathbf{L} = \mathbf{I}_{\ell_1}, \mathbf{R}^T\mathbf{R} = \mathbf{I}_{\ell_2}$$

where  $\mathbf{L} \in \mathbb{R}^{r \times \ell_1}$ ,  $\mathbf{R} \in \mathbb{R}^{c \times \ell_2}$ ,  $\mathbf{D}_j \in \mathbb{R}^{\ell_1 \times \ell_2}$ . First  $\mathbf{L}$  and  $\mathbf{R}$  are found iteratively, and then the matrices  $\mathbf{D}_j$  are found via  $\mathbf{D}_j = \mathbf{L}^T\mathbf{H}_j\mathbf{R}$ . The matrices  $\mathbf{L}$  and  $\mathbf{R}$  are common to all  $\hat{\mathbf{H}}_j$ ,  $j = 1, 2, \dots, n$ . While  $\mathbf{L}$  and  $\mathbf{R}$  are orthogonal, much like  $\mathbf{U}$  and  $\mathbf{V}$  of a traditional singular value decomposition (SVD),  $\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ ,  $\mathbf{D}_j$  is not necessarily, in contrast to  $\mathbf{S}$ , diagonal. It is also worth noting that  $\ell_1$  and  $\ell_2$  do not have to be equal, yielding extended modeling freedom. By considering  $\ell_1 < r$ ,  $\ell_2 < c$ , and a large number of RIRs,  $n$ , the number of coefficients needed to represent the approximated matrices  $\hat{\mathbf{H}}_j$ , can be made significantly smaller than the number of coefficients needed to represent the original matrices  $\mathbf{H}_j$ . The original GLRAM algorithm can be found in [175].

### 6.2.3 Proposed Room Compression Methods

We propose two distinct methods for the simultaneous compression of multi-channel RIRs, which will be introduced in this section. We will from hereon distinguish between the set of RIRs used for finding  $\mathbf{L}$  and  $\mathbf{R}$ , denoted  $\mathcal{H}_{\text{Model}}$ , and the set of RIRs to be compressed, denoted  $\mathcal{H}_{\text{Comp}}$ . We will denote the cardinalities of these sets  $n_{\text{Model}} = |\mathcal{H}_{\text{Model}}|$  and  $n_{\text{Comp}} = |\mathcal{H}_{\text{Comp}}|$ , respectively. To reflect the distinction between finding  $\mathbf{L}$  and  $\mathbf{R}$ , summarized in Algorithm 5, and finding the matrices  $\mathbf{D}_j$  for all the RIRs of  $\mathcal{H}_{\text{Comp}}$ , summarized in Algorithm 6, we will divide the original algorithm from [175], into two distinct algorithms. For Algorithm 5, we will initialize  $\mathbf{L}$  as suggested in [175], by using

$$\mathbf{L}^{(0)} = \begin{bmatrix} \mathbf{I}_{\ell_1} \\ \mathbf{0} \end{bmatrix} \quad (6.4)$$

where  $\mathbf{0}$  is a matrix of all zeroes, of appropriate size. As indicated in [175], the algorithm generally converges in very few iterations. We will therefore not consider a stopping criterion, but rather a maximum number of iterations,  $I = 3$ . It should be noted that when considering only one matrix, i.e.  $n_{\text{Model}} = 1$  and by letting  $\ell_1 = \ell_2 = R$ , GLRAM is equivalent to an  $R$ -truncated SVD. The first proposed method, *Low-rank Compression of Room Impulse Responses for Low-latency Convolution* (LoCo-LoCo), consists of running Algorithm 5 and then Algorithm 6.

**Algorithm 5: Finding  $\mathbf{L}$  and  $\mathbf{R}$** 

- 
- 1: Input:  $\{\mathbf{H}_j\}_{j \in \mathcal{H}_{\text{Model}}}$ ,  $\mathbf{L}^{(0)}$ ,  $I$
  - 2: Output:  $\mathbf{L}$ ,  $\mathbf{R}$
  - 3: **for**  $i = 1, 2, \dots, I$  **do**
  - 4:    $\mathbf{M}_R = \sum_{j \in \mathcal{H}_{\text{Model}}} \mathbf{H}_j^T \mathbf{L}^{(i-1)} \mathbf{L}^{(i-1)T} \mathbf{H}_j$
  - 5:   Compute  $\ell_2$  first eigenvectors  $\{\phi_k^R\}_{k=1}^{\ell_2}$  of  $\mathbf{M}_R$
  - 6:    $\mathbf{R}^{(i)} \leftarrow [\phi_1^R, \dots, \phi_{\ell_2}^R]$
  - 7:    $\mathbf{M}_L = \sum_{j \in \mathcal{H}_{\text{Model}}} \mathbf{H}_j \mathbf{R}^{(i)} \mathbf{R}^{(i)T} \mathbf{H}_j^T$
  - 8:   Compute  $\ell_1$  first eigenvectors  $\{\phi_k^L\}_{k=1}^{\ell_1}$  of  $\mathbf{M}_L$
  - 9:    $\mathbf{L}^{(i)} \leftarrow [\phi_1^L, \dots, \phi_{\ell_1}^L]$
  - 10: **end for**
  - 11:  $\mathbf{L} \leftarrow \mathbf{L}^{(i)}$
  - 12:  $\mathbf{R} \leftarrow \mathbf{R}^{(i)}$
- 

**Algorithm 6: Finding  $\mathbf{D}_j$** 

- 
- 1: Input:  $\{\mathbf{H}_j\}_{j \in \mathcal{H}_{\text{Comp}}}$ ,  $\mathbf{L}$ ,  $\mathbf{R}$
  - 2: Output:  $\{\mathbf{D}_j\}_{j=1}^{n_{\text{Comp}}}$
  - 3: **for**  $j = 1, \dots, n_{\text{Comp}}$  **do**
  - 4:    $\mathbf{D}_j \leftarrow \mathbf{L}^T \mathbf{H}_j \mathbf{R}$
  - 5: **end for**
- 

Next, the difference in spatial variability between the matrices  $\mathbf{U}$  and  $\mathbf{V}$  of an SVD of an RIR matrix  $\mathbf{H}$ , and by extension, the matrices  $\mathbf{L}$  and  $\mathbf{R}$  of the GLRAM, will be discussed. In preliminary simulations it was observed that for two separate SVDs of matricized RIRs corresponding to closely spaced receiver positions, the  $\mathbf{V}$  matrices for the respective SVDs were much more similar than the respective  $\mathbf{U}$  matrices. An example of this is displayed in Fig. 6.1. There it can be seen that for three matricized RIRs, corresponding to closely spaced receiver positions, taken from [1], the columns of  $\mathbf{U}$  (left) appear to change much more from one RIR to another, as compared to the columns of  $\mathbf{V}$  (right). In Fig. 6.1 we display the first (top) and second (bottom) columns of  $\mathbf{U}$  and  $\mathbf{V}$ .

The merit of this will be further discussed in Section 6.3.2. For now, it motivates the proposal of the second method, *Low-rank Compression of Room Impulse Responses for Low-latency Convolution with Partially Invariant Transformation* (LoCo-LoCo-PIñaTa). This method consists of first finding the matrix  $\mathbf{R}$  (common to all the compressed RIRs) by running Algorithm 5 (but ignoring  $\mathbf{L}$ ), and then finding

$$\mathbf{W}_j = \mathbf{L}_j \mathbf{D}_j = \mathbf{L}_j \mathbf{L}_j^T \mathbf{H}_j \mathbf{R}, \quad (6.5)$$

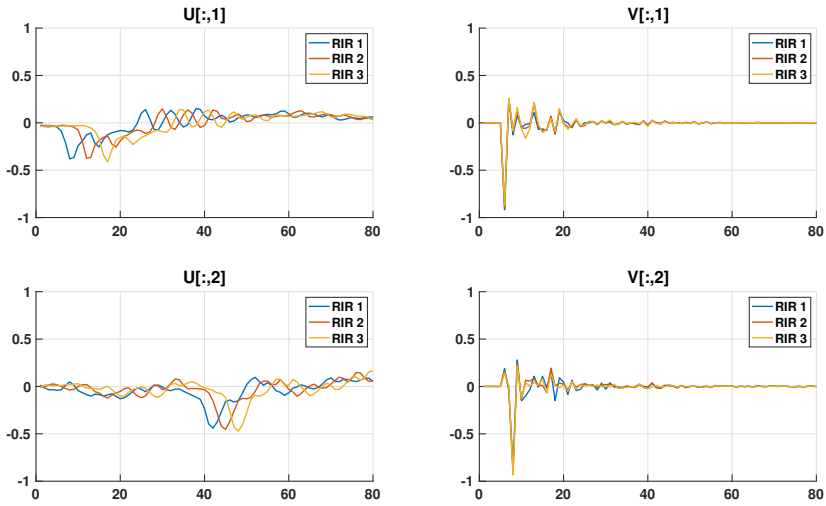


Figure 6.1: Similarity of SVD for closely spaced RIRs. Columns of  $\mathbf{U}$  (left) and  $\mathbf{V}$  (right). First column (top), second column (bottom).

$\mathbf{W}_j \in \mathbb{R}^{r \times \ell_2}$ , by running Algorithm 7.

---

**Algorithm 7:** Finding  $\mathbf{W}_j$

---

- 1: Input:  $\{\mathbf{H}_j\}_{j \in \mathcal{H}_{\text{Comp}}}$ ,  $\mathbf{R}$
  - 2: Output:  $\{\mathbf{W}_j\}_{j \in \mathcal{H}_{\text{Comp}}}$
  - 3: **for**  $j = 1, 2, \dots, n_{\text{Comp}}$  **do**
  - 4:  $\mathbf{M}_L = \mathbf{H}_j \mathbf{R} \mathbf{R}^T \mathbf{H}_j^T$
  - 5: Compute  $\ell_1$  first eigenvectors  $\{\phi_k^L\}_{k=1}^{\ell_1}$  of  $\mathbf{M}_L$
  - 6:  $\mathbf{L}_j \leftarrow [\phi_1^L, \dots, \phi_{\ell_1}^L]$
  - 7:  $\mathbf{W}_j = \mathbf{L}_j \mathbf{L}_j^T \mathbf{H}_j \mathbf{R}$
  - 8: **end for**
- 

## 6.2.4 Computational Complexity and Compression

The computationally most expensive steps of the GLRAM algorithm are the formation of the matrices  $\mathbf{M}_R$  and  $\mathbf{M}_L$ , of  $\mathcal{O}(\ell_1 c(r+c)n_{\text{Model}})$  and  $\mathcal{O}(\ell_2 r(r+c)n_{\text{Model}})$ , respectively [175], where  $\mathcal{O}(\cdot)$  refers to the limiting number of multiplications. For GLRAM, and the algorithms considered here, the eigenvalue decomposition of  $\mathbf{M}_R$  and  $\mathbf{M}_L$ , are also expensive, bounded at  $\mathcal{O}(c^3)$

and  $\mathcal{O}(r^3)$ , respectively [176]. For Algorithms 5 and 7, the most expensive step therefore depends on the values of  $c$ ,  $r$ ,  $n_{\text{Model}}$ , and  $n_{\text{Comp}}$ . For Algorithm 6, the creation of  $\mathbf{D}_j$   $j = 1, 2, \dots, n_{\text{Comp}}$  is of order  $\mathcal{O}(n_{\text{Comp}}r\ell_2(c + \ell_1))$  [175].

With compression rate, we refer to the reduction in computational storage. For a single RIR, this is defined as

$$C(\hat{\mathbf{h}}) = 1 - \frac{\Upsilon(\hat{\mathbf{h}})}{n_h}, \quad (6.6)$$

where  $n_h$  is the number of coefficients of the recorded and truncated RIR and  $\Upsilon(\hat{\mathbf{h}})$  is the number of coefficients for the compressed RIR. A compression rate close to zero means nearly no compression, whereas a compression rate closer to one means a high degree of compression. The benefit of GLRAM is that it allows us to consider  $\mathbf{L}$  and/or  $\mathbf{R}$  for the compression of multiple matrixized RIRs simultaneously. In light of this, and denoting the number of  $\mathbf{L}$  matrices by  $n_{\mathbf{L}}$ , the number of  $\mathbf{R}$  matrices by  $n_{\mathbf{R}}$ , we instead consider the compression rate for a set of  $n_{\text{Comp}}$  RIRs as

$$C(\hat{\mathcal{H}}) = 1 - \frac{n_{\mathbf{L}}r\ell_1 + n_{\text{Comp}}\ell_1\ell_2 + n_{\mathbf{R}}c\ell_2}{n_h n_{\text{Comp}}}, \quad (6.7)$$

where  $\hat{\mathcal{H}}$  is the set of compressed RIRs. When considering the scenario where  $\mathbf{R}$  is common to all the RIRs of the room, but each RIR has its own  $\mathbf{W}_j$ , the expression for the compression rate is reduced to

$$C(\hat{\mathcal{H}}) = 1 - \frac{n_{\text{Comp}}r\ell_2 + c\ell_2}{n_h n_{\text{Comp}}}. \quad (6.8)$$

## 6.2.5 Fast Low-latency Convolution

How low-rank structure to the reshaped RIR can be leveraged for fast, low-latency convolution has been explored in previous work [126, 141]. Here we consider an input signal  $\mathbf{x} \in \mathbb{R}^{n_x}$  and an output signal  $\mathbf{y} = \mathbf{h} * \mathbf{x}$ ,  $\mathbf{y} \in \mathbb{R}^{n_y}$ , where  $n_y = n_h + n_x - 1$ . In the simplest case, where the RIR  $\mathbf{h}$  is reshaped into a rank-1 matrix  $\mathbf{H}$ , the filtering operation with a very long filter (i.e., the RIR) is replaced by two filtering operations with significantly shorter filters, corresponding to the columns of  $\mathbf{W}_j$  and  $\mathbf{R}$ . This unveils another benefit of a matrix  $\mathbf{R}$  common to all the RIRs. The convolution between an audio signal and  $\mathbf{R}$  needs to be done only once and the result can then be reused for all the considered RIRs. The extension to arbitrary rank of the matrix  $\mathbf{H}$  is straightforward as a rank- $R$  matrix is the sum of  $R$  rank-1 matrices. The filtering of the signal through the entire RIR can be replaced by the filtering through  $2R$  shorter filters.

The approach is summarized in Algorithm 8.<sup>1</sup> For Algorithm 8, we consider complexity in terms of number of multiply-add instructions. The creation of  $\mathbf{P}$  requires  $(n_x + (c - 1)r)\ell_2(\lfloor \frac{n_x}{r} \rfloor + 1)$  multiply-add instructions and the creation of  $\{\mathbf{y}_j\}_{j=1}^{n_{\text{Comp}}}$  requires in total  $n_{\text{Comp}}n_y\ell_2r$  multiply-add instructions. For most relevant scenarios  $(n_x + (c - 1)r)\ell_2(\lfloor \frac{n_x}{r} \rfloor + 1) \ll n_{\text{Comp}}n_y\ell_2r$ . Traditional time-domain multi-channel convolution, assuming  $n_h \geq n_x$ , requires in total  $n_{\text{Comp}}n_y n_h$  multiply-add instructions, hence the proposed approach yields a reduction of the computational cost by a factor of  $\approx \frac{c}{\ell_2}$ . A similar algorithm, exploiting both similar  $\mathbf{L}$  and  $\mathbf{R}$ , could be envisaged, but as will be shown in Section 6.3, LoCo-LoCo-PIñaTa is better suited for the purpose of compressing the RIRs of an entire room for low-latency convolution. For the sake of brevity, such an algorithm will therefore not be considered.

---

**Algorithm 8:** Multi-channel Low-rank Convolution
 

---

```

1: Input:  $\{\mathbf{W}_j\}_{j \in \mathcal{H}_{\text{Comp}}}, \mathbf{R}, \mathbf{x}$ 
2: Output:  $\{\mathbf{y}_j\}_{j=1}^{n_{\text{Comp}}}$ 
3: for  $j = 1, \dots, n_x + (c - 1)r$  do
4:   for  $\ell = 1, \dots, \ell_2$  do
5:      $R_{\text{Low}} = \max(\lceil \frac{j - n_x}{r} \rceil + 1, 1)$ 
6:      $R_{\text{High}} = \min(\lceil \frac{j}{r} \rceil, c)$ 
7:      $x_{\text{Low}} = \max(\text{mod}(j - 1, r) + 1, j - (c - 1)r)$ 
8:      $x_{\text{High}} = \min(n_x - r + 1 + \text{mod}(j - 1 - n_x, r), j)$ 
9:      $\mathbf{P}[j, \ell] = \mathbf{R}[R_{\text{Low}} : R_{\text{High}}, \ell] \mathcal{I}(\mathbf{x}[x_{\text{Low}} : r : x_{\text{High}}])$ 
10:   end for
11: end for
12: for  $j = 1, \dots, n_{\text{Comp}}$  do
13:   for  $k = 1, \dots, n_y$  do
14:      $P_{\text{Low}} = \max(k - r + 1, 1)$ 
15:      $P_{\text{High}} = \min(k, n_y - r + 1)$ 
16:      $W_{\text{Low}} = \max(k - (n_y - r), 1)$ 
17:      $W_{\text{High}} = \min(k, r)$ 
18:     for  $\ell = 1, \dots, \ell_2$  do
19:        $\mathbf{p} = \mathbf{P}[P_{\text{Low}} : P_{\text{High}}, \ell]$ 
20:        $\mathbf{w} = \mathcal{I}(\mathbf{W}_j[W_{\text{Low}} : W_{\text{High}}, \ell])$ 
21:        $\mathbf{y}_j[k] = \mathbf{y}_j[k] + \mathbf{w}^T \mathbf{p}$ 
22:     end for
23:   end for
24: end for

```

---

<sup>1</sup>A Matlab implementation of Algorithm 8 is available at [https://github.com/m-jalmy/multichannel\\_conv\\_example](https://github.com/m-jalmy/multichannel_conv_example)

## 6.3 Numerical Results

### 6.3.1 Compact-array RIR Measurements

In the first example we will be using RIRs from the *single- and multichannel audio recordings database* (SMARD) [1]. These are recorded at 48 kHz in a 60 m<sup>2</sup> room, with a reverberation time of approximately 0.15 s. The RIRs will be truncated at  $n_h = 6400$  samples, corresponding to 0.13 seconds. We will consider the RIRs recorded with orthogonal arrays, which in turn consist of 3 uniform linear arrays (ULA) with 7 microphones (5 cm spacing) for each, recorded at 24 different source-receiver position configurations, yielding a total of 504 RIRs. In this scenario  $\mathcal{H}_{\text{Model}} = \mathcal{H}_{\text{Comp}}$ , i.e., the set of matrices used for finding  $\mathbf{L}$  and  $\mathbf{R}$  will be the same as the set of matrices to be compressed. We will consider two different versions of LoCo-LoCo and LoCo-LoCo-PIñaTa, respectively. For the first ones, which we denote LoCo-LoCo 7 and LoCo-LoCo-PIñaTa 7, we will consider a different  $\mathcal{H}_{\text{Model}}$  for each ULA of the orthogonal array, i.e.  $n_{\text{Model}} = 7$ . For the other ones, LoCo-LoCo 21 and LoCo-LoCo-PIñaTa 21,  $\mathcal{H}_{\text{Model}}$  is the entire orthogonal array, i.e.  $n_{\text{Model}} = 21$ . These two versions of LoCo-LoCo will be compared to two other approaches. Firstly, a low-rank approximation using a truncated SVD, for each RIR. The  $R$ -truncated SVD of a matrix  $\mathbf{H}$  is the closest rank- $R$  matrix to  $\mathbf{H}$  in both the 2-norm and in the Frobenius norm [177]. The drawback from a compression point of view, however, is that for each compressed matricized RIR  $\hat{\mathbf{H}}_j = \mathbf{U}_j[:, 1 : R]\mathbf{S}_j[1 : R, 1 : R]\mathbf{V}_j[:, 1 : R]^T$ , the three matrices  $\mathbf{U}_j$ ,  $\mathbf{S}_j$ , and  $\mathbf{V}_j$  are unique to each approximated RIR  $\hat{\mathbf{H}}_j$  and have to be stored. This means that for a fixed compression rate, the rank for the truncated-SVD approximation method will be significantly lower than  $\ell_1$  and  $\ell_2$  for LoCo-LoCo and LoCo-LoCo-PIñaTa. For the  $R$ -truncated SVD,  $C(\hat{\mathbf{h}}) = 1 - R(c + r)/n_h$ . For LoCo-LoCo, LoCo-LoCo-PIñaTa, and SVD, the truncated RIR vectors are reshaped into  $80 \times 80$ -matrices. We will also consider the state-of-the-art Opus interactive speech and audio codec [144, 145]. The Opus codec is created for the compression of audio, but has recently been considered for the compression of RIRs too [149]. The Opus encoding was done using Matlab's *audiowrite*. Although Opus shrinks the file size of the stored RIR, the number of coefficients remains the same. The error is measured in terms of normalized misalignment,

$$\mathcal{M}_{\text{dB}}(\hat{\mathbf{h}}) = 20 \log_{10} \left( \frac{1}{n_{\text{Comp}}} \sum_{j=1}^{n_{\text{Comp}}} \frac{\|\hat{\mathbf{h}}_j - \mathbf{h}_j\|_2}{\|\mathbf{h}_j\|_2} \right). \quad (6.9)$$

The results are shown in Fig. 6.2. There it can be seen that for the RIRs corresponding to the closely spaced microphones of SMARD, on either the

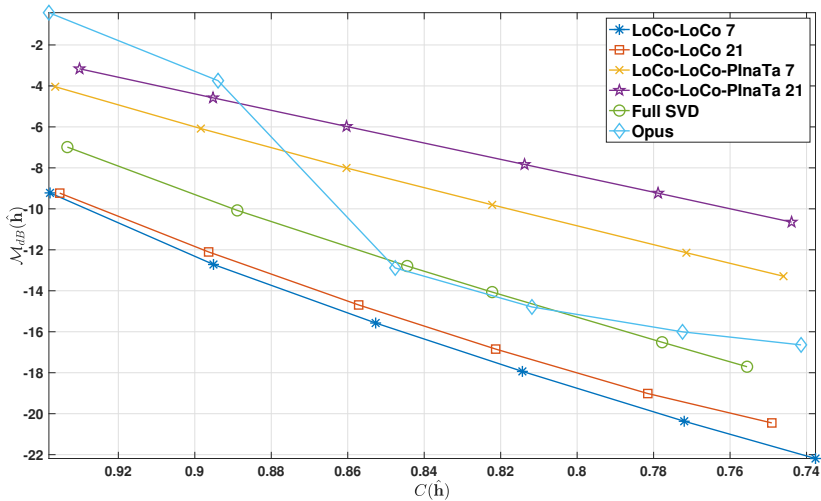


Figure 6.2: Normalized misalignment,  $\mathcal{H}_{\text{Model}} = \mathcal{H}_{\text{Comp}}$ .

full array or one for each ULA, LoCo-LoCo performs the best. LoCo-LoCo-PiNaTa, for which  $\ell_1$  and  $\ell_2$  is much smaller than for LoCo-LoCo, given a fixed compression rate, fails to perform on par with the benchmark methods, except for very high compression rates, where the performance of Opus declines considerably.

### 6.3.2 Spatial Variation of $\mathbf{U}$ and $\mathbf{V}$

The difference in how much of the spatial variation is reflected in  $\mathbf{U}$  and  $\mathbf{V}$ , respectively, is highlighted in the following example. We consider the same RIRs as in Section 6.3.1 and use an  $R$ -truncated SVD with  $R = 12$ . For each ULA of the orthogonal array, separate SVDs are made of the matricized RIRs, denote these  $\mathbf{U}_n$  and  $\mathbf{V}_n$ , respectively,  $n = 1, 2, \dots, 7$ . It is then explored what error is induced by using either  $\mathbf{U}_1$  or  $\mathbf{V}_1$ , in place of  $\mathbf{U}_n$  and  $\mathbf{V}_n$ , respectively, for the low-rank approximation. The averaged results are found in Fig. 6.3 (top). For reference, using both  $\mathbf{U}_n$  and  $\mathbf{V}_n$  resulted in a normalized misalignment of  $\sim -18$  dB. As expected, using  $\mathbf{U}_n$  and  $\mathbf{V}_n$  is by far the best, but the substantial outperformance by the case where  $\mathbf{V}_1$  is kept constant, as compared to when  $\mathbf{U}_1$  is kept constant, is an indication that more of the spatial invariance of adjacent RIRs is captured by  $\mathbf{V}$ .

This is also seen when considering the subspace angles (see, e.g., [178]). For

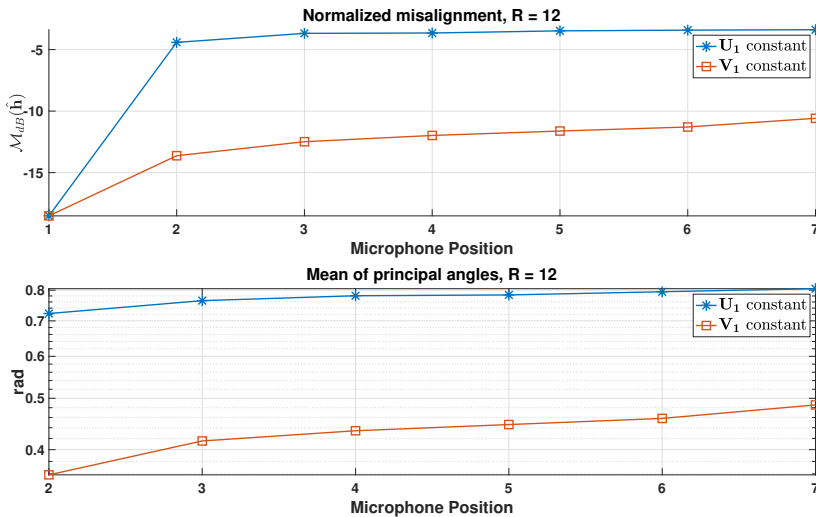


Figure 6.3: Spatial variation of  $\mathbf{U}$  and  $\mathbf{V}$  (top) and mean of principal angles between subspaces spanned by the columns of  $\mathbf{U}_1$  and  $\mathbf{U}_n$ , and  $\mathbf{V}_1$  and  $\mathbf{V}_n$ , respectively.

a matrix  $\mathbf{U} \in \mathbb{R}^{m \times n}$ , let  $\mathcal{U} = \{\mathbf{U}[:, 1], \mathbf{U}[:, 2], \dots, \mathbf{U}[:, n]\}$  denote the set of its column vectors, and  $U$  the subspace of  $\mathbb{R}^m$  that the vectors of  $\mathcal{U}$  span. For two matrices,  $\mathbf{U} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{V} \in \mathbb{R}^{m \times p}$ , the principal angles  $\theta_1 \leq \theta_2 \leq \dots \leq \theta_{\min(n,p)} \leq \pi/2$  between the two subspaces  $U$  and  $V$  and the corresponding principal directions  $\mathbf{u}_k \in U$  and  $\mathbf{v}_k \in V$  are defined recursively as

$$\begin{aligned} \cos(\theta_k) &= \mathbf{u}_k^T \mathbf{v}_k = \\ & \max_{\mathbf{u} \in U, \mathbf{v} \in V} \mathbf{u}^T \mathbf{v} \\ \text{s.t. } & \|\mathbf{u}\| = \|\mathbf{v}\| = 1 \\ & \mathbf{u}^T \mathbf{u}_i = \mathbf{v}^T \mathbf{v}_i = 0, i = 1, \dots, k-1. \end{aligned} \tag{6.10}$$

In Fig. 6.3 (bottom) we see the mean principal angles, averaged over the three ULA's of the 24 source-receiver position configurations, as a function of microphone position. The large degree of similarity between the subspaces  $V_1$  and  $V_n$ ,  $n = 2, 3, \dots, 7$ , compared to  $U_1$  and  $U_n$ ,  $n = 2, 3, \dots, 7$ , is evident. We will now consider how this can be exploited.

Table 6.1: Damping coefficients and reverberation times, Section 6.3.3

Surface / Hz	125	250	500	1000	2000	4000
Walls	0.1	0.2	0.4	0.6	0.5	0.6
Ceiling	0.02	0.02	0.03	0.03	0.04	0.07
Floor	0.02	0.02	0.03	0.03	0.04	0.07
$T_{60}$	1.23	0.63	0.33	0.22	0.26	0.21

### 6.3.3 Synthetic Room Impulse Responses

We will here consider synthetically generated RIRs, in order to further investigate the spatial variation captured by  $\mathbf{L}$  and  $\mathbf{R}$ , respectively. This allows for a scenario where the location of the loudspeaker generating the signal is fixed, and we consider the RIRs for microphones placed on a finely spaced grid throughout the room. The RIRs considered in this section have been generated using the image source method [7, 179]. The three-dimensional room is 3.7 m  $\times$  3.1 m  $\times$  3.2 m, the loudspeaker is placed at [2.62, 1.4, 1.6] m, and microphones are placed on a two-dimensional Cartesian grid with spacing 0.06 m, also at 1.6 m above the floor, starting 0.2 m from each wall. The frequencies of the excitation signal, the damping coefficients for the respective surfaces and frequencies, and the reverberation time,  $T_{60}$ , for the respective frequencies, are stated in Table 6.1. The RIRs are truncated to 0.33 s, corresponding to  $n_h = 15625$  samples, given the sampling rate of 48 kHz.

In this scenario we initially considered LoCo-LoCo and LoCo-LoCo-PIñaTa, and truncated SVDs as a baseline. For LoCo-LoCo,  $\ell_1 = \ell_2 = 56$ , for LoCo-LoCo-PIñaTa,  $\ell_1 = \ell_2 = 21$ , and for truncated SVD,  $R = 12$ , all corresponding to  $C(\hat{\mathbf{h}}) \approx 0.8$ , due to the fact that the number of coefficients shared by the compressed multi-channel RIRs varies strongly between the methods. For LoCo-LoCo and LoCo-LoCo-PIñaTa, the matrices common to all the compressed RIRs ( $\mathbf{L}$  and  $\mathbf{R}$  for LoCo-LoCo and  $\mathbf{R}$  for LoCo-LoCo-PIñaTa) were found using a randomly chosen subset,  $\mathcal{H}_{\text{Model}}$ , with  $n_{\text{Model}} = 25$ , of the all RIRs to be compressed,  $\mathcal{H}_{\text{Comp}}$ , with  $n_{\text{Comp}} = 2576$ . Preliminary simulations showed poor performance by LoCo-LoCo. Despite the comparatively large  $\ell_1$  and  $\ell_2$ , the matrices  $\mathbf{L}$  and  $\mathbf{R}$  are unable to capture the RIR variability throughout the room. Therefore, the results for LoCo-LoCo will not be shown here.

LoCo-LoCo-PIñaTa, on the other hand, is able to make use of the invariance of  $\mathbf{R}$  throughout the room, while capturing the variability in  $\mathbf{W}_j$ . In order to be able to reliably represent RIRs from anywhere in the room, it is important to not only use RIRs that are far away from the source in  $\mathcal{H}_{\text{Model}}$ . If only RIRs too far from the source are used when modeling  $\mathbf{R}$ , the entries of the top row

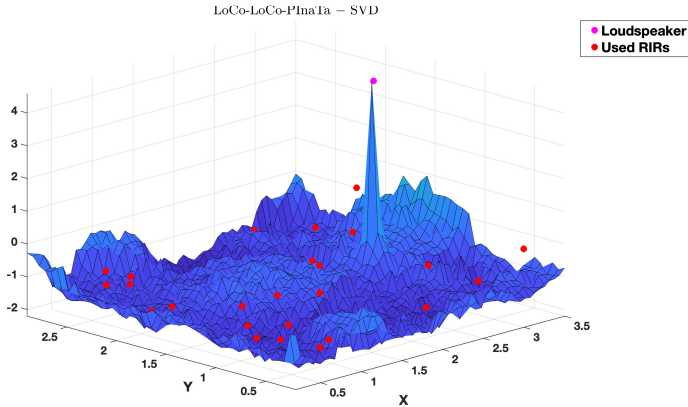


Figure 6.4: Difference between normalized misalignment for LoCo-LoCo-PIñaTa and normalized misalignment for SVD.

of  $\mathbf{R}$  will be 0 which, in turn, will cause the first  $r$  taps of  $\hat{\mathbf{h}}$  to be 0 as well. If then the direct component of an RIR in the room is at one of the first  $r$  samples, LoCo-LoCo-PIñaTa will be unable to faithfully restore the direct component, causing large misalignment.

The difference in normalized misalignment for LoCo-LoCo-PIñaTa and the SVD approximation is displayed in Fig. 6.4. The locations at which the RIRs of  $\mathcal{H}_{\text{Model}}$  for LoCo-LoCo-PIñaTa are sampled are marked in red. In Fig. 6.4 it can be seen that LoCo-LoCo-PIñaTa performs better than the baseline, with the exception of a very small area in direct proximity of the source. For reference, the average normalized misalignment was  $-19.98$  dB for LoCo-LoCo-PIñaTa and  $-18.73$  dB for the baseline truncated SVD.

### 6.3.4 Distributed-array RIR Measurements

The findings from Section 6.3.3 are confirmed when applying the same methods to real data. In this section we consider RIRs from the dataset S32-M441 from [52]. Here, RIRs are recorded by microphones in a  $1 \text{ m} \times 1 \text{ m}$  planar grid, every  $0.05 \text{ m}$ , from  $\{x, y | -0.5 \leq x, y \leq 0.5\}$ , yielding 441 RIRs, with a source at  $[1 \ 0.45 \ -0.1]$ , relative to the middle of the microphone grid. The RIRs are recorded in a room with approximate dimensions of  $7.0 \text{ m} \times 6.4 \text{ m} \times 2.7 \text{ m}$ , with  $T_{60} = 0.19 \text{ s}$ . We consider  $n_h = 8100$ , corresponding to  $0.17 \text{ s}$ , and the RIRs are reshaped into  $90 \times 90$  matrices. Much like in Section 6.3.3,

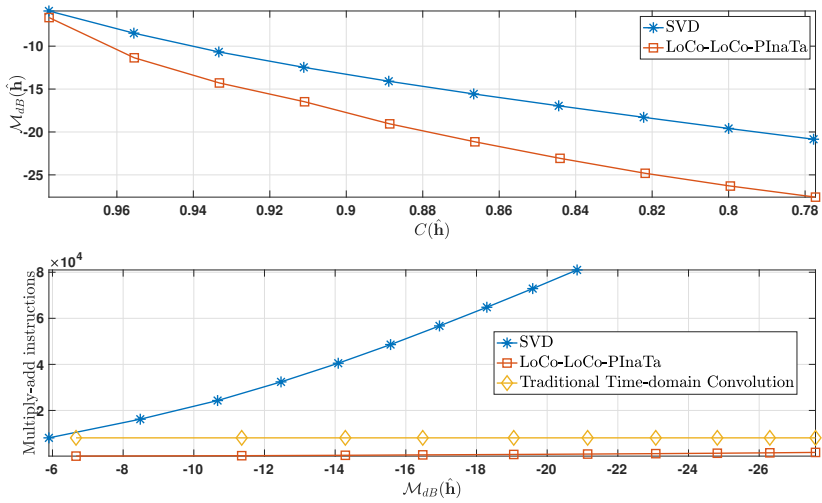


Figure 6.5: Normalized misalignment for distributed-array RIR measurements.

LoCo-LoCo fell short in preliminary simulations and will not be considered further. For LoCo-LoCo-PIñaTa,  $\mathbf{R}$  was found using 40 randomly chosen RIRs. The average normalized misalignment is displayed in Fig. 6.5, in the top plot as a function of compression rate and in the bottom plot as a function of convolution complexity, in terms of number of multiply-add instructions, when considering convolution with a signal of length  $n_x = 100$ . An example RIR, and the respective approximations, are displayed in Fig. 6.6, zoomed in at samples 2800 – 4000 in the time domain and the range 0 – 6000 Hz in the frequency domain. LoCo-LoCo-PIñaTa’s superior performance is evident in the time domain at the samples from 3200 to 3400, as it is able to capture a longer part of the original RIR. In the frequency domain, LoCo-LoCo-PIñaTa is able to better represent the dominant modal peaks.

There is a noticeable difference in improvement for using LoCo-LoCo-PIñaTa, as opposed to a traditional SVD, for the scenarios considered in Sections 6.3.3 and 6.3.4. This is likely due to the fact that the synthetic RIRs of Section 6.3.3 display very little modal behavior, as compared to the recorded RIRs of Section 6.3.4, as the low-rank property of a matrix such as the one in (6.2) is dependent on modal behavior, as described in (6.1). This is displayed in Fig. 6.7, where the magnitude response of a measured RIR from [52] (top) and a synthetic RIR (bottom) are plotted. The magnitude of the synthetic RIR is dominated by the cavity mode at 0 Hz, whereas the measured RIR from [52] contains several distinct modal peaks.

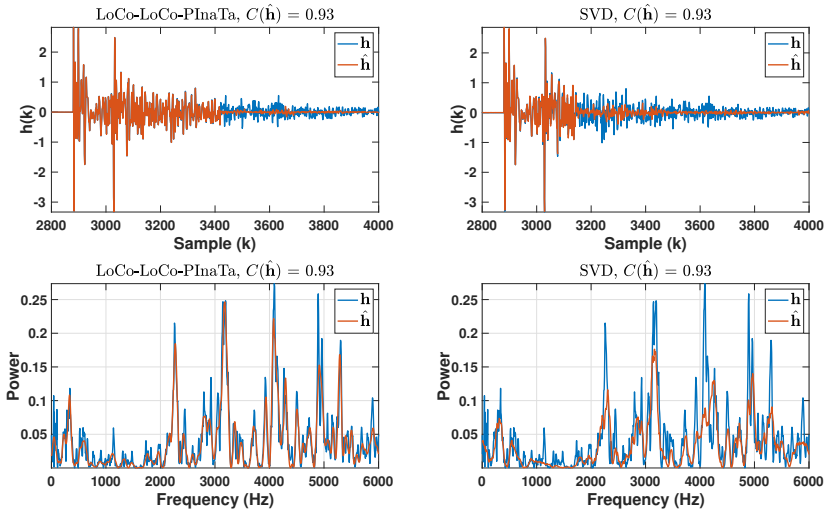


Figure 6.6: Example RIR from [52] and compressed RIRs obtained with LoCo-LoCo-PIñaTa (left) and SVD (right), in the time domain (top) and in the frequency domain (bottom).

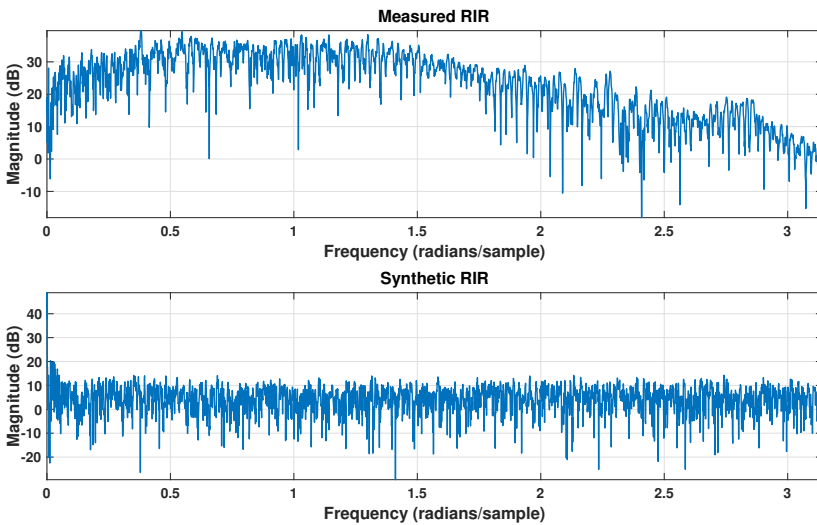


Figure 6.7: Example RIR from [52] (top) and synthetic RIR (bottom).

Table 6.2: Changes in source position

Measure	Common $\mathbf{R}$	Unique $\mathbf{R}$
$C(\hat{\mathbf{h}})$	0.82	0.82
$\ell_1 = \ell_2$	16	16
$\mathcal{M}_{\text{dB}}(\hat{\mathbf{h}})$	-23.95	-23.96
Multiply-add instructions per output sample	$1.44 \cdot 10^3$	$1.44 \cdot 10^3$
Retrieval of $\mathbf{R}$	0.18 s	2.87 s

### 6.3.5 Changes in Source Position

In this section we investigate how the different methods work for a scenario where the position of the source changes. The dataset S32-M441 from [52] that we will be using, contains RIR measurements from 441 different microphones and 32 different sources, for a total of  $441 \cdot 32 = 14112$  RIRs. The source position is varied along two separate rectangles around the microphone grid, one placed 0.1 m above the microphones, the other placed 0.1 m below the microphones, each containing 16 source positions. Again we consider  $n_h = 8100$ , each truncated RIR is reshaped into a  $90 \times 90$  matrix, and we let  $\ell_1 = \ell_2 = 16$ . We consider the difference in the quality of RIR approximation between two different versions of LoCo-LoCo-PIñaTa. In the first one, called *Common  $\mathbf{R}$* , only one matrix  $\mathbf{R}$  is found for each rectangle of sources, using 40 randomly selected RIRs out of the 441 RIRs measured for one of the 16 source positions. For the second one, called *Unique  $\mathbf{R}$* , a separate  $\mathbf{R}$  will be used for each source position, found using the same 40 randomly selected RIRs as for the first version. The compression rate is 0.82 in both cases. Averaged over all RIRs, the normalized misalignment is -23.95 dB for the *Common  $\mathbf{R}$*  version and -23.96 dB for the *Unique  $\mathbf{R}$*  version. The number of multiply-add instructions per output sample when considering convolution with a signal of length  $n_x = 100$  is  $1.44 \cdot 10^3$  in both scenarios. The main difference is the time spent retrieving  $\mathbf{R}$ . This needs to be done only twice for the *Common  $\mathbf{R}$*  version, but 32 times for the *Unique  $\mathbf{R}$*  version. Simulations were done using Matlab 2022b on a 2018 MacBook Pro with a 2.7 GHz QuadCore Intel Core i7 processor. Averaged over 50 Monte Carlo-simulations, the *Common  $\mathbf{R}$*  version spent 0.18 s on the retrieval of  $\mathbf{R}$  whereas *Unique  $\mathbf{R}$*  spent 2.87 s. The results are summarized in Table 6.2. The minute difference in misalignment between the two considered versions indicates that the same matrix  $\mathbf{R}$  can be used for the compression of multi-channel RIRs corresponding to multiple source positions throughout a room.

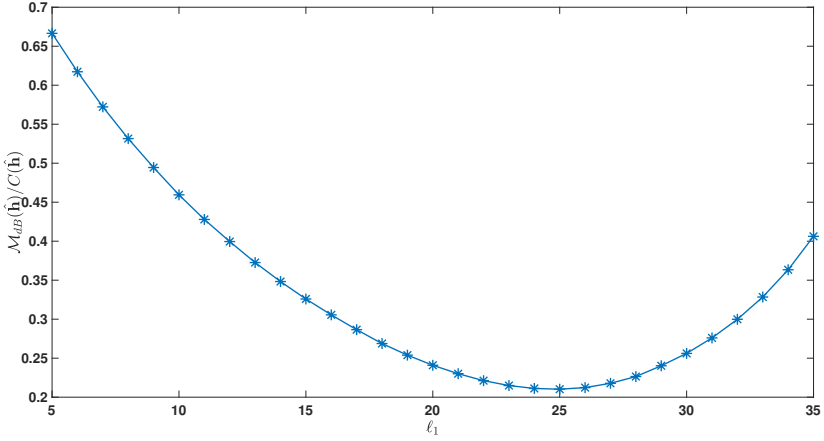


Figure 6.8: Normalized misalignment per compression rate, for varying  $\ell_1$  and  $\ell_2$ .

### 6.3.6 The Relationship Between $\ell_1$ and $\ell_2$

Up until this point we have considered only the case where  $\ell_1 = \ell_2$ , where we remind the reader that  $\ell_1$  and  $\ell_2$  define the size of  $\mathbf{D}_j \in \mathbb{R}^{\ell_1 \times \ell_2}$  in the approximated RIR  $\hat{\mathbf{H}}_j = \mathbf{L}\mathbf{D}_j\mathbf{R}^T$ . One of the advantages of the GLRAM, as compared to the SVD is that  $\ell_1$  and  $\ell_2$  can be chosen independently from each other. The previously established higher degree of similarity of the  $\mathbf{R}$  matrices, as compared to the  $\mathbf{L}$  matrices, for RIRs corresponding to adjacent receiver positions is an indication that  $\ell_1 = \ell_2$  might not necessarily be the best choice. To demonstrate this, using the RIRs from [1] as described in Section 6.3.1, we vary  $\ell_1$  between 5 and 35, while  $\ell_2 = 40 - \ell_1$ . The results are shown in Fig. 6.8. In order to be able to illustrate both normalized misalignment and compression in the same figure, the  $y$ -axis in Fig. 6.8 is average normalized misalignment per compression rate. Noticeable in Fig. 6.8 is that the minimum does not occur at  $\ell_1 = \ell_2 = 20$ , but rather at  $\ell_1 = 25, \ell_2 = 15$ . This further strengthens the conclusion that it is favorable to yield more modeling capacity to  $\mathbf{L}$ , given that it absorbs more of the spatial variation. This is beneficial when considering the multi-channel low-rank convolution in Algorithm 8. As previously concluded, with a common  $\mathbf{R} \in \mathbb{R}^{c \times \ell_2}$ , the matrices  $\mathbf{L}_j \in \mathbb{R}^{r \times \ell_1}$  and  $\mathbf{D}_j \in \mathbb{R}^{\ell_1 \times \ell_2}$  can be stored together as  $\mathbf{W}_j \in \mathbb{R}^{r \times \ell_2}$ . A smaller  $\ell_2$  decreases the overall complexity of the convolution.

## 6.4 Conclusions

In this paper we propose two novel methods for the joint compression of multiple RIRs by use of joint low-rank approximations. The first proposed method, LoCo-LoCo, proved better than the benchmark methods, truncated SVD and state-of-the-art audio compression standard Opus, in scenarios where the RIRs to be approximated correspond to very closely spaced receivers. In scenarios where the receivers are farther apart, the other proposed method, LoCo-LoCo-PIñaTa, outperformed the benchmark methods, exploiting the demonstrated spatial invariability in one of the components of the GLRAM decomposition used throughout this paper. The compressed multi-channel RIRs yielded by the proposed methods are amenable to fast low-latency multi-channel convolution and for multi-channel RIRs compressed with LoCo-LoCo-PIñaTa we provide an explicit convolution algorithm.

Previous research has revealed a significant improvement in performance when considering a 3D tensor approximation of a single RIR, as opposed to a 2D matrix approximation, given a fixed compression rate. This gives reason to believe that tensor approximations could lead to an improvement of the joint compression of multiple RIRs considered in this paper and should be the focus of future research.



# Chapter 7

## Valorization

No research is created in a void. Needs spark innovation and examples of how innovation has changed our lives throughout history are plentiful. In this chapter we will look at three possible areas for valorization of the research conducted in this thesis, virtual reality (VR) and augmented reality (AR), sound zone control, and the cultural heritage preservation of important acoustic spaces.

### 7.1 VR/AR

Through commercial products like Meta Quest, PlayStation VR, and HTC Vive, simulated experiences like VR are quickly transforming from futuristic science fiction to something for every-day use. The rapid growth up until this point is due to improvement in the underlying technology, in form of greater processing power and richer audio [180]. The market for VR has grown rapidly in recent years, and the growth is expected to continue [29]. The global market was valued at \$19.44 billion in 2022 and is projected to grow to \$165.91 billion by 2030, with a compounded annual growth rate of 31%, with live virtual entertainment being the main driving factor [181]. VR can be used for a wide range of purposes, including gaming and entertainment [29], architectural acoustics [66], skill training [31], decision making in building design [30], and mental health therapy [32].

Although VR is becoming more common, it is still not close to other digital devices, such as PCs, TVs, tablets, and smartphones. VR becoming as commonplace as these hinges on it being able to do things the other devices

cannot, by delivering truly immersive experiences [180]. Although the visual aspect is undoubtedly the most striking thing about VR, good reverberation is just as important for an immersive experience [33]. For most VR applications, however, modalities apart from the visuals are presented as effects, without proper physical based reference to real-world properties [66]. Artificial reverberation for VR is challenging. The storage space of real-world products is limited [173]. The processing needs to be light, as for most VR applications, the lion's share of the computational resources are dedicated to the visual rendering [30]. Latency from motion tracking, etc. needs to be taken into account [66]. Different reverberation simulation techniques have their strengths and weaknesses. Ray-based methods simulate early reflections well, but struggle when it comes to the late reverberation. Wave-based methods are accurate and can achieve good spatial awareness, but become prohibitively computationally expensive when considering higher frequencies [182].

For the assessment of VR, there are two important terms, immersion and presence. They are often used interchangeably, but whereas immersion refers to the technology and its ability to generate vivid illusions of reality, presence refers to the sense of truly being in the virtual environment [183]. VR is known to potentially cause VR sickness, with symptoms like nausea, fatigue, and disorientation. This effect can be mitigated, however, by the inclusion of multi-sensory information, such as auditory, olfactory, and tactile information [184]. Good audio rendering in VR has been shown to increase the sense of presence [185], which in turn is negatively correlated with VR sickness [186].

Similar to VR, AR is different in that it does not create an entirely simulated environment, but rather augments information to the already existing world. As opposed to VR, for AR, the real-world surroundings have to be taken into account and the acoustic properties have to be estimated. This is not seldomly done under adverse conditions, as setting up loudspeakers and microphones for proper RIR measurements in such situations is hardly practical [187]. Considering the merit shown by the low-rank models under adverse conditions and conditions of uncertainty, the employment of low-rank models in AR could prove very useful. In simulated experiences, a distinction is made between *plausibility* and *authenticity*. For an experience to be authentic it has to be perceptually identical to an external reference, whereas for it to be plausible it is enough that it is indistinguishable from our expectations, or internal reference. For VR, as opposed to for AR, no external reference is available, why authenticity is considered more important in AR [187].

The research conducted in this thesis revolves around the compact representation of the impact of a room on a sound signal, for efficient storage and fast processing. As been shown throughout this thesis, low-rank modeling of RIRs allow for very compact storage of RIRs, without considerable quality loss. In the context

of AR and VR, this is paramount, due to the mentioned limited storage on real-world products. The algorithms for low-rank, low-latency convolution provides a framework for the so crucial fast processing. The fact that the RIRs can be used for fast low-latency convolution on their compressed form is a very appealing property and gives reason to believe that the ideas of exploiting low-rank in the modeling of RIRs explored in this thesis could help push the boundaries of AR and VR.

## 7.2 Sound Zone Control

In sound zone control, the aim is to achieve a desired sound field within a restricted area, while affecting other areas of the space as little as possible. Typically, the room is divided into two sets of areas. The bright zones, where a certain sound is to be reproduced, and the dark zones, in which the acoustic energy is to be minimized. Multiple microphones and loudspeakers are used to control the reproduced sound field [39]. Described as a headphone experience without headphones [188], it is making its way out of the experimental lab and into our homes [189]. Its usefulness has also been demonstrated in, among others, airplanes [190] and car cabins [191]. Accurate knowledge of the RIR is imperative to good sound zone control [192]. Sound zone control and is an active market with companies like NTT Sonority, Harman, USound, and Hyundai.

Closely related to the field of sound zone control is that of active noise control (ANC), in which, by means of destructive interference, sound is canceled out in a particular area [193]. The performance of the reproduction of the sound field is sensitive to errors in the acoustic transfer function. For ANC algorithms, convergence speed, computational complexity, and robustness are of the highest concern. The design of adaptive sound field control systems, to be able to deal with a variable acoustic environment, is believed to be a fruitful area of future research within sound zone control [194]. The applicability of low-rank decompositions of RIRs for multi-channel ANC was recently demonstrated in [60].

The findings of Chapter 6 show how the low-rank ideas explored in this thesis can be used for compact storage and fast multi-channel convolution with audio signals. Therein we present two different methods for the joint compression of RIRs. The first one, LoCo-LoCo, where both the left and right transformation is common to the compressed RIRs, proved more useful when the receiver positions of the RIRs are closely spaced. The second one, LoCo-LoCo-PIñaTa, was preferred when the receivers are farther apart. This way, great flexibility

is provided depending on the scenario considered, as both methods yield approximations amenable to fast low-latency convolution.

### 7.3 Cultural Heritage Preservation of Historically Important Buildings

Music and architecture are two cultural expressions that have always been interlaced through acoustics. The Notre-Dame fire of 2019 was undoubtedly a tragic event, leaving most of the roof and spire of the cathedral destroyed. However, it serves as an excellent example of the importance of the digital preservation of the acoustics of historically important buildings. While photos and drawings can help us restore the visuals, with a building like Notre-Dame, the acoustics of the space is a big part of what makes it special. Luckily for us, extensive measurements of the acoustics of Norte-Dame were made a couple of years prior to the fire. With the help of computer simulations and post-fire measurements, the pre-fire measurements can help to guide the rebuilding of Notre-Dame in order to restore it as well as possible to its prior condition [195, 196]. In a similar fashion, Teatro La Fenice in Venice has been reconstructed after the fire that destroyed it in 1996, thanks to measurements made just two months before the fire, helping to restore the acoustics of the original building [197].

Archaeoacoustics studies archeological sites through their acoustic properties, or the application of acoustics in archeological spaces. The number of studies of sound in archeological sites is quickly increasing and providing us with invaluable knowledge about the behavior of ancient societies [198]. Examining the acoustics of significant cultural monuments has been deemed to be of immense importance for the exploration of historical and cultural contexts [199]. Archaeoacoustics can provide an empirical complement to historical archeology and can allow us to demonstrate the acoustics of a historical location not only in situ, but possibly also off site. Acoustic measurements can, when properly interpreted and in conjunction with archeological evidence, help make the speculative plausible [200].

Taking the acoustics into account when constructing a building long precedes our modern-day understanding of acoustics [197, 198]. Despite acoustics being carefully considered from an architectural point of view, the acoustics of many historical spaces are yet to be examined. Deepened knowledge about the acoustics of a historical site can provide important understanding about the life and customs of the people that used it [198]. For instance, through modeling and measurements, researchers have been able to make conclusions about the

purpose of buildings at the Forum Romanum, based on the speech intelligibility in different parts of the buildings and how many people could have heard a speaker, granting additional insight to the life in ancient Rome [201].

As described in Chapter 1, the storage of RIRs quickly becomes problematic, when considering large, reverberant spaces. The measurement standard of today (ISO 3382-1:2009) calls only for the characteristics of an acoustic space, in the form of parameters such as reverberation time, early decay time, center time, etc. Suggestions have been made to make the storage more comprehensive [199]. The ideas for compact RIR representation proposed in this thesis could facilitate such an endeavor. This is a valorization pathway that will be further explored by our research group in close collaboration with the Alamire Foundation, an international study centre for early music in the Low Countries.



## Chapter 8

# Conclusions and Future Research

This thesis is a first step in the exploration of the possibilities of low-rank modeling in room acoustics. It started with the physical justification for the use of a low-rank model of the RIR and a confirmation of its usefulness when applied to real-life RIRs. It continued with a demonstration of how this low-rank structure can be exploited when estimating RIRs from input-output data and then how the low-rank properties can be leveraged when convolving the RIR with an audio signal. Next, it was confirmed that the low-rank modeling of RIRs is good also with respect to objective quality measures. Finally, it was shown how the low-rank structure can be exploited for the joint compression of, and convolution with, multiple RIRs. In this chapter, conclusions of the research are presented, as well as suggestions for future research.

### 8.1 Conclusions

Up until the publication of the paper Chapter 2 is based upon, exploiting the low-rank structure of RIRs was a fairly unexplored idea in room acoustics. The sum-of-sinusoids model had been widely used before, and here the connection between the physics governing room acoustics, the sum-of-sinusoids model, and the possibility of using low-rank as a modeling tool, is presented. The connection between these three parts is built on a number of model approximations, but it is here confirmed that the use of low-rank is relevant for real-life RIRs. Low-rank is shown to be a better option than truncation and thresholding for all

considered cases, except for the very highest level of compression, with respect to the normalized misalignment between the recorded and approximated RIR. Further it is shown that low-rank modeling demonstrates very useful robustness qualities, in that the standard deviation is lower throughout for the low-rank approximation than for the comparable methods. Finally, one of the most interesting observations in this chapter, which came to influence the subsequent research significantly, is shown in Fig 2.2. Here, the benefit of higher-order tensorization, as opposed to just 2-D matricization, is made evident. By using a 3-D tensor approximation instead of a 2-D matrix approximation for the RIR, for similar compression rates, a significant reduction in normalized misalignment can be attained.

Once the real-life applicability of low-rank modeling in room acoustics has been established, it seems only natural to investigate if low rank can be used to improve the estimation of RIRs from input-output data. This way, the RIRs can be estimated directly on a low-rank form, instead of acquiring the entire RIR without the use of a physically motivated prior, only to throw away a large portion of the data after making a low-rank approximation. In Chapter 3 it is shown that the inclusion of low-rank regularization improves estimation performance in general, but particularly in scenarios with either low SNR or poor spectral excitation. Two different low-rank promoting penalty functions, nuclear norm and quadratic envelope, are compared to ordinary least squares and Tikhonov regularized least squares. It is found that the two low-rank promoting methods in most cases perform very similarly, see e.g., Fig. 3.4, but the quadratic envelope has the added benefit of being particularly robust to the case where the regularization parameter has either been over- or underestimated, and is therefore especially useful in cases where oracle knowledge of the SNR can not be assumed, see Fig. 3.6.

Granted that the reshaped RIR has a low-rank structure, it would be appealing to use that structure in the forward problem of convolving it with speech or music. Therefore, in this chapter, it is demonstrated how the RIRs on low-rank form can be used for fast, low-latency convolution. Exploiting low-rank structure in convolution has been considered before for the 2-D case, but given the significant improvement in going from 2-D matricization to higher order tensorization shown in Chapter 2, the extension to 3-D tensorization provided in this chapter, is a very relevant one. It is shown that the improvement in normalized misalignment in channel approximation observed in Chapter 2 between 2-D and 3-D persists for the normalized misalignment in an output signal sense, both when considering shorter RIRs convolved with speech signals, see Fig. 4.1, and when considering longer RIRs convolved with music, see Fig. 4.2.

The evaluation that has been considered up until this point in the thesis

has been purely mathematical, normalized misalignment between either the approximated/estimated channel and the recorded RIR, or between the output of convolution with the approximated/estimated RIR and with the recorded RIR. This is promising, but the true test for any attempted improvement in the field of acoustic signal enhancement or reproduction, is whether it yields an improvement in signal quality and in the experience of a listener. In Chapter 5, low-rank modeling is compared to truncation, thresholding, and the state-of-the-art audio compression Opus, with respect to a multitude of objective quality measures, channel-based as well as output-based. With very few exceptions among the many considered scenarios, low-rank modeling compares very well. In Chapter 5, RIRs from different rooms, and therefore with different reverberation times, are considered. It is found that the low-rank modeling, compared to the other considered methods, is more useful for shorter RIRs compared to longer ones, and more competitive for compression rates of  $< 0.9$ . Low-rank approximation preserves measures such as reverberation time, early decay time, and center time as well as, or better than, Opus for these scenarios, see Fig. 5.4, Fig. 5.7, and Fig. 5.9, respectively. It is shown to preserve the TOA of the direct component very well, see Fig. 5.8, something that is believed to be very promising for future research in the context of spatial RIRs. Finally, the excellent performance of low-rank models with respect to perceptually weighted measures such as frequency-weighted log-spectral signal distortion and ViSQOLAudio, see Fig. 5.10, Fig. 5.11, and Fig. 5.12, gives reasons to believe that low-rank models will perform very well also in subjective listening tests.

Numerous applications consider closely spaced RIRs in a room, in order to provide immersive experiences to a listener. As previously established, however, the storage of many and long RIRs is strenuous, and the processing using said RIRs, under strict latency constraints, is challenging. In Chapter 6 we demonstrate how multiple RIRs of a room can be compressed jointly. It is shown how one of the components of the GLRAM-decomposition varies slowly throughout the room, see Fig. 6.3 and Fig. 6.4, and how this can be exploited for reduced storage and fast multi-channel time-domain convolution. One of the proposed methods, LoCo-LoCo, is shown to be superior to SVD and state-of-the-art Opus for the compression of multiple RIRs corresponding to closely spaced receivers, see Fig. 6.2. The other proposed method, LoCo-LoCo-PIñaTa, is shown to be better when the receivers are farther apart, see Fig. 6.7, granting great flexibility. The exploitation of the similarity of one of the matrices of the GLRAM decomposition between adjacent RIRs yields a significant reduction of the computational complexity of the multi-channel convolution between the RIRs and an audio signal, see Fig. 6.8.

Taken together, the main contribution of the research presented in this thesis is providing in-depth introduction to the novel perspective of exploiting low

rank in the modeling of RIRs. Considering the wide applicability of RIRs in acoustic signal processing, the prospect of its usability is exciting. For this to come into full fruition, however, a lot of research is still necessary, which will now be further discussed.

## 8.2 Future Research

Two major lines run through this thesis. The first one is the 2-D matrix vs. the 3-D, 4-D, and 5-D tensors. The second one being the single-channel case vs. the multi-channel case. These lines have mainly been explored separately. The full potential of the ideas explored in this thesis is likely still ahead of us, by simultaneously considering multiple RIRs and higher order tensorization.

In Chapter 2, a distinct improvement in approximation accuracy, given a fixed compression rate, was seen when going from 2-D matricization to 3-D tensorization. This improvement was confirmed for output error in Chapter 4, and the several objective channel-based and signal-based measures considered Chapter 5. A similar jump in accuracy was, however, not present when going from a 3-D tensorization to a 4-D or 5-D tensor. Considering that it comes with a similar reduction in the number of coefficients needed to store a rank-1 component, as when going from 2-D matricization to 3-D tensorization, one would expect to see improvement there as well. Shedding light on this is an interesting possible path for future research.

For Chapters 3 and 6 of this thesis, we considered only 2-D matricization. However, in Chapters 2, 4, and 5, great improvement was seen when moving from 2-D matricization to 3-D tensorization of the RIR vectors. Therefore, it would be highly relevant to explore low-rank room impulse response estimation for higher-order tensors and joint compression of multiple tensorized RIRs, as extension of the ideas in Chapters 3 and 6, respectively.

In Chapter 3 we compared the performance of the low-rank estimator to that of the CRLB, the lower bound on the MSE of an unbiased estimator. However, for the bound to be correctly specified, this had to be done on low-rank approximations of the matricized RIRs, rather than the matricized RIRs themselves. The balance between the bias introduced by a misspecified model, and the reduction in variance, therefore remains an open question. Given that the application for this reaches far beyond just room acoustics, it looks like a very interesting topic for future research.

For decompositions and approximations of higher-order tensors, the practical decision was made to always use the CPD, as the particular choice of tensor

decomposition was beyond the scope of this thesis. It would therefore be very interesting to see a methodical review of different tensor decomposition methods and their suitability for low-rank modeling in room acoustics.

Throughout this thesis, with a brief exception in Chapter 6, only square matrices and tensors were considered, i.e., when  $n_{s_1} = n_{s_2} = \dots = n_{s_D}$ . It is justified by the fact that this way, the number of coefficients needed to store a rank-1 component is the smallest. That does, however, not mean that this is an optimal choice in every conceivable scenario, as the findings in Chapter 6 indicate. A systematic exploration of the impact of the sizes of the tensor dimensions, and possible physically-based motivations therefore, appears fruitful.

In this thesis we have considered RIRs and signals sampled at fairly high sampling rates, 44.1 kHz and 48 kHz. This forces a decision between an RIR that is long in terms of time and containing many coefficients or short in terms of time and containing few coefficients. In applications like sound zone control, it is not uncommon to consider a significantly lower sampling rate, often on the order of  $10^3$  Hz, in order to have filters that are longer with respect to time, without them also demanding a large number of coefficients. A study of the performance of the low-rank methods at lower sampling rates would therefore be interesting.

Spatial audio has not been considered in this thesis. The results of Chapter 5, however, suggest that the low-rank structure to RIRs considered in this thesis could be very useful when considering spatial audio. The times of arrival of the direct component and early reflections are very important for source localization. The results in Chapter 5 show that the low-rank approximations considered preserve the time of arrival of the direct component very well. Verifying that this indeed translates into improved modeling also of spatial RIRs is a very interesting area for future research.



# Acknowledgements

This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of KU Leuven internal funds VES/19/004 and C14/21/075 and FWO projects G0A2721N and S005319N. The research leading to these results has received funding from the Research Foundation – Flanders (FWO, grant 12ZD622N) and from the European Research Council under the European Union’s Horizon 2020 research and innovation program / ERC Consolidator Grant: SONORA (no. 773268). This thesis reflects only the author’s views and the Union is not liable for any use that may be made of the contained information.



# Bibliography

- [1] 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)*, 2014.
- [2] V. Välimäki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, “Fifty years of artificial reverberation,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 20, no. 5, pp. 1421–1448, 2012.
- [3] D. D. Rife and J. Vanderkooy, “Transfer-function measurement using maximum-length sequences,” in *Proc. 83rd Audio Eng. Soc. Conv.*, 1987.
- [4] D. D. Rife and J. Vanderkooy, “Transfer-function measurement with maximum-length sequences,” *J. Audio Eng. Soc.*, vol. 37, no. 6, pp. 419–444, 1989.
- [5] A. Farina, “Simultaneous measurement of impulse response and distortion with a swept-sine technique,” in *Proc. 108th Audio Eng. Soc. Conv.*, 2000.
- [6] L. Savioja and U. P. Svensson, “Overview of geometrical room acoustic modeling techniques,” *J. Acoust. Soc. Amer.*, vol. 138, no. 2, pp. 708–730, 2015.
- [7] 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.
- [8] J. Borish, “Extension of the image model to arbitrary polyhedra,” *J. Acoust. Soc. Amer.*, vol. 75, no. 6, pp. 1827–1836, 1984.
- [9] 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.

- [10] 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.
- [11] D. Schröder, *Physically based real-time auralization of interactive virtual environments*. PhD thesis, RWTH Aachen, 2011.
- [12] T. Funkhouser, N. Tsingos, I. Carlbom, G. Elko, M. Sondhi, J. E. West, G. Pingali, P. Min, and A. Ngan, "A beam tracing method for interactive architectural acoustics," *J. Acoust. Soc. Amer.*, vol. 115, no. 2, pp. 739–756, 2004.
- [13] T. Okuzono, T. Yoshida, and K. Sakagami, "Efficiency of room acoustic simulations with time-domain FEM including frequency-dependent absorbing boundary conditions: Comparison with frequency-domain FEM," *Appl. Acoust.*, vol. 182, 2021.
- [14] 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.
- [15] 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.
- [16] S. Kirkup, "The boundary element method in acoustics: A survey," *Appl. Sci.*, vol. 9, no. 8, 2019.
- [17] M. Vorländer, *Auralization : fundamentals of acoustics, modelling, simulation, algorithms and acoustic virtual reality*. Berlin: Springer, 2020.
- [18] M. Aretz and M. Vorländer, "Combined wave and ray based room acoustic simulations of audio systems in car passenger compartments, part I: Boundary and source data," *Appl. Acoust.*, vol. 76, pp. 82–99, 2014.
- [19] M. Aretz and M. Vorländer, "Combined wave and ray based room acoustic simulations of audio systems in car passenger compartments, part II: Comparison of simulations and measurements," *Appl. Acoust.*, vol. 76, pp. 52–65, 2014.
- [20] "How humans conquered echo." <https://www.theatlantic.com/entertainment/archive/2012/06/how-humans-conquered-echo/258557/>, 2012.

- [21] M. Schroeder and B. Logan, "Colorless" artificial reverberation," *IRE Trans. Audio.*, vol. AU-9, no. 6, pp. 209–214, 1961.
- [22] M. R. Schroeder, "Natural sounding artificial reverberation," in *Proc. 13th Audio Eng. Soc. Conv.*, 1961.
- [23] B. De Man, K. McNally, and J. D. Reiss, "Perceptual evaluation and analysis of reverberation in multitrack music production," *J. Audio Eng. Soc.*, vol. 65, no. 1/2, pp. 108–116, 2017.
- [24] N. Tsingos, "Pre-computing geometry-based reverberation effects for games," in *Proc. 35th Int. Conf. Audio Games*, 2009.
- [25] K. B. Christensen and T. Lund, "Room simulation for multichannel film and music," in *Proc. 107th Audio Eng. Soc. Conv.*, 1999.
- [26] J.-M. Jot and A. Chaigne, "Digital delay networks for designing artificial reverberators," in *Proc. 90th Audio Eng. Soc. Conv.*, 1991.
- [27] E. De Sena, H. Hacıhabiboglu, Z. Cvetkovic, 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.
- [28] M. Kleiner, B.-I. Dalenbäck, and P. Svensson, "Auralization - an overview," *J. Audio Eng. Soc.*, vol. 41, no. 11, pp. 861–875, 1993.
- [29] I. Wohlgenannt, A. Simons, and S. Stieglitz, "Virtual reality," *Bus. & Inf. Syst. Eng.*, vol. 62, no. 5, pp. 455–461, 2020.
- [30] H. S. Llopis, F. Pind, and C.-H. Jeong, "Development of an auditory virtual reality system based on pre-computed b-format impulse responses for building design evaluation," *Build. Environ.*, vol. 169, 2020.
- [31] B. Xie, H. Liu, R. Alghofaili, Y. Zhang, Y. Jiang, F. D. Lobo, C. Li, W. Li, H. Huang, M. Akdere, C. Mousas, and L. Yu, "A review on virtual reality skill training applications," *Front. Virtual Reality*, vol. 2, 2021.
- [32] P. M. Emmelkamp and K. Meyerbröker, "Virtual reality therapy in mental health," *Annu. Rev. Clin. Psychol.*, vol. 17, no. 1, pp. 495–519, 2021.
- [33] T. Potter, Z. Cvetković, and E. De Sena, "On the relative importance of visual and spatial audio rendering on VR immersion," *Front. Signal Process.*, vol. 2, 2022.
- [34] S. Cecchi, A. Carini, and S. Spors, "Room response equalization – a review," *Appl. Sci.*, vol. 8, no. 1, 2018.

- [35] V. Välimäki and J. D. Reiss, “All about audio equalization: Solutions and frontiers,” *Appl. Sci.*, vol. 6, no. 5, 2016.
- [36] J. Mourjopoulos, “Digital equalisation of room acoustics,” *J. Audio Eng. Soc.*, vol. 42, 1994.
- [37] M. Karjalainen, T. Paatero, J. Mourjopoulos, and P. Hatziantoniou, “About room response equalization and dereverberation,” in *Proc. 2005 IEEE Workshop Appl. of Signal Process. Audio Acoust. (WASPAA)*, pp. 183–186, 2005.
- [38] P. A. Naylor and N. D. Gaubitch, *Speech Dereverberation*. Springer, 2010.
- [39] 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.
- [40] J.-H. Chang and F. Jacobsen, “Experimental validation of sound field control with a circular double-layer array of loudspeakers,” *J. Acoust. Soc. Amer.*, vol. 133, no. 4, pp. 2046–2054, 2013.
- [41] M. F. Simón Gálvez, S. J. Elliott, and J. Cheer, “Time domain optimization of filters used in a loudspeaker array for personal audio,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 11, pp. 1869–1878, 2015.
- [42] “Vectors | chapter 1, Essence of linear algebra.” [https://www.youtube.com/watch?v=fNk\\_zzaMoSs](https://www.youtube.com/watch?v=fNk_zzaMoSs), 2016.
- [43] M. Udell and A. Townsend, “Why are big data matrices approximately low rank?,” *SIAM J. Math. Data Sci.*, vol. 1, no. 1, pp. 144–160, 2019.
- [44] M. Boussé, O. Debals, and L. De Lathauwer, “A tensor-based method for large-scale blind source separation using segmentation,” *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 346–358, 2017.
- [45] M. Boussé, O. Debals, and L. De Lathauwer, “Tensor-based large-scale blind system identification using segmentation,” *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5770–5784, 2017.
- [46] C. Paleologu, J. Benesty, and S. Ciochină, “Linear system identification based on a Kronecker product decomposition,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 10, pp. 1793–1808, 2018.
- [47] Z. Hu, F. Nie, R. Wang, and X. Li, “Low rank regularization: A review,” *Neural Netw.*, vol. 136, pp. 218–232, 2021.

- [48] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?,” *J. ACM*, vol. 58, no. 3, 2011.
- [49] Q. Yao, J. Kwok, T. Wang, and T. Liu, “Large-scale low-rank matrix learning with nonconvex regularizers,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2628–2643, 2019.
- [50] L. D. Lathauwer, “Blind separation of exponential polynomials and the decomposition of a tensor in rank- $(L_r, L_r, 1)$  terms,” *SIAM J. Matrix Anal. Appl.*, vol. 32, no. 4, pp. 1451–1474, 2011.
- [51] T. Ajdler, L. Sbaiz, and M. Vetterli, “The plenacoustic function and its sampling,” *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3790–3804, 2006.
- [52] S. Koyama, T. Nishida, K. Kimura, T. Abe, N. Ueno, and J. Brunnström, “MESHRIR: A dataset of room impulse responses on meshed grid points for evaluating sound field analysis and synthesis methods,” in *Proc. 2021 IEEE Workshop Appl. of Signal Process. Audio Acoust. (WASPAA)*, pp. 1–5, 2021.
- [53] J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex fourier series,” *Math. Comput.*, vol. 19, pp. 297–301, 1965.
- [54] T. G. Stockham, “High-speed convolution and correlation,” in *Proc. Spring Joint Comput. Conf.*, pp. 229–233, 1966.
- [55] Rafaely, B., Tourbabin, V., Habets, E., Ben-Hur, Z., Lee, H., Gamper, H., Arbel, L., Birnie, L., Abhayapala, T., and Samarasinghe, P., “Spatial audio signal processing for binaural reproduction of recorded acoustic scenes - review and challenges,” *Acta Acust.*, vol. 6, p. 47, 2022.
- [56] R. Gupta, J. He, R. Ranjan, W.-S. Gan, F. Klein, C. Schneiderwind, A. Neidhardt, K. Brandenburg, and V. Välimäki, “Augmented/mixed reality audio for hearables: Sensing, control, and rendering,” *IEEE Signal Process. Mag.*, vol. 39, no. 3, pp. 63–89, 2022.
- [57] M. Vorländer, *Auralization: fundamentals of acoustics, modelling, simulation, algorithms and acoustic virtual reality*. Springer Nature, 2020.
- [58] C. Schissler, P. Stirling, and R. Mehra, “Efficient construction of the spatial room impulse response,” in *Proc. 2017 IEEE Virtual Reality (VR) Conf.*, pp. 122–130, 2017.
- [59] M. B. Møller and J. Østergaard, “A moving horizon framework for sound zones,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 256–265, 2020.

- [60] J. Brunnström, M. Jälmy, T. van Waterschoot, and M. Moonen, “Fast low-rank filtered-x least mean squares for multichannel active noise control.” accepted for publication in Proc. 2023 Asilomar Conf. Signals Syst. Comput., 2023. [https://ftp.esat.kuleuven.be/pub/stadius/jbrunnst/2023\\_asilomar\\_low\\_rank\\_fxlms\\_006.pdf](https://ftp.esat.kuleuven.be/pub/stadius/jbrunnst/2023_asilomar_low_rank_fxlms_006.pdf).
- [61] A. Carôt and C. Werner, “Network music performance-problems, approaches and perspectives,” in *Proc. of the "Music in the Global Village"-Conf.*, vol. 162, pp. 10–23, 2007.
- [62] B. D. Kulp, “Digital equalization using Fourier transform techniques,” *J. Audio Eng. Soc.*, 1988.
- [63] W. G. Gardner, “Efficient convolution without input/output delay,” *J. Audio Eng. Soc.*, vol. 43, pp. 127–136, 1995.
- [64] F. Wefers, *Partitioned Convolution Algorithms for Real-time Auralization*. PhD thesis, RWTH Aachen, 2015.
- [65] G. García, “Optimal filter partition for efficient convolution with short input/output delay,” *J. Audio Eng. Soc.*, 2002.
- [66] M. Vorländer, D. Schröder, S. Pelzer, and F. Wefers, “Virtual reality for architectural acoustics,” *J. Build. Perform. Simul.*, vol. 8, no. 1, pp. 15–25, 2015.
- [67] J. Y. C. Wen, N. D. Gaubitch, E. A. P. Habets, T. Myatt, and P. A. Naylor, “Evaluation of speech dereverberation algorithms using the MARDY database,” in *Proc. 2006 Int. Workshop Acoustic Signal Enhancement (IWAENC)*, 2006.
- [68] E. Hadad, F. Heese, P. Vary, and S. Gannot, “Multichannel audio database in various acoustic environments,” in *Proc. 2014 Int. Workshop Acoustic Signal Enhancement (IWAENC)*, 2014.
- [69] G. W. Elko, E. Diethorn, and T. Gaensler, “Room impulse response variation due to thermal fluctuation and its impact on acoustic echo cancellation,” in *Proc. 2003 Int. Workshop Acoustic Signal Enhancement (IWAENC)*, 2003.
- [70] J. S. Abel, S. Coffin, and K. Spratt, “A modal architecture for artificial reverberation with application to room acoustics modeling,” in *Proc. 137th Audio Eng. Soc. Conv.*, 2014.
- [71] G. Vairetti, E. De Sena, M. Catrysse, S. H. Jensen, M. Moonen, and T. van Waterschoot, “A scalable algorithm for physically motivated and

- sparse approximation of room impulse responses with orthonormal basis functions,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 7, pp. 1547–1561, 2017.
- [72] C. Huszty, N. Bukuli, Á. Torma, and F. Augusztinovicz, “Effects of filtering of room impulse responses on room acoustics parameters by using different filter structures,” *J. Acoust. Soc. Amer.*, vol. 123, p. 3617, 2008.
- [73] G. Vairetti, *Efficient parametric modeling, identification and equalization of room acoustics*. PhD thesis, KU Leuven, 2018.
- [74] H. Kuttruff, *Room Acoustics*. London: Spon Press, 2009.
- [75] 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.
- [76] T. P. Vu and H. Lissek, “Sound field reconstruction in a room using low-rank approximation of mode shapes: validation and application,” in *Proc. 26th Int. Congr. Sound Vib. (ICSV)*, 2019.
- [77] H. P. Tukuljac, T. P. Vu, H. Lissek, and P. Vandergheynst, “Joint estimation of the room geometry and modes with compressed sensing,” in *Proc. 2018 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 6882–6886, 2018.
- [78] F. Jacobsen and P. M. Juhl, *Fundamentals of General Linear Acoustics*. Wiley, 2013.
- [79] 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.
- [80] S. Bilbao, *Numerical Sound Synthesis*. Wiley, 2009.
- [81] L. S. H. Ngia, “Recursive identification of acoustic echo systems using orthonormal basis functions,” *IEEE Trans. Speech Audio Process.*, vol. 11, no. 3, pp. 278–293, 2003.
- [82] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
- [83] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, “Tensor decompositions for signal processing applications: From two-way to multiway component analysis,” *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.

- [84] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, “Tensorlab 3.0.” <https://www.tensorlab.net>, 2016.
- [85] C. Evers, H. W. Löllmann, H. Mellmann, A. Schmidt, H. Barfuss, P. A. Naylor, and W. Kellermann, “The LOCATA challenge: Acoustic source localization and tracking,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 1620–1643, 2020.
- [86] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, “A consolidated perspective on multimicrophone speech enhancement and source separation,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 4, pp. 692–730, 2017.
- [87] J. Mourjopoulos and M. Paraskevas, “Pole and zero modeling of room transfer functions,” *J. Sound Vib.*, vol. 146, no. 2, pp. 281–302, 1991.
- [88] K. Shi, X. Ma, and G. Tong Zhou, “An efficient acoustic echo cancellation design for systems with long room impulses and nonlinear loudspeakers,” *Signal Processing*, vol. 89, no. 2, pp. 121–132, 2009.
- [89] L. Krishnan, P. D. Teal, and T. Betlehem, “A robust sparse approach to acoustic impulse response shaping,” in *Proc. 2015 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 738–742, 2015.
- [90] L. Ljung, T. Chen, and B. Mu, “A shift in paradigm for system identification,” *Int. J. Control*, vol. 93, no. 2, pp. 173–180, 2020.
- [91] T. van Waterschoot, G. Rombouts, and M. Moonen, “Optimally regularized adaptive filtering algorithms for room acoustic signal enhancement,” *Signal Processing*, vol. 88, no. 3, pp. 594–611, 2008.
- [92] M. Boussé and L. De Lathauwer, “Large-scale autoregressive system identification using Kronecker product equations,” in *Proc. 2018 IEEE Glob. Conf. Signal Inf. Process. (GlobalSIP)*, pp. 1348–1352, 2018.
- [93] A. Marconato, L. Ljung, Y. Rolain, and J. Schoukens, “Linking regularization and low-rank approximation for impulse response modeling,” *Proc. 19th IFAC World Congr.*, vol. 47, no. 3, pp. 4999–5004, 2014.
- [94] A. Marconato and J. Schoukens, “Impact of regularization in FIR estimation for short and long data records,” in *Proc. 2015 IEEE Int. Instrum. Meas. Technol. Conf.*, pp. 789–793, 2015.
- [95] M. Jälmy, F. Elvander, and T. van Waterschoot, “Low-rank tensor modeling of room impulse responses,” in *Proc. 29th European Signal Process. Conf. (EUSIPCO)*, pp. 111–115, 2021.

- [96] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*. Springer Series in Computational Mathematics, Springer Berlin Heidelberg, 2012.
- [97] C. Elisei-Iliescu, C. Paleologu, J. Benesty, C. Stanciu, C. Anghel, and S. Ciochină, “Recursive least-squares algorithms for the identification of low-rank systems,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 5, pp. 903–918, 2019.
- [98] L.-M. Dogariu, C. Paleologu, J. Benesty, and S. Ciochină, “An efficient Kalman filter for the identification of low-rank systems,” *Signal Processing*, vol. 166, p. 107239, 2020.
- [99] Y. Xiang, S. and Zhu, X. Shen, and J. Ye, “Optimal exact least squares rank minimization,” in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 480–488, 2012.
- [100] M. Carlsson, “On convex envelopes and regularization of non-convex functionals without moving global minima,” *J. Optim. Theory Appl.*, vol. 183, no. 1, pp. 66–84, 2019.
- [101] V. Larsson and C. Olsson, “Convex low rank approximation,” *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 194–214, 2016.
- [102] C. Olsson, M. Carlsson, F. Andersson, and V. Larsson, “Non-convex rank/sparsity regularization and local minima,” in *Proc. 2017 IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 332–340, 2017.
- [103] J. Caprani and M. Carlsson, “Quadratic envelope regularization for structured low rank approximation,” in *Proc. 2019 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 8217–8221, 2019.
- [104] P. C. Hansen, “Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank,” *SIAM J. Sci. Stat. Comput.*, vol. 11, no. 3, pp. 503–518, 1990.
- [105] N. Antonello, L. Stella, P. Patrinos, and T. van Waterschoot, “Proximal gradient algorithms: Applications in signal processing.” arXiv: 1803.01621, 2020.
- [106] M. Fazel, H. Hindi, and S. Boyd, “A rank minimization heuristic with application to minimum order system approximation,” in *Proc. 2001 Amer. Control Conf.*, vol. 6, pp. 4734–4739, 2001.
- [107] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

- [108] L. Gu, S. and Zhang and X. Zuo, W. and Feng, “Weighted nuclear norm minimization with application to image denoising,” in *Proc. 2014 IEEE Conf. Comput. Vis. Pattern Recogn.*, pp. 2862–2869, 2014.
- [109] C. Lu, C. Zhu, C. Xu, S. Yan, and Z. Lin, “Generalized singular value thresholding,” in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015.
- [110] E. Soubies, A. Chinatto, P. Larzabal, J. Romano, and L. Blanc-Féraud, “Direction-of-arrival estimation through exact continuous  $\ell_{2,0}$ -norm relaxation,” *IEEE Signal Process. Lett.*, vol. 28, pp. 16–20, 2021.
- [111] E. Vasquez-Ortiz and P. Rodriguez, “Fast gradient-based algorithm for a quadratic envelope relaxation of the  $\ell_0$  gradient regularization,” in *Proc. 2021 XXIII Symp. Image, Signal Process. Artif. Vis. (STSIVA)*, pp. 1–6, 2021.
- [112] C.-H. Zhang, “Nearly unbiased variable selection under minimax concave penalty,” *Ann. Stat.*, vol. 38, no. 2, pp. 894 – 942, 2010.
- [113] E. Soubies, L. Blanc-Féraud, and G. Aubert, “A continuous exact  $\ell_0$  penalty (CELO) for least squares regularized problem,” *SIAM J. Imaging Sci.*, vol. 8, no. 3, pp. 1607–1639, 2015.
- [114] M. Jälmbly, F. Elvander, and T. van Waterschoot, “Low-rank room impulse response estimation,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 957–969, 2023.
- [115] P. L. Combettes and J.-C. Pesquet, *Proximal Splitting Methods in Signal Processing*, pp. 185–212. New York: Springer, 2011.
- [116] N. Parikh and S. Boyd, “Proximal algorithms,” *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.
- [117] H. Attouch, J. Bolte, and B. Svaiter, “Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods,” *Math. Program.*, vol. 137, no. 1, pp. 91–129, 2013.
- [118] 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.
- [119] R. Gray, “Toeplitz and circulant matrices: A review,” *Found. Trends Commun. Inf. Theory*, vol. 2, no. 3, pp. 155–239, 2006.
- [120] R. Badeau, “Unified stochastic reverberation modeling,” in *Proc. 26th European Signal Process. Conf. (EUSIPCO)*, pp. 2175–2179, 2018.

- [121] M. R. Schroeder, “Frequency-correlation functions of frequency responses in rooms,” *J. Acoust. Soc. Amer.*, vol. 34, no. 12, pp. 1819–1823, 1962.
- [122] J. Moorer, “About this reverberation business,” *Computer Music Journal*, vol. 3, pp. 605–639, 1985.
- [123] P. Kabal, “TSP speech database,” tech. rep., McGill University, 2002.
- [124] G. Tang and A. Nehorai, “Lower bounds on the mean-squared error of low-rank matrix reconstruction,” *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4559–4571, 2011.
- [125] J. Cai, E. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
- [126] J. Atkins, A. Strauss, and C. Zhang, “Approximate convolution using partitioned truncated singular value decomposition filtering,” in *Proc. 2013 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 176–180, 2013.
- [127] W.-C. Lee, C.-M. Liu, C.-H. Yang, and J.-I. Guo, “Fast perceptual convolution for room reverberation,” in *Proc. 6th Int. Conf. Digital Audio Effects (DAFx)*, 2003.
- [128] N. Jillings, J. D. Reiss, and R. Stables, “Zero-delay large signal convolution using multiple processor architectures,” in *Proc. 2017 IEEE Workshop Appl. of Signal Process. Audio Acoust. (WASPAA)*, pp. 339–343, 2017.
- [129] X. Zhou, C. Yang, H. Zhao, and W. Yu, “Low-rank modeling and its applications in image analysis,” *ACM Comput. Surv.*, vol. 47, dec 2014.
- [130] G. Huang, J. Benesty, J. Chen, C. Paleologu, S. Ciochină, W. Kellermann, and I. Cohen, “Acoustic system identification with partially time-varying models based on tensor decompositions,” in *Proc. 2022 Int. Workshop Acoustic Signal Enhancement (IWAENC)*, 2022.
- [131] M. Jeub, M. Schafer, and P. Vary, “A binaural room impulse response database for the evaluation of dereverberation algorithms,” in *Proc. 16th Int. Conf. Digit. Signal Process.*, pp. 1–5, 2009.
- [132] G. Waters, “Sound quality assessment material—recordings for subjective tests: User’s handbook for the EBU–SQAM compact disk,” *European Broadcasting Union (EBU), Tech. Rep.*, pp. 1–13, 1988.
- [133] F. Brinkmann, L. Aspöck, D. Ackermann, S. Lepa, M. Vorländer, and S. Weinzierl, “A round robin on room acoustical simulation and auralization,” *J. Acoust. Soc. Amer.*, vol. 145, no. 4, pp. 2746–2760, 2019.

- [134] S. Goetze, E. Albertin, M. Kallinger, A. Mertins, and K.-D. Kammeyer, "Quality assessment for listening-room compensation algorithms," in *Proc. 2010 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 2450–2453, 2010.
- [135] T. Rossing, *Springer Handbook of Acoustics*. Springer, 2014.
- [136] H. Hacihabiboglu, E. De Sena, Z. Cvetkovic, J. Johnston, and J. O. Smith III, "Perceptual spatial audio recording, simulation, and rendering: An overview of spatial-audio techniques based on psychoacoustics," *IEEE Signal Process. Mag.*, vol. 34, no. 3, pp. 36–54, 2017.
- [137] B. F. G. Katz, D. Murphy, and A. Farina, "The past has ears (PHE): XR explorations of acoustic spaces as cultural heritage," in *Augmented Reality, Virtual Reality, and Computer Graphics*, pp. 91–98, Springer International Publishing, 2020.
- [138] A. Primavera, S. Cecchi, L. Romoli, P. Peretti, and F. Piazza, "A low latency implementation of a non-uniform partitioned convolution algorithm for room acoustic simulation," *Signal, Image Video Process.*, vol. 8, no. 5, pp. 985–994, 2014.
- [139] B. Holm-Rasmussen, H. Lehtonen, and V. Välimäki, "A new reverberator based on variable sparsity convolution," in *Proc. 16th Int. Conf. Digital Audio Effects (DAFx)*, 2013.
- [140] M. Jälmy, F. Elvander, and T. van Waterschoot, "Multi-channel low-rank convolution of jointly compressed room impulse responses." KU Leuven ESAT-STADIUS Technical Report 23-150, 2023. <https://ftp.esat.kuleuven.be/pub/stadius/mjalmy/23-150.pdf>.
- [141] M. Jälmy, F. Elvander, and T. van Waterschoot, "Fast low-latency convolution by low-rank tensor approximation," in *Proc. 2023 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 1–5, 2023.
- [142] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions." arXiv: 1405.3866, 2014.
- [143] L. Sorber, M. Van Barel, and L. De Lathauwer, "Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank- $(L_r, L_r, 1)$  terms, and a new generalization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 695–720, 2013.
- [144] J.-M. Valin, K. Vos, and T. Terriberry, *Definition of the Opus audio codec*, IETF, September 2012.

- [145] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, “High-quality, low-delay music coding in the Opus codec,” *J. Audio Eng. Soc.*, 2013.
- [146] K. Vos, S. Jensen, and K. Soerensen, “Silk speech codec. draft-vos-silk-02,” 2010.
- [147] J.-M. Valin, T. B. Terriberry, and G. Maxwell, “A full-bandwidth audio codec with low complexity and very low delay,” in *Proc. 17th European Signal Process. Conf. (EUSIPCO)*, pp. 1254–1258, 2009.
- [148] J.-M. Valin, T. B. Terriberry, C. Montgomery, and G. Maxwell, “A high-quality speech and audio codec with less than 10-ms delay,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 18, no. 1, pp. 58–67, 2010.
- [149] H. Ren, C. Ritz, J. Zhao, and D. Jang, “Impact of compression on the performance of the room impulse response interpolation approach to spatial audio synthesis,” in *Proc. 2022 Asia-Pacific Signal and Information Process. Association Annual Summit and Conf. (APSIPA ASC)*, pp. 442–448, 2022.
- [150] A. Hines, J. Skoglund, A. Kokaram, and N. Harte, “ViSQOL: The virtual speech quality objective listener,” in *Proc. 2012 Int. Workshop Acoustic Signal Enhancement (IWAENC)*, 2012.
- [151] A. Hines, J. Skoglund, A. C. Kokaram, and N. Harte, “ViSQOL: an objective speech quality model,” *EURASIP J. Audio, Speech, Music Process.*, vol. 2015, no. 1, 2015.
- [152] A. Hines, E. Gillen, D. Kelly, J. Skoglund, A. Kokaram, and N. Harte, “ViSQOLaudio: An objective audio quality metric for low bitrate codecs,” *J. Acoust. Soc. Amer.*, vol. 137, no. 6, 2015.
- [153] M. R. Schroeder, “New method of measuring reverberation time,” *J. Acoust. Soc. Amer.*, vol. 37, no. 3, pp. 409–412, 1965.
- [154] F. Rumsey, *Spatial audio*. Routledge, 2012.
- [155] J. Abel and P. Huang, “A simple, robust measure of reverberation echo density,” in *Proc. 121th Audio Eng. Soc. Conv.*, 2006.
- [156] M. Karjalainen and H. Järveläinen, “More about this reverberation science: Perceptually good late reverberation,” in *Proc. 111th Audio Eng. Soc. Conv.*, 2001.
- [157] K. MacWilliam, F. Elvander, and T. van Waterschoot, “Simultaneous acoustic echo sorting and 3-d room geometry inference,” in *Proc. 2023 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 1–5, 2023.

- [158] H. Rosseel and T. van Waterschoot, “Improved acoustic source localization by time delay estimation with subsample accuracy,” in *Proc. 2021 Immersive 3D Audio (I3DA)*, pp. 1–8, 2021.
- [159] M. Cartwright, B. Pardo, G. J. Mysore, and M. Hoffman, “Fast and easy crowdsourced perceptual audio evaluation,” in *Proc. 2016 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 619–623, 2016.
- [160] A. Spriet, K. Eneman, M. Moonen, and J. Wouters, “Objective measures for real-time evaluation of adaptive feedback cancellation algorithms in hearing aids,” in *Proc. 16th European Signal Process. Conf. (EUSIPCO)*, pp. 1–5, 2008.
- [161] A. Hines and N. Harte, “Speech intelligibility prediction using a neurogram similarity index measure,” *Speech Commun.*, vol. 54, no. 2, pp. 306–320, 2012.
- [162] Rec.ITU-R.BS.1534-1:, *Method for the Subjective Assessment of Intermediate Sound Quality (MUSHRA)*. International Telecommunication Union, Geneva, 2003.
- [163] M. Narbutt, A. Allen, J. Skoglund, M. Chinen, and A. Hines, “AMBIQUAL - a full reference objective quality metric for ambisonic spatial audio,” in *Proc. 2018 10th Int. Conf. Qual. Multimedia Exp.*, pp. 1–6, 2018.
- [164] M. Narbutt, J. Skoglund, A. Allen, M. Chinen, D. Barry, and A. Hines, “AMBIQUAL: Towards a quality metric for headphone rendered compressed ambisonic spatial audio,” *Appl. Sci.*, vol. 10, no. 9, 2020.
- [165] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, “Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs,” in *Proc. 2001 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 749–752, 2001.
- [166] J. Beerends, C. Schmidmer, J. Berger, M. Obermann, R. Ullmann, J. Pomy, and M. Keyhl, “Perceptual objective listening quality assessment (POLQA), the third generation ITU-T standard for end-to-end speech quality measurement part I — temporal alignment,” *J. Audio Eng. Soc.*, vol. 61, no. 6, pp. 366–384, 2013.
- [167] T. Dietzen, R. Ali, M. Taseska, and T. van Waterschoot, “MYRiAD: a multi-array room acoustic database,” *EURASIP J. Audio, Speech, Music Process.*, vol. 2023, no. 1, 2023.

- [168] J. P. Paulo, C. R. Martins, and J. Bento Coelho, “A hybrid MLS technique for room impulse response estimation,” *Appl. Acoust.*, vol. 70, no. 4, pp. 556–562, 2009.
- [169] D. G. Ćirić and M. Janković, “Correction of room impulse response truncation based on a nonlinear decay model,” *Appl. Acoust.*, vol. 132, pp. 210–222, 2018.
- [170] M. Crocco and A. Del Bue, “Room impulse response estimation by iterative weighted  $L_1$ -norm,” in *Proc. 23rd European Signal Process. Conf. (EUSIPCO)*, pp. 1895–1899, 2015.
- [171] M. Chen and C.-M. Lee, “The optimal determination of the truncation time of non-exponential sound decays,” *Buildings*, vol. 12, no. 5, 2022.
- [172] N. D. Gaubitch, H. W. Löllmann, M. Jeub, T. H. Falk, P. A. Naylor, P. Vary, and M. Brookes, “Performance comparison of algorithms for blind reverberation time estimation from speech,” in *Proc. 2012 Int. Workshop Acoustic Signal Enhancement (IWAENC)*, 2012.
- [173] J. Zhao, X. Zheng, C. Ritz, and D. Jang, “Interpolating the directional room impulse response for dynamic spatial audio reproduction,” *Appl. Sci.*, vol. 12, no. 4, 2022.
- [174] M. Jälmy, F. Elvander, and T. van Waterschoot, “Compression of room impulse responses for compact storage and fast low-latency convolution.” KU Leuven ESAT-STADIUS Technical Report 23-149, 2023. <https://ftp.esat.kuleuven.be/pub/stadius/mjalmy/23-149.pdf>.
- [175] J. Ye, “Generalized low rank approximations of matrices,” in *Proc. Twenty-First Int. Conf. Machine Learning*, pp. 112–119, 2004.
- [176] V. Y. Pan and Z. Q. Chen, “The complexity of the matrix eigenproblem,” in *Proc. of the Thirty-First Annual ACM Symposium on Theory of Computing*, pp. 507–516, 1999.
- [177] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.
- [178] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory — Implementation — Applications*. Springer, 2012.
- [179] “Room impulse response simulation with the image-source method and HRTF interpolation.” <https://nl.mathworks.com/help/audio/ug/room-impulse-response-simulation-with-image-source-method-and-hrtf-interpolation.html>.

- [180] “Will VR go from niche to mainstream? It all depends on compelling VR content.” <https://www2.deloitte.com/us/en/insights/industry/technology/technology-media-and-telecom-predictions/2023/vr-content-development-lagging-vr-hardware-market.html>, 2022.
- [181] “Virtual reality market.” <https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-market-101378>, 2022.
- [182] “Creating compelling reverberations for virtual reality.” <https://blog.audiokinetic.com/en/creating-compelling-reverberations-for-virtual-reality/>, 2017.
- [183] M. Slater and S. Wilbur, “A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments,” *Presence: Teleoperators Virtual Environ.*, vol. 6, pp. 603–616, 1997.
- [184] E. Chang, H. T. Kim, and B. Yoo, “Virtual reality sickness: A review of causes and measurements,” *Int. J. Hum.-Comput. Interact.*, vol. 36, pp. 1658 – 1682, 2020.
- [185] A. C. Kern and W. Ellermeier, “Audio in VR: Effects of a soundscape and movement-triggered step sounds on presence,” *Front. Robot AI*, vol. 7, p. 20, 2020.
- [186] S. Weech, S. Kenny, and M. Barnett-Cowan, “Presence and cybersickness in virtual reality are negatively related: A review,” *Front. Psychol.*, vol. 10, p. 158, 2019.
- [187] H. Kim, L. Remaggi, P. J. Jackson, and A. Hilton, “Immersive spatial audio reproduction for VR/AR using room acoustic modelling from 360 ° images,” in *Proc. 2019 IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, pp. 120–126, 2019.
- [188] “Personal audio space: The headphones experience sans headphones.” <https://www.microsoft.com/en-us/research/blog/personal-audio-space-headphones-experience-sans-headphones/>, 2007.
- [189] R. M. Jacobsen, K. Fangel Skov, S. S. Johansen, M. B. Skov, and J. Kjeldskov, “Living with sound zones: A long-term field study of dynamic sound zones in a domestic context,” in *Proc. 2023 Conf. Hum. Factors Comput. Syst.*, 2023.
- [190] M. Jones and S. J. Elliott, “Personal audio with multiple dark zones,” *J. Acoust. Soc. Amer.*, vol. 124, no. 6, pp. 3497–3506, 2008.
- [191] J. Cheer and S. Elliott, “Design and implementation of a personal audio system in a car cabin,” *J. Acoust. Soc. Amer.*, vol. 133, 2013.

- [192] M. B. Møller, J. K. Nielsen, E. Fernandez-Grande, and S. K. Olesen, "On the influence of transfer function noise on sound zone control in a room," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 9, pp. 1405–1418, 2019.
- [193] S. Elliott and P. Nelson, "Active noise control," *IEEE Signal Process. Mag.*, vol. 10, no. 4, pp. 12–35, 1993.
- [194] J. Yang, M. Wu, and L. Han, "A review of sound field control," *Appl. Sci.*, vol. 12, no. 14, 2022.
- [195] "How to restore the legendary acoustics of Notre Dame." <https://www.sciencenews.org/article/notre-dame-cathedral-fire-legendary-acoustics-restoration>, 2020.
- [196] B. F. Katz and A. Weber, "An acoustic survey of the Cathédrale Notre-Dame de Paris before and after the fire of 2019," *Acoustics*, vol. 2, no. 4, pp. 791–802, 2020.
- [197] B. F. G. Katz, D. Murphy, and A. Farina, "Exploring cultural heritage through acoustic digital reconstructions," *Phys. Today*, vol. 73, no. 12, pp. 32–37, 2020.
- [198] G. Navas-Reascos, L. M. Alonso-Valerdi, and D. I. Ibarra-Zarate, "Archaeoacoustics around the world: A literature review (2016-2022)," *Appl. Sci.*, vol. 13, no. 4, 2023.
- [199] P. Brezina, "Acoustics of historic spaces as a form of intangible cultural heritage," *Antiquity*, vol. 87, no. 336, pp. 574–580, 2013.
- [200] M. A. Kolar, "Archaeoacoustics: Re-sounding material culture," *Acoust. Today*, vol. 14, 2018.
- [201] K. Kopij and A. Pilch, "The acoustics of contiones, or how many romans could have heard speakers," *Open Archaeol.*, vol. 5, no. 1, pp. 340–349, 2019.



# List of Publications

## International Journal Papers

1. M. Jälmy, F. Elvander, and T. van Waterschoot, "Low-rank room impulse response estimation," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 957–969, 2023.
2. M. Jälmy, F. Elvander, and T. van Waterschoot, "Compression of Room Impulse Responses for Compact Storage and Fast Low-latency Convolution," Submitted for publication, Dec. 2023.
3. M. Jälmy, F. Elvander, and T. van Waterschoot, "Multi-channel Low-rank Convolution of Jointly Compressed Room Impulse Responses," Submitted for publication, Dec. 2023.

## International Conference Papers

1. M. Jälmy, F. Elvander, and T. van Waterschoot, "Low-rank tensor modeling of room impulse responses," in *Proc. 29th European Signal Process. Conf. (EUSIPCO)*, pp. 111–115, 2021.
2. M. Jälmy, F. Elvander, and T. van Waterschoot, "Low-Rank Acoustic Room Impulse Response Estimation," in *Proc. 2022 Workshop Low-Rank Models Appl. (LRMA)*, 2022.
3. M. Jälmy, F. Elvander, and T. van Waterschoot, "Fast low-latency convolution by low-rank tensor approximation," in *Proc. 2023 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 1–5, 2023.
4. J. Brunnström, M. Jälmy, T. van Waterschoot, and M. Moonen "Fast low-rank filtered-x least mean squares for multichannel active noise control," in *Proc. 57th Asilomar Conf. Signals Syst. Comput.*, 2023.





FACULTY OF ENGINEERING TECHNOLOGY  
DEPARTMENT OF ELECTRICAL ENGINEERING  
STADIUS CENTER FOR DYNAMICAL SYSTEMS, SIGNAL PROCESSING AND DATA ANALYTICS  
Kasteelpark Arenberg 10  
B-3001 Leuven  
<https://www.esat.kuleuven.be/stadius/>

