Extension and Evaluation of Multi-Channel Diffuse PSD Estimators

Marvin Tammen

July 17, 2018
Version: final

Carl von Ossietzky University of Oldenburg

Dept. of Medical Physics and Acoustics
Institute of Medicine and Health Sciences
Signal Processing Group

Engineering Physics (Master)
Master Thesis

Extension and Evaluation of Multi-Channel Diffuse PSD Estimators

Marvin Tammen

1. Reviewer Prof. Dr. ir. Simon Doclo
Signal Processing Group
Dept. of Medical Physics and Acoustics
Carl von Ossietzky University of Oldenburg

2. Reviewer Dr. Ing. Ina Kodrasi
Speech and Audio Processing Group
Idiap Research Institute

July 17, 2018

Marvin Tammen

Extension and Evaluation of
Multi-Channel Diffuse PSD Estimators

Engineering Physics (Master)
Master Thesis, July 17, 2018
Reviewers: Prof. Dr. ir. Simon Doclo and Dr. Ing. Ina Kodrasi

Carl von Ossietzky University of Oldenburg
Signal Processing Group
Institute of Medicine and Health Sciences
Dept. of Medical Physics and Acoustics
Küpkersweg 74
26129 Oldenburg

Abstract

In noisy and reverberant environments, speech enhancement techniques such as the multi-channel Wiener filter (MWF) can be used to improve speech quality and intelligibility. Typically, the MWF is implemented as the concatenation of a minimum variance distortionless response (MVDR) beamformer and a spectro-temporal postfilter. Assuming that reverberation and ambient noise can be modeled as diffuse sound fields, estimates of the relative early transfer function (RETF) vector of the target speaker and of the diffuse power spectral density (PSD) are required to implement the MWF. RETF vector and diffuse PSD estimation methods are often decoupled, i.e., one of the quantities is estimated assuming that the other quantity is known.

First, we aim at jointly estimating the RETF vector and the diffuse PSD by minimizing the Frobenius norm of an error matrix based on the presumed signal model. To solve this minimization problem, we propose to use an alternating least squares approach. Simulation results using artificial and real data show that the proposed method leads to a better performance than a state-of-the-art method based on covariance whitening and the eigenvalue decomposition (EVD)-based diffuse PSD estimator.
Since this estimator does not require knowledge of the RETFs of the target speaker, it has been shown to be advantageous to other state-of-the-art estimators in the presence of RETF mismatches. However, computing the EVD can be computationally expensive, particularly when the number of microphones is large.
This is why, second, we propose to reduce the complexity of the EVD-based PSD estimator by using the iterative power method to compute the eigenvalues. Since the EVD-based PSD estimator only requires the largest eigenvalues, the full EVD is not required and the power method is a well suited computationally efficient technique to estimate these eigenvalues. Experimental results show that using the PSD estimated via the power method in an MWF yields a very similar performance as using the PSD estimated via the full EVD.

Zusammenfassung

In akustischen Umgebungen, die von Rauschen und Nachhall geprägt sind, können Sprachverbesserungs-Methoden wie der mehrkanalige Wiener Filter (MWF) benutzt werden, um Sprachqualität und -verständlichkeit zu verbessern. Typischerweise wird der MWF als die Verkettung eines Minimum Variance Distortionless Response (MVDR) Beamformers und eines spektral-temporalen Postfilters implementiert. Unter der Annahme, dass Nachhall sowie Umgebungsstörgeräusche als ein diffuses Schallfeld modelliert werden können, werden Schätzungen der relativen frühen Transferfunktionen (RETFs) des Zielsprechers sowie der diffusen spektralen Leistungsdichte (PSD) benötigt, um den MWF zu implementieren. Methoden zum Schätzen der RETFs und der PSDs sind üblicherweise entkoppelt, so dass eine der Größen unter der Annahme geschätzt wird, dass die andere Größe bekannt ist.

Erstens erstreben wir, den RETF Vektor sowie die diffuse PSD gemeinsam zu schätzen, indem wir die Frobeniusnorm einer Fehlermatrix minimieren, welche auf dem angenommenen Signalmodell basiert. Um dieses Minimierungsproblem zu lösen, schlagen wir vor, eine alternierende Methode der kleinsten Quadrate zu nutzen. Simulationsergebnisse mit künstlichen und realen Daten zeigen, dass die vorgeschlagene Methode zu einer besseren Leistung führt als eine kürzlich vorstellte Methode, welche auf Kovarianz-Whitening und dem Eigenwertzerlegung (EVD)-basierten diffusen PSD-Schätzer aufbaut. Dieser Schätzer nutzt die EVD der vorgeweißten PSD-Matrix der Mikrofonsignale. Da er kein Wissen über die RETFs des Zielsprechers benötigt, weist er Vorteile gegenüber anderen aktuellen Schätzern in Anwesenheit von RETF-Schätzfehlern auf. Das Berechnen der EVD stellt jedoch aus Gründen der rechnerischen Komplexität ein Problem dar, besonders, wenn die Anzahl der betrachteten Mikrofone groß ist.

Aus diesem Grunde schlagen wir zweitens vor, die Komplexität des oben genannten EVD-basierten PSD-Schätzers zu reduzieren, indem die iterative Potenzmethode eingesetzt wird, um die Eigenwerte zu berechnen. Da der EVD-basierte PSD-Schätzer lediglich die größten Eigenwerte benötigt, ist eine vollständige EVD nicht notwendig, und die Potenzmethode stellt eine passende und effiziente Methode dar, diese Eigenwerte zu schätzen. Experimentelle Ergebnisse zeigen, dass das Ersetzen der vollständigen EVD durch die Potenzmethode zu sehr ähnlichen Ergebnissen führt.

Acknowledgement

This master thesis has been written at the Signal Processing Group in the Department of Medical Physics and Acoustics of the Carl von Ossietzky University in Oldenburg, Germany. Here, I would like to thank all the people that contributed to the finalization of this thesis.

First of all, I would like to thank my first examiner Simon Doclo for giving me the opportunity to dive into a different field than what I had originally planned, as well as for his continuous support and stream of ideas.

Also, I would like to express my gratitude to my supervisor Ina Kodrasi for all the time she sacrificed introducing me into the topic, giving suggestions as well as valuable comments.

Moreover, I want to thank Kai Siedenburg for participating in the examination committee on such short notice.

Of course, also a huge “cheers” goes out to all my present and previous SigProc colleagues, who welcomed me into the group, and who made working here a fun experience.

At the end, I want to thank my family and friends, who always supported me without exception.

Acronyms

ALS	alternating least squares 19
BFGS	Broyden-Fletcher-Goldfarb-Shanno 20
CD	cepstral distance 17
CW	covariance whitening 15
DFT	discrete Fourier transform 5
DNR	diffuse-to-noise ratio 44
EVD	eigenvalue decomposition 12
FFT	fast Fourier transform 32
FIR	finite impulse response 4
flop	floating point operation 29
fwsSNR	frequency-weighted segmental SNR 16
GAXPY	general matrix $\mathbf{A}$ times vector $\mathbf{x}$ plus vector $\mathbf{y}$ 57
LLR	log-likelihood ratio 18
LPC	linear predictive coding 18
LS	least squares 14
LTI	linear time-invariant 2
MMSE	minimum mean square error 7
MVDR	minimum variance distortionless response 7
MWF	multi-channel Wiener filter 4
OPT	unconstrained optimization 44
PESQ	perceptual evaluation of speech quality 16
PSD	power spectral density 4
RETF	relative early transfer function 4
RIR	room impulse response 4
SNR	signal-to-noise ratio 9
STFT	short-time Fourier transform 4
SVD	singular value decomposition 21

Mathematical Notation

x	scalar
$\mathbf{x}$	vector
$\mathbf{X}$	matrix
$\mathbf{X}(j,:)$	j-th row of matrix $\mathbf{X}$
$\mathbf{X}(:,j)$	j-th column of matrix $\mathbf{X}$
a_{ij}	element in i-th row and j-th column of matrix $\mathbf{A}$
$\hat{\circ}$	estimate of $\circ$
$\lambda_i\{\circ\}$	i-th eigenvalue of $\circ$
$\mathcal{E}\{\circ\}$	expected value of $\circ$
$\mathrm{Re}\{\circ\}$	real part of $\circ$
$\mathrm{Im}\{\circ\}$	imaginary part of $\circ$

$A * B$ convolution of A and B

$A := B$ A defined as B

$\circ^H$ Hermitian (conjugate transpose) of $\circ$

$\circ^*$ complex conjugate of $\circ$

$||\circ||_F$ Frobenius norm of $\circ$

Fixed Symbols

$\mathbf{I}_M$ $M \times M$-dimensional identity matrix

$\mathbf{\Gamma}$ spatial coherence matrix

$\mathbf{a}$ RETF vector

m microphone index

n discrete time index

l time frame index

k frequency bin index

$s[n]$ target component in microphone signal

$x[n]$ speech component in microphone signal

$d[n]$ diffuse noise component in microphone signal

$v[n]$ uncorrelated noise component in microphone signal

$i[n]$ interference component in microphone signal (including diffuse and non-diffuse noise)

$y[n]$ microphone signal

$z[n]$ output signal

$z_x[n]$ speech component in output signal

$z_e[n]$ direct and early reverberation component in output signal

$z_r[n]$ late reverberation component in output signal

$z_v[n]$ uncorrelated noise component in output signal

$\mathbf{\Phi_x}$ power spectral density matrix of $\mathbf{x}$, i.e., $\mathbf{\Phi_x} = \mathcal{E}\left\{\mathbf{x}\mathbf{x}^H\right\}$

$\phi_\mathbf{x}$ power spectral density of $\mathbf{x}$, i.e., $\phi_\mathbf{x} = \frac{1}{M}\mathcal{E}\left\{\mathbf{x}^H\mathbf{x}\right\}$

Contents

List of Figures

List of Tables

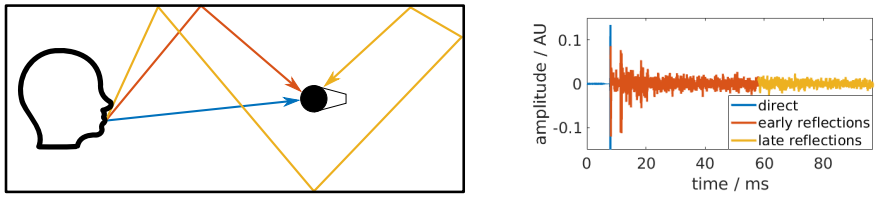
1 Introduction

1.1 Motivation

Recent developments in hand-held devices have led to an increased interaction of humans and machines, be it as a ways of communication (e.g. Skype, Google Duo, Apple FaceTime) or with regard to assisted living technologies, where automatic speech recognition is of utmost importance. Commonly, these systems involve a small set of microphones used to acquire a speech signal in a room of limited size. However, in addition to the desired speech, also interferences are captured by the microphones, including ambient noise and reverberation. This may result in the degradation of speech quality, a decrease in listening comfort [1, 2], and in a performance deterioration of automatic speech recognition systems [3]. Hence, speech enhancement techniques with the ability of suppressing both reverberation and ambient noise are required. To this end, many single- and multi-channel techniques have been proposed, with the latter ones being generally preferred, since they are capable of additionally exploiting spatial information. The multi-channel Wiener filter (MWF) is a commonly used speech enhancement technique, which minimizes the mean square error between a target signal and the output signal [4, 5, 6]. The MWF can be implemented as a minimum variance distortionless response (MVDR) beamformer followed by a single-channel Wiener postfilter [7]. Modeling reverberation and ambient noise as diffuse sound fields, the implementation of the MVDR beamformer and the Wiener postfilter requires estimates of the relative early transfer functions (RETFs) of the target speaker and of the diffuse power spectral density (PSD). Several RETF estimation procedures have been proposed, such as the covariance whitening (CW) method [8, 9], the covariance subtraction method [9, 10], or the least squares (LS) method [11]. In addition, several multi-channel diffuse PSD estimators have been proposed, such as maximum likelihood-based estimators [12, 13], Frobenius norm-based estimators [14, 15], or the eigenvalue decomposition (EVD)-based estimator [16]. In [17] it has been shown that using the CW method to estimate the RETFs and the EVD-based PSD estimator to estimate the diffuse PSD results in a high dereverberation and noise reduction performance. A major drawback of this procedure is its computational complexity, which prevents it from being used in real-time scenarios, where computational resources are usually limited.

The objective of this thesis is a) to propose a technique to reduce the computational complexity of the EVD-based PSD estimator, and b) to couple the PSD and RETF estimation within a joint noise reduction and dereverberation framework.

Reverberation Since the negative effects of reverberation are a major part of what motivates the work in this thesis, a short description of reverberation is provided in the following. In general, reverberation originates from the superposition of acoustical waves reflected at



(a) illustration **(b)** measured RIR, $T_{60} = 610\,\mathrm{ms}$

Fig. 1.1.: a), b): direct path, early reflections and late reflections

surfaces, including walls and other objects. For the purpose of illustration, and to arrive at a more technical definition, the construct of room impulse responses (RIRs) is used. In principle, RIRs are used to model the acoustical path between the source and the microphones in a room as a linear time-invariant (LTI) system, or, figuratively speaking, they represent a black box that receives an input (the source) and outputs the microphone signals. An example of a RIR is depicted in Figure 1.1b, which depicts one channel of a measured RIR used in this thesis. Three stages can be identified (as illustrated in Figure 1.1a):

direct path including a brief period of near-zero values[1], followed by the arrival of the acoustical waves at the microphone on a direct path

early reverberation typically said to be the first $\approx 50\,\mathrm{ms}$ after the direct path; contains first reflections from the wall, which may be well distinguished (as can be seen by the distinct extrema); said to be beneficial for speech intelligibility [18]

late reflections constitute the main negative impact of reverberation, and thus are to be avoided; characterized by the fact that a large number of (multiple) reflections with relatively small amplitudes arrive at the microphones simultaneously, leading to its diffuse nature

A common measure used to quantify the amount of reverberation introduced by a room is the reverberation time T_{60}, which is defined as the time necessary for a signal to drop by 60 dB after it is cut off. Usually, as also done in this work, the reverberation time is given as a single number, although, to be more precise, it is a frequency-dependent quantity.

1.2 Outline

In Chapter 2, some theoretical background is provided, laying the foundation of this thesis. The time domain signal model and the notation used throughout this work are introduced. Since most of the methods described in this work are derived in the time-frequency domain, the short-time Fourier transform (STFT) is presented, which is a transformation used to obtain the time-frequency representation of time domain signals. Corresponding changes to the signal model are described. A brief overview of the MWF is provided, which is the speech enhancement technique that the main contributions of this thesis are dedicated to. The EVD-based diffuse PSD estimator is the basis of one of these contributions, and hence it is explained briefly, followed by a least squares-based competing estimator. Furthermore, a method for the joint estimation of the RETF vector and the diffuse and target PSD is introduced. Finally, the methods used for the objective evaluation of the MWF performance are presented.

Chapter 3 contains a description of the main contributions of this thesis. We propose an alternating least squares (ALS) approach for the joint estimation of the diffuse and the target

[1] The values are not exactly zero due to sampling.

PSDs as well as the time-varying RETF vector. Regarding the EVD-based diffuse PSD estimator, we propose a variant with significantly reduced computational complexity.

In Chapter 4, the performance of the proposed methods is evaluated and compared to reference approaches. The computationally more viable variant of the EVD-based diffuse PSD estimator is compared to the original estimator in terms of run speed, PSD estimation accuracy, as well as resulting MWF performance in different reverberant acoustical scenarios including different types of noise. Similarly, the ALS approach is compared to a state-of-the-art approach in terms of diffuse PSD and RETF vector estimation accuracy, both for artificial data and in the acoustical scenarios mentioned above.

Finally, in Chapter 5, the thesis is concluded, and possibilities for further research are pointed out.

2 Theoretical Prerequisites

The aim of this chapter is to present the reader with the theoretical background that is necessary to follow what is done in this work. First, the signal model is presented in the time domain, and the reader is introduced to the used notation in Section 2.1. Second, since all of the processing in this thesis is done in the time-frequency domain, the short-time Fourier transform (STFT) is introduced, and the signal model is extended to the time-frequency representation of the time domain signals in Section 2.2. A brief overview of the multi-channel Wiener filter (MWF) is provided in Section 2.3, which is the speech enhancement technique used throughout this work. In the considered implementation, it requires estimates of the relative early transfer function (RETF) vector and the diffuse power spectral density (PSD), which are the subjects in Section 2.4 and Section 2.5. To evaluate the performance of the MWF using the various estimators, objective measures are presented in Section 2.6.

2.1 Time Domain Signal Model

We consider a single spatially stationary speech source and M microphones in a reverberant and noisy acoustical scenario. The signal at the m-th microphone and time index n can be written as the superposition of the desired speech and an interference term including reverberation and noise as

$$y_m[n] = \underbrace{h_m[n] * s[n]}_{x_m[n]} + i_m[n]. \tag{2.1}$$

Here, the desired speech component $x_m[n] := h_m[n] * s[n]$ is obtained as the filtered speech source, where $h_m[n]$ contains the direct path and early reflections of the room impulse response (RIR) from the speech source $s[n]$ to the m-th microphone, $*$ denotes the convolution operator, and $i_m[n]$ is the interference term. Note that in this model, only *one* source is considered. Modeling the RIR as a finite impulse response (FIR) filter $\mathbf{h}_m = [h_m(0), h_m(1), \ldots, h_m(L_h - 1)]^T$ with length L_h, the convolution can be expressed as

$$x_m[n] = \sum_{l=0}^{L_h-1} h_m[l]s[n-l]. \tag{2.2}$$

The term $i_m[n]$ contains any interference to the desired signal that is to be removed or attenuated, i.e., late reverberation, background noise and/or sensor noise. The interference component is further separated into a diffuse component $d_m[n]$ (that can be used to model, e.g., babble noise and late reverberation) and a component $v_m[n]$, which is used to model uncorrelated noise, e.g., sensor noise, such that Equation 2.1 becomes

$$y_m[n] = x_m[n] + d_m[n] + v_m[n]. \tag{2.3}$$

2.2 Short-Time Fourier Transform

Since a large part of this work involves inspection of the time-frequency representation of time domain signals, this section is devoted to briefly introducing the reader to the methods used in this context. The following derivations are mainly based on [19].

2.2.1 Discrete Fourier Transform

Recall the definition of the Fourier transform of the continuous-time signal $x(t)$

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi\omega t} dt, \tag{2.4}$$

with $j^2 = -1$, ω the continuous angular frequency, and t the continuous time variable. The techniques described in this work focus on discrete, and hence computer-representable, signals. $x(t)$ is replaced by a periodically sampled version, $x[n] := x(nT)$, where n corresponds to the time index and T to the sample period. Analogously, the spectrum $X(\omega)$ is replaced by the *sampled* version $X[k] := X(\omega = 2\pi k/N)$ (denoted discrete Fourier transform (DFT)) with spacing $2\pi/N$ between two samples. Thus, Equation 2.4 is modified to

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k = 0, ..., K-1, \tag{2.5}$$

where N is the signal length and K is the number of frequency bins of the spectrum. The inverse DFT, which yields the time domain signal when being applied to its spectrum, is then given as

$$x[n] = \frac{1}{K} \sum_{k=0}^{K-1} X[k] e^{j2\pi kn/N}, \quad n = 0, ..., N-1. \tag{2.6}$$

Note that choosing the DFT length $K = N$ is sufficient for retaining the spectral information of $x[n]$; however for visualization purposes it may be useful to zero-pad the signal to length $N > K$. This does not add any information, but it leads to a finer interpolation of the spectrum and any visual spectral characteristics may become accessible more easily.

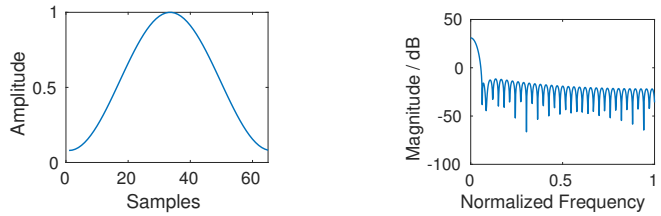
2.2.2 Application in Long Signals

In some situations, it may prove useful to obtain overall spectral characteristics of a signal, hence requiring the application of Equation 2.5. On the other side, crucial information may be highly localized in time. Speech, e.g., is often assumed to be stationary within time frames of $\approx (20 - 30)\,\mathrm{ms}$. This makes it necessary to investigate the spectrum of *subsets* of the signal sequence, which are obtained via windowing. Considering different segments in time, the signal is transformed into the time-frequency domain by applying the STFT. The l-th segment at time n is then given by

$$x_l[n] = w[n] x[n + lR], \quad n = 0, ..., N_w - 1, \tag{2.7}$$

where R corresponds to the shift, i.e., the number of samples between two segments, and $w[n]$ is a window sequence with length N_w. As such, the representation of the l-th time frame and the k-th bin of the signal becomes

$$X(k, l) = \sum_{n=0}^{N_w-1} w[n] x[n + lR] e^{-j2\pi nk/N_w}, \quad k = 0, ..., K-1. \tag{2.8}$$



(a) shape **(b)** frequency response

Fig. 2.1.: a) Hamming window and b) its spectrum

The window choice has a large influence on the STFT. Due to the fact that the observed signals have finite lengths, the frequency resolution of their STFTs is limited. Indeed, a higher spectral resolution requires more samples and hence leads to a lower temporal resolution, and vice versa. Furthermore, signal energy concentrated at one specific frequency f may not only be found in the corresponding frequency bin, but it may be spread to neighboring bins, termed spectral "leakage". This is an effect caused by the *side lobes* of the window spectrum, which can be seen as the local maxima in Figure 2.1b. In addition, due to the width of the main lobe of the window spectrum, it may occur that two separate spectral peaks may not be resolved individually. Choosing a window function means a trade-off between frequency resolution and leakage. In this work, $w[n]$ represents a Hamming window of length N_w, given by (cf. Figure 2.1a)

$$w[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N_w - 1}\right), \quad n = 0,\ \ldots,\ N_w - 1 \tag{2.9}$$

with the property of *minimizing* the *nearest side lobe* (cf. Figure 2.1b).

Concluding, it can be said that the STFT parameters (window type, DFT length N, shift R) need to be chosen carefully with regard to the application in order to yield satisfying results. For instance, in case of speech analysis, a suitable trade-off for the temporal and spectral resolution needs to be found that satisfies both the need for large spectral resolution as well as the assumption that speech can be considered stationary only at small time scales.

2.2.3 STFT-Domain Signal Model

Since all of the processing in this work is done in the time-frequency domain, the discussion of the signal model is continued as such. This transition brings with it the advantages that filtering becomes computationally less expensive (as given by the convolution theorem). Applying the STFT to the signal components in Equation 2.3 and using the property of linearity yields

$$Y_m(k,l) = X_m(k,l) + D_m(k,l) + V_m(k,l), \tag{2.10}$$

where k and l are the frequency and time indices, $Y_m(k,l)$ is the STFT of $y_m[n]$ and $X_m(k,l)$, $D_m(k,l)$ and $V_m(k,l)$ are defined similarly. For a more convenient notation, $\mathbf{y}(k,l)$ is defined as the stacked components vector

$$\mathbf{y}(k,l) = [Y_1(k,l),\, Y_2(k,l),\, \ldots,\, Y_M(k,l)]^T, \tag{2.11}$$

such that Equation 2.10 can be written simultaneously for all microphones as

$$\boxed{\mathbf{y}(k,l) = \mathbf{x}(k,l) + \mathbf{d}(k,l) + \mathbf{v}(k,l),} \tag{2.12}$$

with $\mathbf{x}(k,l)$, $\mathbf{d}(k,l)$, and $\mathbf{v}(k,l)$ similarly defined.

2.3 Multi-Channel Wiener Filter

One commonly used technique with the aim of dereverberation and denoising of an input signal is the multi-channel Wiener filter (MWF), described, e.g., in [20, 7]. It achieves interference reduction by taking into account the statistical properties of the signals and minimizing the mean square error between a target and the output signal [21], yielding an optimal solution in the minimum mean square error (MMSE) sense. In this work, the MWF is implemented as a two-stage technique, with a structure that is briefly described in the following (cf. Figure 2.2).

In the first stage of the MWF, a beamformer with $\mathbf{w}_{\text{MVDR}}(k,l) = [W_1(k,l),\ \ldots, W_M(k,l)]^T$ as its filter coefficients (cf. Section 2.3.1) is applied, exploiting spatial information of the signal. At the output of this first stage, an enhanced single-channel signal remains, which is given by

$$Z(k,l) = \mathbf{w}_{\text{MVDR}}^H(k,l)\mathbf{y}(k,l), \tag{2.13}$$

where $\circ^H$ denotes the Hermitian operator. To further enhance the output, a single-channel spectral postfilter $G(k,l)$ (cf. Section 2.3.2) is applied to the output of the minimum variance distortionless response (MVDR) beamformer, resulting in the final target estimate

$$\hat{S}(k,l) = G(k,l)Z(k,l) \tag{2.14}$$

$$= \underbrace{G(k,l)\mathbf{w}_{\text{MVDR}}^H(k,l)}_{\mathbf{w}_{\text{MWF}}^H(k,l)}\mathbf{y}(k,l), \tag{2.15}$$

where $\mathbf{w}_{\text{MWF}}(k,l)$ are the coefficients of the MWF. Note that the computations are done independently for each frequency bin, which is why for the remainder of this thesis, the frequency index k is omitted wherever possible.

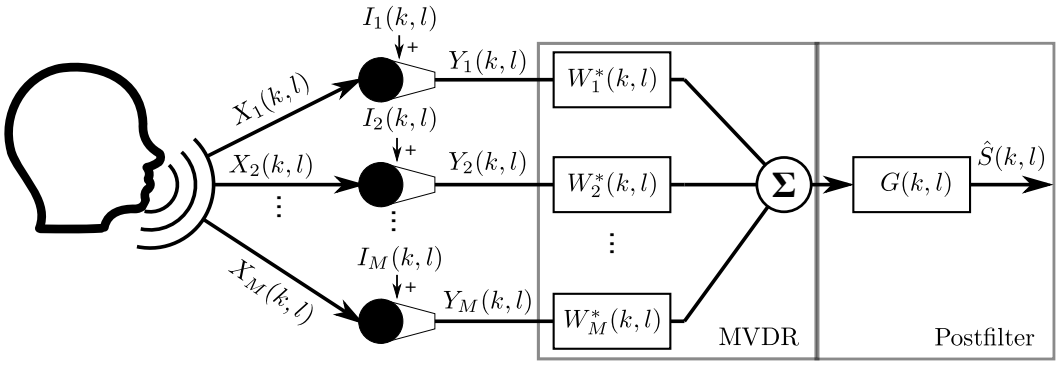


Fig. 2.2.: acoustical system configuration

2.3.1 Minimum Variance Distortionless Response Beamformer

The MVDR beamformer aims at obtaining filter coefficients solving the optimization problem

$$\mathbf{w}_{\text{MVDR}}(l) = \underset{\mathbf{w}\in\mathbb{C}^M}{\operatorname{argmin}}\{\mathbf{w}^H\mathbf{\Phi}_{\mathbf{i}}(l)\mathbf{w}\}, \quad \text{s. t.} \quad \mathbf{w}^H\mathbf{c}(l) = 1, \tag{2.16}$$

where $\mathbf{\Phi}_{\mathbf{i}}(l)$ is the $M \times M$-dimensional interference PSD matrix and $\mathbf{c}(l)$ is an M-dimensional constraint vector. $\mathbf{c}(l)$ defines the path that remains undistorted by the MVDR (addressed

by "distortionless response") and needs to be chosen carefully. When aiming for both dereverberation and noise reduction, e.g., one may choose the reverberant speech component to remain undistorted. In this work, $\mathbf{c}(l)$ is chosen to be the possibly time-dependent RETF vector $\mathbf{a}(l)$, which results in an alignment of the direct and early reverberation paths from the reference microphone to all microphones at the MVDR output.

The solution to the optimization problem in Equation 2.16 may be obtained using a *Lagrange multiplier* λ as follows, starting with the derivative w.r.t. the filter weights $\mathbf{w}$:

$$\begin{aligned}\frac{d}{d\mathbf{w}}\left(\mathbf{w}^H\mathbf{\Phi}_{\mathbf{i}}(l)\mathbf{w}+\lambda(\mathbf{w}^H\mathbf{c}(l)-1)\right) &= \left(\mathbf{\Phi}_{\mathbf{i}}(l)+\mathbf{\Phi}_{\mathbf{i}}^H(l)\right)\mathbf{w}+\lambda\mathbf{c}(l)\\ &= 2\mathbf{\Phi}_{\mathbf{i}}(l)\mathbf{w}+\lambda\mathbf{c}(l)\\ &\overset{!}{=} 0 \leftrightarrow\\ \mathbf{w}_{\text{MVDR}}(l) &= -\frac{\lambda}{2}\Phi_{\mathbf{i}}^{-1}(l)\mathbf{c}(l),\end{aligned} \tag{2.17}$$

where $\mathbf{\Phi}_{\mathbf{i}}(l) = \mathbf{\Phi}_{\mathbf{i}}^H(l)$ is exploited in the second line (cf. Equation 2.30). Setting the derivative w.r.t. the Lagrange multiplier λ to zero and afterwards inserting $\mathbf{w}_{\text{MVDR}}$ yields

$$\begin{aligned}\frac{d}{d\lambda}\left(\mathbf{w}_{\text{MVDR}}^H(l)\mathbf{\Phi}_{\mathbf{i}}(l)\mathbf{w}_{\text{MVDR}}(l)+\lambda(\mathbf{w}_{\text{MVDR}}^H(l)\mathbf{c}(l)-1)\right) &= \mathbf{w}_{\text{MVDR}}^H\mathbf{c}(l)-1\\ &= -\frac{\lambda}{2}\mathbf{c}^H(l)\mathbf{\Phi}_{\mathbf{i}}^{-H}(l)\mathbf{c}(l)-1\\ &\overset{!}{=} 0 \leftrightarrow\\ \lambda &= \left(-\frac{1}{2}\mathbf{c}^H(l)\mathbf{\Phi}_{\mathbf{i}}^{-H}(l)\mathbf{c}(l)\right),\end{aligned} \tag{2.18}$$

which, when inserted back into $\mathbf{w}_{\text{MVDR}}(l)$, leads to the solution

$$\begin{aligned}\mathbf{w}_{\text{MVDR}}(l) &= \frac{\mathbf{\Phi}_{\mathbf{i}}^{-1}(l)\mathbf{c}(l)}{\mathbf{c}^H(l)\mathbf{\Phi}_{\mathbf{i}}^{-H}(l)\mathbf{c}(l)}\\ &= \frac{\mathbf{\Phi}_{\mathbf{i}}^{-1}(l)\mathbf{c}(l)}{\mathbf{c}^H(l)\mathbf{\Phi}_{\mathbf{i}}^{-1}(l)\mathbf{c}(l)},\end{aligned} \tag{2.19}$$

where $\mathbf{\Phi}_{\mathbf{i}}(l) = \mathbf{\Phi}_{\mathbf{i}}^H(l)$ is exploited again in the last equality. Note that in the absence of uncorrelated noise, $\mathbf{\Phi}_{\mathbf{i}}(l)$ may be replaced with the spatial coherence matrix $\mathbf{\Gamma}$, which, in the case of a time-independent constraint vector $\mathbf{c}$, leads to stationary MVDR coefficients (cf. Section 2.3.4).

The single-channel MVDR output $Z(l)$, which corresponds to an estimate of the speech signal containing the direct and early reverberation component, is then given by (cf. Figure 2.2)

$$\begin{aligned}Z(l) &= \mathbf{w}_{\text{MVDR}}^H(l)\mathbf{y}(l)\\ &= Z_e(l)+Z_d(l)+Z_v(l).\end{aligned} \tag{2.20}$$

Since no perfect dereverberation and denoising can be achieved by the MVDR [22][1], in addition to the desired direct and early reverberation output component $Z_e(l)$, the estimate contains residual late reverberation and noise, which is represented by the diffuse output component $Z_d(l)$ and the uncorrelated noise output component $Z_v(l)$. Additional enhance-

[1] One intuitive reason for this is that the MVDR depends on PSD matrices, which contain only absolute values and no complex phases.

ment can be achieved by applying a single-channel postfilter to the MVDR output, which is described in the following section.

2.3.2 Single-Channel Postfilter

The performance of the beamformer described in the previous section strongly depends on the acoustical scenario that it is applied in. While coherent noise may usually be well suppressed, diffuse noise (such as babble noise) is reduced at a smaller degree [23], and improvement upon the MVDR is still possible. This is why a postfilter aiming at further signal enhancement is applied to the MVDR output. In general, the postfilter is given by a real-valued gain function $G(l)$, which weights the STFT bins with their significance for the desired signal, resulting in the target estimate

$$\hat{S}(l) = G(l)Z(l). \tag{2.21}$$

Several techniques for the computation of the gain $G(l)$ exist, which commonly depend on the *a posteriori* signal-to-noise ratio (SNR)

$$\gamma(l) := \frac{\mathcal{E}\left\{|Z(l)|^2\right\}}{\mathcal{E}\left\{|Z_d(l) + Z_v(l)|^2\right\}} = \frac{\mathcal{E}\left\{|Z(l)|^2\right\}}{\phi_{zi}(l)}, \tag{2.22}$$

which relates the estimated speech PSD (corresponding to the MVDR output) to the interference PSD $\phi_{zi}(l)$, with $\mathcal{E}\{\circ\}$ the expected value operator, and / or the *a priori* SNR

$$\xi(l) := \frac{\mathcal{E}\left\{|S(l)|^2\right\}}{\mathcal{E}\left\{|Z_d(l) + Z_v(l)|^2\right\}} = \frac{\phi_s(l)}{\phi_{zi}(l)}, \tag{2.23}$$

which relates the estimated target PSD $\phi_s(l)$ to the interference PSD $\phi_{zi}(l)$ at the output of the MVDR. In this work, the gain is computed as

$$G(l) = \max\left(\frac{\hat{\xi}(l)}{1+\hat{\xi}(l)}, G_{\min}\right),\ 0 < G_{\min} < 1, \tag{2.24}$$

where the *a priori* SNR estimate $\hat{\xi}(l)$ is obtained using the decision-directed approach [24](cf. Section 2.3.3), such that $G(l) \in [G_{\min},\, 1)$. The minimum gain $G_{\min}$ prevents a zero or negative output, which would typically have a negative influence on subjective signal quality [25] (termed “spectral flooring”).

2.3.3 A Priori SNR Estimation

As can be seen in Equation 2.24, the single-channel spectral postfilter requires an estimate of the *a priori* SNR $\xi(l)$. A common method used to obtain an estimate of this quantity is the decision-directed approach. In the following, the method will be described in a qualitative fashion; for a more detailed description, please refer to [24].

To arrive at the estimate, a statistical model of the STFT coefficients is assumed. In particular, the coefficients are modeled as independent Gaussians with a zero-mean and a time-varying variance, which originates from the strong non-stationary character of speech. The assumption of independent coefficients is valid only in the extreme case of an infinitely long STFT analysis window. Since in practice the windows have a length in the range of 64 ms, this assumption does not hold. However, by applying a Hamming window to the signal during the STFT analysis, this problem is mitigated to some extent due to its small side lobe at the cost of a large main lobe (cf. Figure 2.1).

As mentioned above, the decision-directed approach aims at estimating the *a priori* SNR $\xi(l)$ (defined in Equation 2.23). A simple calculation, again assuming that the interference component and the speech component are uncorrelated, leads to its relation to the *a posteriori* SNR (cf. Equation 2.22):

$$\xi(l) = \frac{\phi_s(l)}{\phi_{zi}(l)} = \frac{\mathcal{E}\left\{|Z(l)|^2\right\} - \phi_{zi}(l)}{\phi_{zi}(l)} = \frac{\mathcal{E}\left\{|Z(l)|^2\right\}}{\phi_{zi}(l)} - 1 = \gamma(l) - 1. \tag{2.25}$$

Hence, using a linear combination of the definition in Equation 2.23 and the result from above, the *a priori* SNR can be written as

$$\xi(l) = \beta\frac{\phi_s(l)}{\phi_{zi}(l)} + (1-\beta)(\gamma(l)-1), \quad \beta \in \mathbb{R}. \tag{2.26}$$

In the decision-directed approach, a practical estimate of Equation 2.26 is obtained by substituting the expectation values (cf. Equation 2.22 and Equation 2.23) with the estimates of the previous frame, where the MVDR output corresponds to an estimate of the target signal, resulting in

$$\hat{\xi}(l) = \beta\frac{|Z(l-1)|^2}{\hat{\phi}_{zi}(l-1)} + (1-\beta)\max\left\{\hat{\gamma}(l)-1, 0\right\}, \tag{2.27}$$

where β is called "smoothing factor". The $\max$ operation is used to ensure a non-negative second term. In the original paper proposing this algorithm [24], the smoothing factor $\beta = 0.98$ is suggested.

Intuitively, by smoothing, the decision-directed approach reduces overly rapid PSD estimate changes, resulting in a larger subjective quality of the filter output.

2.3.4 Spatial Coherence Matrix

To describe the effect of a particular microphone array geometry on the coherence of the individual microphone signals, the spatial coherence matrix $\mathbf{\Gamma}$ is used, which is often assumed to be time-invariant, since microphone arrays commonly have a fixed arrangement. Several noise types can be modeled as a diffuse noise field [14, 26, 13, 27, 12, 15] including, e.g., late reverberation and babble noise. In the scope of this work, a spherically diffuse noise field[1] is assumed, such that the spatial coherence matrix can be computed as [28]

$$\mathbf{\Gamma}_{ij}(k) = \operatorname{sinc}\left(2\pi f_k l_{ij}/c\right). \tag{2.28}$$

Here, the entry $\mathbf{\Gamma}_{ij}(k)$ describes the coherence between the i-th and j-th microphone (which have a distance of l_{ij}) in the k-th frequency bin with center frequency f_k, and $c \approx 340.29\,\mathrm{m/s}$ is the speed of sound. Note that for further processing (cf. Section A.3), $\mathbf{\Gamma}(k)$ needs to be symmetric and positive definite. Under the used assumption, its symmetry is inherent, which is easily seen in Equation 2.28 ($\mathbf{\Gamma}_{ij}(k) = \mathbf{\Gamma}_{ji}(k)$, since $l_{ij} = l_{ji}$).

Remark At this point, it should be emphasized that the considered implementation of the MWF requires several quantities. Its first stage, the MVDR beamformer, relies on estimates of the interference PSD matrix $\mathbf{\Phi_i}(l)$ (or, in case the model includes only diffuse noise, the spatial coherence matrix $\mathbf{\Gamma}$), as well as the RETF vector $\mathbf{a}(l)$. Its second stage, the spectro-temporal postfilter relying on the decision-directed approach, requires an estimate of the diffuse PSD $\phi_{\mathbf{d}}(l)$.

[1] A spherically diffuse noise field models the sound as coming from all directions, distributed equally.

2.4 Power Spectral Density Estimation

Subject of this section is the estimation of PSDs and PSD matrices, which are required for the implementation of the MWF (cf. Equation 2.19 and Equation 2.24). In general, the scalar PSD $\phi_{\mathbf{x}}(l)$ of a stacked vector $\mathbf{x}(l) = [X_1(l),\ \ldots,\ X_M(l)]^T$ is defined as the expected value of signal energy in an STFT bin averaged over the microphones, i.e.,

$$\begin{aligned}\phi_{\mathbf{x}}(l) &= \frac{1}{M}\mathcal{E}\left\{\mathbf{x}^H(l)\mathbf{x}(l)\right\} \\ &= \frac{1}{M}\mathcal{E}\left\{\sum_{m=1}^{M} X_m^*(l)X_m(l)\right\},\end{aligned} \tag{2.29}$$

while the $M \times M$-dimensional PSD *matrix* $\mathbf{\Phi}_{\mathbf{x}}(l)$ is defined as

$$\mathbf{\Phi}_{\mathbf{x}}(l) = \mathcal{E}\left\{\mathbf{x}(l)\mathbf{x}^H(l)\right\} = \mathcal{E}\left\{\begin{bmatrix} X_1(l)X_1^H(l) & \ldots & X_1(l)X_M^H(l) \\ \vdots & \ddots & \vdots \\ X_M(l)X_1^H(l) & \ldots & X_M(l)X_M^H(l) \end{bmatrix}\right\} = \mathbf{\Phi}_{\mathbf{x}}^H(l). \tag{2.30}$$

Since it is common to model STFT coefficients as zero-mean Gaussian random variables [24], "signal energy" and "variance" (within a bin) have the same definition and are used interchangeably in literature. The methods used for the estimation of the relevant PSDs in this work are briefly outlined in the following sections.

2.4.1 Microphone and Uncorrelated Noise PSD Matrices

The microphone PSD matrix is estimated directly from the microphone signals, where the expectation operation is approximated using recursive averaging with one signal realization as

$$\hat{\mathbf{\Phi}}_{\mathbf{y}}(l) = \begin{cases} (1-\alpha)\hat{\mathbf{\Phi}}_{\mathbf{y}}(l-1) + \alpha\ \mathbf{y}(l)\mathbf{y}^H(l), & l > 0 \\ \mathbf{y}(l)\mathbf{y}^H(l), & l = 0, \end{cases} \tag{2.31}$$

where α is referred to as "forgetting" or "smoothing" factor.

The estimation of the uncorrelated noise PSD matrix $\mathbf{\Phi}_{\mathbf{v}}(l)$, which contains sensor noise in the performed simulations, is based on the assumption that it is slowly time-varying [22]. Hence, in the considered scenarios, it is sufficient to estimate it in noise-only phases of the signal using, e.g., recursive averaging as described above.

In both cases, the associated smoothing factor α depends on how the STFT coefficients are obtained; in this work we use

$$\alpha = \exp\left(\frac{-R}{f_s\tau}\right) \tag{2.32}$$

with the window shift R (in samples), the sampling frequency f_s, and the time constant τ (influenced by the STFT settings).

2.4.2 Diffuse Noise PSD

In contrast to the uncorrelated noise PSD matrix described above, the assumption of a slowly varying diffuse noise PSD matrix is not valid. An intuitive reasoning for this is the fact that late reverberation, which is modeled as a diffuse noise field[1], originates from the (speech)

[1] The intuition behind modeling late reverberation as a diffuse sound field results from the fact that it usually consists of the combination of reflections arising from all directions.

source, which in general is *highly* time-varying. Hence, more sophisticated approaches are required to obtain reasonable approximations of the diffuse noise PSD. This work focuses on the eigenvalue decomposition (EVD)-based approach [17, 29], which is described in the following section. As an alternative, the Frobenius norm-based approach [15] is discussed in Section 2.4.3.

Eigenvalue Decomposition-Based Approach

Many of the state-of-the-art diffuse PSD estimators, e.g. [14, 13, 27, 12, 15], require accurate estimates of the RETF vector, which may be difficult to obtain. Especially in reverberant scenarios, RETF vector mismatches are likely to occur, leading to PSD estimation errors and hence a worse MWF performance. The EVD-based diffuse PSD estimator described and evaluated in [17, 29] circumvents this by not relying on RETF vector estimates, leading to advantages especially in scenarios with RETF estimation errors.

To simplify the derivation, only diffuse noise is considered in the following, i.e., uncorrelated sensor noise is neglected. Note, however, that, as long as an estimate of the uncorrelated noise PSD matrix $\mathbf{\Phi}_{\mathbf{v}}(l)$ is available, the approach may be applied to the general case as well (cf. Section 2.4.1). Thus, the stacked microphone vector can be written as

$$\begin{aligned}\mathbf{y}(l) &= \mathbf{x}(l) + \mathbf{d}(l) + \underbrace{\mathbf{v}(l)}_{=\mathbf{0}} \\ &= \mathbf{x}(l) + \mathbf{d}(l).\end{aligned} \tag{2.33}$$

Assuming that both components are uncorrelated, the corresponding $M \times M$-dimensional PSD matrix may be written as

$$\begin{aligned}\mathbf{\Phi}_{\mathbf{y}}(l) &= \mathcal{E}\left\{\mathbf{y}(l)\mathbf{y}^H(l)\right\} \\ &= \mathbf{\Phi}_{\mathbf{x}}(l) + \mathbf{\Phi}_{\mathbf{d}}(l).\end{aligned} \tag{2.34}$$

Note that this assumption is clearly not fulfilled, since late reverberation contains a transformed version of the target signal, and thus is indeed correlated with it to some extent. Still, this model leads to good results and hence is common in literature.

The speech component is modeled as the target signal filtered with the RETFs, which are defined as the RIRs mentioned in Equation 2.1, normalized w.r.t. one of the microphone paths, i.e., $\mathbf{x}(l) = S(l)\mathbf{a}(l)$. Hence, we can rewrite $\mathbf{\Phi}_{\mathbf{x}}(l)$:

$$\begin{aligned}\mathbf{\Phi}_{\mathbf{x}}(l) &= \mathcal{E}\left\{\mathbf{x}(l)\mathbf{x}^H(l)\right\} \\ &= \mathcal{E}\left\{S(l)\mathbf{a}(l)\left[S(l)\mathbf{a}(l)\right]^H\right\} \\ &= \mathcal{E}\left\{S(l)\mathbf{a}(l)\mathbf{a}^H(l)S^*(l)\right\} \\ &= \underbrace{\mathcal{E}\left\{|S(l)|^2\right\}}_{=:\phi_s(l)}\mathcal{E}\left\{\mathbf{a}(l)\mathbf{a}^H(l)\right\} \quad (S(l),\ \mathbf{a}(l) \text{ uncorrelated}) \\ &= \phi_s(l)\mathbf{a}(l)\mathbf{a}^H(l),\end{aligned} \tag{2.35}$$

where the expectation operator disappears in the last equality because the RETF vector $\mathbf{a}(l)$ is deterministic. The diffuse PSD matrix $\mathbf{\Phi}_{\mathbf{d}}(l)$ can be represented as the (assumed-to-be time-invariant) scaled spatial coherence matrix $\mathbf{\Gamma}$ (cf. Section 2.3.4) as

$$\mathbf{\Phi}_{\mathbf{d}}(l) = \phi_{\mathbf{d}}(l)\mathbf{\Gamma}, \tag{2.36}$$

with the scaling factor given by the diffuse PSD $\phi_{\mathbf{d}}(k,l)$. Using Equation 2.35 and Equation 2.36, Equation 2.34 can be rewritten as

$$\boxed{\mathbf{\Phi}_{\mathbf{y}}(l) = \underbrace{\phi_s(l)\mathbf{a}(l)\mathbf{a}^H(l)}_{\mathbf{\Phi}_{\mathbf{x}}(l)} + \underbrace{\phi_{\mathbf{d}}(l)\mathbf{\Gamma}}_{\mathbf{\Phi}_{\mathbf{d}}(l)}.} \tag{2.37}$$

To obtain an estimate of the diffuse PSD, a prewhitened version of the microphone PSD matrix $\mathbf{\Phi}_{\mathbf{y}}(l)$ is obtained using the inverse of the Cholesky decomposition $\mathbf{\Gamma} = \mathbf{L}\mathbf{L}^H$ of the spatial coherence matrix:

$$\begin{aligned}
\mathbf{\Phi}_{\mathbf{y}}^{\mathrm{w}}(l) &= \mathbf{L}^{-1}\mathbf{\Phi}_{\mathbf{y}}(l)\mathbf{L}^{-H} \\
&= \mathbf{L}^{-1}\mathbf{\Phi}_{\mathbf{x}}(l)\mathbf{L}^{-H} + \mathbf{L}^{-1}\mathbf{\Phi}_{\mathbf{d}}(l)\mathbf{L}^{-H} \\
&= \phi_s(l)\underbrace{\mathbf{L}^{-1}\mathbf{a}(l)}_{=:\mathbf{b}(l)}\underbrace{\mathbf{a}^H(l)\mathbf{L}^{-H}}_{=\mathbf{b}^H(l)} + \phi_{\mathbf{d}}(l)\underbrace{\mathbf{L}^{-1}\mathbf{\Gamma}\mathbf{L}^{-H}}_{=\mathbf{I}_M} \\
&= \underbrace{\phi_s(l)\mathbf{b}(l)\mathbf{b}^H(l)}_{\text{rank-1}} + \phi_{\mathbf{d}}(l)\mathbf{I}_M
\end{aligned} \tag{2.38}$$

The Cholesky decomposition, which is briefly described in Section A.3, requires $\mathbf{\Gamma}$ to be symmetric and positive definite.

As can be seen in the last line, the prewhitened microphone PSD matrix is the sum of a rank-1 matrix and a scaled identity matrix, which is exploited to obtain a diffuse PSD estimate as follows. Regarding the rank-1 matrix, it can easily be shown that it has only a single non-zero eigenvalue $\sigma > 0$ (cf. Theorem A.2.1).

Now, the contribution of the scaled identity matrix is considered.

Theorem 2.4.1. *If a matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ has eigenvalues λ_i, $1 \leq i \leq M$, then $\mathbf{B} = \mathbf{A} + \mu\mathbf{I}_M$ has eigenvalues $\lambda_i + \mu$, $1 \leq i \leq M$.*

Proof. Consider the eigenvalue problem $\mathbf{A}\mathbf{u}_i = \lambda_i\mathbf{u}_i$. Define $\mathbf{B} := \mathbf{A} + \mu\mathbf{I}_M$, then:

$$\begin{aligned}
\mathbf{B}\mathbf{u}_i &= (\mathbf{A} + \mu\mathbf{I}_M)\mathbf{u}_i \\
&= \mathbf{A}\mathbf{u}_i + \mu\mathbf{I}_M\mathbf{u}_i \\
&= \lambda_i\mathbf{u}_i + \mu\mathbf{I}_M\mathbf{u}_i \\
&= \lambda_i\mathbf{u}_i + \mu\mathbf{u}_i \\
&= (\lambda_i + \mu)\mathbf{u}_i
\end{aligned}$$

□

Combining these two results, it can be concluded that the eigenvalues of the prewhitened microphone PSD matrix are given by

$$\begin{cases}
\lambda_1\{\mathbf{\Phi}_{\mathbf{y}}^{\mathrm{w}}(l)\} = \sigma(l) + \phi_{\mathbf{d}}(l) & (2.39) \\
\lambda_i\{\mathbf{\Phi}_{\mathbf{y}}^{\mathrm{w}}(l)\} = \phi_{\mathbf{d}}(l) \quad \forall i \in \{2,...,M\}. & (2.40)
\end{cases}$$

Here and in the remainder of this work, $\lambda_i\{\circ\}$ denotes the i-th eigenvalue of $\circ$ arranged in descending order.

In [29] it is proposed to make use of either the first or the second eigenvalue in estimating the diffuse PSD $\phi_{\mathbf{d}}(l)$. Conferring with Equation 2.40, we have

$$\boxed{\phi_{\mathbf{d}}^{\text{EIG2}}(l) := \lambda_2\{\mathbf{\Phi}_{\mathbf{y}}^{\text{w}}(l)\}.} \tag{2.41}$$

Alternatively, Equation 2.39 can be exploited by using the trace equality to estimate the diffuse PSD as follows:

$$\begin{aligned} \text{trace}\{\mathbf{\Phi}_{\mathbf{y}}^{\text{w}}(l)\} &= \sum_{m=1}^{M} \lambda_i(l) \\ &= \lambda_1(l) + \sum_{m=2}^{M} \lambda_i(l) \\ &= \lambda_1(l) + (M-1)\phi_{\mathbf{d}}(l) \end{aligned} \tag{2.42}$$

$$\boxed{\rightarrow \phi_{\mathbf{d}}^{\text{EIG1}}(l) := \frac{\text{trace}\{\mathbf{\Phi}_{\mathbf{y}}^{\text{w}}(l)\} - \lambda_1(l)}{M-1}} \tag{2.43}$$

In the case that all assumptions made in this model are perfectly valid, it is clear that both diffuse PSD estimates are equal, i.e., $\phi_{\text{d}}^{\text{EIG1}}(l) = \phi_{\text{d}}^{\text{EIG2}}(l)$. In practice, however, the model described in Equation 2.37 does not perfectly hold, since the rank-1 term describes a point source, which does not exist in practice, and the assumption of a diffuse noise field is merely an approximation. Thus, the prewhitened diffuse PSD matrix in Equation 2.38 *cannot* be written as a scaled identity matrix, and Equation 2.40 loses its exactness.

Note that the EVD-based PSD estimator does not yield the speech PSD $\phi_{\mathbf{s}}(l)$ by itself. Instead, the decision-directed approach, which is described in Section 2.3.3, is used to obtain a corresponding estimate.
Furthermore, in Algorithm 3.2, a computationally more efficient variant of the EVD-based PSD estimator is proposed, which does not rely on computing a full EVD in each time-frequency bin.

2.4.3 Joint Diffuse and Target PSD Estimation

As an alternative to the EVD-based method described in Section 2.4.2, the least squares (LS)-based method proposed in [15], which provides an estimate of both $\phi_s(l)$ and $\phi_d(l)$ at the cost of requiring an estimate of the RETFs, is to be presented in this section. Starting from the model discussed above, i.e.,

$$\begin{cases} \mathbf{y}(l) & = \mathbf{x}(l) + \mathbf{d}(l) + \underbrace{\mathbf{v}(l)}_{=\mathbf{0}} \\ \mathbf{\Phi}_{\mathbf{y}}(l) & = \underbrace{\phi_s(l)\mathbf{a}(l)\mathbf{a}^H(l)}_{\mathbf{\Phi}_{\mathbf{x}}(l)} + \underbrace{\phi_{\mathbf{d}}(l)\mathbf{\Gamma}}_{\mathbf{\Phi}_{\mathbf{d}}(l)} + \underbrace{\mathbf{\Phi}_{\mathbf{v}}(l)}_{=\mathbf{0}}, \end{cases} \tag{2.44}$$

a cost function $J(l)$ is defined in order to fit the observed data $\hat{\mathbf{\Phi}}_{\mathbf{y}}(l)$ to the model as

$$J(l) = \left\| \hat{\mathbf{\Phi}}_{\mathbf{y}}(l) - \left(\phi_s(l)\mathbf{a}(l)\mathbf{a}^H(l) + \phi_{\mathbf{d}}(l)\mathbf{\Gamma}\right) \right\|_F^2, \tag{2.45}$$

where $||\mathbf{A}||_F$ is the Frobenius norm of $\mathbf{A} \in \mathbb{C}^{M \times M}$, defined as

$$||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{M} |a_{ij}|^2}. \tag{2.46}$$

The above cost function $J(l)$ provides a set of M^2 equations, $M(M+1)/2$ of which are independent due to the symmetry of the considered matrices, with only 2 unknowns $\phi_s(l)$ and $\phi_{\mathbf{d}}(l)$. Hence, for

$$\frac{M(M+1)}{2} \geq 2, \quad M \in \mathbb{N} \quad \rightarrow \quad M > 2, \tag{2.47}$$

i.e., for a microphone array comprising more than 2 microphones, the equation can be solved in a least squares sense. Stacking the unknowns in a vector and redefining as

$$\boldsymbol{\phi}(l) := \begin{bmatrix} \phi_s(l) \\ \phi_{\mathbf{d}}(l) \end{bmatrix}, \mathbf{A}(l) := \begin{bmatrix} \left(\mathbf{a}(l)^H \mathbf{a}(l)\right)^2 & \mathbf{a}^H(l)\boldsymbol{\Gamma}\mathbf{a}(l) \\ \mathbf{a}^H(l)\boldsymbol{\Gamma}\mathbf{a}(l) & \text{trace}\{\boldsymbol{\Gamma}^H\boldsymbol{\Gamma}\} \end{bmatrix}, \mathbf{b}(l) := \begin{bmatrix} \text{Re}\left\{\mathbf{a}^H(l)\hat{\boldsymbol{\Phi}}_{\mathbf{y}}(l)\mathbf{a}(l)\right\} \\ \text{Re}\{\text{trace}\{\boldsymbol{\Phi}_{\mathbf{y}}\boldsymbol{\Gamma}^H\}\} \end{bmatrix}, \tag{2.48}$$

the estimates $\hat{\phi}_s(l)$ and $\hat{\phi}_{\mathbf{d}}(l)$ are obtained by minimizing $J(l)$, which is rewritten using $\mathbf{A}(l)$, $\mathbf{b}(l)$, $\boldsymbol{\phi}(l)$ as

$$\begin{aligned} \hat{\boldsymbol{\phi}}(l) &= \underset{\boldsymbol{\phi}}{\text{argmin}}\, J(l) \\ &= \underset{\boldsymbol{\phi}}{\text{argmin}} \left(\boldsymbol{\phi}^T(l)\mathbf{A}(l)\boldsymbol{\phi}(l) - 2\mathbf{b}^T(l)\boldsymbol{\phi}(l) + C\right), \end{aligned} \tag{2.49}$$

with C an independent constant. Setting the derivative w.r.t. $\boldsymbol{\phi}$ to 0 yields the solution

$$\boxed{\hat{\boldsymbol{\phi}}(l) = \mathbf{A}^{-1}(l)\mathbf{b}(l).} \tag{2.50}$$

2.5 Relative Early Transfer Function Estimation

Assuming that reverberation and ambient noise can be modeled as diffuse sound fields, the implementation of the MVDR beamformer and the Wiener postfilter require, in addition to the diffuse PSD, an estimate of the RETF vector of the target speaker. In the past, several RETF vector estimation procedures have been proposed, e.g., based on the least squares method [30, 11], the covariance subtraction method [10, 9, 31], or the covariance whitening (CW) method [9, 32, 8].

The CW method [8] for estimating the (time-varying) RETF vector is briefly outlined in the following. For a more detailed derivation, it is referred to the literature.

Consider again the decomposition of the prewhitened microphone PSD matrix in Equation 2.38:

$$\begin{aligned} \boldsymbol{\Phi}_{\mathbf{y}}^{\text{w}}(l) &= \phi_s(l) \underbrace{\mathbf{L}^{-1}\mathbf{a}(l)}_{=:\mathbf{b}(l)} \underbrace{\mathbf{a}^H(l)\mathbf{L}^{-H}}_{=\mathbf{b}^H(l)} + \phi_{\mathbf{d}}(l)\mathbf{L}^{-1}\boldsymbol{\Gamma}\mathbf{L}^{-H} \\ &= \phi_s(l)\mathbf{b}(l)\mathbf{b}^H(l) + \phi_{\mathbf{d}}(l)\mathbf{I}_M, \end{aligned}$$

where it can be seen that the rank-1 term is constructed as the outer product of a transformed version of the RETF vector $\mathbf{a}(l)$ with itself. Computing the EVD of $\boldsymbol{\Phi}_{\mathbf{y}}^{w}(l)$, as is required for the diffuse PSD estimation described in Section 2.4.2, also yields its eigenvectors, which hence do not need to be computed in an extra step. A simple calculation (cf. Theorem A.2.4) shows

that the dominant eigenvector indeed corresponds to the vector $\mathbf{b}(l)$, from which the RETF vector $\mathbf{a}(l)$ may be extracted through inverse transformation and normalization w.r.t. its first component as

$$\boxed{\hat{\mathbf{a}}_{\text{CW}}(l) = \frac{\mathbf{L}\mathbf{b}(l)}{\mathbf{e}_1^T \mathbf{L}\mathbf{b}(l)},} \tag{2.51}$$

where $\mathbf{e}_1$ is a selection vector with its first component[1] equal to one and the others equal to zero.

2.6 Performance Measures

In this section, the instrumental performance measures used for the evaluation of the considered speech enhancement procedures are briefly presented. Note that to evaluate the effectiveness of a technique, *improvements* in performance measures are computed, corresponding to the difference of the measures *before* and *after* processing. All of the considered measures are *intrusive*, i.e., they require a reference signal which the target signal is compared to. In the scope of this work, the reference is given by the target component $S(l)$.

2.6.1 Perceptual Evaluation of Speech Quality

Being the ITU-T-recommended method for predicting subjective quality of speech signals (P.862) [33], the perceptual evaluation of speech quality (PESQ) measure is widely used in the speech enhancement community. Like the measures described in the following sections, PESQ takes into account the human perception of speech. To achieve this, first both signals are brought into a representation that is motivated by human perception. The obtained representations of the reference and the target signal are then used to compute difference measures, which are mapped to a single number in $[-0.5,\, 4.5]$, although values in $[1.0,\, 4.5]$ result in typical situations. A higher PESQ-value corresponds to a higher perceptual speech quality.

2.6.2 Frequency-Weighted Segmental SNR

In signal processing, the signal-to-noise ratio (SNR) is a commonly-used measure, in which the energy level of a signal is compared to that of its noise background. There are several variants, only one of which is used in the scope of this work, since it incorporates information on human perception, i.e., the frequency-weighted segmental SNR (fwsSNR) [34]. This measure yields a single number, which is computed as

$$\text{fwsSNR} = \frac{10}{K} \sum_{k=1}^{K} \frac{\sum_{l=1}^{L} W(k,l) \log_{10}\left(\frac{|S(k,l)|^2}{(|S(k,l)| - |\hat{S}(k,l)|)^2}\right)}{\sum_{l=1}^{L} W(k,l)}, \tag{2.52}$$

in which $W(k,l)$ gives the weight of the (k,l)-th time-frequency bin, L corresponds to the number of time frames, and K gives the number of frequency bins. As a more qualitative description, the measure involves the following steps:

(i) For the STFT, a Gaussian window is used [34].

(ii) The windowed spectra are distributed into 25 frequency bands, whose spacing is determined by human perception.

[1] The selection vector is constructed based on the choice of the reference microphone.

(iii) The windowed, distributed spectra are normalized such that the sum over the absolute values in all frequency bands equals one, followed by summing the absolute values in each band over the time frames.

In [34], the weighting function $W(k,l) = |X(k,l)|^\gamma$ is proposed, with γ chosen so as to achieve maximum correlation between this measure and subjective experiments. Furthermore, to ensure that the resulting value is not biased due to noise-only frames or no-noise frames, the individual summands $\log_{10}\left(\frac{|X(k,l)|^2}{(|X(k,l)|-|\hat{X}(k,l)|)^2}\right)$ are constrained to be within $[\mathrm{SNR}_{min}, \mathrm{SNR}_{max}]$, with, e.g., $\mathrm{SNR}_{min} = -10\,\mathrm{dB}$ and $\mathrm{SNR}_{max} = 35\,\mathrm{dB}$ [35].
These combined modifications to the standard SNR make the fwsSNR a more suitable tool for the evaluation of perceptual speech quality, as demonstrated in [34], where a high correlation between perceived signal distortion and the fwsSNR is shown. The implementation used in this work is the one distributed by the REVERB challenge organizers [36].

2.6.3 Cepstral Distance

To understand the concept of the cepstral distance (CD), and how it relates to human perception, firstly the cepstrum itself is briefly described [38]. To be exact, the *power cepstrum* $c[n]$ of a signal $y[n]$ is discussed, since it is commonly used in audio processing:

$$c[n] := \mathrm{IDFT}\{\log|\mathrm{DFT}\{y[n]\}|^2\}, \quad (2.53)$$

with (I)DFT$\{\cdot\}$ denoting the (inverse) discrete Fourier transform (cf. section 2.2). Clearly, any phase information is lost due to the absolute-value operation.
To motivate this definition, recall the filtered source component in the signal model in Equation 2.1 and apply the widely-known convolution theorem to it:

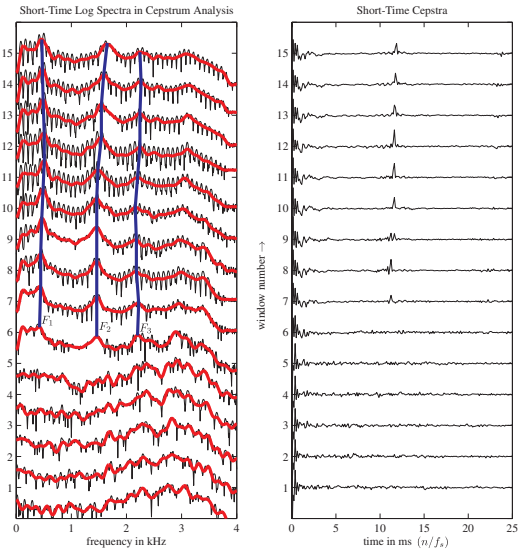


Fig. 2.3.: visualization of cepstrum, from [37]

$$\begin{aligned}
\underbrace{\mathrm{DFT}\{y[n]\}}_{=:Y(w)} = \mathrm{DFT}\{h[n] * s[n]\} &= \underbrace{\mathrm{DFT}\{h[n]\}}_{=:H(w)} \cdot \underbrace{\mathrm{DFT}\{s[n]\}}_{=:S(w)} \\
\Leftrightarrow Y(\omega) &= H(\omega)\cdot S(\omega) \\
\Rightarrow |Y(\omega)|^2 &= |H(\omega)\cdot S(\omega)|^2 \\
\Leftrightarrow \log|Y(\omega)|^2 &= \log|H(\omega)\cdot S(\omega)|^2 \\
\Leftrightarrow 2\log|Y(\omega)| &= 2\log|H(\omega)\cdot S(\omega)| \\
\Leftrightarrow \log|Y(\omega)| &= \log(|H(\omega)|) + \log(|S(\omega)|) \\
\Rightarrow \underbrace{\mathrm{IDFT}\{\log|Y(\omega)|\}}_{=:\tilde{y}[n]} &= \underbrace{\mathrm{IDFT}\{\log(|H(\omega)|)\}}_{=:\tilde{h}[n]} + \underbrace{\mathrm{IDFT}\{\log(|S(\omega)|)\}}_{=:\tilde{s}[n]} \\
\Leftrightarrow \tilde{y}[n] &= \tilde{h}[n] + \tilde{s}[n]
\end{aligned} \quad (2.54)$$

It can be seen that computing the cepstrum resembles a *deconvolution operation*, where the cepstra of the convolved sequences $h[n]$ and $s[n]$ are *added*. If the frequency ranges of

both log-spectra are well-distinguished, deconvolution can easily be performed by linear separation, e.g., through *liftering*. To explain the attractiveness of CD in terms of being a perceptually viable quality measure, consider Figure 2.3. Signals containing voiced speech exhibit approximately periodic waveform repetitions leading to local maxima in the corresponding log-spectra, as is highlighted in the left part of the figure. The cepstrum responds to these "periodic ripples" [39] with a peak as seen in the right part. These peaks can hence be used for the identification of frames including voiced speech, and as such represent information which is relevant to human perception.

To motivate the definition of the CD, consider two sequences $s_1[n]$ and $s_2[n]$ with cepstra $\tilde{s}_1[n]$ and $\tilde{s}_1[n]$, which are both convolved with the sequence $h[n]$, resulting in $y_i[n] = h[n] * s_i[n],\ i = \{1, 2\}$. Remembering the last line in Equation 2.54, an intuitive explanation of the CD can be given by noting that the difference between both cepstra, i.e., $\tilde{y}_1[n] - \tilde{y}_2[n] = \tilde{s}_1[n] - \tilde{s}_2[n]$, is described without taking the transfer function $h[n]$ into account. This leads to the definition of the (Euclidian) CD

$$d_{\text{cep}} = \sum_{n=1}^{N} \left(c_{\text{tar}}[n] - c_{\text{ref}}[n]\right)^2, \tag{2.55}$$

with c_{tar} and c_{ref} representing the cepstral coefficients of the target and the reference signal, respectively. A larger value corresponds to a larger distortion of the target signal w.r.t. the reference, usually constrained to satisfy $d_{\text{cep}} \in [0\,\text{dB}, 10\,\text{dB}]$

2.6.4 Log-Likelihood Ratio

As a fourth performance measure, the log-likelihood ratio (LLR) is used in this work. Being a linear predictive coding (LPC)-based measure, first the LPC vectors $\mathbf{a}_{\text{ref}}$ and $\mathbf{a}_{\text{tar}}$ of both the reference and the enhanced signal are computed. Afterwards, the difference between those vectors is computed as [34]

$$d_{\text{LLR}}(\mathbf{a}_{\text{ref}}, \mathbf{a}_{\text{tar}}) = \log\left(\frac{\mathbf{a}_{\text{ref}}\mathbf{\Phi}_{\text{ref}}\mathbf{a}_{\text{tar}}^T}{\mathbf{a}_{\text{tar}}\mathbf{\Phi}_{\text{ref}}\mathbf{a}_{\text{ref}}^T}\right) \tag{2.56}$$

with the target autocorrelation matrix $\mathbf{\Phi}_{\text{ref}}$. Since LPC is not otherwise a part of this work, it is referred to the literature for a more detailed description.

Remark on Performance Measures The measures above were chosen because they are the objective measures[1] that correlate the most with subjective “overall quality”, “signal distortion” and “background distortion” as reported in [34]. In this context, it needs to be said that performance evaluations relying on those or other methods should still be treated with caution, since no measure is “perfect” in the sense that it correlates with human perception in all scenarios.

[1] I.e., not including hybrid approaches, in which multiple objective performance measures are combined.

3 Contributions

In this chapter, the original contributions of this thesis are presented. First, an alternating least squares (ALS) approach to jointly estimate the RETF vector and the diffuse and target PSDs based on the minimization of a model-derived cost function is proposed. Second, a computationally efficient variant of the EVD-based diffuse PSD estimator is presented.

3.1 Frobenius Norm-Based Joint Estimation of RETF Vector and Diffuse PSD

Although the diffuse PSD estimator described in Section 2.4.2 does not directly depend on an estimate of the RETF vector $\mathbf{a}(l)$, the preceding MVDR beamformer, which is a crucial part of the MWF, *does*. This means that in practice, estimating $\mathbf{a}(l)$ is not circumvented by relying on the EVD-based PSD estimator, since it is usually incorporated in an MWF framework for larger effectiveness.

3.1.1 Coupling the RETF Vector and the Diffuse PSD Estimation

Many RETF vector and PSD estimation methods are decoupled, i.e., a) the RETF vector is estimated either assuming that the diffuse PSD is known [11] or without requiring knowledge of the diffuse PSD [30, 10, 8, 9, 31, 32]; b) the diffuse PSD is estimated either assuming that the RETF vector is known [12, 13, 40, 15] or without requiring knowledge of the RETF vector [16]. In [17] it has been shown that jointly estimating both the RETF vector and the diffuse PSD based on CW results in a high dereverberation and noise reduction performance.

As an extension of the Frobenius norm-based PSD estimator in [15] (cf. Section 2.4.3), in this section a method to jointly estimate the (time-varying) RETF vector and diffuse PSD by minimizing the Frobenius norm of an error matrix constructed from the presumed signal model is described. Since no closed-form solution exists for the RETF vector *and* the diffuse PSD to the best of our knowledge, the minimization is performed in an iterative fashion using an ALS approach. By coupling the RETF vector and PSD estimation procedures, they are expected to yield estimates which fit the signal model in Equation 2.37 more accurately.

To arrive at an iterative procedure which aims at improving the accuracy of the estimates $\hat{\phi}_s(l)$, $\hat{\phi}_d(l)$ and $\hat{\mathbf{a}}(l)$, it is necessary to evaluate the synergy of the utilized methods, i.e., whether or not the estimates are interdependent. Specifically, the considered systems comprise

(i) a method estimating $\phi_{\mathbf{d}}(l)$ and $\phi_s(l)$ based on the estimate $\hat{\mathbf{a}}(l)$.

(ii) a method estimating the RETF vector $\mathbf{a}(l)$ based on the estimates $\hat{\phi}_d(l)$ and $\hat{\phi}_s(l)$.

Thus, an improvement at the output of *one method* may result in an improvement at the output of *the other method* for each iteration, which can be considered an ALS approach.

For this purpose, the following combinations are used:

OPT Frobenius norm-based diffuse PSD estimation (Section 2.4.3) and Frobenius norm-based RETF vector estimation using *unconstrained optimization* (Section 3.1.2)

LS Frobenius norm-based diffuse PSD estimation (Section 2.4.3) and Frobenius norm-based RETF vector estimation using *rank-1 approximation* (Equation 3.6)

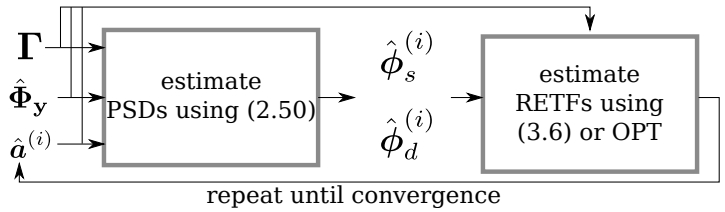


Fig. 3.1.: schematics of OPT and LS method

These combinations are depicted in Figure 3.1, where the only difference is given by the implementation of the second block.

As a baseline approach, the EVD-based diffuse PSD estimator (cf. Section 2.4.2) is used in conjunction with the CW-based RETF vector estimator (cf. Section 2.5), both of which make use of the eigenvalue decomposition of the prewhitened estimated microphone PSD matrix $\hat{\mathbf{\Phi}}_{\mathbf{y}}(l)$. In [17] it was shown that this combination leads to a high dereverberation and denoising performance, especially when using the mean of the last $M-1$ eigenvalues for the diffuse PSD estimate (i.e., ϕ_d^{EIG1}, cf. Equation 2.41).

Simulation results, in which the performance of the considered methods is compared in terms of artificial and real data, are presented in Section 4.2.1.

3.1.2 Frobenius Norm-Based RETF Vector Estimation

Quite intuitively, and in analogy to the Frobenius norm-based PSD estimator mentioned in Section 2.4.3, the RETF estimator in this section is based on the minimization of the cost function $J(l)$ in Equation 2.45, which is a measure of the difference between the observed data and the model in Equation 2.44. Hence, the estimate is given by

$$\begin{aligned}\hat{\mathbf{a}} &= \underset{\mathbf{a}}{\operatorname{argmin}}\, J \\ &= \underset{\mathbf{a}}{\operatorname{argmin}}\, || \underbrace{\mathbf{\Phi}_{\mathbf{y}} - \phi_{\mathbf{d}}\mathbf{\Gamma}}_{=:\mathbf{A}} - \phi_s \mathbf{a}\mathbf{a}^H ||_F^2 \\ &= \underset{\mathbf{a}}{\operatorname{argmin}}\, \operatorname{trace}\left\{ (\mathbf{A} - \phi_s \mathbf{a}\mathbf{a}^H)^H (\mathbf{A} - \phi_s \mathbf{a}\mathbf{a}^H) \right\},\end{aligned} \tag{3.1}$$

where Theorem A.2.2 is used in the last step and the time index l is omitted for brevity.

Unconstrained Optimization

Since an analytical solution of this problem is rather hard to find (in case that one does exist), a numerical unconstrained minimization procedure is utilized based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [41]. As a member of the quasi-Newton methods, the

BFGS algorithm relies on the gradient of $J(l)$ to be zero, where the used implementation either accepts a gradient function, or it estimates the gradient numerically. Thus, to lower the chance of converging to a local optimum due to gradient inaccuracies, the analytical gradient is computed in the following.

Since the BFGS implementation expects real-valued inputs and outputs, it is necessary to separate the involved quantities into their respective imaginary and real parts, denoted with superscripts $\circ^R$ and $\circ^I$ for brevity. For the Frobenius norm, the possibility of separating the real and the imaginary part, which is applied in deriving the gradient, is shown in Theorem A.2.3. In this fashion, also the gradient itself is separated, resulting in the four following terms contributing to the full gradient:

$$\begin{aligned}
\nabla_{\mathbf{a}^R}||\mathrm{Re}\{J\}||_F^2 &= -4\phi_s\mathbf{A}^R\mathbf{a}^R+4\phi_s^2(\mathbf{a}^R\mathbf{a}^{R,T}\mathbf{a}^R - \mathbf{a}^I\mathbf{a}^{I,T}\mathbf{a}^R)\\
\nabla_{\mathbf{a}^I}||\mathrm{Re}\{J\}||_F^2 &= 4\phi_s\mathbf{A}^R\mathbf{a}^I \quad -4\phi_s^2(\mathbf{a}^R\mathbf{a}^{R,T}\mathbf{a}^I - \mathbf{a}^I\mathbf{a}^{I,T}\mathbf{a}^I)\\
\nabla_{\mathbf{a}^R}||\mathrm{Im}\{J\}||_F^2 &= 4\phi_s\mathbf{A}^I\mathbf{a}^I \quad -4\phi_s^2(\mathbf{a}^I\mathbf{a}^{R,T}\mathbf{a}^I - \mathbf{a}^R\mathbf{a}^{I,T}\mathbf{a}^I)\\
\nabla_{\mathbf{a}^I}||\mathrm{Im}\{J\}||_F^2 &= -4\phi_s\mathbf{A}^I\mathbf{a}^R +4\phi_s^2(\mathbf{a}^I\mathbf{a}^{R,T}\mathbf{a}^R - \mathbf{a}^R\mathbf{a}^{I,T}\mathbf{a}^R),
\end{aligned}$$

yielding

$$\nabla_{\mathbf{a}}||J||_F^2 = \nabla_{\mathbf{a}^R}\left(||\mathrm{Re}\{J\}||_F^2 + ||\mathrm{Im}\{\mathrm{J}\}||_\mathrm{F}^2\right) + j\nabla_{\mathbf{a}^I}\left(||\mathrm{Re}\{J\}||_F^2 + ||\mathrm{Im}\{J\}||_F^2\right), \tag{3.2}$$

where the linearity of the gradient operator is exploited. As a starting vector, a vector with normally distributed complex-valued components (except for its first component, which is equal to 1 by definition) is chosen. The successful[1] optimization result is then normalized w.r.t. its first component and accepted as the current estimate $\hat{\mathbf{a}}_{\mathrm{OPT}}$. If, on the other hand, the optimization fails, the estimate of the last frame is used instead, which is assumed to be at least equally fitting as the random starting vector.

Rank-1 Approximation

Another possible route to solving the minimization problem in Equation 3.1 is seeing it as a rank-1 approximation problem. Thus, the problem becomes finding a rank-1 matrix $\tilde{\mathbf{B}}$ that solves the problem

$$\tilde{\mathbf{B}} = \underset{\mathbf{B}}{\mathrm{argmin}}\|\mathbf{A} - \mathbf{B}\|_F^2. \tag{3.3}$$

The solution to this problem in the MMSE sense is given in [42], where it is shown to correspond to a part of its singular value decomposition (SVD)

$$\mathbf{A} \approx \tilde{\mathbf{B}} = \sigma_1\mathbf{t}_1\mathbf{v}_1, \tag{3.4}$$

where σ_1 is its first singular value, and $\mathbf{t}_1$ as well as $\mathbf{v}_1$ are its first left-singular and right-singular vectors, respectively. Considering that $\mathbf{A}$ is Hermitian in this application, and assuming that it is also positive-definite (which is not always the case in practice, possibly leading to estimation errors), the SVD of $\mathbf{A}$ merges with its EVD, such that

$$\mathbf{A} \approx \tilde{\mathbf{B}} = \sigma_1\mathbf{t}_1\mathbf{v}_1 = \lambda_1\mathbf{u}_1\mathbf{u}_1^H, \tag{3.5}$$

[1] "Successful" in this case denotes that the 2-norm of the gradient is below some limit, which, in the computations within this work, is chosen to be the square root of machine precision.

with λ_1 ($\in \mathbb{R}$, cf. Theorem A.2.2) and $\mathbf{u}_1$ the first eigenvalue and eigenvector of $\mathbf{A}$. Hence, by comparing Equation 3.1 and Equation 3.5, we find $\lambda_1 \mathbf{u}_1 \mathbf{u}_1^H \triangleq \phi_s \mathbf{a}\mathbf{a}^H$, and thus the first eigenvector is a scaled version of the RETF vector:

$$\boxed{\hat{\mathbf{a}}_{\text{LS}} = \sqrt{\frac{\lambda_1}{\hat{\phi}_s}}\mathbf{u}_1} \tag{3.6}$$

The full algorithm utilizing this estimate is presented in the following, where the RETF vector initialization is done using normally distributed complex-valued components, and 1 as the first component.

Algorithm 3.1: ALS approach to jointly estimate the RETF vector and PSDs

Input: $\mathbf{\Gamma}(k),\ \hat{\mathbf{\Phi}}_{\mathbf{y}}(k,\ l),$ num. iterations $N,$ init. $\hat{\mathbf{a}}^{(1)}(k, 1)$
Output: $\hat{\mathbf{a}}(k,\ l),\ \hat{\boldsymbol{\phi}}_{\text{LS}} = [\hat{\phi}_{s,\text{LS}},\ \hat{\phi}_{d,\text{LS}}]^T$
for *all* k **do**
  for *all* l **do**
    for $i = 1 : N$ **do**
      compute $\mathbf{A}^{(i)}(k,\ l)$ and $\mathbf{b}^{(i)}(k,\ l)$; // (2.48)
      $\hat{\boldsymbol{\phi}}^{(i)}(k,\ l) = \left(\mathbf{A}^{(i)}\right)^{-1}(k,\ l)\mathbf{b}^{(i)}(k,\ l)$; // (2.50)
      constrain $\hat{\boldsymbol{\phi}}^{(i)}(k,\ l)$; // (4.2)
      $\hat{\mathbf{\Phi}}_{\mathbf{x}}^{(i)}(k,\ l) = \hat{\mathbf{\Phi}}_{\mathbf{y}}(k,\ l) - \hat{\phi}_d^{(i)}(k,\ l)\mathbf{\Gamma}(k)$
      $\hat{\mathbf{\Phi}}_{\mathbf{x}}^{(i)}(k,\ l) = \mathbf{U}^{(i)}(k,\ l)\mathbf{\Lambda}^{(i)}(k,\ l)\mathbf{U}^{(i),H}(k,\ l)$; // EVD
      $\hat{\mathbf{a}}^{(i)}(k,\ l) = \sqrt{\lambda_1^{(i)}(k,\ l)/\hat{\phi}_s^{(i)}(k,\ l)}\mathbf{u}_1^{(i)}(k,\ l)$; // (3.6)
    $\hat{\mathbf{a}}^{(1)}(k, l+1) = \hat{\mathbf{a}}^{(N)}(k,\ l)/(\mathbf{e}^T\hat{\mathbf{a}}^{(N)}(k,\ l))$; // for next frame

3.2 Complexity Reduction of EVD-Based Diffuse PSD Estimator

The diffuse PSD estimator discussed in Section 2.4 relies on the eigenvalues of an $M \times M$-dimensional Hermitian matrix $\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}$. Since the number of sensors in a microphone array M may be large, a closed-form solution as discussed in Section 3.2.4 is not viable in practice, and approximative schemes need to be applied. We choose the first one that is presented here — termed "power method" — to be the most suitable method as discussed in Section 4.1.2. To list alternatives and their drawbacks when compared to the power method in the considered application, additionally some of the most common methods for the determination of the eigenvalues (and eigenvectors) of $\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}$ are discussed, i.e., the Arnoldi iteration and the QR method.

3.2.1 Power Method

One of the first — and simplest — numerical eigenvector/eigenvalue approximation algorithms is the *power method*, which is explained briefly in this section. In its most basic form, it may be utilized to find the *dominant* eigenvalue, i.e., the eigenvalue λ_1 for which

$$|\lambda_1| > |\lambda_i| \quad \forall i \in \{2, ..., M\}. \tag{3.7}$$

It is instructive, however, to begin with the Rayleigh quotient.

Rayleigh Quotient

To obtain the eigenvalue corresponding to the dominant eigenvector acquired in the next section, the Rayleigh quotient may be utilized [43].

Theorem 3.2.1. *If* $\mathbf{u}$ *is an eigenvector of* $\mathbf{A}$*, its associated eigenvalue* λ *is given by the Rayleigh quotient*

$$\lambda = \frac{\mathbf{u}^H \mathbf{A}\mathbf{u}}{\mathbf{u}^H \mathbf{u}}.$$

Proof. $\mathbf{A}$ has eigenvector $\mathbf{u}$ associated with eigenvalue λ, hence

$$\begin{aligned} \mathbf{A}\mathbf{u} &= \lambda\mathbf{u} \leftrightarrow \\ \mathbf{u}^H\mathbf{A}\mathbf{u} &= \mathbf{u}^H\lambda\mathbf{u} \leftrightarrow \\ \frac{\mathbf{u}^H\mathbf{A}\mathbf{u}}{\mathbf{u}^H\mathbf{u}} &= \lambda\frac{\mathbf{u}^H\mathbf{u}}{\mathbf{u}^H\mathbf{u}} = \lambda. \end{aligned}$$

□

Algorithm

The power method is implemented as shown in Algorithm 3.2. At the start of every iteration, first the starting vector $\mathbf{u}^{(0)}$ is initialized. In this work, the following initialization schemes are considered:

(i) complex-valued normally distributed components

(ii) deterministic components, i.e., $\mathbf{u}^{(0)} = [1,\ 0,\ \ldots,\ 0]^T$

(iii) the estimated dominant eigenvector from the previous time frame

Afterwards, $\mathbf{u}^{(0)}$ is multiplied from the left with the input matrix $\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)$, resulting in the transformed vector $\mathbf{t}^{(0)}$. The current eigenvalue estimate is then obtained by computing the Rayleigh quotient using a normalized version of $\mathbf{t}$ and the input matrix $\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)$. This process is repeated with the transformed vector as the starting vector of the next iteration, i.e., $\mathbf{u}^{(1)} = \frac{\mathbf{t}^{(0)}}{\left\|\mathbf{t}^{(0)}\right\|_2}$, until a convergence criterion is reached, which is chosen to be a fixed number of iterations N (cf. Section 4.1.4).

Algorithm 3.2: power method for computing the first P largest eigenvalues

In: $\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l) \in \mathbb{C}^{M\times M}$, number N of iterations, number P of eigenvalues to be estimated
Out: P eigenvalue estimates $\left\{\hat{\lambda}_1\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)\},\ \ldots\ \hat{\lambda}_P\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)\}\right\}$
for $p = 1$ **to** P **do**
  initialize $\mathbf{u}_p^{(0)} \in \mathbb{C}^M$;
  for $n = 1$ **to** N **do**
    $\mathbf{t} = \hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)\mathbf{u}_p^{(n-1)}$;
    $\mathbf{u}_p^{(n)} = \mathbf{t}/||\mathbf{t}||_2$; `// normalization`
    $\lambda_p^{(n)} = \mathbf{u}_p^{(n),H}\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)\mathbf{u}_p^{(n)}$; `// Rayleigh quotient`
  $\hat{\lambda}_p\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)\} = \lambda_p^{(N)}$; `// matrix rank reduction`
  $\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l) = \hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l) - \hat{\lambda}_p\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)\}\mathbf{u}_p^{(N)}\mathbf{u}_p^{(N),H}$;

The Rayleigh quotient reduces to $\mathbf{u}_p^{(n),H}\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)\mathbf{u}_p^{(n)}$ due to the fact that $\mathbf{u}_p^{(n)}$ is normalized and, as such, the denominator is always equal to 1. For $P > 1$, Algorithm 3.2 includes

an extension that allows obtaining more than the dominant eigenvalue, which is based on Wielandt deflation [44]. Once the largest eigenvalue has been computed, the column space of the input matrix $\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)$ is reduced using the corresponding estimated eigenvector and the same iterations as for computing the largest eigenvalue are repeated. This process can be continued to obtain all M eigenpairs of $\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)$ as shown in Algorithm 3.2. One has to note, however, that this algorithm has limitations that become apparent especially when not focusing on the dominant eigenpair, as is shown in the next section.

Convergence Analysis

In this section, the convergence properties of the power method are discussed, and Wielandt deflation is proven to be a means of gaining access to the inner eigenvalues.

First Eigenpair For this convergence proof [43], it is assumed that the input matrix is not defective and hence that it has M linearly independent eigenvectors. Thus, this set of eigenvectors spans $\mathbb{R}^M$. By choosing a vector $\mathbf{b}^{(0)} \in \mathbb{R}^M$ with normally distributed real-valued components, it can be expressed as a linear combination of the eigenvectors $\mathbf{x}_m$ (sorted according to the absolute values of their associated eigenvalues) as

$$\mathbf{b}^{(0)} = \sum_{m=1}^{M} c_m \mathbf{x_m} \tag{3.8}$$

with some coefficients c_m. Multiplying from the left with $\mathbf{A}$ yields

$$\begin{aligned} \mathbf{A}\mathbf{b}^{(0)} &= \mathbf{A} \sum_{m=1}^{M} c_m \mathbf{x_m} = \sum_{m=1}^{M} c_m \lambda_m \mathbf{x_m} \\ \rightarrow \mathbf{A}^n\mathbf{b}^{(0)} &= \sum_{m=1}^{M} c_m \lambda_m^n \mathbf{x_m} \\ &= \lambda_1^n \sum_{m=1}^{M} c_m \left(\frac{\lambda_m}{\lambda_1}\right)^n \mathbf{x_m}. \end{aligned} \tag{3.9}$$

As a side note, the span[1] of the set of vectors $\left\{\mathbf{A}^0\mathbf{b}^{(0)}, \mathbf{A}^1\mathbf{b}^{(0)}, \ldots, \mathbf{A}^N\mathbf{b}^{(0)}\right\}$ is called the Krylov subspace $\mathcal{K}_{N+1}(\mathbf{A}, \mathbf{b}^{(0)})$, which reappears in the Arnoldi iteration in Section 3.2.3.

By increasing the number of maximum iterations N and considering that the algorithm has its benefit in finding the *dominant* eigenpair, the fractions $\left\{\left(\frac{\lambda_m}{\lambda_1}\right)^n\right\}$, $2 \leq m \leq M$, approach zero for larger n, yielding the eigenvector estimate

$$\begin{aligned} \lim_{n\to\infty} \mathbf{A}^n\mathbf{b}^{(0)} &= \lambda_1^n \sum_{m=1}^{M} c_m \left(\frac{\lambda_m}{\lambda_1}\right)^n \mathbf{x_m} \\ &= \lambda_1^n \left[c_1 \underbrace{\left(\frac{\lambda_1}{\lambda_1}\right)^n}_{=1} \mathbf{x}_1 + \sum_{m=2}^{M} c_m \underbrace{\left(\frac{\lambda_m}{\lambda_1}\right)^n}_{\to 0} \mathbf{x_m} \right] \\ &\approx \lambda_1^n c_1 \mathbf{x_1}. \end{aligned} \tag{3.10}$$

[1] The span is the set of all possible linear combinations of the arguments, i.e., $\text{span}\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} = \left\{\sum_{n=1}^{N} a_n \mathbf{x}_n | a_n \in \mathbb{C}, \mathbf{x}_n \in \mathbb{C}^M\right\}$.

Note that a multiple of an eigenvector is still an eigenvector. This also shows that when $|\lambda_1| \gtrapprox |\lambda_2|$, convergence may be slow, especially when one aims at finding *all* eigenpairs. Furthermore, the algorithm fails if there is *no* dominant eigenvalue, which is assumed not to be the case in the application within this work.

Wielandt Deflation This proof is developed based on the convergence properties in the previous section. In Wielandt deflation [44], the input matrix $\mathbf{A}$ is modified such that the application of additional power iterations on it yields the subsequent eigenvalue of $\mathbf{A}$. Assume that the non-defective, Hermitian matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ has eigenvalues λ_i and eigenvectors $\mathbf{x}_i, \quad ||\mathbf{x}_i||_2 = 1, \quad 1 \leq i \leq M$, sorted according to the magnitude of the corresponding eigenvalue. Hence, any vector $\mathbf{b}$ can be written as a linear combination of said eigenvectors (cf. Equation 3.8). Multiplying from the left with $(\mathbf{A} - \lambda_1 \mathbf{x}\mathbf{x}_1^H)$ yields

$$\begin{aligned}
(\mathbf{A} - \lambda_1 \mathbf{x}\mathbf{x}_1^H)\mathbf{b} &= (\mathbf{A} - \lambda_1 \mathbf{x}\mathbf{x}_1^H) \sum_{m=1}^{M} c_m \mathbf{x_m} \\
&= \mathbf{A} \sum_{m=1}^{M} c_m \mathbf{x_m} - \lambda_1 \mathbf{x}\mathbf{x}_1^H \sum_{m=1}^{M} c_m \mathbf{x_m} \\
&= \sum_{m=1}^{M} c_m \lambda_m \mathbf{x_m} - \lambda_1 \mathbf{x}_1 \sum_{m=1}^{M} c_m \underbrace{\mathbf{x}_1^H \mathbf{x_m}}_{=\delta_{1m}} \\
&= \sum_{m=1}^{M} c_m \lambda_m \mathbf{x_m} - c_1 \lambda_1 \mathbf{x}_1 \\
&= \sum_{m=2}^{M} c_m \lambda_m \mathbf{x_m},
\end{aligned} \tag{3.11}$$

where δ_{1m} is the Kronecker-δ, defined as

$$\delta_{ij} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j. \end{cases} \tag{3.12}$$

$\mathbf{x}_1^H \mathbf{x_m} = \delta_{1m}$ results from the fact that the considered matrices $\mathbf{A}$ are Hermitian and hence its eigenvectors are orthogonal (cf. proof in A.2.1). The result in Equation 3.11 shows that

$$\text{span}\{\mathbf{A} - \lambda_1 \mathbf{x}\mathbf{x}_1^H\} = \text{span}\{\mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_M\}, \tag{3.13}$$

such that the eigenpair corresponding to the first eigenvalue is removed, and the *second* eigenpair becomes the dominant one. Note that this derivation is based on the exactness of λ_1 and $\mathbf{x}_1$; in case of deviations, the first eigenpair is *not* completely removed, resulting in an increased error in the estimation of the *second* eigenpair.

Deterministic Initialization Since we propose to use a non-Gaussian, deterministic initialization of the vector $\mathbf{b}^{(0)}$ later in Section 4.1.4 to further reduce the computational complexity, the corresponding convergence behavior and some additional information is provided in this paragraph. Specifically, we have

$$\mathbf{b}^{(0)} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{3.14}$$

Because the eigenvectors of the input matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ do not generally correspond to the M Cartesian unit vectors, $\mathbf{b}^{(0)}$ is given as a linear combination of the eigenvectors with non-zero coefficients as in Equation 3.8, which means that the same convergence proof can be applied to this case. Investigating the steps in the power method, one small difference can be made out: In the first matrix-vector multiplication, we have

$$\left(\mathbf{A}\mathbf{b}^{(0)}\right)_i = \sum_{k=1}^{M} A_{ik} b_k = \sum_{k=1}^{M} A_{ik}\delta_{1k} = A_i, \tag{3.15}$$

such that the multiplication results in the first column of $\mathbf{A}$ and does not need execution, reducing the computational complexity by an (insignificant) amount. Since the first column of $\mathbf{A}$ in general consists of nonzero components, the deterministic and Gaussian initialization do not differ significantly from this point on.

3.2.2 QR Method

As the method of choice for the eigenvalue decomposition of a "not too large" matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ [45], the QR method, which produces a full set of eigenvectors and eigenvalues, is outlined briefly in this section. Note that, while it may be used in conjunction with other techniques to increase efficiency (cf. Arnoldi iteration in the following section), it can also be utilized on its own. It is based on the Schur decomposition of $\mathbf{A}$:

$$\mathbf{Q}^H\mathbf{A}\mathbf{Q} = \mathbf{T}, \tag{3.16}$$

which is a unitary similarity transform with $\mathbf{Q} \in \mathbb{C}^{M\times M}$ a unitary matrix (i.e., $\mathbf{Q}^H\mathbf{Q} = \mathbf{Q}\mathbf{Q}^H = \mathbf{I}_M$), and $\mathbf{T}$ an upper triangular matrix. Hence, as proven in Theorem 3.2.2, extracting the eigenvalues of $\mathbf{T}$ yields the eigenvalues of $\mathbf{A}$. This is computationally much less complex, as is demonstrated in Theorem A.2.5.

In the QR method, which is an iterative method, a sequence of unitary transformations is desired that yields the Schur form of the transformed matrix:

$$\begin{aligned}\mathbf{T}_k &= \underbrace{(\mathbf{Q}_k^H \dots \overbrace{\mathbf{Q}_1^H}^{=:\mathbf{T}_1}\mathbf{A}(\mathbf{Q}_1 \dots \mathbf{Q}_k)}_{=:\mathbf{T}_k} \\ &= \mathbf{Q}_k^H\mathbf{T}_{k-1}\mathbf{Q}_k,\end{aligned} \tag{3.17}$$

where $\mathbf{T}_k$ approaches triangular form with larger k.

Since the aim is computing the eigenvalues of $\mathbf{A}$, it is important for the second stage that the following is true:

Theorem 3.2.2. *If $\mathbf{A}$ and $\mathbf{T} \in \mathbb{C}^{M\times M}$ are unitarily similar, they have the same eigenvalues.*

Proof. Assume that $\mathbf{A}$ has eigenvalue λ and eigenvector $\mathbf{u}$. Then

$$\begin{aligned}&\mathbf{T} = \mathbf{Q}^H\mathbf{A}\mathbf{Q}, \quad \text{with } \mathbf{Q}^H\mathbf{Q} = \mathbf{Q}\mathbf{Q}^H = \mathbf{I}_M \quad \text{and } \mathbf{A}\mathbf{u} = \lambda\mathbf{u} \\ \leftrightarrow\ &\mathbf{T}\mathbf{Q}^H\mathbf{u} = \mathbf{Q}^H\mathbf{A}\underbrace{\mathbf{Q}\mathbf{Q}^H}_{=\mathbf{I}}\mathbf{u} = \mathbf{Q}^H\mathbf{A}\mathbf{u} \\ &= \mathbf{Q}^H\lambda\mathbf{u} = \lambda\underbrace{\mathbf{Q}^H\mathbf{u}}_{:=\mathbf{v}} \\ \leftrightarrow\ &\mathbf{T}\mathbf{v} = \lambda\mathbf{v}.\end{aligned}$$

□

Hence, $\mathbf{T}$ possesses the same eigenvalues and corresponding multiplicities as $\mathbf{A}$ with associated eigenvectors that are versions of the eigenvectors of $\mathbf{A}$ modified by the transform $\mathbf{Q}$ (i.e., rotated).

The unitary transformation matrices are obtained using the QR decomposition (which gives the method its name):

$$\mathbf{T}_{k-1} = \mathbf{Q}_k \mathbf{R}_k, \tag{3.18}$$

with $\mathbf{Q}_k$ a unitary and $\mathbf{R}_k$ an upper triangular matrix. The next iterate $\mathbf{A}_k$ can then be obtained by reversing the multiplication order in Equation 3.18, since (using the unitarity of $\mathbf{Q}_k$ and the definition of $\mathbf{Q}_k$ and $\mathbf{R}_k$)

$$\begin{aligned} \mathbf{T}_k &= (\mathbf{Q}_1 \dots \mathbf{Q}_k)^H \mathbf{A} (\mathbf{Q}_1 \dots \mathbf{Q}_k) \\ &= \mathbf{Q}_k^H \underbrace{\mathbf{T}_{k-1}}_{=\mathbf{Q}_k \mathbf{R}_k} \mathbf{Q}_k \\ &= \underbrace{\mathbf{Q}_k^H \mathbf{Q}_k}_{=\mathbf{I}_M} \mathbf{R}_k \mathbf{Q}_k \\ &= \mathbf{R}_k \mathbf{Q}_k, \end{aligned} \tag{3.19}$$

where, comparing with Equation 3.17, it can be seen that the current iterate $\mathbf{A}_k$ and the input $\mathbf{A}$ are unitarily similar. What remains to be shown is that the elements on the strictly lower triangle approach zero, and hence $\mathbf{A}_k$ approaches upper triangular form, for which it is referred to [43]. The implementation used in this work (in addition to the MATLAB implementation) is shown in Algorithm 3.3. Note that there is a number of ways

Algorithm 3.3: simple QR algorithm to obtain eigenvalues of matrix $\mathbf{A}$

Input: matrix $\mathbf{A} \in \mathbb{C}^{M \times M}$, number of iterations N
Output: eigenvalues λ_i of $\mathbf{A}$
init $\mathbf{A}_0 = \mathbf{A}$;
for $k = 1 : N$ **do**
    $[\mathbf{Q}, \mathbf{R}] = \mathrm{qr}\{\mathbf{A}\}$; `// QR decomposition`
    $\mathbf{A}_k = \mathbf{R}\mathbf{Q}$; `// tends to triangular form`
$\boldsymbol{\lambda} = \mathrm{diag}\{\mathbf{A}_N\}$ `// after convergence: lambda is vector of eigenvalues`

to check for convergence, e.g., the size of elements on the strict lower triangle of $\mathbf{A_k}$ or the change in the norm of the eigenvalue vector, i.e., $||\boldsymbol{\lambda}||_2$. For reasons of comparison, however, a fixed iteration number N is chosen. Furthermore, this implementation is used to demonstrate the functionality of the QR method. Variants exist that are far superior in terms of convergence rate (from linear to cubic) and overall computational efficiency (working, e.g., with shifts). To include these optimized variants in the simulations in Section 4.1, the MATLAB implementation is used.

3.2.3 Arnoldi Iteration

The Arnoldi iteration [46] is a *two-stage approach* to finding the eigenvalues of a matrix $\mathbf{A} \in \mathbb{C}^{M \times M}$. In the *first stage*, a unitary similarity transform is applied to $\mathbf{A}$ (similar to the previous section), such that

$$\mathbf{U}^H \mathbf{A} \mathbf{U} = \mathbf{H}, \quad \mathbf{U}^H \mathbf{U} = \mathbf{U}\mathbf{U}^H = \mathbf{I}_M \quad (\mathbf{U} \text{ unitary}), \tag{3.20}$$

where $\mathbf{H}$ has Hessenberg form, i.e.,

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1M} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2M} \\ 0 & h_{32} & h_{33} & \cdots & h_{3M} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{MM-1} & h_{MM} \end{pmatrix}.$$

To arrive at an algorithm (modified from [43]) that can be used to obtain the Hessenberg form $\mathbf{H}$ of $\mathbf{A}$, the unitarity of $\mathbf{U}$ is used to rewrite Equation 3.20:

$$\begin{aligned} \mathbf{U}^H\mathbf{A}\mathbf{U} &= \mathbf{H} \\ \leftrightarrow \mathbf{U}\mathbf{U}^H\mathbf{A}\mathbf{U} &= \mathbf{U}\mathbf{H} \\ \leftrightarrow \mathbf{A}\mathbf{U} &= \mathbf{U}\mathbf{H} \end{aligned} \tag{3.21}$$

Defining $\mathbf{u}_k$ as the columns of $\mathbf{U}$ such that $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]$, and having h_{ij} as the ij-th element of $\mathbf{H}$, Equation 3.21 can be rewritten to obtain the transformation matrix column-wise as

$$\begin{aligned} \mathbf{A}\mathbf{u}_k &= \sum_{i=1}^{k+1} h_{ik}\mathbf{u}_i, \quad 1 \leq k \leq M-1 \\ &= h_{k+1,k}\mathbf{u}_{k+1} + \sum_{i=1}^{k} h_{ik}\mathbf{u}_i \\ \leftrightarrow \mathbf{u}_{k+1} &= \frac{1}{h_{k+1,k}} \underbrace{\left(\mathbf{A}\mathbf{u}_k - \sum_{i=1}^{k} h_{ik}\mathbf{u}_i\right)}_{=:\mathbf{r}_k} \\ &= \frac{\mathbf{r}_k}{h_{k+1,k}}. \end{aligned} \tag{3.22}$$

Here, the vectors $\mathbf{u}_k$, also termed "Arnoldi" vectors, form an orthonormal basis for the Krylov subspace $\mathcal{K}\{\mathbf{A}, \mathbf{q}_1, k\} = \text{span}\{\mathbf{q}_1, \mathbf{A}\mathbf{q}_1, \ldots, \mathbf{A}^{k-1}\mathbf{q}_1\}$ (cf. Section 3.2.1). This leads to the algorithm presented in Algorithm 3.4, in which the Arnoldi vectors are made orthonormal using the Gram-Schmidt process. A break condition is introduced to test for convergence

Algorithm 3.4: Arnoldi iteration to compute the Hessenberg form of $\mathbf{A}$

Input: matrix $\mathbf{A} \in \mathbb{C}^{M \times M}$, max. iteration number $N \leq M$, tolerance
Output: unitarily similar Hessenberg matrix $\mathbf{H} \in \mathbb{R}^{M \times M}$
`// initialization`
init $\mathbf{u}_1 = \mathbf{0}^M, \mathbf{H} = \mathbf{0}^{M \times M}, \mathbf{r}_1 \in \mathbb{C}^M$;
for $k = 2 : N$ **do**
  $h_{k,k-1} = ||\mathbf{r}_{k-1}||_2$;
  $\mathbf{q}_k = \mathbf{r}_{k-1}/h_{k,k-1}$;
  $\mathbf{r}_k = \mathbf{A}\mathbf{q}_k$;
  for $j = 1 : k$ **do**
    `// Gram-Schmidt process`
    $h_{jk} = \mathbf{q}_j^H \mathbf{r}_k$;
    $\mathbf{r}_k = \mathbf{r}_k - h_{jk}\mathbf{q}_j$;
  if $||\mathbf{r}_k||_2 <$ *tolerance* **then**
    `// checks whether approximate Hessenberg form is achieved`
    break

even before N iterations are executed[1]. The initialization of $\mathbf{r}_1 \in \mathbb{C}^M$ is an important aspect to consider when aiming for quick convergence, especially in the case of high-dimensional matrices [47]. However, since the investigated matrices in this work are of rather small dimensionality, a vector with normally distributed real coefficients is chosen.

Having brought $\mathbf{A}$ into Hessenberg form, the computational complexity of the *second stage* is reduced, which consists of using the QR method (cf. previous section) to bring $\mathbf{H}$ to upper triangular form, from which the eigenvalues can readily be extracted.

3.2.4 Closed-Form Solution

Obtaining a closed-form solution for the eigenvalues of an M-dimensional matrix $\mathbf{A}$ is based upon finding the roots of a corresponding M-th degree polynomial, implying a restriction on the allowed matrix dimension $M \leq 4$, since the matrices concerned here are of no particular structure (cf. Abel–Ruffini theorem, [48]). For the case $M = 2$, the solution is acquired as follows (starting from the eigenvalue equation).

$$\begin{aligned} &\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \leftrightarrow \\ &(\mathbf{A} - \lambda\mathbf{I}_M)\mathbf{x} = 0, \end{aligned}$$

where the identity matrix $\mathbf{I}_M$ is introduced to write the problem as a set of linear equations. A non-trivial solution (i.e., $\mathbf{x} \neq \mathbf{0}$) exists iff $(\mathbf{A} - \lambda\mathbf{I}_M)$ is singular and hence

$$\det\{\mathbf{A} - \lambda\mathbf{I}_M\} = 0, \tag{3.23}$$

which is called the *characteristic equation* of $\mathbf{A}$. Its M roots correspond to the eigenvalues of $\mathbf{A}$. Hence, for $M = 2$, they are obtained as

$$\begin{aligned} &\det\{\mathbf{A} - \lambda\mathbf{I}_M\} = \lambda^2 + \lambda(a_{22} - a_{11}) + (a_{11}a_{22} - a_{12}a_{21}) \stackrel{!}{=} 0 \\ \leftrightarrow &\lambda_{1,2} = \frac{a_{11} - a_{22}}{2} \pm \sqrt{\left(\frac{a_{22} - a_{11}}{2}\right)^2 + a_{12}a_{21} - a_{11}a_{22}} \end{aligned}$$

For the case $M = 3$, Cardano's method [49] can be used. For $M = 4$, there are several solution schemes, though none was investigated in this work.

3.2.5 Computational Complexity Comparison

To motivate the use of power iterations in Section 4.1 instead of the full EVD obtained using the MATLAB function `eig` or the mentioned alternatives, this section deals with the theoretical complexity of this problem. The computational complexity is given in terms of the number of floating point operations (flops), with each real basic arithmetic operation counted as 1 flop.

For the power method, one iteration of the inner `for` loop requires $8M^2 - 2M - 3$ additions, $8M^2 + 2M$ multiplications, $2M$ divisions, and 1 square root operation, which results in total in $16M^2 + 2M - 2$ flops. Hence, if only the largest eigenvalue is computed using N iterations, $N(16M^2 + 2M - 2)$ flops are required. If the second largest eigenvalue is computed as well, additional operations are required for the matrix rank reduction and N iterations of the inner `for` loop should be repeated. Reducing the rank of $\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)$ requires $2M^2$ additions and $3M^2$ multiplications, yielding in total $5M^2$ flops. Hence, using the power method to estimate

[1] Note that the Arnoldi iteration is a method that may provide sufficient results even before going through all M columns, which is why the maximum iteration number N should be chosen to be smaller than or equal to the number of columns M.

$\lambda_2\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}(l)\}$ requires $N(16M^2 + 2M - 2) + 5M^2$ flops. Overall, the complexity of the power method is $\mathcal{O}(M^2)$.

Although many algorithms exist for computing the full EVD, the QR decomposition-based algorithm [43] is considered here, which is one of the most widely used algorithms to compute eigenvalues. The complexity of the QR decomposition-based algorithm for Hermitian matrices is $\mathcal{O}(M^3)$ [50], also when the matrix is first transformed into real tridiagonal form using Householder reflections [43] (cf. Section 3.2.3).

Hence, using the power method to compute the eigenvalues of interest instead of the full EVD reduces the complexity from $\mathcal{O}(M^3)$ to $\mathcal{O}(M^2)$, which can be advantageous for a real-time implementation of the diffuse PSD estimator, particularly when the number of microphones M is large. For an experimental validation, see Section 4.1.2.

4 Experimental Results

This chapter deals with the experimental validation of the proposed methods. First, in Section 4.1, the performance of the computationally less expensive variant of the EVD-based diffuse PSD estimator is evaluated in terms of its runtime, its convergence behavior, as well as the performance of the MWF using its estimates in different reverberant acoustical scenarios. As a baseline, the diffuse PSD estimator utilizing the *full* EVD is chosen. Second, in Section 4.2, the ALS approach to jointly estimate the RETF vector and the diffuse and target PSDs is compared to a state-of-the-art approach, i.e., the CW approach. The convergence behavior, estimation accuracy, as well as the resulting MWF performance are investigated using the same acoustical scenarios mentioned above.

4.1 PSD Estimation using EVD

In this section, the performance of the EVD-based approach laid out in Section 2.4.2 is evaluated, and the different methods to obtain an EVD described in Section 3.2 are briefly compared in terms of their accuracy and runtime. The acoustical systems used in this evaluation are presented, along with the corresponding algorithm settings. Afterwards, the resulting improvements in terms of the quality measures discussed in Section 2.6 are shown, where the improvements are always computed w.r.t. the target signal.

4.1.1 Evaluation Setup and Algorithmic Settings

In this evaluation, three different acoustical scenarios are investigated, each consisting of a single spatially stationary speech source and a microphone array with $M \in \{4,\ 6\}$ microphones. The first microphone is considered as the reference microphone. The details for each system are summarized in Table 4.1. The reverberant microphone signals are obtained by convolving a 38 s long anechoic speech signal with the measured RIRs at a sampling frequency of 16 kHz. All acoustical systems are investigated *without* any additional noise in Section 4.1.3. Further, they are considered including additive diffuse babble noise at different input SNRs, and with uncorrelated noise in addition to reverberation and diffuse noise at different input SNRs, in Section 4.1.4. In the latter case, the noisy signal is preceded

	array geometry	mic. distance	θ	T_{60}
AS_1 [51]	linear	$d = 8\,\mathrm{cm}$	$45\,^\circ$	$0.61\,\mathrm{s}$
AS_2 [36]	circular	$r = 10\,\mathrm{cm}$	$45\,^\circ$	$0.73\,\mathrm{s}$
AS_3 [52]	linear	$d = 6\,\mathrm{cm}$	$-15\,^\circ$	$1.25\,\mathrm{s}$

Tab. 4.1.: configuration of considered acoustical scenarios; d: inter-microphone distance, r: circle radius, θ: speaker direction of arrival, reverberation time T_{60} (cf. Section 1)

by a 1 s noise-only signal used for the estimation of the uncorrelated noise PSD matrix $\mathbf{\Phi_v}(l)$. Note that in the following, the time frame index l is omitted wherever possible.

The spatial coherence matrix $\mathbf{\Gamma}$ is computed assuming spherically diffuse noise, such that the (i,j)-th component is given by Equation 2.28. For the spatial coherence matrix to be invertible (cf., e.g., Equation 2.38, Equation 4.6), it needs to be non-singular. However, when computed as shown in Equation 2.28, numerical imprecisions especially in the case of low frequencies may lead to a singular $\mathbf{\Gamma}(k)$ (depending on the microphone array geometry)[1]. Hence, regularization of the matrices is performed as

$$\mathbf{\Gamma} = (1-\rho)\mathbf{\Gamma} + \rho\mathbf{I}_M, \tag{4.1}$$

where the regularization constant $\rho = 0.1$ is used in this work.

Since PSDs can only assume positive values, the PSD estimates are lower-bounded by the machine precision `eps`. Furthermore, since neither the diffuse nor the target PSD can have larger values than the microphone signal PSD, also an upper bound is applied, i.e.,

$$\texttt{eps} \leq \{\hat{\phi}_s,\, \hat{\phi}_d\} \leq \frac{1}{M}\mathbf{y}^H\mathbf{y}. \tag{4.2}$$

The RETF vector $\mathbf{a}$ necessary for the MVDR filter computation is assumed to be known. It is obtained from the first 8 ms of the measured RIRs. The transformation of the time domain signals to the time-frequency domain (cf. Section 2.2) is done using the `dgtreal` function from the LTFAT MATLAB toolbox [53]. The fast Fourier transform (FFT) length is set to 1024 with an overlap of 75 % between successive frames, and a tight Hamming window is used. Recursive averaging (as described in Equation 2.31; $\alpha = 0.6703$) is utilized to estimate the microphone PSD matrix $\mathbf{\Phi_y}$. In the scenarios including uncorrelated noise, the uncorrelated noise PSD matrix $\mathbf{\Phi_v}$ is estimated using recursive averaging with forgetting factor $\alpha = 0.5$ during the noise-only phase at the beginning, and the estimate of the last time frame is subtracted from the microphone PSD matrix, i.e., it is assumed to be stationary. In the next step, the diffuse PSD $\phi_\mathbf{d}$ is estimated using the EVD-based approach as described in Section 3.2, using either the full EVD obtained via the `eig` function of MATLAB or the eigenvalue(s) computed using a given number of power iterations. This leads to four variations that are discussed in the following sections:

EIG1 denotes the approach utilizing the first eigenvalue as estimated with the `eig` function and the trace of the pre-whitened estimated microphone PSD matrix to compute the diffuse PSD:

$$\hat{\phi}_d^{\text{EIG1}} = \frac{1}{M-1}\left(\text{tr}\{\hat{\mathbf{\Phi}}_\mathbf{y}^w\} - \lambda_1\{\hat{\mathbf{\Phi}}_\mathbf{y}^w\}\right) \tag{4.3}$$

PI1 uses the above equation for the PSD estimate, replacing $\lambda_1\{\hat{\mathbf{\Phi}}_\mathbf{y}^w\}$ with the estimate $\hat{\lambda}_1\{\hat{\mathbf{\Phi}}_\mathbf{y}^w\}$ obtained via $N = 2$ power iterations.

EIG2 makes use of the second eigenvalue as estimated with MATLAB's `eig` function, resulting in

$$\hat{\phi}_d^{\text{EIG2}} = \lambda_2\{\hat{\mathbf{\Phi}}_\mathbf{y}^w\}, \tag{4.4}$$

[1] More intuitively, for small microphone distances and low frequencies / large wavelengths, the microphone signals are extremely coherent, leading to only minute differences between the entries of $\mathbf{\Gamma}(k)$.

which is — in theory — equal to $\hat{\phi}_d^{\text{EIG1}}$. In practice, however, the individual eigenvalues are distinct[1], i.e.,

$$\lambda_i\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}\} \neq \lambda_j\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}\}, \quad i,j \in [2,M]; \quad i \neq j. \tag{4.5}$$

PI2 uses a power iteration-based procedure to obtain an estimate $\hat{\lambda}_2\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}\}$ of the second eigenvalue, which is then used to estimate the diffuse PSD as in Equation 4.4.

Following the decision-directed approach described in Section 2.3.3, the diffuse PSD estimate $\hat{\phi}_d$ is used to estimate the target PSD ϕ_s.
The MWF coefficients are computed — depending on the investigated scenario — as

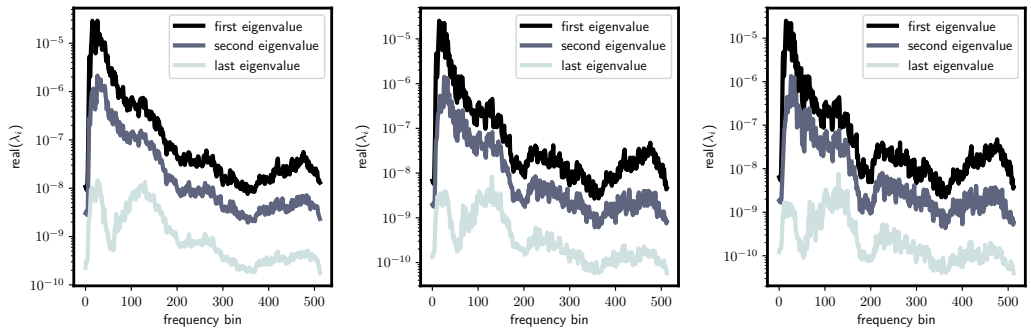
$$\mathbf{w}_{\text{MWF}} = \underbrace{\frac{\mathbf{\Gamma}^{-1}\mathbf{a}}{\mathbf{a}^H\mathbf{\Gamma}^{-1}\mathbf{a}}}_{\mathbf{w}_{\text{MVDR}}} \underbrace{\frac{\hat{\phi}_s(l)}{\hat{\phi}_s(l) + \hat{\phi}_d(l)/(\mathbf{a}^H\mathbf{\Gamma}^{-1}\mathbf{a})}}_{G(l)} \tag{4.6}$$

in the case of either no noise or diffuse noise, and for additional uncorrelated noise as

$$\mathbf{w}_{\text{MWF}} = \underbrace{\frac{\left(\hat{\phi}_d(l)\mathbf{\Gamma} + \hat{\mathbf{\Phi}}_{\mathbf{v}}(l)\right)^{-1}\mathbf{a}}{\mathbf{a}^H\left(\hat{\phi}_d(l)\mathbf{\Gamma} + \hat{\mathbf{\Phi}}_{\mathbf{v}}(l)\right)^{-1}\mathbf{a}}}_{\mathbf{w}_{\text{MVDR}}(l)} \underbrace{\frac{\hat{\phi}_s(l)}{\hat{\phi}_s(l) + \left(\mathbf{a}^H(\hat{\phi}_d(l)\mathbf{\Gamma} + \hat{\mathbf{\Phi}}_{\mathbf{v}}(l)^{-1}\mathbf{a}\right)^{-1}}}_{G(l)} \tag{4.7}$$

yielding time-independent MVDR coefficients in Equation 4.6 and time-dependent ones in Equation 4.7. The minimum gain of the single-channel postfilter is set to $G_{\min} = -10\,\text{dB}$ (cf. Section 2.3.2).

Typical Eigenvalues Since the method is based on computing the eigenvalues of the prewhitened estimated microphone PSD matrix, the actual values that are typically encountered in practical scenarios are of interest due to the possibility of numerical precision problems. For this reason, typical eigenvalues of $\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}$ are displayed in this section, including the first, second and last eigenvalue, averaged over all time frames. Acoustical scenario AS_1 with $M = 4$ is considered at different input SNRs, with added diffuse babble noise in Figure 4.1 and added uncorrelated noise in Figure 4.2. It can be observed that the eigenvalues (especially the

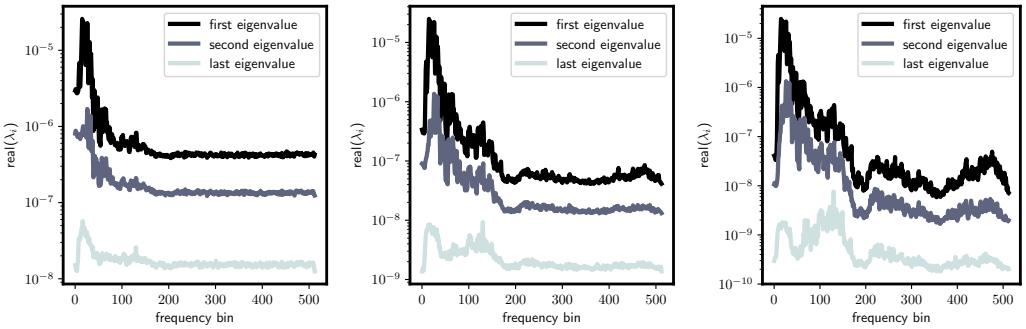


(a) diffuse noise at 0 dB **(b)** diffuse noise at 10 dB **(c)** diffuse noise at 20 dB

Fig. 4.1.: typical eigenvalues of prewhitened microphone PSD matrix, AS_1, diffuse noise at different input SNRs

first and second one) behave similarly for the same frequencies, with only a constant scaling between them[2]. This hints at the fact that there is always a dominant eigenvalue, which

[1] For more details see Section 2.4.2.
[2] Note that the y-axis is plotted logarithmically.



(a) uncorrelated noise at 0 dB **(b)** uncorrelated noise at 10 dB **(c)** uncorrelated noise at 20 dB

Fig. 4.2.: typical eigenvalues of prewhitened microphone PSD matrix, AS_1, uncorrelated noise at different input SNRs

justifies the use of the power method in the computation of the eigenvalues. Furthermore, the plot reveals, as already mentioned above, that the signal model in Equation 2.44 is not perfectly true, and hence the last $M-1$ eigenvalues are not equal, with the relative difference between the second and the last eigenvalue larger than the relative difference between the first and second eigenvalue. Comparing the eigenvalues in the diffuse noise and uncorrelated noise scenarios, it becomes apparent that for a low input SNR (cf. Figure 4.2a) and high frequencies (i.e., $f > f_{200} = 200\frac{8000\,\text{Hz}}{513} \approx 3100\,\text{Hz}$, with $f_k = k\frac{8000\,\text{Hz}}{513}$ the center frequency in the k-th frequency bin with the used STFT settings), the eigenvalues remain approximately constant. This is to be expected, since the diffuse noise includes late reverberation (i.e., filtered *speech*) such that it will have low energy at high frequencies. Thus, it will become hidden in the uncorrelated noise floor at low input SNRs.

4.1.2 Comparison of EVD Algorithms

In this section, the various EVD algorithms presented in Section 3.2 are compared in terms of their PSD estimation accuracy and their runtime on an Intel i7-2600 processor running Linux Mint 18.3 (64 Bit) measured using MATLAB's `tic toc` function. Note that none of the implementations are optimized in terms of their computational efficiency except for the MATLAB function `eig`; however, the presented values can be used to get a general idea of their complexity relative to each other. The closed-form solutions, which exist up to a sensor number $M = 4$, are not included due to their limited use and the fact that they are not iterative like the other algorithms. In order for the comparison to be representative, this evaluation is done in different scenarios, and the average values are presented. To be able to compare the estimated diffuse PSDs with the "true" ones[1], reverberant scenarios $\{\text{AS}_1, \text{AS}_2, \text{AS}_3\}$ with diffuse noise at 20 dB input SNR are chosen (cf. Section 4.1.1). The "true" PSD is then computed intrusively using recursive averaging (cf. Equation 2.31) as

$$\hat{\phi}_{\mathbf{d}}(l) = \begin{cases} (1-\alpha)\hat{\phi}_{\mathbf{d}}(l-1) + & \frac{\alpha}{M}\mathbf{d}^H(l)\mathbf{d}(l), \quad l > 1 \\ & \frac{1}{M}\mathbf{d}^H(l)\mathbf{d}(l), \quad l = 1, \end{cases} \tag{4.8}$$

[1] A "true" diffuse PSD is not defined, since it is arbitrary where exactly to split a RIR into its early and late reverberation part, hence influencing what the diffuse PSD describes. A common choice is 8 ms (corresponding to 128 samples at $f_s = 16\,\text{kHz}$), which is adopted here.

	$\phi_{\mathbf{d}}$ estimation error / dB		av. time per computation / 10^{-5} s	
iteration number	2 its	5 its	2 its	5 its
Arnoldi Iteration	5.389	5.632	17.935	22.517
Power Method, First Eigenvalue	5.663	5.714	2.331	2.426
Power Method, Second Eigenvalue	4.653	4.603	2.090	2.253
QR Method	5.682	5.716	5.391	10.426
QZ Method (MATLAB's `eig` function)	5.721	5.721	3.583	3.583

Tab. 4.2.: accuracy and speed comparison of considered EVD algorithms

where the diffuse component vector $\mathbf{d}(l)$ contains late reverberation and diffuse babble noise. The PSD estimation accuracy is evaluated using the average PSD estimation error over all time-frequency bins [54] (denoted "total error"), i.e.,

$$\epsilon = \frac{1}{KL} \sum_{k=1}^{K} \sum_{l=1}^{L} 10 \log_{10} \frac{\phi(k,l)}{\hat{\phi}(k,l)}. \tag{4.9}$$

For all but MATLAB's method, a fixed number of iterations $N = 2$ is chosen, which is determined to be a suitable compromise between speed and accuracy in case of the power method (see Figure 4.5). In addition, the comparison is repeated for $N = 5$, since the number of power iterations required for a suitable PSD estimation accuracy is increased in the presence of uncorrelated noise.

Without giving extensive meaning to the exact values, the results in Table 4.2 demonstrate that both power method variants at both iteration numbers $N = \{2,\ 5\}$ are slightly faster than the `eig` function and significantly faster than the Arnoldi iteration and QR method[1].

The PSD estimation error is determined using the first eigenvalue, i.e., following Equation 2.42; the result in "Power Method, Second Eigenvalue" obtained via Equation 2.41 is given only for reference, since computing the second eigenvalue with the power method involves a larger computational complexity.

4.1.3 Dereverberation Performance

The dereverberation performance of the MWF using either the full EVD (using the MATLAB function `eig`) or the more tractable power method in noiseless, reverberant scenarios is investigated in this section. For reasons of comparison, also the performance of the MVDR without the additional postfilter as depicted. The choice of parameters for these simulations (random initialization, $N = 2$ power iterations, $M = 4$ microphones) is derived from the following section.

The results, which are displayed in Figure 4.3, demonstrate that two power iterations are sufficient for providing the same MWF performance as the full EVD using `eig` in the evaluated scenarios and performance measures. Informal listening tests confirm this observation.

[1] The measured time per computation for the power method using the second eigenvalue is smaller than for the power method using the first eigenvalue, which is a sign of the non-exactness of the measurement, since the computation of the second eigenvalue requires the first one and hence should take longer in every case.

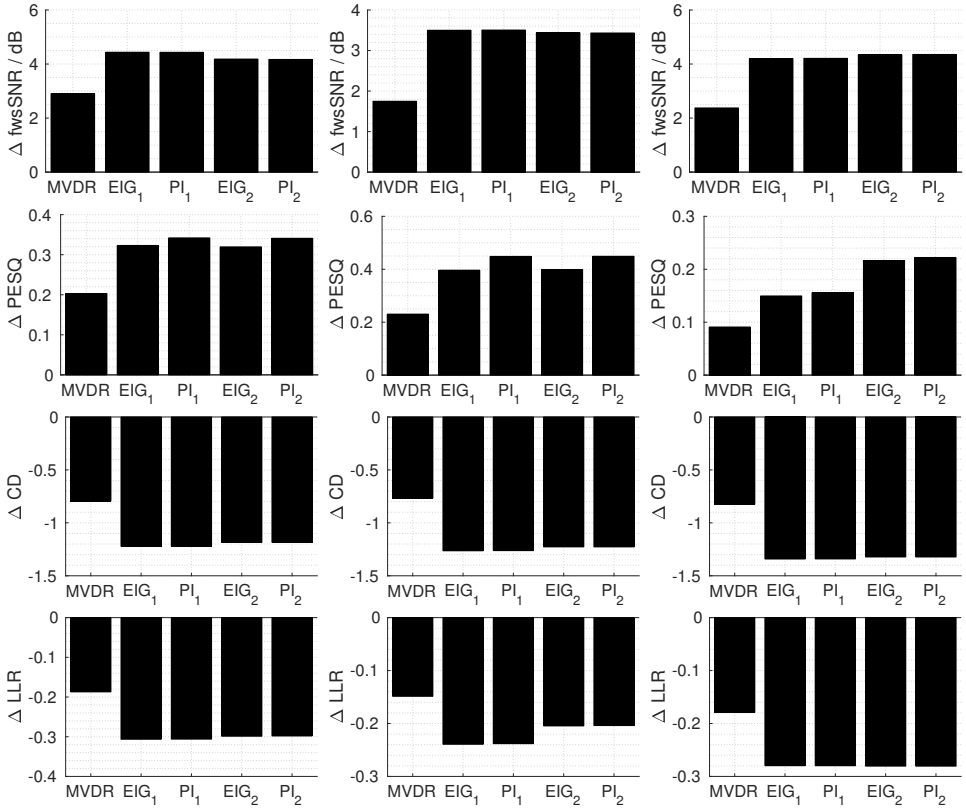


Fig. 4.3.: dereverberation performance in noiseless scenarios, $M = 4$, $N = 2$; left to right column: AS_1, AS_2, AS_3

4.1.4 Joint Dereverberation and Denoising

As an extension of the previous section, in this section the performance of the MWF is evaluated in scenarios containing noise in addition to reverberation. Following the model in Equation 2.3, the investigation is split into a part dealing with diffuse noise, which the derivation of the EVD-based MWF is based upon, and a part in which additional uncorrelated noise is considered as well.

Diffuse Noise

Acoustical systems $\{\mathrm{AS}_1, \mathrm{AS}_2, \mathrm{AS}_3\}$ with additional diffuse babble noise generated as described in Section 4.1.1 are evaluated in this section. First, the influence of using an increased number of microphones is shown. Next, different power method initialization schemes are compared, presented in terms of convergence speed and accuracy. After obtaining a suitable set of parameters, the method is applied to the acoustical scenarios mentioned above, and the resulting MWF performance is compared to that of the full eigendecomposition.

Influence of Number of Used Microphones In this section, the performance of the MWF is compared for different numbers of used microphones ($M = \{4, 6\}$). The acoustical system AS_1 with diffuse babble noise at input SNR $10\,\mathrm{dB}$ is considered, and the number of power iterations is set to $N = 2$ as suggested in Figure 4.5. The results show that the MWF with $M = 6$ consistently outperforms the one with $M = 4$, which can be explained by the fact that more spatial information is exploited. Simulations for acoustical system AS_2 confirm this finding (cf. Figure A.1).

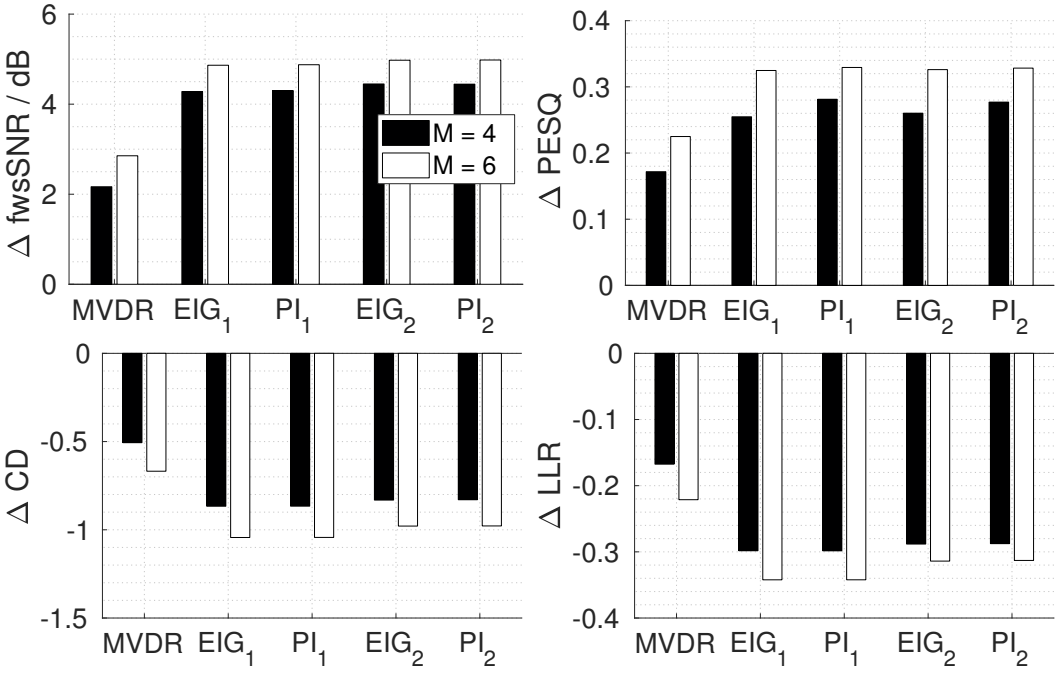


Fig. 4.4.: MWF performance for $M = \{4, 6\}$, AS_1, diffuse babble noise at 10 dB input SNR

Initialization and Convergence Speed of Power Method Since the power method is an iterative procedure, a termination criterion needs to be imposed that, in general, ensures a sufficient estimation accuracy. While there is a variety of options to consider, such as the angle between successive estimated dominant eigenvectors or the relative incremental change in the dominant eigenvalue, setting a fixed iteration number remains a computationally cheap and easily comparable alternative. To investigate the required number of iterations, in this section the diffuse PSD estimation accuracy versus the number of power iterations is evaluated in terms of the total PSD estimation error (cf. Equation 4.9) and compared to a benchmark. This benchmark is chosen to be MATLAB's `eig` function, since it is thoroughly tested.

In addition, the influence of the initialization of $\mathbf{b}_p^{(0)}$ (cf. Algorithm 3.2) is tested. On the one hand, a *random* initialization using *normally distributed real-valued vector components* is chosen, marked with "-R". On the other hand, a *deterministic variant* $\mathbf{b}_p^{(0)} = [1,\, 0,\, \ldots,\, 0]^T$ is tested, marked with "-D". As a last option, initializing with the estimated eigenvector of the *previous time frame* is evaluated[1], which aims at exploiting the temporal correlation of neighbored STFT bins (marked "-P" and termed "previous initialization" from here on).

For the random and previous initialization, five simulations are carried out with the same set of parameters, and the corresponding mean and standard deviation values are displayed. The same is provided for the deterministic initialization as a proof of concept, although it is clear that the standard deviation equals zero, since no random number generator is involved in this case. To reduce the number of plots, only the results in the acoustical system AS_1 with diffuse babble noise at input SNRs $\{10,\, 40\}\,\text{dB}$ and using $M = 4$ microphones are presented (see Figure 4.5), although the other tested diffuse noise scenarios exhibit the same results.

First and foremost, it can be seen that in all instances the power method converges to the same diffuse PSD estimation accuracy as the `eig` function after maximally $N = 2$ power iterations ($N = 1$ for previous initialization). Note that the error in case of $\text{PI}_2 - \{R, D\}$ after *exactly one* iteration is very large, which is why the corresponding line is cut off.
As expected from the discussion in Section 3.2.1, there is no significant difference between using either the random or the deterministic initialization scheme. In addition, the variance when using random initialization is close to zero, which points to the robustness of the algorithm with regard to the starting values. Although these methods already perform quite well, exploiting the correlation of STFT bins in the previous initialization seems beneficial, since the required number of iterations can be reduced from $N = 2$ to $N = 1$ at no significant additional computational complexity. In summary, the three tested initialization schemes all lead to quick convergence to the expected solution as dictated by the MATLAB function `eig`, with the previous initialization performing the best.

MWF Performance In this section, the parameters determined in the previous sections (i.e., $M = 4$ microphones, $N = 2$ power iterations) are chosen to evaluate the performance of the MWF utilizing the different diffuse PSD estimates more thoroughly. Random initialization is chosen to describe the convergence behavior in a "worst-case" scenario. In addition to the previously used fwsSNR and PESQ, also the CD is considered (cf. Section 2.6.3). Figure 4.4 depicts the denoising performance of the MWF using the different PSD estimates described in Section 4.1.1 in the presence of diffuse babble noise. It can be observed that no significant discrepancy between using either the full EVD or the power method occurs in any of the

[1] Note that for the first time frame, where *no* previous estimate is available, random initialization is chosen.

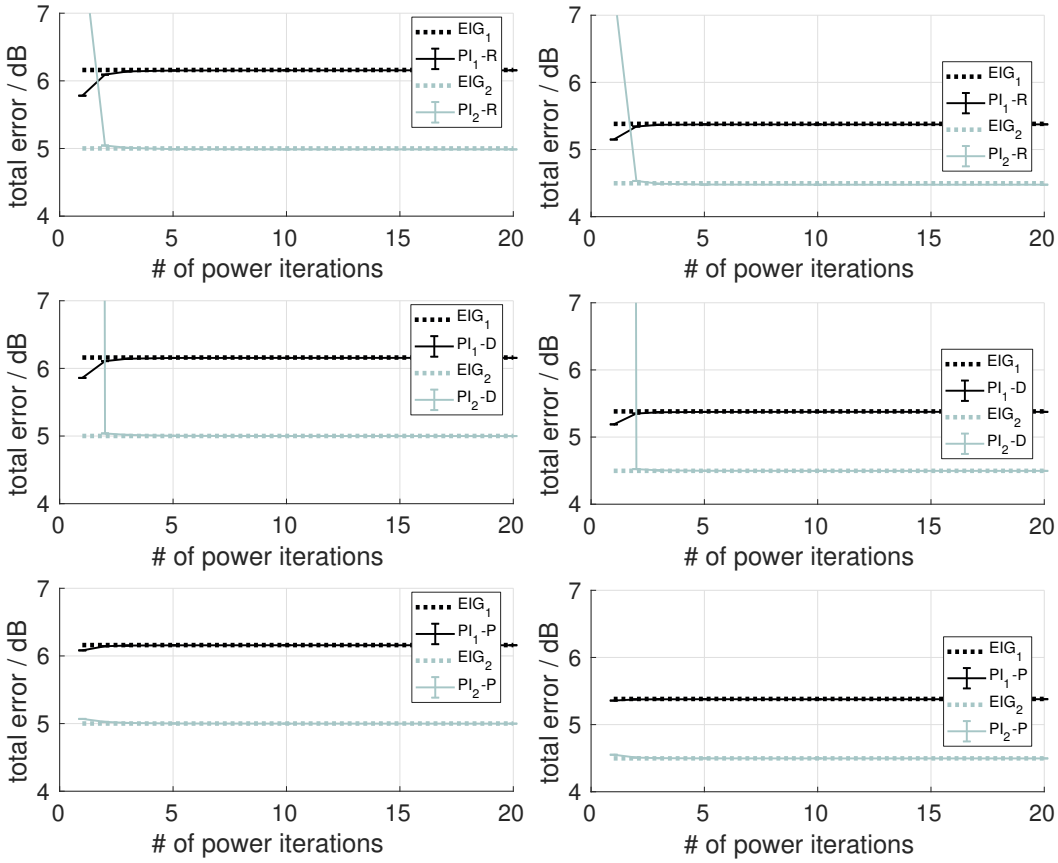


Fig. 4.5.: convergence speed of power method with random, deterministic and previous initialization; input SNRs: left: 10 dB, right: 40 dB

considered scenarios, as expected when considering the convergence discussion above. Only when comparing the PESQ values, the differences become visible, which are, however, fairly small. Here, the power method-based variants lead to an even higher quality increase, which can also be seen in the STFT representations of the MWF output signals, depicted in Figure A.3, showing a higher contrast between the noise floor and the speech components when using the power method-based diffuse PSD estimate. This may be a random occurrence, since no better performance is expected.

Furthermore, the Wiener postfilter adds to the performance of the MVDR beamformer in every case, with, e.g., additional improvements in terms of fwsSNR of about 2 dB in AS_1.

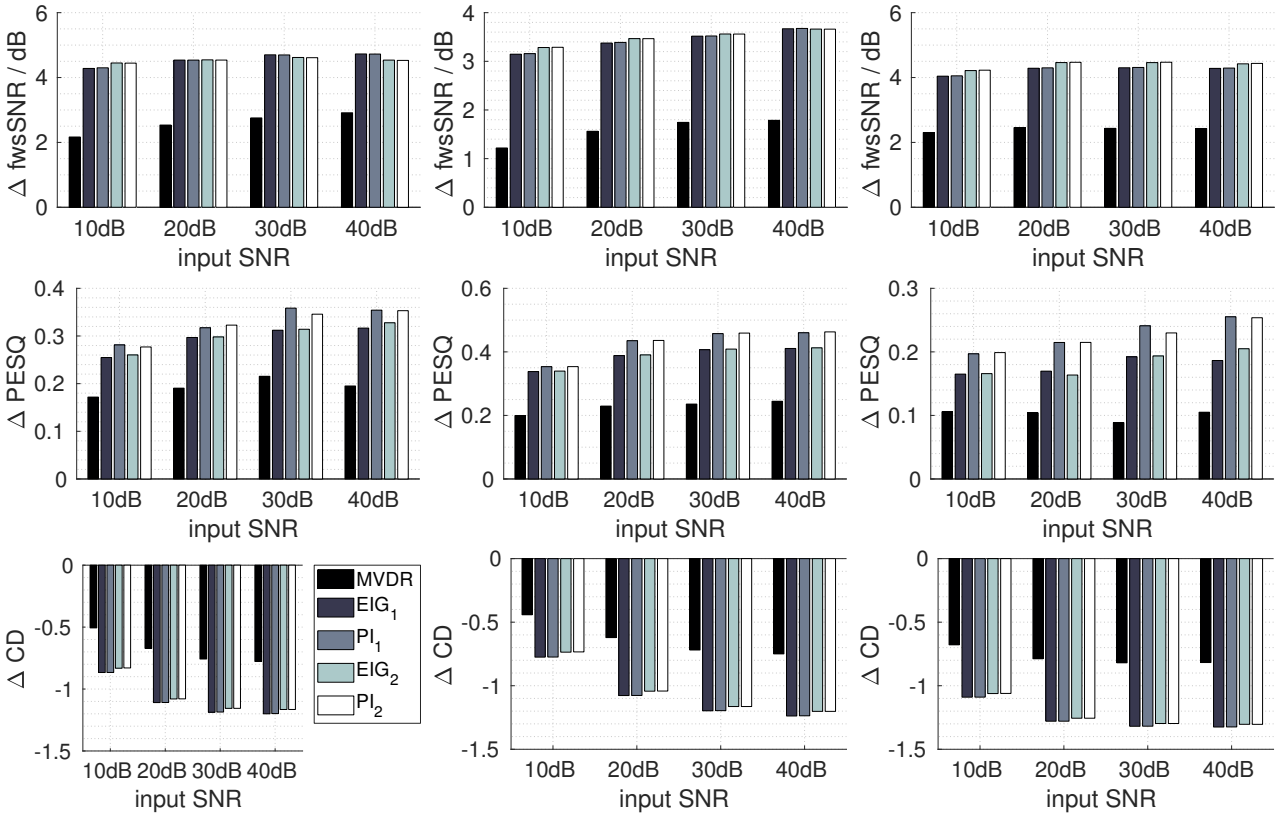


Fig. 4.6.: joint dereverberation and diffuse noise reduction at different input SNRs, $M = 4$, $N = 2$; left to right column: AS_1, AS_2, AS_3

Uncorrelated Noise

Initialization and Convergence Speed of Power Method To demonstrate the convergence behavior of the power method with its different initializations in scenarios containing uncorrelated white noise, the discussion in Section 4.1.4 is repeated. However, note that access to the true interference PSD is not available here, and hence a different measure to evaluate the convergence speed is required. For this purpose, the performance of the MWF utilizing the diffuse PSD estimates $\hat{\phi}_d(l)$ as given by the PESQ improvement is chosen. The remaining performance measures used above are not included in this discussion, since they show the same behavior.

The results, which are displayed in Figure 4.7, show a different convergence behavior for the power method, when uncorrelated noise is present. In case of a lower input SNR (10 dB), the power method-based variants converge to different PESQ improvements than the `eig`-based ones. The methods based on the second eigenvalue seem to be affected by this to a larger extent, which can be explained by the fact that estimation of the second eigenvalue in the power method requires an estimate of the first eigenvalue, which itself is error-prone. In addition, there is a noticeable but still rather small variation between simulations as shown by the error bars, which is different in the diffuse noise case.

Initializing the power method deterministically yields similar results. Also for the previous initialization, convergence to a different limit can be observed, however faster than in the other variants. It may be interesting to note that independent of the initialization method, the level that the power method-based variants converge to is approximately the same.

Although power method-based variants using the second eigenvalue seem to perform better in the case of 10 dB input SNR than the `eig`-based methods, it needs to be noted that the reason for using the power method is complexity reduction, not increasing the performance. Indeed, a different performance only hints at an inaccurate estimate, which is why the deterministic initialization receives a lower rating in this scenario. To check this finding, the simulation is repeated for a different scenario, i.e., using AS_2 instead of AS_1 and otherwise using the same parameters. This simulation confirms the previously mentioned results (see Figure A.2).

Another finding, which is recurring throughout the simulations including uncorrelated noise, is that the estimates based on the second eigenvalue tend to a better MWF performance than the ones based on the mean of the smallest $M-1$ eigenvalues in case of a high input SNR, and vice versa for a lower input SNR. This may be explained by the fact that, as $\lambda_2\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}\} \geq \frac{1}{M-1}(\mathrm{trace}\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}\}) - \lambda_1\{\hat{\mathbf{\Phi}}_{\mathbf{y}}^{w}\}$, using λ_2 leads to a more aggressive postfilter. In case of a lower input SNR and hence larger estimation error, the *less* aggressive filter may be preferred, since it introduces fewer signal distortions at the cost of providing smaller noise reduction.

In conclusion, additive uncorrelated noise at a significant input level makes a larger number of power iterations ($N \approx 3$ for previous initialization, otherwise $N \approx 5$) necessary in order to provide a sufficient accuracy. There is no significant difference between the investigated initialization schemes after convergence, although using previous initialization seems to be the preferable choice due to its faster convergence speed. Since the intention behind using the power method is reducing computational complexity, it is noted here that even for $N = 5$, the power method is significantly faster than the other evaluated

algorithms, and that its use hence is still advantageous in terms of computational complexity. (cf. Section 3.2.5, Table 4.2).

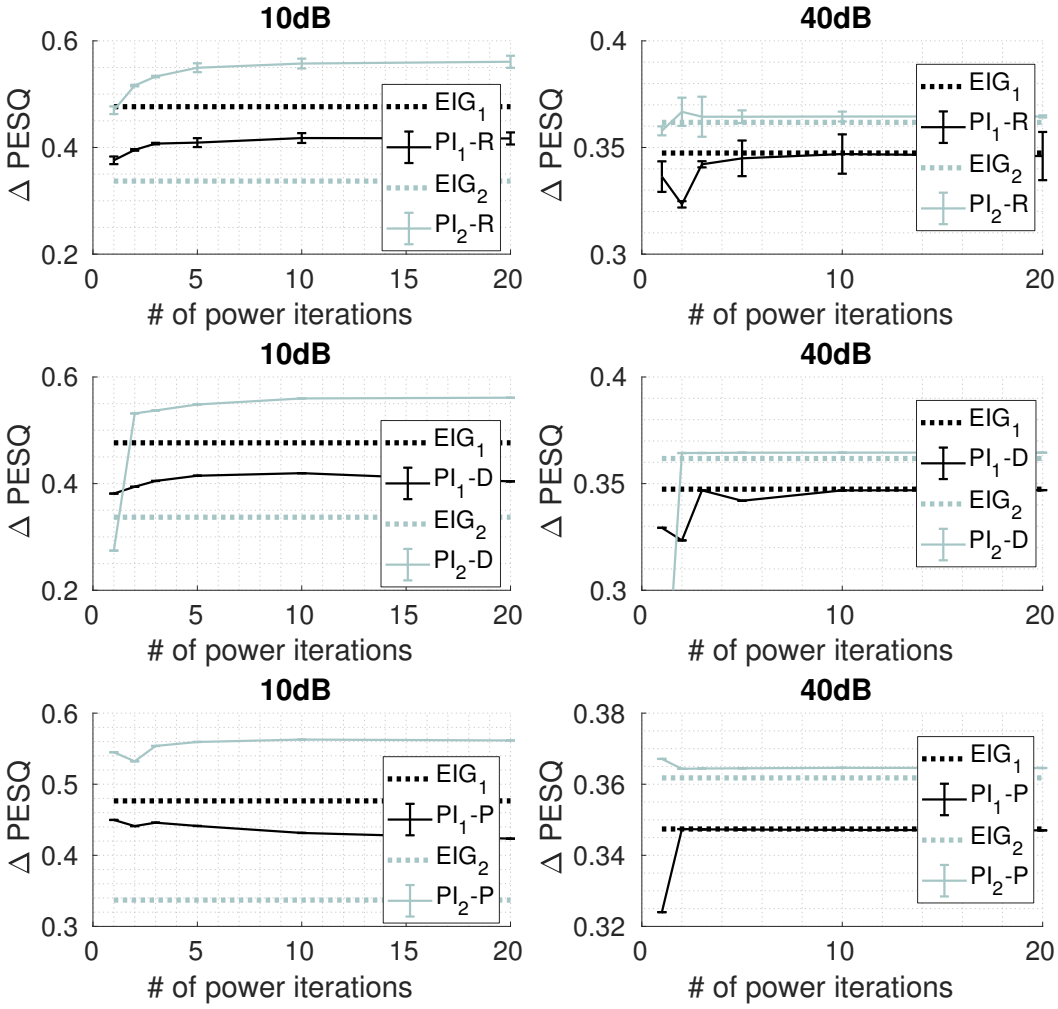


Fig. 4.7.: power method convergence speed with random, deterministic and previous initialization at different uncorrelated noise input SNRs evaluated using MWF performance; AS_1

MWF Performance Similar to the discussion in Figure 4.1.4, in this section the overall performance of the considered methods is discussed, again making use of the fwsSNR, PESQ and CD measures. Instead of using $N = 5$ power iterations, as suggested in the previous convergence discussion, $N = 2$ iterations are chosen here as well, in order to allow a better comparison to the results in Figure 4.1.4.

In contrast to the scenarios with diffuse noise, significant differences between using the full EVD-based or the power method-based diffuse PSD estimates can be detected using fwsSNR and CD. This is especially the case, when the input SNR is low, i.e., when the observed data differ from the model to a larger extent. However, there is no clear trend as to whether this difference leads to a higher or lower MWF performance. Considering the convergence discussion above, in which it was shown that for uncorrelated noise, the power method converges to a different solution at a slower pace, this behavior is expected. However, note that in practical scenarios, uncorrelated noise such as sensor noise does not usually reach

critical levels such as 10 dB input SNR, and hence the small number of iterations $N = 2$ is still justified.

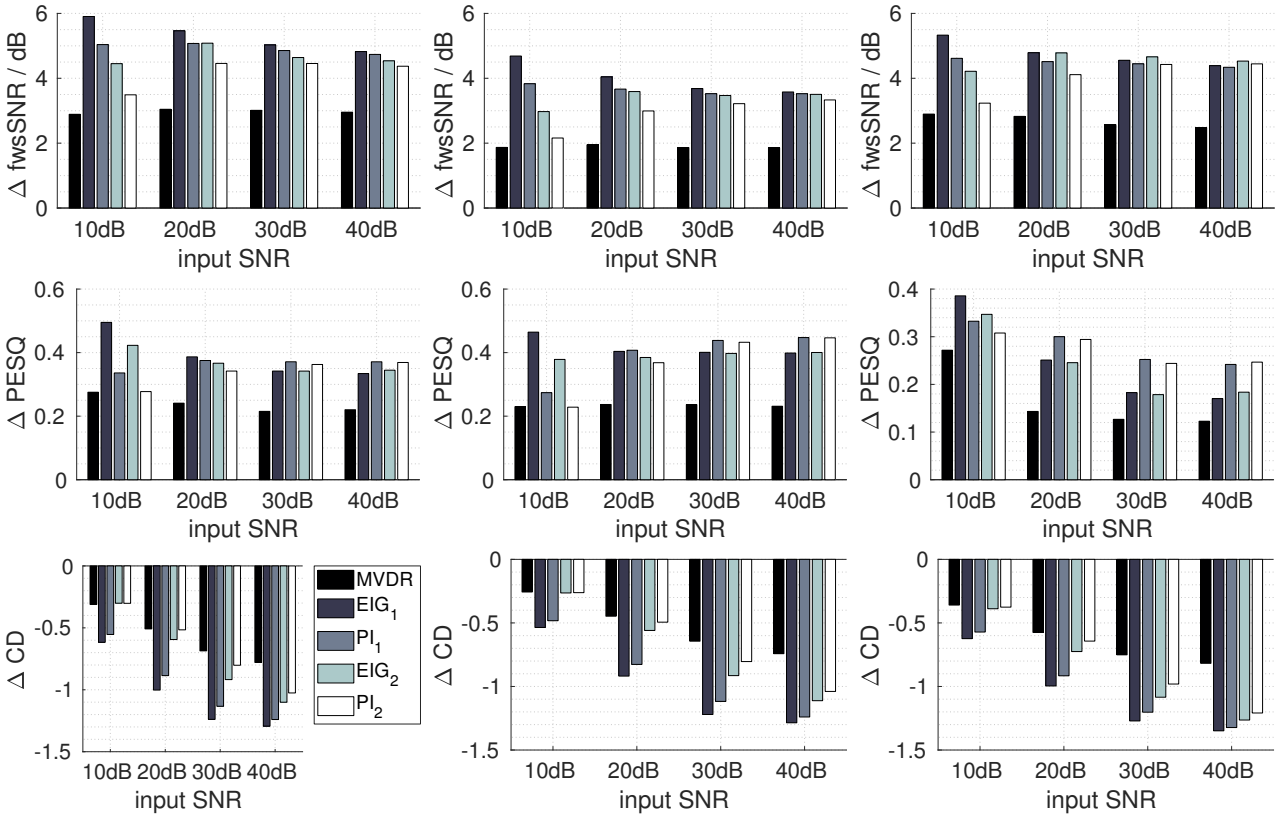


Fig. 4.8.: joint dereverberation and noise reduction, $M = 4$, $N = 2$; left to right column: AS_1, AS_2, AS_3

4.1.5 Summary

In the previous sections, the performance of the EVD-based methods for estimating the diffuse PSD $\phi_{\mathbf{d}}$ is investigated in different reverberant acoustical scenarios with and without uncorrelated white noise or diffuse babble noise. The results show that the extension of the MVDR beamformer to the MWF increases the performance consistently, with a larger number of utilized microphones corresponding to larger improvements. Considering the initialization method, it can be observed that all used variants converge to the same estimation error, although utilizing the dominant eigenvector from the previous time frame results in the fastest convergence.

While the behavior of the full EVD-based methods is without significant differences to the power iteration-based ones in the noiseless as well as diffuse noise scenarios, there *are* such differences in the uncorrelated noise cases. However, under the assumption of a large SNR (i.e., $\gtrapprox 20\,\mathrm{dB}$), these differences become negligible, such that the power method can still be used for complexity reduction.

4.2 Iterative Joint Estimation of PSDs and RETFs

This section deals with validating the performance of the methods for joint RETF vector and diffuse PSD estimation discussed in Section 2.5 and Section 3.1. First, the RETF vector and diffuse PSD estimation accuracy of the considered methods is evaluated for different input diffuse-to-noise ratios (DNRs) using simulated data. Considering the Frobenius norm-based RETF vector estimators, it is decided whether to follow the unconstrained optimization (OPT) or the rank-1 approximation approach LS. In addition, the performance of an MWF in realistic acoustical scenarios when using the various RETF vector and diffuse PSD estimates is compared.

4.2.1 Validation Using Artificial Data

To evaluate the estimation accuracy and the convergence speed of the discussed methods, artificial data is generated according to the assumed signal model in Equation 2.44, such that *oracle information* is available. In total, $M = 4$ microphones are simulated for $K = 513$ frequency bins and $L = 100$ time frames. Specifically, the target and diffuse PSDs at each time-frequency bin are drawn from a scaled and squared normal distribution, i.e., $\phi_s(k,\, l),\ \phi_{\mathbf{d}}(k,\, l) \sim 10^{-7}\,(\mathcal{N}(\mu = 0,\ \sigma = 1))^2$. The scaling is chosen to be in the range of real data (cf. Figure 4.1, Figure 4.2). The RETF vectors $\mathbf{a}(k)$ are assumed to be time-invariant and are generated using normally distributed complex-valued components, and the first element is set equal to 1. For the spatial coherence matrix $\mathbf{\Gamma}(k)$, a random positive-definite matrix is added to a random diagonal matrix with positive values, and the resulting matrix is scaled such that the diagonal elements are equal to 1.

Since in typical acoustical scenarios the late reverberation and the ambient noise are not perfectly diffuse, and since the early reverberation and diffuse components are not perfectly uncorrelated, the signal model in Equation 2.44 is typically violated. In order to evaluate the robustness of the proposed method to model mismatches, an $M \times M$-dimensional scaled error matrix $\mathbf{\Xi}$ is added, i.e.,

$$\mathbf{\Phi}_{\mathbf{y}}(k,l) = \phi_s(k,l)\mathbf{a}(k)\mathbf{a}^H(k) + \phi_{\mathbf{d}}(k,l)\mathbf{\Gamma}(k) + \delta\mathbf{\Xi}(k,l); \quad \delta = \frac{10^{-7}}{10^{\text{input DNR / 10}}}, \tag{4.10}$$

where δ determines the strength of the model mismatch, and with the diffuse-to-noise ratio (DNR). The error matrix $\mathbf{\Xi}(k,\, l)$ is generated as $\mathbf{\Xi}(k,l) = \mathbf{n}(k,l)\mathbf{n}^H(k,l)$, with $\mathbf{n}(k,l)$ an M-dimensional vector with complex-valued normally distributed components. The considered DNR values range from -50 dB to 50 dB.

For the PSD estimation accuracy, the error defined in Equation 4.9 is considered again. The RETF vector estimation accuracy is evaluated using the average Hermitian angle between the oracle vector $\mathbf{a}(k,\, l)$ and the RETF vector estimate $\hat{\mathbf{a}}(k,\, l)$ as [31]

$$\Delta\theta = \frac{1}{KL}\sum_{k=1}^{K}\sum_{l=1}^{L}\arccos\left(\frac{\left|\hat{\mathbf{a}}^H(k,l)\mathbf{a}(k)\right|}{\|\hat{\mathbf{a}}(k,l)\|_2\|\mathbf{a}(k)\|_2}\right)\frac{360\,^\circ}{2\pi}, \tag{4.11}$$

which is a measure disregarding the length difference of both vectors. Figure 4.9 depicts the diffuse PSD estimation error and the Hermitian angle versus the number of ALS iterations for a DNR of 0 dB. In addition, the performance of the CW method is depicted for reference. For the LS method, it can be observed that, while there is no significant change in the Hermitian angle for an increasing number of ALS iterations, the diffuse PSD estimation error

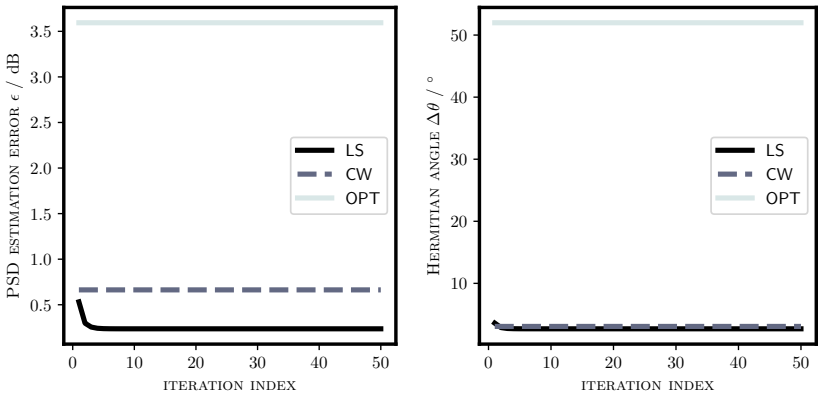
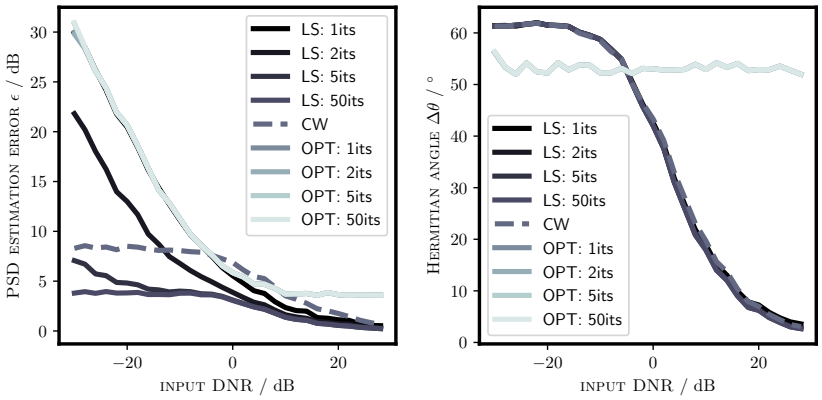


Fig. 4.9.: diffuse PSD and RETF vector estimation errors vs. the iteration index ($M = 4$, DNR $= 0\,\mathrm{dB}$)



(a) PSD estimation error **(b)** Hermitian angle

Fig. 4.10.: diffuse PSD and RETF vector estimation errors vs. input DNR for different numbers of iterations ($M = 4$)

is decreased by about 2 dB after convergence, which is reached after approximately 5 ALS iterations. Furthermore, the figure shows that using the LS method, lower PSD and RETF vector estimation errors are obtained than using CW. What is more, using unconstrained optimization to estimate the RETF vector (OPT) results in no performance increase for more iterations.

For different DNRs, Figure 4.10 compares the diffuse PSD and RETF vector estimation accuracy of the CW, OPT and LS method for $\{1,\ 2,\ 5,\ 50\}$ ALS iterations. In terms of the diffuse PSD estimation error, it can be observed that for only a few ALS iterations (i.e., 1 or 2), the LS as well as the OPT method perform significantly worse than the CW method. After convergence, however, the LS method clearly outperforms the CW and the OPT method for low input DNRs, while resulting in a similar estimation accuracy as CW at high input DNRs. 5 ALS iterations is observed to be a good compromise between computational complexity and estimation accuracy (note that the only significant difference between 5 and 50 iterations is observable for low input DNRs $< -10\,\mathrm{dB}$, cf. Figure 4.10a).

In terms of the Hermitian angle, there is no significant difference between the LS and the CW approach for all considered DNRs, with OPT again showing the worst accuracy. Hence, the ALS approach utilizing rank-1 approximation improves the PSD estimation accuracy with increasing number of ALS iterations, while one ALS iteration seems to suffice in terms of RETF vector estimation accuracy.

Concluding, the LS method exhibits a large robustness to model noise in artificial data, outperforming both the CW as well as the OPT method in terms of the diffuse PSD estimation accuracy, while leading to a comparable performance as CW in terms of the RETF vector estimation accuracy. Thus, utilizing the rank-1 approximation for the Frobenius norm-based RETF vector estimation proves superior to using unconstrained optimization[1], which is why OPT will not be considered in the validation using recorded data.

4.2.2 Validation Using Recorded Data

To evaluate the performance of the ALS and CW approaches on measured data, the acoustical scenarios that were already used in the evaluation of the power method-based diffuse PSD estimators are utilized (cf. Table 4.1). This enables a closer comparison with the results in Section 4.1, in which the RETF vector is assumed to be known. As mentioned above, the OPT method will not be considered in this validation, since it does not exhibit any advantages w.r.t. the LS method while aiming at minimizing the same cost function.

A spatially stationary speech source is considered in the presence of reverberation and diffuse babble noise. The reverberant multi-channel speech signals are obtained by convolving an 8.8 s long anechoic speech signal with the measured RIRs corresponding to AS_{1-3}. As for the simulated data, $M = 4$ microphones are used. Diffuse babble noise is generated as described in [28] and added at 10 dB input SNR w.r.t. the reference microphone.

As already mentioned, it should be noted that in realistic acoustical scenarios, deviations from the signal model in Equation 2.44 are to be expected, since the perfectly diffuse sound field model is generally violated and since the individual components are not perfectly uncorrelated. To investigate the impact of additional model mismatch, spatially uncorrelated noise is added to the microphone signals at different input DNRs ranging from 10 dB to 40 dB.

The signals are processed in their STFT representation, which is obtained using the parameters described in Section 4.1.1. The estimated microphone PSD matrix $\hat{\mathbf{\Phi}}_{\mathbf{y}}(l)$ is obtained using recursive averaging as described in Equation 2.31 with a smoothing constant $\alpha = 0.67$, corresponding to approximately 40 ms. Note that *in the first part* of this investigation, the uncorrelated noise PSD matrix is *not* estimated and subtracted from the microphone PSD matrix, as, e.g., done in [15, 16, 40] and Section 4.1.1, such that the sensitivity to uncorrelated noise can be evaluated. In a second part, it *is* estimated and subtracted from the microphone PSD matrix (cf. Section 4.1.1) to be able to compare the performance of both approaches in a more realistic setting.

The performance comparison is done by using the obtained RETF vector and diffuse PSD estimates in an MWF.

It was shown in [55, 15] that using the decision-directed approach [24] to obtain an estimate of the a priori SNR results in a better MWF performance than directly utilizing the Frobenius norm-based target PSD estimate. For this reason, the target PSD estimation accuracy has not been evaluated in Section 4.2.1, and the decision-directed approach is used to implement the MWF in this section (as also done in the simulations in Section 4.1). The a priori SNR $\xi(l)$ is estimated using Equation 2.27 with the smoothing constant $\rho = 0.98$. The MVDR beamformer coefficients $\mathbf{w}_{\text{MVDR}}(l)$ are computed as in Equation 4.6 when *not* subtracting the noise PSD

[1] Furthermore, the rank-1 approximation-based variant is far less computationally complex than the unconstrained optimization-based variant.

matrix, and as in Equation 4.7 *when subtracting* the noise PSD matrix. The postfilter $G(l)$ is computed using the a priori SNR estimate as

$$G(l) = \frac{\hat{\xi}(l)}{1 + \hat{\xi}(l)}, \tag{4.12}$$

with a minimum gain of -10 dB. The corresponding simulations are performed 5 times using *one* noise realization, such that the influence of the random initialization in case of the LS method is reduced. The mean values are displayed in addition to the standard deviation as error bars.

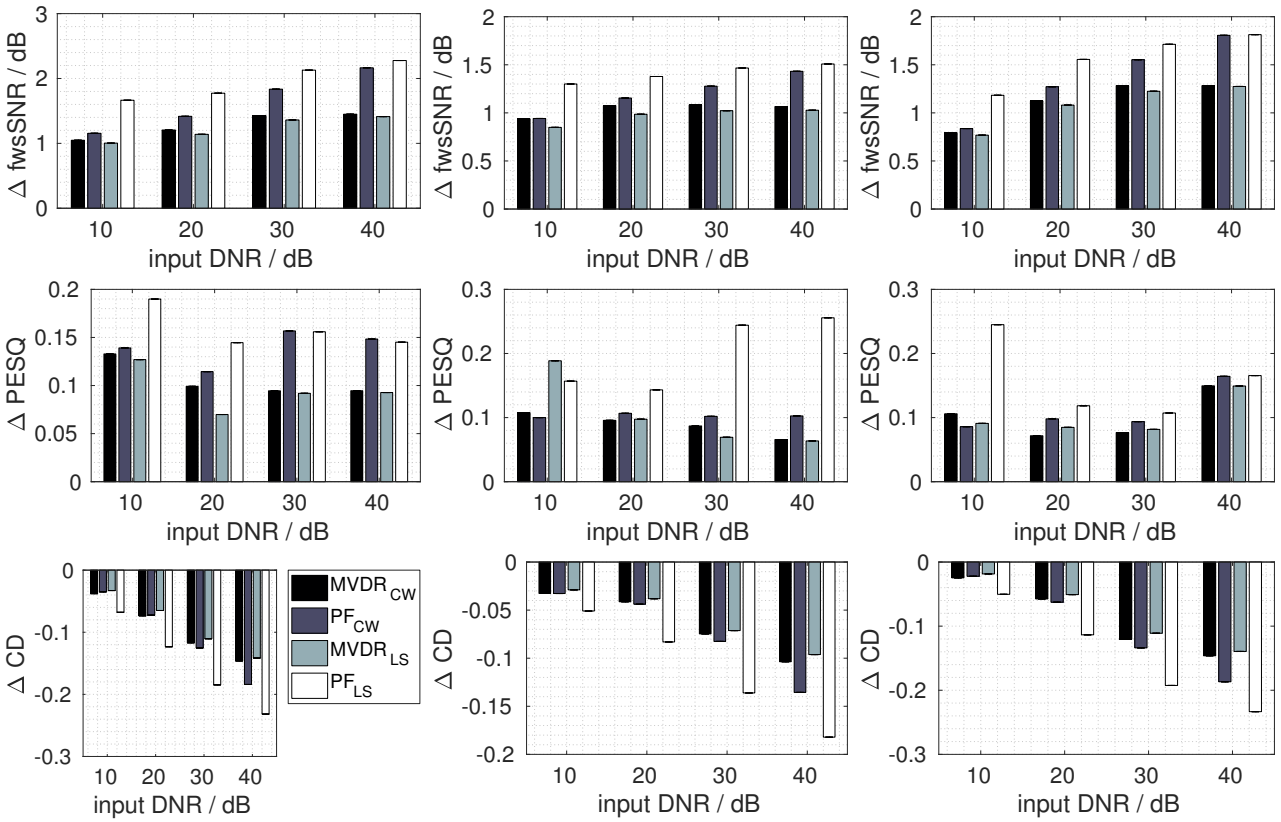


Fig. 4.11.: joint dereverberation and noise reduction *without* subtracting noise PSD matrix, $M = 4$, 5 ALS iterations; left to right column: AS_1, AS_2, AS_3

Without Subtraction of Noise PSD Matrix The quality of the MWF output signal, which is evaluated with fwsSNR, PESQ and CD using the target signal as the reference, is presented in Figure 4.11 for several input DNRs. 5 ALS iterations are used, as determined in the simulations using artificial data in the previous section. It can be observed that the LS method outperforms the CW method for all considered DNRs in terms of the considered performance measures, except for the PESQ values in AS_1 at 30 dB and 40 dB input DNR.
In terms of PESQ, a better performance of up to 0.16 is obtained using the LS method, whereas in terms of fwsSNR, a better performance of up to 0.4 dB is obtained. Also in terms of fwsSNR, the performance difference between both methods decreases for larger input DNRs, which is in line with the results from Figure 4.10. In terms of PESQ, the differences become most apparent.

Comparing the MVDR performance, which only depends on the RETF estimation accuracy when the noise PSD matrix is not subtracted (cf. Equation 4.6), confirms the result from Figure 4.10b, which demonstrated a similar Hermitian angle between the ground truth and the estimates. Noting that in almost all cases, the performance is increased when the postfilter is added, also the higher diffuse PSD estimation accuracy of the LS method is confirmed.

The error bars are invisible in almost all cases, leading to the conclusion that random initialization of the RETF vector in case of the LS method does not have a significant impact.

In summary, if strong model deviations are present, the proposed LS method yields a significantly better performance than the CW method when used in an MWF, confirming the advantages of coupling the RETF vector and diffuse PSD estimation in an ALS approach.

With Subtraction of Noise PSD Matrix The results when subtracting the uncorrelated noise PSD matrix from the microphone PSD matrix are presented in Figure 4.12. Otherwise, the simulation settings are equal to those from the previous section.

When compared to the results *without* noise PSD matrix subtraction, some differences can be observed. Considering the performance of the MVDR only, no significant difference between the methods can be made out. This confirms the results from the validation using artificial data (cf. Section 4.2.1), in which the Hermitian angle between the estimates and the reference was approximately the same for the LS and the CW method. Note, however, that the MVDR implementation here also includes the diffuse PSD estimate (cf. Equation 4.7).
Focusing on the performance of the MWF, the CW method tends to slightly better results (although mostly insignificant). Hence, in the case of real data, the diffuse PSD estimate of the LS method seems to be slightly more inaccurate than the one of the CW method, which contradicts the results from the validation using artificial data. A possible explanation of this observation is a faulty scaling of the RETF vector during the ALS iterations ($\hat{\mathbf{a}} = \sqrt{\lambda_1 / \hat{\phi}_s}\mathbf{u}_1$), which depends on the target PSD estimate $\hat{\phi}_s$. During the simulations, $\hat{\phi}_s$ tends to take on values in the range of machine precision, which may lead to numerical inaccuracies and thus to wrongly scaled RETF vector estimates. While such a faulty scaling does not affect the *direction* of the vector (and hence not the Hermitian angle), it *does* affect the PSD estimation (note, e.g., the first element of $\mathbf{A}(l)$ in Equation 2.48).

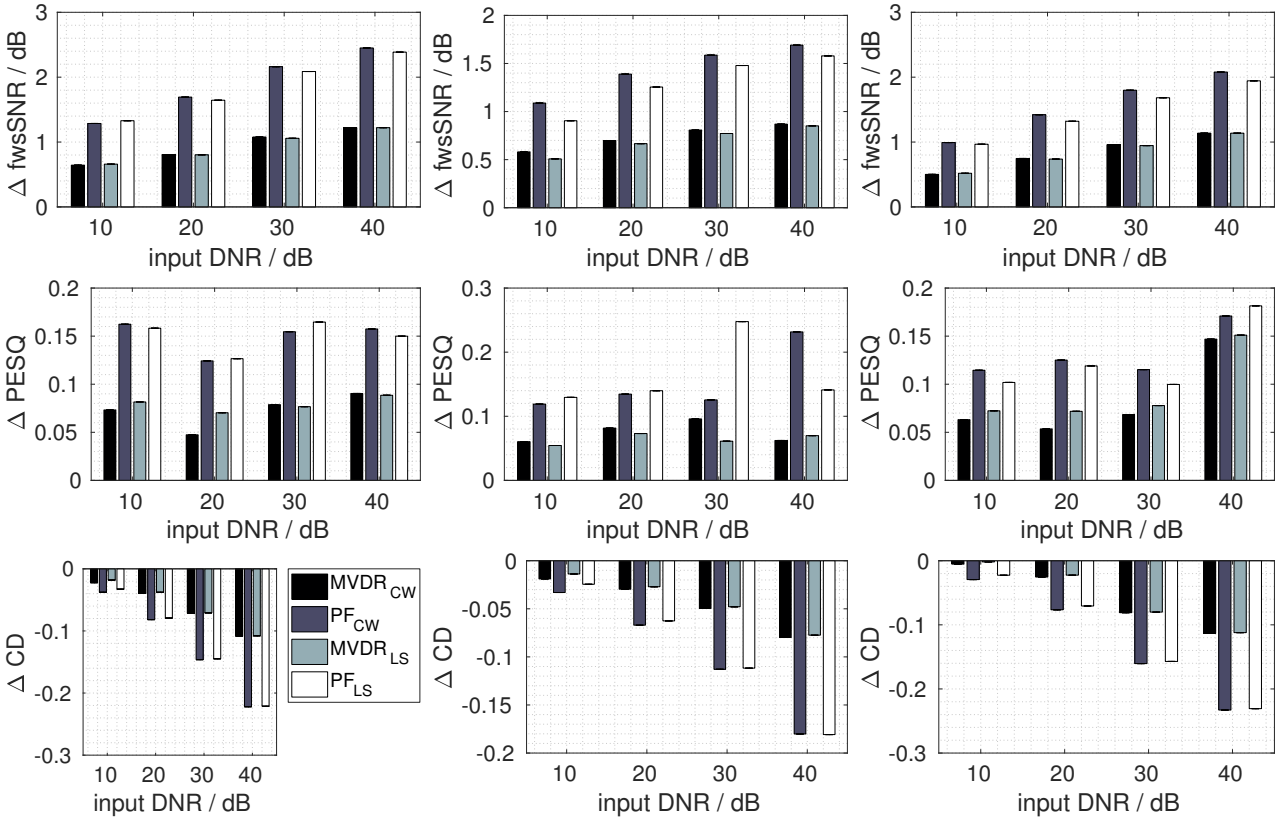


Fig. 4.12.: joint dereverberation and noise reduction *with* subtracting noise PSD matrix, $M = 4$, 5 ALS iterations; left to right column: AS_1, AS_2, AS_3

Summary

In this section, the evaluation results for the joint estimation of the RETF vector and diffuse PSD estimates are summarized. Firstly, in the case of artificial data, it is shown that the LS method leads to a higher PSD estimation accuracy than the CW method, when uncorrelated noise is present. For the RETF vector estimation accuracy, no difference can be made out.
This result is confirmed for real data, as long as the noise PSD matrix is *not* subtracted, hinting at the larger robustness of the LS method to model deviations.

On the other hand, when the noise PSD matrix *is* subtracted, and hence the strength of the model deviations is decreased, the LS method has a similar RETF vector estimation accuracy as the CW method (leading to a similar MVDR performance), while its diffuse PSD estimation accuracy is slightly lower (leading to a slightly lower MWF performance). This discrepancy may be caused by numerical imprecisions during the scaling of the RETF vector.

As in the previous section, there is no significant variation between independent simulations, confirming that the LS method does not depend on favorably initialized RETF vectors.

In conclusion, the CW method seems to have a higher potential for joint dereverberation and denoising, as long as the model deviation is low. However, with increasing model deviation, the LS method will become advantageous, depending on the scenario.
Since it is possible to estimate the noise PSD matrix in noise-only phases, it is reasonable to prefer the CW method in this state. For future research, however, it remains interesting to incorporate the PSD estimator described in [55] in the LS method, which can additionally provide an estimate of the uncorrelated noise PSD, possibly improving the performance.

5 Conclusion and Outlook

In this chapter, the main contributions of this thesis are summarized, and opportunities for further research are pointed out.

5.1 Conclusion

Recent developments in hand-held devices have led to an increased interaction of humans and machines, be it as a ways of communication or with regard to assisted living technologies, where automatic speech recognition is of utmost importance. Commonly, these systems involve a small set of microphones used to acquire a speech signal in a room of limited size. However, in addition to the desired speech, also interferences are captured by the microphones, including ambient noise and late reverberation. A commonly used technique for joint dereverberation and denoising is the MWF. Modeling late reverberation and ambient noise as a diffuse sound field, it requires estimates of the RETFs and the diffuse PSD.

The first aim of this thesis was to reduce the computational complexity of an existing diffuse PSD estimator, such that its practical implementation becomes more feasible. For this problem, we have proposed to use the power method. Regarding the initialization method, it was shown that using the estimated dominant eigenvector from the previous time frame results in the fastest convergence. In terms of estimation error, however, no significant difference between the different initialization methods could be observed.
In the case that no, or only diffuse, noise is present, even a small number of iterations, i.e., $N = 2$, is sufficient to obtain the same diffuse PSD estimation accuracy as the full EVD. In the presence of additional non-diffuse noise, the required amount of iterations increases, and for a low input SNR, the diffuse PSD estimation accuracy of the full EVD and the power method diverges to a small extent. Considering that in many realistic scenarios the input SNR does not fall too low, the power method remains a suitable and computationally more efficient variant of obtaining the required eigenvalues.

The second aim was to jointly estimate the RETFs and the diffuse PSD, since the implementation of the considered speech enhancement framework, i.e., the MWF, requires *both* of these quantities. For this purpose, we proposed an ALS approach to jointly estimate the RETFs as well as the diffuse and target PSDs. It was shown that using these estimates in an MWF leads to a high dereverberation and denoising performance. When compared to a state-of-the-art approach based on CW, a higher robustness against model deviations could be observed even for a small number of ALS iterations. In the case that these model deviations were estimated in noise-only segments and then subtracted from the microphone signal, however, a similar performance of both techniques was obtained.

5.2 Further Research

The signal model used throughout this work includes any noise that may be represented by a diffuse sound field. Hence, any noise which does *not* satisfy this assumption, embodies a deviation from the signal model, leading to a decrease in the performance of the proposed estimators.

While it is not straightforward to extend the EVD-based diffuse PSD estimator such that it includes non-diffuse noise inherently, the case for the ALS approach based on the minimization of the Frobenius norm of an error matrix is different. Modeling the noise PSD matrix as a spatially homogeneous coherence matrix with time-varying PSD, it is possible to incorporate non-diffuse noise into the model, which is highly interesting for future work. Furthermore, since the ALS approach yields the *time-varying* RETFs, the performance in the case of a moving speaker is subject of possible further investigation.

Appendix

A

A.1 Supplementary Plots

In this section, figures are depicted that support the findings made in the experimental analysis in Section 4.

A.1.1 PSD Estimation Using EVD

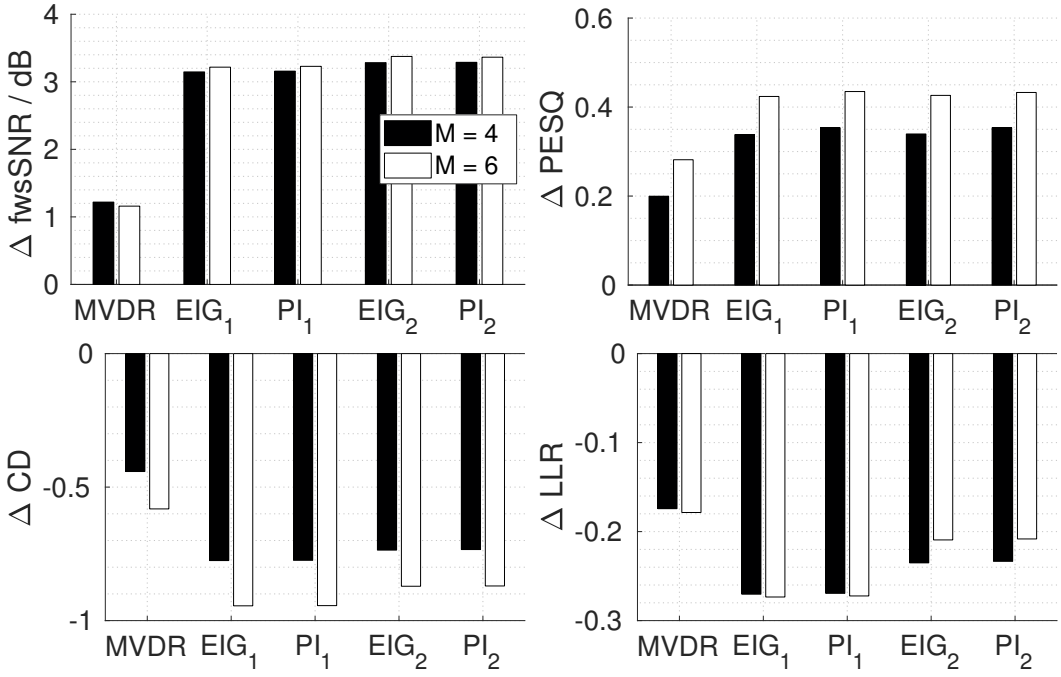


Fig. A.1.: MWF performance for $M = \{4, 6\}$, AS_2, diffuse babble noise at 10 dB input SNR

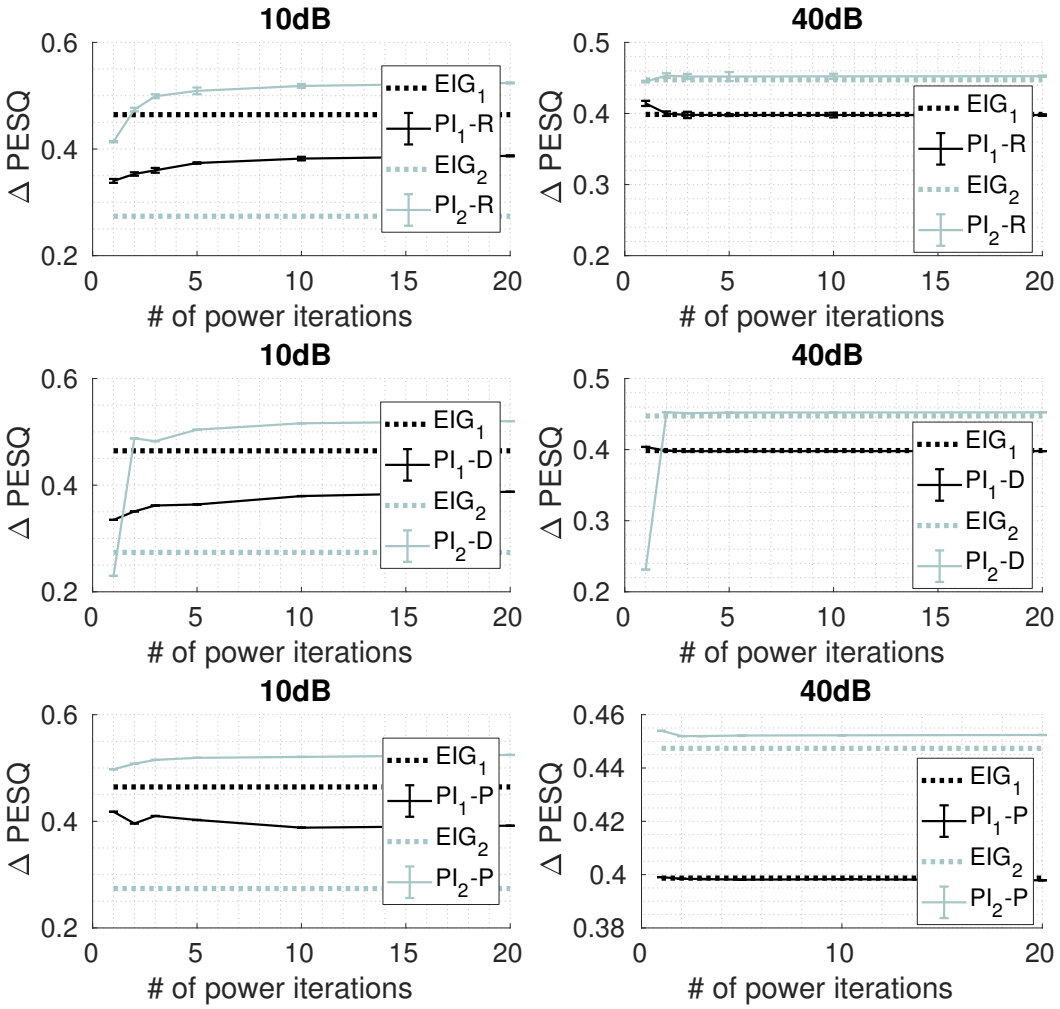
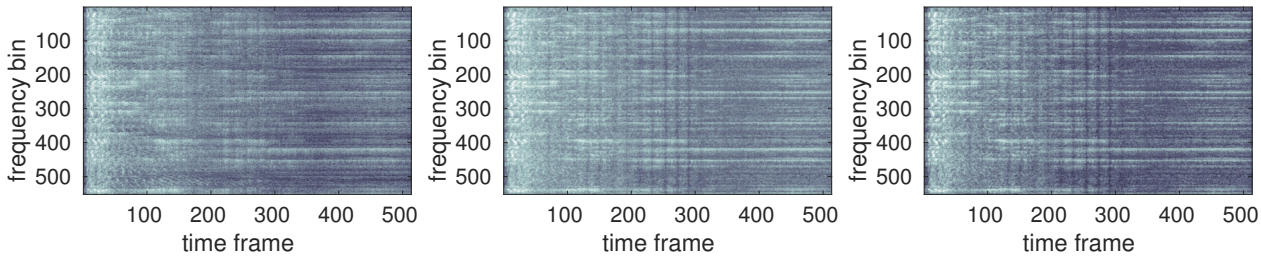


Fig. A.2.: power method convergence rate with random, deterministic and previous initialization at different uncorrelated noise input SNRs evaluated using MWF performance; AS_2



(a) reference microphone signal

(b) output of MWF using diffuse PSD estimate $\hat{\phi}_d^{\text{EIG2}}$

(c) output of MWF using diffuse PSD estimate $\hat{\phi}_d^{\text{PI2}}$

Fig. A.3.: comparison of STFTs for AS_1; diffuse babble noise at 10 dB input SNR

A.2 Some Proofs

In this section, proofs are presented that would otherwise have disturbed the reading flow.

Theorem A.2.1. *If a matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ has a rank equal to 1 — i.e., it can be written as $\mathbf{A} = \mathbf{u}\mathbf{u}^H$ with $\mathbf{u}$ being an M-dimensional non-zero vector — it has only one non-zero eigenvalue σ.*

Proof. Knowing that $\mathbf{u}$ is an eigenvector of the matrix $\mathbf{A} = \mathbf{u}\mathbf{u}^H$, the non-zero eigenvalue of $\mathbf{A}$ is easily obtained as

$$\mathbf{A}\mathbf{u} = \mathbf{u}\mathbf{u}^H\mathbf{u} = \mathbf{u}\underbrace{(\mathbf{u}^H\mathbf{u})}_{=:\sigma>0} = \sigma\mathbf{u}.$$

For the other eigenvalues, by the rank-nullity theorem we have

$$\dim(\ker(\mathbf{A})) = M - 1,$$

where $\dim(\circ)$ and $\ker(\circ)$ are the dimensionality and kernel of $\circ$, respectively. Noting the definition of the kernel

$$\ker(\mathbf{A}) = \{\mathbf{x} \in \mathbb{C}^M | \mathbf{A}\mathbf{x} = \mathbf{0}\} =: \mathbf{x}_0 \leftrightarrow \mathbf{A}\mathbf{x}_0 = \mathbf{0} = 0\mathbf{x}_0,$$

we can see that it corresponds to the eigenspace of $\mathbf{A}$ corresponding to the eigenvalue 0, i.e., the eigenspace of $\mathbf{A}$ corresponding to the eigenvalue 0 has dimensionality $M - 1$. Hence, $\mathbf{A}$ has one non-zero eigenvalue σ, with the eigenvalue 0 having an algebraic multiplicity of $M - 1$. □

Theorem A.2.2. *The squared Frobenius norm $||\mathbf{A}||_F^2$ of matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ may be rewritten using the trace as $||\mathbf{A}||_F^2 = \mathrm{tr}\left\{\mathbf{A}^{\mathrm{H}}\mathbf{A}\right\} = \mathrm{tr}\left\{\mathbf{A}\mathbf{A}^{\mathrm{H}}\right\}$.*

Proof.

$$\begin{aligned}
\mathrm{tr}\{\mathbf{A}^{\mathrm{H}}\mathbf{A}\} &= \sum_{i=1}^{M}(\mathbf{A}^H\mathbf{A})_{ii} \\
&= \sum_{i=1}^{M}\sum_{k=1}^{M} a_{ki}^* a_{ki} \left(= \sum_{i=1}^{M}\sum_{k=1}^{M} a_{ki} a_{ki}^* = \mathrm{tr}\{\mathbf{A}\mathbf{A}^{\mathrm{H}}\}\right) \qquad \text{(A.1)} \\
&= \sum_{i=1}^{M}\sum_{k=1}^{M} |a_{ki}|^2 \equiv ||\mathbf{A}||_F^2
\end{aligned}$$

□

Theorem A.2.3. *The squared Frobenius norm $||\mathbf{A}||_F^2$ of matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ may be separated into its imaginary and real part according to $||\mathbf{A}||_F^2 = ||\mathrm{Re}\{\mathbf{A}\}||_{\mathrm{F}}^2 + ||\mathrm{Im}\{\mathbf{A}\}||_{\mathrm{F}}^2$.*

Proof.

$$
\begin{aligned}
||\mathbf{A}||_F^2 &= \sum_{i=1}^{M}\sum_{j=1}^{M}|a_{ij}|^2 \\
&= \sum_{i=1}^{M}\sum_{j=1}^{M}a_{ij}a_{ij}^* \\
&= \sum_{i=1}^{M}\sum_{j=1}^{M}(a_{ij}^R + ja_{ij}^I)(a_{ij}^R - ja_{ij}^I) \\
&= \sum_{i=1}^{M}\sum_{j=1}^{M}(a_{ij}^{R,2} \underbrace{-ja_{ij}^R a_{ij}^I + ja_{ij}^I a_{ij}^R}_{=0} + a_{ij}^{I,2}) \\
&= \sum_{i=1}^{M}\sum_{j=1}^{M}a_{ij}^{R,2} + \sum_{i=1}^{M}\sum_{j=1}^{M}a_{ij}^{I,2} \\
&= ||\mathrm{Re}\{\mathbf{A}\}||_{\mathrm{F}}^2 + ||\mathrm{Im}\{\mathbf{A}\}||_{\mathrm{F}}^2,
\end{aligned}
\tag{A.2}
$$

□

where a_{ij}^R and a_{ij}^I denote the real and imaginary part of the (i,j)-th component of $\mathbf{A}$, respectively.

Theorem A.2.4. *The matrix $\mathbf{A} = \phi_s\mathbf{b}\mathbf{b}^H + \phi_d\mathbf{I}_M$ with ϕ_s, $\phi_d \in \mathbb{R}^+$, $\mathbf{b} \in \mathbb{C}^M$ has $\mathbf{b}$ as its dominant eigenvector.*

Proof. The proof starts by showing that $\mathbf{b}$ satisfies the eigenvalue equation.

$$
\begin{aligned}
(\phi_s\mathbf{b}\mathbf{b}^H + \phi_d\mathbf{I}_M)\mathbf{b} &= \phi_s\mathbf{b}\overbrace{\mathbf{b}^H\mathbf{b}}^{=:b^*>0} + \phi_d\mathbf{I}_M\mathbf{b} \\
&= (\phi_s b^* + \phi_d)\mathbf{b}
\end{aligned}
\tag{A.3}
$$

Now, using that the eigenvalues of $\mathbf{A}$ can be identified as $\lambda_1\{\mathbf{A}\} = \sigma + \phi_d > \lambda_i\{\mathbf{A}\} = \phi_d$, $i = \{2,\ldots,M\}$ (see Section 2.4.2), $\mathbf{b}$ is shown to be the eigenvector of $\mathbf{A}$ corresponding to its largest eigenvalue. □

Theorem A.2.5. *If a matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ is (upper) triangular, then its eigenvalues λ_i and the corresponding multiplicities are given as its diagonal entries a_{ii}.*

Proof. We use that the determinant of any (upper) triangular matrix is given by the product of its diagonal entries, i.e.,

$$\det\{\mathbf{A}\} = \prod_{i=1}^{M} a_{ii}.$$

Since the eigenvalues of $\mathbf{A}$ are given by the solutions of its characteristic polynomial, we have

$$
\begin{aligned}
\det\{\mathbf{A} - \lambda\mathbf{I}_M\} =: \det\{\mathbf{B}\} \quad & \mathbf{B} \text{ (also (upper) triangular)} \\
&= \prod_{i=1}^{M} b_{ii} \\
&= (b_{11})(b_{22})\ldots(b_{MM}) \\
&= (a_{11} - \lambda)(a_{22} - \lambda)\ldots(a_{MM} - \lambda),
\end{aligned}
$$

where the roots are clearly given by the diagonal entries themselves. □

Lemma A.2.1. *If a matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ is Hermitian, i.e., $\mathbf{A}^H = \mathbf{A}$, then its eigenvectors are orthogonal.*

Proof. Assume that the matrix $\mathbf{A}$ has eigenvalues λ_i, λ_j and corresponding eigenvectors $\mathbf{u}_i$, $\mathbf{u}_j$. Using that the eigenvalues of the above Hermitian matrix $\mathbf{A}$ are real yields

$$\begin{aligned}\lambda_j(\mathbf{u}_j^H\mathbf{u}_k) = (\lambda_j\mathbf{u}_j)^H\mathbf{u}_k = (A\mathbf{u}_j)^H\mathbf{u}_k = \mathbf{u}_j^H\mathbf{A}^H\mathbf{u}_k = \mathbf{u}_j^H\mathbf{A}\mathbf{u}_k = \mathbf{u}_j^H\lambda_k\mathbf{u}_k = \lambda_k\mathbf{u}_j^H\mathbf{u}_k \\ \rightarrow \lambda_j\mathbf{u}_j^H\mathbf{u}_k = \lambda_k\mathbf{u}_j^H\mathbf{u}_k \\ \leftrightarrow \mathbf{u}_j^H\mathbf{u}_k = 0, \quad \text{since} \quad 0 \neq \lambda_j \neq \lambda_k \neq 0, \quad j \neq k.\end{aligned}$$

□

The above proof uses the fact that Hermitian matrices have real eigenvalues:

Lemma A.2.2. *The eigenvalues of a Hermitian matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ with eigenvector $\mathbf{u}$ are real.*

Proof.

$$\begin{aligned}\lambda(\mathbf{u}^H\mathbf{u}) = \mathbf{u}^H\lambda\mathbf{u} = \mathbf{u}^H\mathbf{A}\mathbf{u} = \mathbf{u}^H\mathbf{A}^H\mathbf{u} = (\mathbf{A}\mathbf{u})^H\mathbf{u} = (\lambda\mathbf{u})^H\mathbf{u} = \lambda^*\mathbf{u}^H\mathbf{u} \\ \rightarrow \lambda = \lambda^* \quad \leftrightarrow \quad \lambda \in \mathbb{R}\end{aligned}$$

□

A.3 Cholesky Decomposition

Since the Cholesky decomposition, which is an example for a square root decomposition of matrices, is an important step in obtaining a favorable structure for the microphone PSD matrix, the technique is outlined briefly in this section, based on [43].

Theorem A.3.1. *If a matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$ is symmetric and positive definite, it can be decomposed as $\mathbf{A} = \mathbf{L}\mathbf{L}^H$, where $\mathbf{L} \in \mathbb{C}^{M\times M}$ is a lower triangular matrix.*

Proof. See [43]. □

There is a variety of algorithms achieving the Cholesky decomposition, with the most efficient ones requiring $M^3/3$ flops. As an example, the so-called "general matrix $\mathbf{A}$ times vector $\mathbf{x}$ plus vector $\mathbf{y}$ (GAXPY)" version is presented in the following, since it relies heavily on GAXPY operations, which are computed efficiently in MATLAB. To arrive at the algorithm, which has

been slightly modified from [43] to account for Hermitian instead of symmetric matrices (i.e., complex instead of real), we first write explicitly the j-th column of $\mathbf{A}$ as

$$\begin{aligned}\mathbf{A}(:,j) &= \sum_{k=1}^{j} \mathbf{L}(:,k)\mathbf{L}^H(k,j) \\ &= \sum_{k=1}^{j} \mathbf{L}(:,k)\mathbf{L}^*(j,k) \\ &= \sum_{k=1}^{j-1} \mathbf{L}(:,k)\mathbf{L}^*(j,k) + \mathbf{L}(:,j)\mathbf{L}^*(j,j) \\ \leftrightarrow \mathbf{L}(:,j)\mathbf{L}^*(j,j) =: \mathbf{v} &= \mathbf{A}(:,j) - \sum_{k=1}^{j-1} \mathbf{L}(:,k)\mathbf{L}^*(j,k).\end{aligned} \tag{A.4}$$

Hence, knowing the input matrix $\mathbf{A}$ and the first $j-1$ columns of $\mathbf{L}$, and noting that $\mathbf{L}^*(j,j) = \sqrt{\mathbf{v}(j)}$, its j-th column can be computed, resulting in the following algorithm:

Algorithm A.1: Cholesky decomposition (GAXPY variant)

Input: Hermitian and positive definite matrix $\mathbf{A} \in \mathbb{C}^{M\times M}$
Output: lower triangular matrix $\mathbf{L}$ such that $\mathbf{A} = \mathbf{L}\mathbf{L}^H$
init $\mathbf{v} \in \mathbf{0}^M, \mathbf{L} \in \mathbf{0}^{M\times M}$;
for $j = 1 : M$ **do**
  $\mathbf{v}(j:M) = \mathbf{A}(j:M, j)$;
  for $k = 1 : j-1$ **do**
    $\mathbf{v}(j:M) = \mathbf{v}(j:M) - \mathbf{L}(j,k)\mathbf{L}(j:M,k)$;
  $\mathbf{L}(j:M, j) = \mathbf{v}(j:M)/\sqrt{\mathbf{v}(j)}$;

A.4 Derivatives

To obtain the gradient in Equation 3.1.2, a number of matrix derivatives w.r.t. vectors is required. To this end, modified results from [56] are presented in this section which are used in the computations.

Tab. A.1.: gradients modified from [56], $\mathbf{A} \in \mathbb{R}^{M\times M}, \mathbf{A} = \mathbf{A}^T, \quad \mathbf{x}, \mathbf{a} \in \mathbb{R}^M$

$f(\mathbf{x})$	$\nabla_{\mathbf{x}} f(\mathbf{x})$
$\mathrm{tr}\{\mathbf{A}\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$	$2\mathbf{A}\mathbf{x}$
$\mathrm{tr}\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{A}\}$	$2\mathbf{A}\mathbf{x}$
$\mathrm{tr}\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}\}$	$4\mathbf{x}\mathbf{x}^T\mathbf{x}$
$\mathrm{tr}\{\mathbf{A}^{\mathrm{T}}\mathbf{x}\mathbf{a}^{\mathrm{T}}\}$	$\mathbf{A}\mathbf{a}$
$\mathrm{tr}\{\mathbf{A}^{\mathrm{T}}\mathbf{a}\mathbf{x}^{\mathrm{T}}\}$	$\mathbf{A}^T\mathbf{a}$
$\mathrm{tr}\{\mathbf{x}\mathbf{a}^{\mathrm{T}}\mathbf{a}\mathbf{x}^{\mathrm{T}}\}$	$2\mathbf{x}\mathbf{a}^T\mathbf{a}$
$\mathrm{tr}\{\mathbf{a}\mathbf{x}^{\mathrm{T}}\mathbf{a}\mathbf{x}^{\mathrm{T}}\}$	$2\mathbf{a}\mathbf{x}^T\mathbf{a}$

Bibliography

[1]R. Beutelmann and T. Brand. „Prediction of speech intelligibility in spatial noise and reverberation for normal-hearing and hearing-impaired listeners“. In: *Journal of the Acoustical Society of America* 120.1 (July 2006), pp. 331–342 (cit. on p. 1).

[2]A. Warzybok, I. Kodrasi, J. O. Jungmann, et al. „Subjective speech quality and speech intelligibility evaluation of single-channel dereverberation algorithms“. In: *Proc. International Workshop on Acoustic Echo and Noise Control*. Antibes, France, Sept. 2014, pp. 333–337 (cit. on p. 1).

[3]T. Yoshioka, A. Sehr, M. Delcroix, et al. „Making Machines Understand Us in Reverberant Rooms: Robustness against Reverberation for Automatic Speech Recognition“. In: *IEEE Signal Processing Magazine* 29.6 (Nov. 2012), pp. 114–126 (cit. on p. 1).

[4]S. Doclo and M. Moonen. „Combined frequency-domain dereverberation and noise reduction technique for multi-microphone speech enhancement“. In: *Proc. International Workshop on Acoustic Echo and Noise Control*. Darmstadt, Germany, Sept. 2001, pp. 31–34 (cit. on p. 1).

[5]E. A. P. Habets and J. Benesty. „A Two-Stage Beamforming Approach for Noise Reduction and Dereverberation“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.5 (May 2013), pp. 945–958 (cit. on p. 1).

[6]B. Cauchi, I. Kodrasi, R. Rehr, et al. „Combination of MVDR beamforming and single-channel spectral processing for enhancing noisy and reverberant speech“. In: *EURASIP Journal on Advances in Signal Processing* 2015.1 (2015) (cit. on p. 1).

[7]K. U. Simmer, J. Bitzer, and C. Marro. „Post-filtering techniques“. In: *Microphone Arrays*. Ed. by M. Brandstein and D. Ward. Berlin, Germany: Springer, 2001 (cit. on pp. 1, 7).

[8]S. Markovich-Golan, S. Gannot, and I. Cohen. „Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.6 (Aug. 2009), pp. 1071–1086 (cit. on pp. 1, 15, 19).

[9]S. Markovich-Golan and S. Gannot. „Performance analysis of the covariance subtraction method for relative transfer function estimation and comparison to the covariance whitening method“. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, Apr. 2015, pp. 544–548 (cit. on pp. 1, 15, 19).

[10]I. Cohen. „Relative transfer function identification using speech signals“. In: *IEEE Transactions on Speech and Audio Processing* 12.5 (2004), pp. 451–459 (cit. on pp. 1, 15, 19).

[11]O. Schwartz, S. Gannot, and E. A. P. Habets. „Multi-Microphone Speech Dereverberation and Noise Reduction Using Relative Early Transfer Functions“. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.2 (Feb. 2015), pp. 240–251 (cit. on pp. 1, 15, 19).

[12]A. Kuklasiński, S. Doclo, S. H. Jensen, and J. Jensen. „Maximum likelihood PSD estimation for speech enhancement in reverberation and noise“. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24.9 (Sept. 2016), pp. 1595–1608 (cit. on pp. 1, 10, 12, 19).

[13]O. Schwartz, S. Braun, S. Gannot, and E. A. P. Habets. „Maximum likelihood estimation of the late reverberant power spectral density in noisy environments“. In: *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. New York, USA, Oct. 2015 (cit. on pp. 1, 10, 12, 19).

[14]S. Braun and E. A. P. Habets. „Dereverberation in noisy environments using reference signals and a maximum likelihood estimator“. In: *Proc. European Signal Processing Conference*. Marrakech, Morocco, Sept. 2013 (cit. on pp. 1, 10, 12).

[15]O. Schwartz, S. Gannot, and E. A. P. Habets. „Joint estimation of late reverberant and speech power spectral densities in noisy environments using Frobenius norm“. In: *Proc. European Signal Processing Conference*. Budapest, Hungary: IEEE, Sept. 2016, pp. 1123–1127 (cit. on pp. 1, 10, 12, 14, 19, 46).

[16]I. Kodrasi and S. Doclo. „Analysis of Eigenvalue Decomposition-Based Late Reverberation Power Spectral Density Estimation“. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26.6 (June 2018), pp. 1106–1118 (cit. on pp. 1, 19, 46).

[17]I. Kodrasi and S. Doclo. „EVD-based multi-channel dereverberation of a moving speaker using different RETF estimation methods“. In: *Proc. Joint Workshop on Hands-Free Speech Communication and Microphone Arrays*. San Francisco, USA: IEEE, Mar. 2017, pp. 116–120 (cit. on pp. 1, 12, 19, 20).

[18]J. S. Bradley, H. Sato, and M. Picard. „On the importance of early reflections for speech in rooms“. In: *Journal of the Acoustical Society of America* 113.6 (June 2003), pp. 3233–3244 (cit. on p. 2).

[19]J. G. Proakis and D. G. Manolakis. *Digital signal processing*. 3rd ed. Prentice Hall, 1996 (cit. on p. 5).

[20]S. Doclo, A. Spriet, J. Wouters, and M. Moonen. „Frequency-domain criterion for the speech distortion weighted multichannel Wiener filter for robust noise reduction“. In: *Speech Communication* 49.7 (July 2007), pp. 636–656 (cit. on p. 7).

[21]T. Van den Bogaert, S. Doclo, J. Wouters, and M. Moonen. „Speech enhancement with multichannel Wiener filter techniques in multimicrophone binaural hearing aids“. In: *The Journal of the Acoustical Society of America* 125.1 (2009), pp. 360–371 (cit. on p. 7).

[22]P. A. Naylor and N. D. Gaubitch, eds. *Speech dereverberation*. London, UK: Springer, 2010 (cit. on pp. 8, 11).

[23]S. Haykin and K. J. R. Liu. *Handbook on array processing and sensor networks*. Vol. 63. John Wiley & Sons, 2010 (cit. on p. 9).

[24]Y. Ephraim and D. Malah. „Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator“. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 32.6 (Dec. 1984), pp. 1109–1121 (cit. on pp. 9–11, 46).

[25]M. Berouti, R. Schwartz, and J. Makhoul. „Enhancement of speech corrupted by acoustic noise“. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Washington, USA, Apr. 1979, pp. 208–211 (cit. on p. 9).

[26]O. Thiergart and E. A. P. Habets. „Extracting reverberant sound using a linearly constrained minimum variance spatial filter“. In: *IEEE Signal Processing Letters* 21.5 (May 2014), pp. 630–634 (cit. on p. 10).

[27]O. Schwartz, S. Gannot, and E. A. P. Habets. „Joint maximum likelihood estimation of late reverberant and speech power spectral density in noisy environments“. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Shanghai, China, Mar. 2016, pp. 151–155 (cit. on pp. 10, 12).

[28]E. A. P. Habets, I. Cohen, and S. Gannot. „Generating nonstationary multisensor signals under a spatial coherence constraint“. In: *Journal of the Acoustical Society of America* 124.5 (Nov. 2008), pp. 2911–2917 (cit. on pp. 10, 46).

[29]I. Kodrasi and S. Doclo. „Late reverberant power spectral density estimation based on an eigenvalue decomposition“. In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*. New Orleans, USA, Mar. 2017, pp. 611–615 (cit. on pp. 12, 14).

[30]S. Gannot, D. Burshtein, and E. E. Weinstein. „Signal enhancement using beamforming and nonstationarity with applications to speech“. In: *IEEE Transactions on Signal Processing* 49.8 (Aug. 2001), pp. 1614–1626 (cit. on pp. 15, 19).

[31]R. Varzandeh, M. Taseska, and E. A. P. Habets. „An iterative multichannel subspace-based covariance subtraction method for relative transfer function estimation“. In: *Proc. Joint Workshop on Hands-Free Speech Communication and Microphone Arrays*. 2017 Hands-free Speech Communications and Microphone Arrays (HSCMA). San Francisco, USA, Mar. 2017, pp. 11–15 (cit. on pp. 15, 19, 44).

[32]E. Warsitz and R. Haeb-Umbach. „Blind Acoustic Beamforming Based on Generalized Eigenvalue Decomposition“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.5 (July 2007), pp. 1529–1539 (cit. on pp. 15, 19).

[33]ITU-T. *Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs P.862*. International Telecommunications Union (ITU-T) Recommendation, Feb. 2001 (cit. on p. 16).

[34]Y. Hu and P. C. Loizou. „Evaluation of Objective Quality Measures for Speech Enhancement“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.1 (Jan. 2008), pp. 229–238 (cit. on pp. 16–18).

[35]J. H. L. Hansen and B. L. Pellom. „An effective quality evaluation protocol for speech enhancement algorithms“. In: *Proc. International Conference on Spoken Language Processing*. Vol. 7. 1998, pp. 2819–2822 (cit. on p. 17).

[36]K. Kinoshita, M. Delcroix, S. Gannot, et al. „A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research“. In: *EURASIP Journal on Advances in Signal Processing* 2016.1 (Jan. 2016) (cit. on pp. 17, 31).

[37]L. R. Rabiner, R. W. Schafer, et al. „Introduction to digital speech processing“. In: *Foundations and Trends® in Signal Processing* 1.1 (2007), pp. 1–194 (cit. on p. 17).

[38]M. P. Norton and D. G. Karczub. *Fundamentals of noise and vibration analysis for engineers*. Cambridge university press, 2003 (cit. on p. 17).

[39]A. M. Noll. „Short-Time Spectrum and "Cepstrum" Techniques for Vocal-Pitch Detection". In: *The Journal of the Acoustical Society of America* 36.2 (1964), pp. 296–302 (cit. on p. 18).

[40]S. Braun and E. A. P. Habets. „A multichannel diffuse power estimator for dereverberation in the presence of multiple sources". In: *EURASIP Journal on Applied Signal Processing* 2015.1 (Dec. 2015) (cit. on pp. 19, 46).

[41]R. Fletcher. „A new approach to variable metric algorithms". In: *The Computer Journal* 13.3 (Mar. 1970), pp. 317–322 (cit. on p. 20).

[42]C. Eckart and G. Young. „The approximation of one matrix by another of lower rank". In: *Psychometrika* 1.3 (1936), pp. 211–218 (cit. on p. 21).

[43]G. H. Golub and C. F. Van Loan. *Matrix Computations*. Vol. 10. 8. Baltimore, USA: The John Hopkins University Press, 1957 (cit. on pp. 23, 24, 27, 28, 30, 57, 58).

[44]H. Wielandt. „Das Iterationsverfahren bei nicht selbstadjungierten linearen Eigenwertaufgaben". In: *Mathematische Zeitschrift* 50.1 (1944), pp. 93–143 (cit. on pp. 24, 25).

[45]G. H. Golub and H. A. Van der Vorst. „Eigenvalue computation in the 20th century". In: *Journal of Computational and Applied Mathematics* 123.1 (2000), pp. 35–65 (cit. on p. 26).

[46]W. E. Arnoldi. „The principle of minimized iterations in the solution of the matrix eigenvalue problem". In: *Quarterly of applied mathematics* 9.1 (1951), pp. 17–29 (cit. on p. 27).

[47]D. C. Sorensen. „Implicit application of polynomial filters in ak-step Arnoldi method". In: *SIAM Journal on Matrix Analysis and Applications* 13.1 (1992), pp. 357–385 (cit. on p. 29).

[48]R. G. Ayoub. „Paolo Ruffini's contributions to the quintic". In: *Archive for history of exact sciences* 23.3 (1980), pp. 253–277 (cit. on p. 29).

[49]G. Cardano. *Ars magna or the rules of algebra*. Dover Publications, 1968 (cit. on p. 29).

[50]J. M. Ortega and H. F. Kaiser. „The LLT and QR methods for symmetric diagonal matrices". In: *The Computer Journal* 6.1 (Jan. 1963), pp. 99–101 (cit. on p. 30).

[51]E. Hadad, F. Heese, P. Vary, and S. Gannot. „Multichannel audio database in various acoustic environments". In: *Proc. International Workshop on Acoustic Echo and Noise Control*. Antibes, France, Sept. 2014, pp. 313–317 (cit. on p. 31).

[52]J. Eaton, N. D. Gaubitch, A. H. Moore, and P. A. Naylor. „The ACE challenge - Corpus description and performance evaluation". In: *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. New York, USA, Oct. 2015, pp. 1–5 (cit. on p. 31).

[53]Z. Pruša, P. Søndergaard, P. Balazs, and N. Holighaus. „LTFAT: A Matlab/Octave toolbox for sound processing". In: *Proc. 10th International Symposium on Computer Music Multidisciplinary Research (CMMR)*. 2013, pp. 299–314 (cit. on p. 32).

[54]R. C. Hendriks, J. Jensen, and R. Heusdens. „Noise tracking using DFT domain subspace decompositions". In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.3 (Mar. 2008), pp. 541–553 (cit. on p. 35).

[55]I. Kodrasi and S. Doclo. „Joint Late Reverberation and Noise Power Spectral Density Estimation in a Spatially Homogeneous Noise Field". In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Calgary, Canada, Apr. 2018, pp. 441–445 (cit. on pp. 46, 50).

[56]K. B. Petersen and M. S. Pedersen. „The Matrix Cookbook". In: *Citeseer* 16.4 (2007), pp. 1–66 (cit. on p. 58).

Colophon

This thesis was typeset with LaTeX 2_ε. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.

Statement of Authorship

I hereby declare that I am the sole author of this master thesis and that I have not used any sources other than those listed in the bibliography and identified as references. I further declare that I have not submitted this thesis at any other institution in order to obtain a degree, and that I have followed the guidelines of good academic practice at the Carl von Ossietzky University of Oldenburg.

Oldenburg, July 17, 2018

Marvin Tammen