



Lukas Pfeifenberger, MSc

# TOWARDS THE EVOLUTION OF NEURAL ACOUSTIC BEAMFORMERS

Leveraging Speech Enhancement and Speaker Separation Performance in  
Time- and Frequency-Domain

## DOCTORAL THESIS

conducted at the

**Signal Processing and Speech Communication Laboratory**  
**Graz University of Technology**

Supervisor:

Prof. Dipl.-Ing. Dr.mont. Franz Pernkopf

Assessors/Examiners:

Prof. Dipl.-Ing. Dr.mont. Franz Pernkopf  
Assoc.Prof. M.S. Dr.Eng. Shinji Watanabe

Graz, April, 2021



## Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral thesis.

31. 3. 2021

---

date

*Heifenzberger Lukas*

---

(signature)





## Abstract (English)

Neural beamforming encompasses the merger of two different scientific disciplines, namely acoustic beamforming, and artificial neural networks. While the former uses statistical signal processing to spatially separate signals such as human speech, the latter uses non-linear function approximators to perform signal classification or regression tasks. Classical beamforming is used in unsupervised tasks such as denoising, or isolating sources with a known position. In these applications, the beam is steered towards the desired source. For tasks such as speaker tracking or blind source separation, the location of the individual speakers is unknown, rendering the problem ill-posed. Neural networks help to solve this class of problems by inferring the missing information from the underlying distribution of the multi-channel audio data. The symbiosis between beamforming and neural networks allows us to tackle hard problems such as the cocktail party scenario.

This thesis explores the evolution of neural beamforming from modest post-filters up to complete blind speaker separation systems, by covering four distinct topics: (i) Mask-based beamforming, which extracts a single speaker from background noise. This method employs a neural network to estimate a speech mask in frequency-domain. This mask is then used to obtain a classical beamformer. Here, we present our *Eigennet* structure which exploits spatial information embedded in the dominant Eigenvector of the spatial power-spectral density matrix of the noisy microphone inputs. (ii) Complex-valued neural beamforming, where complex-valued neural networks are used to predict beamforming weights in frequency-domain. This enables the beamformer to quickly react to location changes such as speaker movement. This concept outperforms classical beamformers, as the neural network directly optimizes the max-SNR objective of the beamformer. We present our *CNBF* architecture, which uses Wirtinger calculus to derive complex-valued recurrent network layers and non-holomorphic functions required for beamforming. (iii) Time-domain neural beamforming, where the concept of cross-domain learning is introduced. It allows to formulate the beamforming principle in a latent space, which is learned by a neural network. The enhanced signal is directly synthesized in time-domain. This approach is completely detached from a physical representation of sound waves, or classical beamforming algorithms. Our *TDNBF* formulation provides solutions for problems such as low-latency beamforming, dereverberation, and non-linear residual echo cancellation. (iv) Blind source separation, where we propose a monolithic, all-in-one solution to perform multi-speaker separation, dereverberation and speaker diarization using a single neural network, termed the *BSSD* architecture. This approach is capable to solve the cocktail party problem with an unknown number of speakers. It uses an analytic or statistic adaption layer, which virtually moves each identified speech source to the coordinate origin of the microphone array, from where it is extracted and dereverberated using a neural network in time-domain. This system was developed with application-driven constraints in mind, such as a reverberant environment with an unknown number of speakers, low latency, and real-time processing using small blocks of audio at a time.

Throughout this thesis, all methods are experimentally evaluated using multi-channel recordings from a variety of acoustic environments. We demonstrate their respective performance using metrics such as the word error rate or the signal-to-distortion ratio.



## Acknowledgements

I wish to express my sincere gratitude to the many people who influenced this thesis, and who contributed in turning a mere piece of scientific work into a passion: First of all, I want to thank my supervisor Franz Pernkopf for being such an experienced and supportive mentor. You always found the time for inspiring discussions about research and about life. I wholeheartedly enjoyed reviewing new ideas and concepts with you. I also want to thank Prof. Shinji Watanabe for serving as external examiner. Thank you, Martin and Hermann for granting me the freedom to pursue my studies in parallel to a full-time position as acoustic signal processing engineer. While this setup has had its minor complications, it also gave me an alternative perspective on my scientific work, which proves to be quite valuable in industry. Further, thank you Gernot Kubin and Franz Zotter, for the many fruitful discussions about beamforming during my Master's thesis. It is your experience and guidance which made me consider pursuing a PhD in this field. Next, I want to thank Matthias Zöhrer, my dear college and friend who joined me on this fascinating research topic. Regardless the time of day, you were always eager to face thought-provoking scientific discussions, or participate in endless coding sessions at the other side of the planet. I greatly appreciate your expertise and endurance!

I also want to thank all members of the SPSC for the friendly atmosphere I experienced during my time here. Thanks to the interdisciplinary environment, there was always an opportunity to discuss diverse topics ranging from graphical models to micro-controller programming. In this sense, Martin, Elmar, Jamilla, Wolfgang, Markus, Vincent, Hannes, Pejman and Sean certainly enriched my time at the SPSC. More importantly, there was always an opportunity for non-scientific talk including movies, books photography or outdoor activities during coffee breaks, which I enjoyed wholeheartedly.

Further, I express my deepest gratitude to my parents, who supported my curiosity in physics and electronics with utmost benevolence since I was a child. I also have to thank my teachers Otto Huber, Franz Klammler and Karl Entacher, who shaped and nourished my interest in mathematics during my high school years. Finally, I want to thank my friends Jeet, Christian, Sebastian, Marvin, Raffaella, Raphael, Magda, Miriam, Margit, Theresa and Andrea for constantly reminding me that there are far more important things in life than science or work.

*“Sometimes it seems as though each new step towards AI, rather than producing something which everyone agrees is real intelligence, merely reveals what real intelligence is not.”*

– Douglas R. Hofstadter, quote from Gödel, Escher, Bach: An Eternal Golden Braid

## Notational Conventions

### General Notation

---

$a$	scalar value
$\mathbf{A}(l)$	vector
$\mathbf{A}(l, k)$	matrix with indices $(l, k)$
$\mathbf{A}(l, k, m)$	tensor with indices $(l, k, m)$

### Operators

---

$\mathbb{R}^d$	$d$ -dimensional space of real numbers
$\mathbb{C}^d$	$d$ -dimensional space of complex numbers
$\text{Re}\{\cdot\}$	real part of a complex expression
$\text{Im}\{\cdot\}$	imaginary part of a complex expression
$\nabla_x$	gradient with respect to $x$
$\nabla_x^2$	Hessian with respect to $x$
$i$	imaginary unit
$ \cdot $	absolute value
$\ \cdot\ _p$	$l_p$ -norm
$\odot$	element-wise multiplication
$\otimes$	convolution operator
$\mathbb{1}(\cdot)$	indicator function
$(\cdot)^*$	conjugate value
$(\cdot)^T$	vector/matrix transpose
$(\cdot)^H$	vector/matrix adjoint
$\text{Tr}\{\cdot\}$	trace operator
$\mathcal{F}(x)$	Fourier transform of $x$
$\angle Z$	phase of the complex-valued argument $Z$
$[x]_+$	clipping of $x$ to positive values

### Probability

---

$X$	random variable
$x$	value of random variable
$P_{X Y}(x, y)$	conditional probability distribution of $X$ given $Y$
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean $\mu$ and variance $\sigma^2$
$\mathbb{E}\{x\}$	expected value of $X$



# Contents

<b>Statutory Declaration</b>	<b>III</b>
<b>Abstract (English)</b>	<b>V</b>
<b>Acknowledgements</b>	<b>VII</b>
<b>Notational Conventions</b>	<b>IX</b>
<b>1 Introduction</b>	<b>15</b>
1.1 Motivation . . . . .	15
1.2 Single-Channel Speech Enhancement . . . . .	15
1.3 Multi-Channel Speech Enhancement . . . . .	16
1.4 Contributions from published papers . . . . .	17
1.5 Outline of this thesis . . . . .	20
<b>2 Background</b>	<b>23</b>
2.1 Microphone Arrays . . . . .	23
2.2 Adaptive Beamforming . . . . .	24
2.2.1 System Model . . . . .	24
2.2.2 Minimum Variance Distortionless Response Beamformer (MVDR) . . . . .	25
2.2.3 Generalized Sidelobe Canceler (GSC) . . . . .	27
2.2.4 Generalized Eigenvalue Beamformer (GEV) . . . . .	28
2.3 Sound fields . . . . .	30
2.3.1 Near-Field . . . . .	30
2.3.2 Far-Field . . . . .	30
2.4 Source Localization . . . . .	32
2.4.1 Steered Response Power Phase Transform . . . . .	32
2.4.2 Direction-Dependent SNR . . . . .	33
2.5 Postfiltering . . . . .	34
2.5.1 Postfilter based on the GCC-PHAT . . . . .	35
2.5.2 Postfilter based on the diffuse noise sound field . . . . .	35
2.6 Spatial Whitening . . . . .	36
2.6.1 Effect on the beamformer . . . . .	36
2.6.2 Effect on the Postfilter . . . . .	38
2.7 Performance Measures . . . . .	39
2.7.1 SNR . . . . .	40
2.7.2 SDR . . . . .	40
2.7.3 WER . . . . .	41
2.7.4 STOI . . . . .	41
2.7.5 PESQ . . . . .	42
2.7.6 PEASS . . . . .	43
<b>3 Mask-based Beamforming</b>	<b>45</b>
3.1 Motivation . . . . .	45
3.2 Speech Masks . . . . .	46
3.2.1 PSD Matrix Estimation . . . . .	48
3.2.2 Speech Mask Estimation . . . . .	48
3.3 Neural Networks . . . . .	48
3.3.1 Neural Network Layers . . . . .	49

3.4	Eigenvector-based Speech Mask Estimation . . . . .	50
3.4.1	Single-speaker Eigennet . . . . .	51
3.4.2	Multi-speaker Eigennet . . . . .	55
3.5	Conclusion . . . . .	56
<b>4</b>	<b>Complex-valued Neural Beamforming</b>	<b>57</b>
4.1	Motivation . . . . .	57
4.2	Complex-valued Back-propagation . . . . .	57
4.2.1	Complex-valued Chain Rule . . . . .	58
4.2.2	Numeric Gradients . . . . .	59
4.2.3	Examples . . . . .	60
4.3	Complex-valued Neural Networks . . . . .	62
4.3.1	Complex-valued Activation Functions . . . . .	62
4.3.2	Complex-valued Neural Network Layers . . . . .	65
4.4	CNBF Architecture . . . . .	67
4.4.1	Performance . . . . .	69
4.5	Conclusion . . . . .	71
<b>5</b>	<b>Time-domain Neural Beamforming</b>	<b>73</b>
5.1	Motivation . . . . .	73
5.2	Cross-domain Learning . . . . .	74
5.3	RIR Recording and Spatialization . . . . .	75
5.3.1	Static RIRs . . . . .	76
5.3.2	Dynamic RIRs . . . . .	76
5.4	TDNBF Architecture . . . . .	78
5.4.1	Performance . . . . .	79
5.5	Dereverberation . . . . .	80
5.5.1	Performance . . . . .	82
5.6	Nonlinear Residual Echo Suppression . . . . .	83
5.6.1	Performance . . . . .	86
5.7	Conclusion . . . . .	88
<b>6</b>	<b>Blind Source Separation</b>	<b>89</b>
6.1	Motivation . . . . .	89
6.2	Speaker Localization . . . . .	90
6.2.1	Performance . . . . .	92
6.3	Selective Attention . . . . .	95
6.4	Speaker Identification . . . . .	96
6.4.1	Distance Measure . . . . .	97
6.5	BSSD Architecture . . . . .	98
6.5.1	Speaker Extraction . . . . .	99
6.5.2	Experiments . . . . .	100
6.6	Conclusion . . . . .	104
<b>7</b>	<b>Discussion and Outlook</b>	<b>105</b>
<b>A</b>	<b>Appendix</b>	<b>107</b>
A.1	Eusipco 2014 . . . . .	107
A.2	InterSpeech 2014 . . . . .	113
A.3	IEEE/ASRU 2015 . . . . .	119
A.4	CHiME4 2016 . . . . .	128
A.5	IEEE/ICASSP 2017 . . . . .	132



A.6 InterSpeech 2017 . . . . .	138
A.7 IEEE/ACM 2019 . . . . .	144
A.8 IEEE/ICASSP 2019 . . . . .	156
A.9 InterSpeech 2020 . . . . .	162
A.10 IEEE/ACM 2020 . . . . .	168



## 1

# Introduction

## 1.1 Motivation

Speech enhancement is concerned with improving the perceived intelligibility and quality of human speech. In our restless world, speech signals are often degraded by additive noise or interfering speakers, e.g. road-noise, manufacturing noise, or a crowded subway train. These noise sources cause discomfort and listener fatigue in many telecommunication applications such as mobile phones, intercoms, or hearing aids. Especially in hands-free scenarios, background noise levels can reach the loudness of the desired speech signal, as there is no handset to shield the ear of the listener from ambient sounds. In human-machine interfaces, degraded speech signals have a negative influence on speech recognition rates. In these applications, misheard voice commands lower the overall user acceptance, e.g. car navigation systems or voice assistants. Consequently, improving the perceptual aspects of speech signals has been an active field of research for many decades. Speech enhancement algorithms are used as preprocessors in a variety of applications, e.g. speech codecs, hearing aids, or noise-canceling headphones.

As mobile devices keep getting cheaper and more powerful, the scientific focus has shifted from Single-Channel Speech Enhancement (SCSE) to Multi-Channel Speech Enhancement (MCSE) methods. This allows addressing more complex types of noise signals, such as interfering speakers. While many noise signals can be modeled using their temporal statistics, separating two speech signals is much harder. The general case of separating multiple speakers who are talking over one another is known as the *cocktail party problem*. By using multiple microphones, the speakers can be spatially separated based on their location. Speaker separation algorithms include acoustic beamforming or blind source separation algorithms, such as Independent Component Analysis (ICA). Fueled by the success of deep learning, both speech enhancement and speaker separation algorithms made compelling advances over the last years. By combining traditional signal processing methods and non-linear function approximators, the performance of these algorithms achieves near-perfect signal reconstruction in many application scenarios. In this work, we elaborate on the fusion of acoustic beamforming and neural networks, which we termed *neural beamforming*.

## 1.2 Single-Channel Speech Enhancement

Historically, speech enhancement has mainly been recognized as a single-channel problem. The reason for this is: (i) Processing multiple signals require expenditures in both acoustics hardware and computing resources. (ii) Array processing algorithms were mostly developed for radar applications, and had yet to be adapted to the acoustic domain [1].

Over the last decades, a vast collection of algorithms has been conceived for SCSE. The most important approaches are based on the Wiener filter [1], [2] and spectral subtraction [3], [4]. Other methods include statistical models based on the Maximum Likelihood (ML), Maximum A-Posteriori (MAP), Minimum Mean-Squared Error (MMSE), Bayesian estimators, or subspace methods using Singular Value Decomposition (SVD) or Eigenvalue Decomposition (EVD). A

comprehensive overview can be found in [5]–[7].

Due to its relatively low complexity, spectral subtraction is by far the most widely used commercial SCSE method, even today. This method relies on the observation that human speech is sparse in both time and frequency [8]. Therefore, the spectrum of unwanted signal components like interfering noise can be estimated during periods where the desired speech signal is absent. The required noise statistics may be obtained by Minimum Statistics (MS) [9], or Improved Minima-Controlled Recursive Averaging (IMCRA) [10]. A central assumption being made is that the noise spectrum is more or less stationary, or at least slowly changing compared to the speech spectrum. The noise spectrum estimate is then subtracted from the spectrum of the mixture, using a spectral gain mask [11], [12]. Only the magnitude is affected by this process, the phase information is left unchanged. Figure 1.1 shows the basic principle of spectral subtraction.

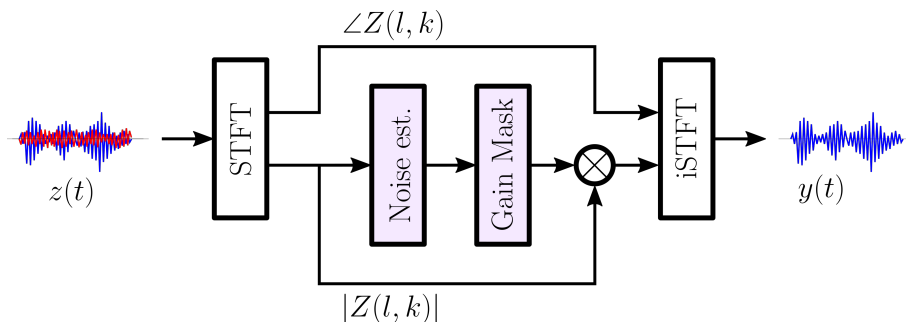


Figure 1.1: General architecture of spectral subtraction algorithms [13].

Recent studies showed that neglecting the phase affects the intelligibility and quality of the enhanced signal [14]–[16]. Further, estimation errors of the noise spectrum also have a severe impact on the speech quality: Under-estimation of the noise spectrum results in randomly distributed residual noise artifacts, lowering the speech quality. Over-estimation of the noise spectrum results in the deletion of spectral components of the desired speech, resulting in a decreased speech intelligibility. As real-world noises are challenging to predict, estimation errors can not be avoided. Consequently, speech quality and intelligibility cannot be maintained at the same time using spectral subtraction [6].

More recently, deep learning has been used for SCSE with great success. In particular, the noise estimation stage in Figure 1.1 is replaced by a Neural Network (NN), which predicts the gain mask directly from the log-power spectrum of the noisy speech data. To train the NN, the ground truth gain mask is required as label. The NN infers the spectral statistics of the noise signal from the training data, which outperforms all of the above model-based noise estimators, resulting in superior speech quality and intelligibility [17]–[24]. Alternative approaches do not use a gain mask at all, but rather rely on speech synthesizers [25]. Here, a Deep Neural Network (DNN) is used to predict a noise-free waveform in time-domain, given a degraded input signal. Examples for such systems are: *Wave-U-Net* [26], *TasNet* [27] and *Conv-TasNet* [28].

### 1.3 Multi-Channel Speech Enhancement

When more than one microphone is available, not only the temporal information but also the spatial features of the sound field can be utilized. This allows for MCSE methods, which can be divided into two main groups: (i) Blind Source Separation (BSS) and (ii) beamforming.

BSS denotes both the supervised and unsupervised separation of multivariate signals into their individual sources, i.e. speech, images, or medical data such as EEG signals. Typically, ICA is used to perform unsupervised BSS, as it maximizes the statistical independence of the estimated

components. This is achieved by maximizing the Kurtosis (non-Gaussianity) of the underlying distribution of the multivariate data [29]. Algorithms for unsupervised speech separation can be found in [30], [31]. Algorithms for supervised speech separation include Non-negative Matrix Factorization (NMF) [32]–[34], and DNNs [27], [35], [36].

While BSS aims at separating all involved sources in the mixture, beamforming enhances only a set of desired sources while treating all others as interference. A beamformer is comprised of a set of spatio-temporal filters, which processes each of the microphone signals followed by a summation operation. If those filters are designed with the objective to extract a desired broadband signal like speech, it is considered as a *broadband* or *superdirective* beamformer. Common beamforming structures are the Minimum Variance Distortionless Response (MVDR) beamformer [37], and its Generalized Sidelobe Canceler (GSC) formulation [38]. Both aim at minimizing the signal power of the interfering signal at the beamformer output, while also minimizing distortions of the target signal [39]–[42]. Another design concept is achieved by Generalized Eigenvalue (GEV) beamformer [43], which trades minimal speech distortions for maximum SNR at the beamformer output. Every beamformer requires a *steering vector* to direct the beam towards the desired signal, i.e. the target speaker. This direction can be estimated using Direction Of Arrival (DOA) algorithms such as PHase Acoustic Transform (PHAT) [12], Multiple Signal Classification (MUSIC) [11], or Direction-Dependent SNR (DD-SNR) [44]. The steering vector is comprised of a set of time delays, which corresponds to the direct line of sight between a sound source and the microphone array. However, sounds do not only propagate via a straight line, but also via multiple reflections caused by the room acoustics [45]. The complete propagation path from the speaker to a microphone is known as Acoustic Transfer Function (ATF) [46], [47]. Depending on the reverberation contained in the ATFs, DOA estimation can be a difficult task.

With recent advances in DNNs, using a DOA algorithm is no longer required, as the steering vector may be replaced by a gain mask. This mask identifies the time-frequency bins that contain the desired signal in the noisy data. The gain mask can be directly estimated from the noisy microphone data. It is used to calculate the spatial Power-Spectral Density (PSD) matrices of the desired and interfering sound sources. The PSD matrix of the desired speech signal contains the ATF of the speaker in its principal Eigenvector [42], [48]. Hence, mask-based beamforming proved to be superior to DOA based approaches [21], [49]–[53]. Alternatively, the complex-valued beamforming weights themselves may be derived from the noisy microphone observations, using DNNs, i.e. [54], [55]. With the help of time-domain NNs, complete BSS systems have been constructed. They perform speaker separation [56], diarization [57], dereverberation [58] and Automatic Speech Recognition (ASR) [59] all at once.

## 1.4 Contributions from published papers

Before the major breakthrough of deep learning, a typical MCSE processing chain consisted of a DOA estimation algorithm [11], [38], an adaptive beamformer such as the GSC [37], [60], a postfilter [12], [61], and an ASR system based on Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs) [62], [63]. The main purpose of the beamforming front-end is to provide a clean, noise-free speech signal for the ASR back-end [64]. Typically, these systems are limited to a single speaker in the near-field of the microphone array. To adapt the beamformer weights, algorithms such as the Normalized Least Mean Squares (NLMS) algorithms are used [2], [12]. Over the last couple of years, each of these building blocks has been replaced with a NN, resulting in impressive performance gains and applications for problems deemed too hard at the time. For example, it has become possible to automatically detect, count and localize an unknown number of sources from a mixture of speakers [56], rendering DOA estimation algorithms obsolete. Further, statistical beamformers such as the MVDR, GSC [38], or GEV [43] have been

outperformed in both the time- and frequency-domain by dedicated NN architectures [26], [53], [58], [65]. In order to do so, complex-valued NN have been devised, making complex-valued back-propagation possible using *Wirtinger Calculus* [66]. This progression allowed for beamformers with new properties and applications such as the Complex-Valued Neural Beamformer (CNBF) [67], which is capable to quickly adapt to multiple sources without relying on temporal signal statistics. Further, time-domain beamformers such as *Beam-TasNet* [68] or the Time-Domain Neural Beamformer (TDNBF) [58] allow for real-time applications with processing delays down to 4ms. In parallel to the acoustic front-end, ASR systems experienced dramatic improvements as well [62], [69]–[71]. More recently, end-to-end solutions which combine far-field speech enhancement and ASR systems have been proposed [59]. The summation of these advances even made it possible to solve the infamous *cocktail party phenomenon* [56], [58]. During the last couple of years, we were able to accompany this impressive journey with our own contributions. All published papers can be found in the appendix in Chapter A.

- "*A multi-channel postfilter based on the diffuse noise sound field*", Lukas Pfeifenberger and Franz Pernkopf, *22nd European Signal Processing Conference, Lisbon, 2014*. In this paper, we proposed a multi-channel postfilter for the MVDR beamformer, which is based on the spatial coherence function of diffuse sound fields. The postfilter exploits the different properties of the near-field and far-field coherence, which can be expressed analytically under certain assumptions. For further details, see Appendix A.1.
- "*Blind source extraction based on a direction-dependent a-priori SNR*", Lukas Pfeifenberger and Franz Pernkopf, *Interspeech 2014 - 15th Annual Conference of the International Speech Communication Association, Singapore, 2014*. In this paper, we propose a concept to estimate the unknown location of a single speaker embedded in diffuse background noise. We formulate an iterative algorithm which maximizes the DD-SNR, thereby identifying the DOA of the desired speech source. As the DD-SNR is related to the gain of the postfilter, we could show that the performance of this algorithm surpasses traditional DOA methods such as MUSIC [11]. For further details, see Appendix A.2.
- "*Multi-channel speech processing architectures for noise robust speech recognition: 3<sup>rd</sup> CHiME challenge results*", Lukas Pfeifenberger, Tobias Schrank, Matthias Zöhrer, Martin Hagemüller and Franz Pernkopf, *IEEE Automatic Speech Recognition and Understanding Workshop, Scottsdale, 2015*. In this paper, we contributed to the CHiME3 speaker separation and speech recognition challenge [72], where a single speaker is embedded in background noise in various acoustic environments. We contributed multiple variants of the MVDR beamformer and the *direction-dependent* Signal to Noise Ratio (SNR) from [44]. Further, we proposed one of the first neural postfilters, which increases the SNR at the output of the beamformer by applying a gain mask in the frequency-domain. We use the DD-SNR as input for this NN. Further, we adapted the Kaldi ASR engine [73] for this type of speech enhancement. For further details, see Appendix A.3.
- "*Deep beamforming and data augmentation for robust speech recognition: Results of the 4<sup>th</sup> CHiME challenge*", Tobias Schrank, Lukas Pfeifenberger and Matthias Zöhrer, Johannes Stahl, Pejman Mowlae, Franz Pernkopf, *4th International Workshop on Speech Processing in Everyday Environments, San Francisco, 2016*. In this paper, we contributed to the CHiME4 challenge with our *Eigenvector* beamforming concept, which uses the principal Eigenvector of the spatial PSD matrix of the noisy multi-channel speech signal to estimate the location of the desired speaker. This concept proved to be beneficial to the neural postfilter, as the Eigenvector contains spatial information about the desired speech signal. For further details, see Appendix A.4.
- "*DNN-based speech mask estimation for Eigenvector beamforming*", Lukas Pfeifenberger, Matthias Zöhrer and Franz Pernkopf, *The 42nd IEEE International Conference on Acous-*

- tics, Speech and Signal Processing, New Orleans, 2017.* In this paper, we extended the *Eigenvector* beamformer to a neural network which estimates a Speech Presence Probability (SPP) mask using the cosine similarity in a temporal sequence of principal Eigenvectors, derived from the noisy multi-channel speech signal. This SPP is then used to obtain the spatial PSD matrices for both the desired speech signal and the unwanted background noise. With these matrices, beamformers such as the MVDR or GEV can be derived in an *offline fashion*, i.e. the beamforming weights are derived based on the signal statistics of a whole utterance. Further, we examined the relation between the SPP and an optimal postfilter in the max-SNR sense. For further details, see Appendix A.5.
- *"Eigenvector-based speech mask estimation using logistic regression", Lukas Pfeifenberger, Matthias Zöhrer and Franz Pernkopf, International Conference on Spoken Language Processing, Stockholm, 2017.* In this paper, we optimized the *Eigenvector* beamformer by using a resource efficient logistic regression, which uses significantly less parameters than our previous implementation in [48]. Further, we proposed the Phase Aware Normalization (PAN) as an alternative to the existing Blind Analytical Normalization (BAN) method, to compensate amplitude distortions caused by the GEV beamformer [43]. For further details, see Appendix A.6.
  - *"Eigenvector-Based speech mask estimation for multi-channel speech enhancement", Lukas Pfeifenberger, Matthias Zöhrer and Franz Pernkopf, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2019.* In this paper, we extend the *Eigenvector* beamformer to multiple speakers, speaker tracking, and *block-online* processing. By selecting the spatial location of the desired and unwanted speakers in the training data, we can train the NN to extract a single speaker at a specific location, or even track a speaker within a limited region of space. While the GEV beamformer still depends on long-term signal statistics, we determine new beamformer weights for short blocks of audio data, thereby getting closer to the real-time application scenario. For further details, see Appendix A.7.
  - *"Deep complex-valued neural beamformers", Lukas Pfeifenberger, Matthias Zöhrer and Franz Pernkopf, The 44th IEEE International Conference on Acoustics, Speech and Signal Processing, Brighton, 2019.* In this paper, we proposed a new beamforming scheme to address the real-time issue and to further increase the signal separation performance. By using complex-valued neural networks, a new set of beamforming weights is estimated for each time frame, without the need for long-term signal statistics. This allows the NN to instantaneously adapt to the desired signal, and to surpass the SNR of statistical beamformers. We further proposed building blocks such as complex-valued Long Short-Term Memory (LSTM) layers and derivatives for non-holomorphic beamforming functions, using *Wirtinger Calculus* [66]. For further details, see Appendix A.8.
  - *"Nonlinear residual echo suppression using a recurrent neural network", Lukas Pfeifenberger and Franz Pernkopf, International Conference on Spoken Language Processing, Shanghai, 2020.* In this paper, we proposed a neural postfilter to suppress the non-linear, residual echo of an Acoustic Echo Canceler (AEC), using a very small, real-time capable NN. While being a classical SCSE method, this work is closely related to a postfilter for a statistical beamformer. For further details, see Appendix A.9.
  - *"Blind speech separation and dereverberation using neural beamforming", Lukas Pfeifenberger and Franz Pernkopf, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020.* In this paper, we proposed a complete framework to tackle the cocktail party problem. Our system addresses blind speaker separation in the far-field, using both complex-valued and time-domain neural beamformers. We iteratively localize and separate each speaker from a mixture of an unknown number of speakers. Further, we perform

dereverberation of the extracted speakers, to enable speaker identification using speaker embedding vectors. We also contribute an algorithm to assign enhanced utterances to speaker identities. Both beamforming and speaker identification are performed by the same, monolithic NN. For further details, see Appendix A.10.

## 1.5 Outline of this thesis

With a scientific field as rich and extensive as speech enhancement, it is a challenging task to compile a comprehensive document that contains all relevant aspects of this topic. In this thesis, we will discuss the evolution of MCSE and BSS methods backed by beamforming and deep learning. Starting with unsupervised adaptive beamformers, we present a coherent story leading towards an all-in-one system for multi-speaker separation, dereverberation and diarization. With a thorough introduction to the basics of beamforming, the interested reader will be equipped with the necessary background to benefit from the presented topics. We start this journey by proposing six problems that are specific to MCSE and BSS, i.e.

1. **Isolate a single speaker from background noise.**
2. **Isolate a single speaker from a mixture of multiple speakers.**
3. **Track moving speakers.**
4. **Isolate and dereverberate a speaker in the far-field.**
5. **Separate all speakers in a mixture of multiple speakers.**
6. **Assign an identity to an isolated speaker.**

In the following chapters, we will discuss properties, solutions, and experiments for each one of these topics. While a large part of our contributions to this list has previously been published, a significant portion has been reworked to align with the structure of this thesis. Each chapter has been enriched with additional experiments and insights accompanying the respective topics.

**Chapter 2** provides an introduction to the topic of multi-microphone speech processing, starting from signal processing with Multiple Input - Multiple Output (MIMO) systems. Next, adaptive beamforming is introduced, including the MVDR, GSC, and GEV beamformers. Then, some basic properties of sound fields are presented, i.e. the near-field and the far-field. Their properties are exploited for signal whitening, which proves to be useful throughout this thesis. Further, we explore various source localization algorithms such as Steered Response Power Phase Transform (SRP-PHAT), Generalized Cross Coherence Phase Transform (GCC-PHAT) and DD-SNR. The chapter is concluded by subjective and objective performance measures for speech enhancement, such as the SNR, Signal to Distortion Ratio (SDR), Word Error Rate (WER), Short-Time Objective Intelligibility measure (STOI), Perceptual Evaluation of Speech Quality (PESQ), and Perceptual Evaluation methods for Audio Source Separation (PEASS).

**Chapter 3** introduces mask-based beamforming. Starting with a gain mask to further enhance the output of a beamformer, this chapter covers the *Eigennet* beamformer, which uses a neural network to estimate a speech mask. This mask is used to estimate the spatial PSD matrices of the involved sound sources. From these PSD matrices, various beamformers may be constructed. This concept is capable to isolate a single speaker from ambient background noise, or from multiple speakers, as long as they are not moving. The chapter is concluded with two experiments using the *Eigennet* beamformer.



**Chapter 4** explores complex-valued neural beamforming, where a neural network is used to predict complex-valued beamforming weights. This concept allows to track and isolate a single speaker, given a spatial region of interest. As a prerequisite, complex-valued back-propagation and Wirtinger Calculus are introduced. Further, complex-valued neural networks are presented. We conclude the chapter with an experiment to compare the performance between the *Eigennet* architecture and complex-valued neural beamforming.

**Chapter 5** addresses time-domain beamforming as an extension to complex-valued neural beamforming. The properties and benefits of time-domain neural networks for speaker separation are presented. This system is capable to track and isolate a moving speaker, as well as performing other tasks such as dereverberation. Further, a comparison between time- and frequency-domain neural beamforming methods is done, by using static and moving speakers in an experiment involving realistic room impulse responses. Therefore, the experimental setup for recording static and dynamic room impulse responses is introduced. We conclude the chapter with three experiments, where the performance of both time- and frequency-domain neural networks is compared, i.e. neural beamforming, dereverberation, and non-linear residual echo suppression.

**Chapter 6** addresses blind source separation, where an unknown number of speakers are isolated from a mixture of multiple speakers, i.e. the meeting room scenario. We construct a monolithic system for speaker localization, separation, dereverberation and identification. This provides solutions to all six problems of the list above. This chapter covers algorithms and models for source localization, selective attention, dereverberation, and speaker identification. We conclude the chapter with experiments on speaker separation and diarization, involving up to four speakers.

**Chapter 7** concludes this thesis, and provides an outlook into future research topics.



## 2

## Background

## 2.1 Microphone Arrays

Acoustic beamformers consist of an array of multiple microphones in a defined geometry or aperture. Ideally, the microphones are considered to be omni-directional sound pressure receivers with perfect linearity and no additive system noise. For the sake of simplicity, we assume an anechoic environment, where no reverberation occurs. Figure 2.1 shows 4 microphones  $M_1 \dots M_4$  in an arbitrary geometry. Further, we randomly place two speakers at locations  $X_S$  and  $X_N$ . They emit the independent waveforms  $s(t)$  and  $n(t)$  as point sound sources [45]. At the  $m^{\text{th}}$  microphone, both waveforms are picked up as superposition

$$z_m(t) = s(t - \tau_{s,m}) + n(t - \tau_{n,m}), \quad (2.1)$$

where  $\tau_{s,m}$  and  $\tau_{n,m}$  denotes the time delay between the respective speaker and the  $m^{\text{th}}$  microphone. The delays are given by

$$\begin{aligned} \tau_{s,m} &= \frac{\|X_S - M_m\|}{c}, \\ \tau_{n,m} &= \frac{\|X_N - M_m\|}{c}, \end{aligned} \quad (2.2)$$

where  $c$  denotes the speed of sound, which is approximately  $343 \frac{m}{s}$  in air at  $20^\circ C$ . At the A/D conversion stage, the signals from the four microphones are quantized with the sampling rate  $f_s$ . For the sake of readability, we use the time index  $t$  for both continuous and quantized signals, as quantization errors do not affect the properties of MIMO signal processing. It can be seen from Figure 2.1, that the signal from speaker  $X_S$  has to travel a shorter distance towards microphone 1, than the signal from speaker  $X_N$ . Hence, the time delay  $\tau_{s,1} < \tau_{n,1}$ . This *time difference of arrival* depends on both the position of the sound sources and the geometry of the microphone array. By exploiting this information, a beamformer is able to differentiate between spatially separated sound sources.

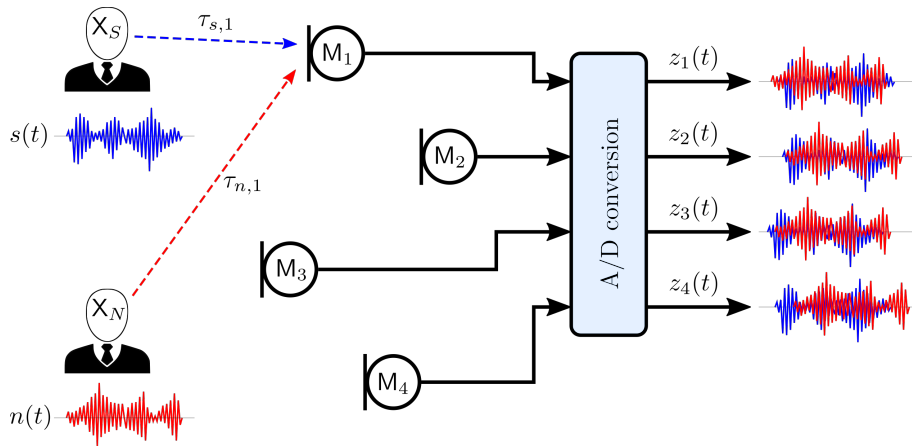


Figure 2.1: Microphone array with four microphones in an arbitrary geometry, and two point sound sources.

## 2.2 Adaptive Beamforming

### 2.2.1 System Model

Based on the physical model shown in Figure 2.1, we define a microphone array of  $M$  microphones, arranged into an arbitrary geometry. Without loss of generality, we define the first source  $X_S$  as the desired speech signal  $s(t)$ , and the other source  $X_N$  as interfering noise signal  $n(t)$  with an unknown location, i.e. ambient sounds or sensor noise. We assume the sources to be stationary, i.e. neither the speaker nor the noise source is moving over time. Figure 2.1 indicates that the sound waves from the speaker to the microphone travel along a straight line. However, in a realistic scenario, we cannot assume that the sound propagation is anechoic, i.e. without reflections. We have to consider the whole acoustic path from the location of the speaker to the microphones, including all reflections and acoustic echoes. This path is known as ATF, and it is typically modeled as a Finite Impulse Response (FIR) filter [12]. With this definition, the signal at the  $m^{\text{th}}$  microphone is written as

$$z_m(t) = s_m(t) + n_m(t), \quad (2.3)$$

where  $s_m(t)$  and  $n_m(t)$  are the speech and noise signals as received by the  $m^{\text{th}}$  microphone. The relation between the speech signal  $s(t)$  at the location of the speaker, and the speech signal  $s_m(t)$  at the  $m^{\text{th}}$  microphone is given by

$$s_m(t) = s(t) \otimes a_m(t), \quad (2.4)$$

where  $a_m(t)$  denotes the ATF from the location of the speaker to the  $m^{\text{th}}$  microphone. For the second sound source in Figure 2.1, an equivalent ATF  $a_n(t)$  can be formulated. However, if  $n_m(t)$  denotes ambient noise without a distinct origin, neither the point source  $n(t)$  nor the ATF  $a_n(t)$  exists. In this case, we refer to  $n_m(t)$  as *diffuse* noise. We can still make some statistical assumptions about this type of noise, as will become clear in the remainder of this chapter. The noisy speech signal  $z_m(t)$  is transformed into the Short-time Fourier Transform (STFT) domain, i.e.  $Z_m(l, k) = \mathcal{F}(z_m(t))$ . By inserting Eq. 2.4 into 2.3, we arrive at

$$Z_m(l, k) = S_m(l, k) + N_m(l, k) = S(l, k)A_m(k) + N_m(l, k), \quad (2.5)$$

where the frequency bin  $k = 1, \dots, K$  and the STFT time frame is denoted by  $l$ . The length of the STFT window needs to be sufficiently long to model the multiplicative filter operation  $S(l, k)A_m(k)$  without aliasing. Note that we do not assign a frame index  $l$  to the ATFs, as we assume them to be constant over time, i.e. the speaker is not moving. By stacking all  $M$  signals to a  $M \times 1$  vector, the signals from all microphones can be written in a more compact notation:

$$\mathbf{Z}(l, k) = S(l, k)\mathbf{A}(l, k) + \mathbf{N}(l, k), \quad (2.6)$$

where

$$\begin{aligned} \mathbf{Z}(l, k) &= [Z_1(l, k), \dots, Z_M(l, k)]^T \\ \mathbf{S}(l, k) &= [S_1(l, k), \dots, S_M(l, k)]^T \\ \mathbf{N}(l, k) &= [N_1(l, k), \dots, N_M(l, k)]^T \\ \mathbf{A}(k) &= [A_1(k), \dots, A_M(k)]^T. \end{aligned} \quad (2.7)$$

With this notation, we can define the most generic beamformer, the *filter-and-sum* beamformer [1], [60] as shown in Figure 2.2. It uses a set of beamforming weights  $\mathbf{W}(k)$  of shape  $M \times 1$  to filter the noisy microphone signals  $\mathbf{Z}(l, k)$ , and add the filtered results to form the output  $Y(l, k)$ , i.e.

$$Y(l, k) = \mathbf{W}^H(k)\mathbf{Z}(l, k), \quad (2.8)$$

where we assume the beamforming weights  $\mathbf{W}(k) \in \mathbb{C}$  to be constant over time, which reflects the requirement of stationary sound sources. With this signal model, we will introduce three of the most common beamformer types in the sequel.

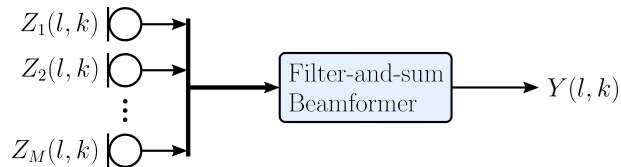


Figure 2.2: Filter-and-sum beamformer. The microphone signals and the beamformer output are denoted as  $Z_m(l, k)$  and  $Y(l, k)$ , respectively.

## 2.2.2 Minimum Variance Distortionless Response Beamformer (MVDR)

We start with some statistical observations, which arise from the stationary setup shown in Figure 2.1. The spatial PSD matrix [74] for the microphone signals  $\mathbf{Z}(l, k)$  is defined as

$$\Phi_{ZZ}(k) \triangleq \mathbb{E}\{\mathbf{Z}(l, k)\mathbf{Z}^H(l, k)\}, \quad (2.9)$$

where the expectation operator is applied to the time dimension, i.e. the frame index  $l$ . For discrete microphone observations  $\mathbf{Z}(l, k)$ , the expectation operator reduces to the average

$$\Phi_{ZZ}(k) = \frac{1}{L} \sum_{l=1}^L \mathbf{Z}(l, k)\mathbf{Z}^H(l, k), \quad (2.10)$$

where  $L$  denotes the total number of frames in the processed *block* of audio data. For uncorrelated speech and noise signals, this PSD matrix can be decomposed into its speech and noise components,

$$\mathbf{\Phi}_{ZZ}(k) = \mathbf{\Phi}_{SS}(k) + \mathbf{\Phi}_{NN}(k). \quad (2.11)$$

According to Eq. 2.6,  $\mathbf{\Phi}_{SS}(k)$  can be decomposed into the speech PSD  $\Phi_S(k)$  and the ATFs  $\mathbf{A}(k)$  from the speaker to the microphones [11], such that:

$$\mathbf{\Phi}_{SS}(k) = \mathbf{A}(k)\mathbf{A}^H(k)\Phi_S(k), \quad (2.12)$$

where  $\Phi_S(k) \triangleq \mathbb{E}\{|S(l, k)|^2\}$ . Note that the amplitude of the ATFs is not necessarily normalized, i.e.  $\|\mathbf{A}\|_2 \neq 1$ . We use the *filter-and-sum* beamformer defined in Eq. 2.8, to obtain the beamformer output  $Y(l, k)$ . The Mean Square Error (MSE) of this output with respect to the desired speech signal  $S(l, k)$  is given by the cost function

$$\mathbf{J}(k) \triangleq \mathbb{E}\{|Y(l, k) - S(l, k)|^2\}. \quad (2.13)$$

Using Eq. 2.8 and the PSD matrices given in Eq. 2.11 and Eq. 2.12, the cost function reduces to

$$\mathbf{J} = \mathbf{W}^H \mathbf{\Phi}_{ZZ} \mathbf{W} + \Phi_S - \mathbf{W}^H \mathbf{A} \Phi_S - \Phi_S \mathbf{A}^H \mathbf{W}, \quad (2.14)$$

where we omitted the frequency index  $k$  for enhanced readability. Setting the derivative of Eq. 2.14 to zero gives

$$\nabla \mathbf{J} = \frac{\partial \mathbf{J}(\mathbf{W})}{\partial \mathbf{W}^H} = -2\mathbf{\Phi}_{ZZ} \mathbf{W} + 2\mathbf{A} \Phi_S \stackrel{!}{=} 0. \quad (2.15)$$

The solution of Eq. 2.15 is known as the MSE-optimal multi-channel Wiener filter  $\mathbf{W}_{OPT}$  [74], [75], i.e.

$$\begin{aligned} \mathbf{W}_{OPT} &= \mathbf{\Phi}_{ZZ}^{-1} \mathbf{A} \Phi_S \\ &= [\mathbf{A} \mathbf{A}^H \Phi_S + \mathbf{\Phi}_{NN}]^{-1} \mathbf{A} \Phi_S. \end{aligned} \quad (2.16)$$

Using the matrix inversion lemma [1], we obtain

$$\begin{aligned} \mathbf{W}_{OPT} &= \left[ \mathbf{\Phi}_{NN}^{-1} - \frac{\Phi_S \mathbf{\Phi}_{NN}^{-1} \mathbf{A} \mathbf{A}^H \mathbf{\Phi}_{NN}^{-1}}{1 + \Phi_S \mathbf{A}^H \mathbf{\Phi}_{NN}^{-1} \mathbf{A}} \right] \mathbf{A} \Phi_S \\ &= \underbrace{\mathbf{\Phi}_{NN}^{-1} \mathbf{A}}_{\mathbf{W}_{MVDR}} \cdot \underbrace{\frac{\Phi_S}{\Phi_S + [\mathbf{A}^H \mathbf{\Phi}_{NN}^{-1} \mathbf{A}]^{-1}}}_{G = \frac{\xi}{1+\xi}}. \end{aligned} \quad (2.17)$$

The filter  $\mathbf{W}_{MVDR}$  can be recognized as the MVDR beamformer [12], [37]. Note that it is equivalent to the Linearly Constrained Minimum Variance (LCMV) beamformer [42], [75] with

the single constraint

$$\begin{aligned} \mathbf{W}_{LCMV} = \underset{\mathbf{W}}{\operatorname{argmin}} \{ & \mathbf{W}^H \boldsymbol{\Phi}_{ZZ} \mathbf{W} \} \\ \text{subject to } & \mathbf{A}^H \mathbf{W} \stackrel{!}{=} 1. \end{aligned} \quad (2.18)$$

The Wiener postfilter  $G = \frac{\xi}{1+\xi}$  depicts a real-valued gain mask, which is applied at the beamformer output. Rearranging Eq. 2.17, leads to

$$\xi = \boldsymbol{\Phi}_S \mathbf{A}^H \boldsymbol{\Phi}_{NN}^{-1} \mathbf{A}, \quad (2.19)$$

which is equivalent to the SNR at the beamformer output, i.e.

$$\xi = \frac{\mathbf{W}_{MVDR}^H \boldsymbol{\Phi}_{SS} \mathbf{W}_{MVDR}}{\mathbf{W}_{MVDR}^H \boldsymbol{\Phi}_{NN} \mathbf{W}_{MVDR}}. \quad (2.20)$$

In practice, both the noise PSD matrix  $\boldsymbol{\Phi}_{NN}(k)$  and the ATFs  $\mathbf{A}(k)$  are not directly observable, which makes the MVDR difficult to implement.

### 2.2.3 Generalized Sidelobe Canceler (GSC)

For implementing the MVDR beamformer, an estimate of the noise PSD matrix  $\hat{\boldsymbol{\Phi}}_{NN}(k)$  is required. The GSC formulation circumvents this requirement by splitting the beamforming filter  $\mathbf{W}_{MVDR}(k)$  into three components: (i) A *steering vector*  $\mathbf{v}_S(k)$ , which provides a spatial focus towards the desired sound source. (ii) A *blocking matrix*  $\mathbf{B}(k)$ , which cancels the desired speaker to obtain a clean reference of the noise signal. (iii) An *Adaptive Interference Canceler (AIC)*  $\mathbf{H}_{AIC}(k)$  which subtracts the noise reference from the summation signal obtained by the steering vector [37], [38]. Figure 2.3 shows the block diagram of the GSC. For its robustness and simplicity, the GSC is used in a wide range of applications [39]–[42]. Its weights are given as

$$\mathbf{W}_{MVDR} \approx \mathbf{W}_{GSC} = \mathbf{v}_S - \mathbf{B} \mathbf{H}_{AIC}. \quad (2.21)$$

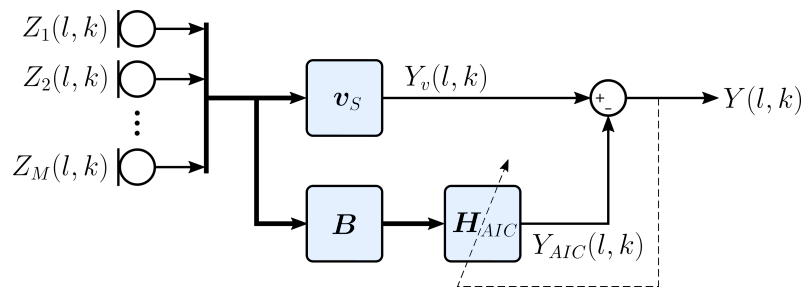


Figure 2.3: Block diagram of the GSC beamformer, with the steering vector  $\mathbf{v}_S(k)$ , the blocking matrix  $\mathbf{B}(k)$  and the AIC  $\mathbf{H}_{AIC}(k)$ .

#### Steering Vector

While the GSC avoids the direct estimation of  $\hat{\boldsymbol{\Phi}}_{NN}(k)$ , the steering vector  $\mathbf{v}_S(k)$  is still a crucial component, as it directs the beamformer towards the general direction of the desired

source signal. Clearly, the ideal steering vector would be the ATFs themselves. However, they are unknown in practice and hard to estimate in reverberant environments [12]. Therefore, the steering vector is usually modeled as a vector of simple time delays [11], i.e.

$$\mathbf{v}_S(k) = [e^{-j\omega_k\tau_1}, e^{-j\omega_k\tau_2}, \dots, e^{-j\omega_k\tau_M}]^T, \quad (2.22)$$

where  $\omega_k = 2\pi \frac{k}{2K} f_s$  is the discrete frequency variable, and  $\tau_m$  denotes the time delay from the desired source to the  $m^{\text{th}}$  microphone [39], [46], [47].

### Blocking Matrix

The blocking matrix  $\mathbf{B}(k)$  is used to obtain noise reference signals which are free of any speech components. This is achieved by steering *nulls* in the direction of the speech source [12], i.e.

$$\mathbf{B}^H \mathbf{A} \stackrel{!}{=} \mathbf{0}_{1 \times M}. \quad (2.23)$$

A blocking matrix that satisfies this constraint is given by:

$$\mathbf{B} = \mathbf{I} - \mathbf{v}_S \mathbf{v}_S^H, \quad (2.24)$$

where  $\mathbf{I}$  is the  $M \times M$  identity matrix. Further variants, like sparse or adaptive blocking matrices, are given in [40], [76]. However, the time-delays in the steering vector in Eq. 2.22 only model the direct path, i.e. the direct line of sight between the speaker and the microphones. Multi-path propagations caused by reverberations and acoustic echoes are not accounted for. This causes the blocking matrix to fail at suppressing the speech signal entirely, leading to *target leakage* [12]. The speech signal leaking through the blocking matrix will be regarded as noise by the AIC, and consequently be subtracted from the beamformer output. This effect limits the overall performance of the GSC beamformer, especially in reverberant or far-field application scenarios.

### Adaptive Interference Canceler

The filters  $\mathbf{H}_{AIC}(l, k)$  of the AIC are calculated adaptively using the NLMS algorithm by minimizing the MSE at the beamformer output  $Y(l, k)$  [2]. Together with the blocking matrix, the AIC adaptively models the unknown spatial noise PSD  $\Phi_{NN}(k)$ . Similar to AECs, a Voice Activity Detector (VAD) has to be used to avoid divergence of the filter weights during speaker activity.

#### 2.2.4 Generalized Eigenvalue Beamformer (GEV)

Another alternative to the MVDR and GSC beamformers is given by the GEV beamformer [43], [77], which constrains the filter weights  $\mathbf{W}(l, k)$  to maximize the SNR  $\xi(l, k)$  at the beamformer output, i.e.

$$\mathbf{W}_{SNR} = \underset{\mathbf{W}}{\operatorname{argmax}} \xi, \quad (2.25)$$



with

$$\xi = \frac{\mathbf{W}^H \boldsymbol{\Phi}_{SS} \mathbf{W}}{\mathbf{W}^H \boldsymbol{\Phi}_{NN} \mathbf{W}}. \quad (2.26)$$

The solution to Eq. 2.25 is found by setting the derivative of Eq. 2.26 to zero:

$$\frac{\partial \xi(\mathbf{W})}{\partial \mathbf{W}^H} = \frac{2\boldsymbol{\Phi}_{SS} \mathbf{W} - 2\xi \boldsymbol{\Phi}_{NN} \mathbf{W}}{\mathbf{W}^H \boldsymbol{\Phi}_{NN} \mathbf{W}} \stackrel{!}{=} 0, \quad (2.27)$$

which leads to the following generalized Eigenvalue problem [77]:

$$\boldsymbol{\Phi}_{NN}^{-1} \boldsymbol{\Phi}_{SS} \mathbf{W} = \xi \mathbf{W}. \quad (2.28)$$

Using the definition of  $\boldsymbol{\Phi}_{SS}$  in Eq. 2.12, a solution for Eq. 2.28 is given by

$$\mathbf{W}_{GEV} = \zeta \boldsymbol{\Phi}_{NN}^{-1} \mathbf{A}, \quad (2.29)$$

where  $\zeta$  is an arbitrary complex scalar.

### Phase Aware Normalization

The beamforming filter  $\mathbf{W}_{GEV}$  will not have a distortionless response, i.e.  $\mathbf{A}^H \mathbf{W}_{GEV} \neq 1$ . We therefore proposed the *PAN* factor in [53]: By comparing the MVDR from Eq. 2.17 and the GEV from Eq. 2.29, it can be seen that both beamforming vectors are identical up to the scalar factor  $G_{PAN}$ , i.e.

$$\mathbf{W}_{MVDR} = G_{PAN} \mathbf{W}_{GEV} = \frac{\boldsymbol{\Phi}_{NN}^{-1} \mathbf{A}}{\mathbf{A}^H \boldsymbol{\Phi}_{NN}^{-1} \mathbf{A}}. \quad (2.30)$$

By rearranging Eq. 2.29 into  $\mathbf{A} = \boldsymbol{\Phi}_{NN} \mathbf{W}_{GEV} \zeta^{-1}$ , and inserting into Eq. 2.30, we arrive at

$$G_{PAN} = \frac{\zeta^*}{\mathbf{W}_{GEV}^H \boldsymbol{\Phi}_{NN} \mathbf{W}_{GEV}}. \quad (2.31)$$

Since  $\zeta$  is an arbitrary complex scalar, we can choose the magnitude of the ATFs freely. By defining  $\|\mathbf{A}\|_2^2 \stackrel{!}{=} 1$ , Eq. 2.29 can be rearranged to  $\zeta = \mathbf{A}^H \boldsymbol{\Phi}_{NN} \mathbf{W}_{GEV}$ , which we insert into Eq. 2.31 to obtain the PAN factor:

$$G_{PAN} = \frac{\mathbf{W}_{GEV}^H \boldsymbol{\Phi}_{NN} \mathbf{A}}{\mathbf{W}_{GEV}^H \boldsymbol{\Phi}_{NN} \mathbf{W}_{GEV}}. \quad (2.32)$$

From Eq. 2.30 it can be seen that the PAN factor turns the GEV beamformer into the MVDR beamformer. However, the GEV avoids the inversion of the noise PSD matrix  $\boldsymbol{\Phi}_{NN}$  by solving the generalized Eigenvalue problem given in Eq. 2.28. This leads to improved numerical stability [78]. However, like with the MVDR beamformer, both the PSD matrices  $\boldsymbol{\Phi}_{SS}$  and  $\boldsymbol{\Phi}_{NN}$  are not directly observable.

## 2.3 Sound fields

In Section 2.2, we introduced the concept of ATFs, which models the acoustic path from a sound source to a microphone as FIR filter. In particular, this filter models all reverberations and echoes caused by reflections off of walls and other obstacles in the acoustic environment. The spatial correlation of these ATFs amongst multiple microphones is known as a *sound field* [45]. To measure a sound field, we use the spatial coherence function  $\mathbf{\Gamma}_{ZZ}(k)$ , which is an  $M \times M$  matrix for  $M$  microphones. Its elements are obtained by

$$\Gamma_{Z_m Z_n}(k) = \frac{\Phi_{Z_m Z_n}(k)}{\sqrt{\Phi_{Z_m Z_m}(k)\Phi_{Z_n Z_n}(k)}}, \quad (2.33)$$

where  $\Phi_{Z_m Z_n}(k)$ ,  $\Phi_{Z_m Z_m}(k)$  and  $\Phi_{Z_n Z_n}(k)$  denote the corresponding elements of the spatial PSD matrix  $\mathbf{\Phi}_{ZZ}(k)$  from Eq. 2.9. If the microphone signals  $\mathbf{Z}(l, k)$  are dominated by directional sounds, the spatial coherence  $\mathbf{\Gamma}_{ZZ}$  represents a *near-field*. If the microphone signals  $\mathbf{Z}(l, k)$  are dominated by diffuse sounds, the spatial coherence  $\mathbf{\Gamma}_{ZZ}$  represents a *far-field*.

### 2.3.1 Near-Field

The near-field of a microphone is considered as the region where  $\frac{\omega_k \cdot r}{c} \ll 1$ , with  $\omega_k = 2\pi \frac{k}{2K} f_s$  being the discrete frequency variable, and  $r$  is the distance from the source to the microphone array [45]. Sounds originating from this region are mostly directional, as the distance  $r$  is required to be small. Hence, we can approximate the signal arriving at the  $m^{\text{th}}$  microphone as a single plane wave, i.e.

$$Z_m(k) = \phi_Z(k) \cdot e^{-i\omega_k r_m / c}, \quad (2.34)$$

where  $\phi_Z(k)$  denotes the amplitude of the signal, and  $c$  denotes the speed of sound. Further, we assume equal amplitudes  $\phi_Z$  at each microphone, i.e. all microphones have the same ideal magnitude response. Inserting Eq. 2.34 into the definition of  $\mathbf{\Phi}_{ZZ}(k)$  in Eq. 2.9 leads to

$$\Phi_{Z_m Z_n}(k) = \mathbb{E}\{Z_m(k)Z_n^*(k)\} = \Phi_S(k) \cdot e^{-i\omega_k d_{mn} \cos \theta / c}, \quad (2.35)$$

where  $d_{mn} \cos \theta = |r_m - r_n| \cos \theta$ , i.e. the distance between the  $m^{\text{th}}$  and  $n^{\text{th}}$  microphone, as seen from the angle of the source  $\theta$ . By assuming equal amplitudes  $\phi_Z$  at all microphones, the energy  $\Phi_S(k)$  is also equal for each element  $\Phi_{Z_m Z_n}(k)$  of the PSD matrix  $\mathbf{\Phi}_{ZZ}(k)$ . Further, inserting into Eq. 2.33 leads to the directional coherence function

$$\Gamma_{Z_m Z_n}(k) = e^{-i\omega_k d_{mn} \cos \theta / c}. \quad (2.36)$$

It can be seen that the coherence purely depends on the Inter-channel Phase Differences (IPDs) of the microphone signals, the signal energy  $\Phi_S(k)$  is not relevant.

### 2.3.2 Far-Field

The far-field of a microphone is considered as the region where  $\frac{\omega_k \cdot r}{c} \gg 1$ . Sounds originating from this region are diffuse, as the distance  $r$  is required to be large, i.e. the microphones will mostly pick up reverberations of the source signal [45]. These reverberations are time-delayed reflections of the source signal, attenuated by the *reflection coefficient* of the walls of the acoustic

enclosure. Every reflection can be thought of a virtual *source image*, whose location is determined by the geometry of the enclosure and the microphone array. Typically, there are thousands of reflections in the far-field, resulting in a spherical (isotropic) distribution of the virtual source images. This distribution can be approximated by uncorrelated signals impinging from every direction with equal amplitude [45], [60]. Hence, we average the cross-spectral density over all spherical directions, i.e.

$$\begin{aligned}\Phi_{Z_m Z_n}(k) &= \frac{1}{4\pi} \int_0^\pi \int_0^{2\pi} \Phi_S \cdot e^{-i\omega_k d_{mn} \cos \theta / c} \cdot \sin \theta d\theta d\Phi \\ &= \Phi_S \frac{\sin(\omega_k d_{mn} / c)}{\omega_k d_{mn} / c},\end{aligned}\tag{2.37}$$

where  $\sin \theta d\theta d\Phi$  can be recognized as the area element of a unit sphere. By inserting this result into Eq. 2.33, we obtain the far-field (diffuse) coherence, i.e.

$$\Gamma_{Z_m Z_n}(k) = \frac{\sin(\omega_k d_{mn} / c)}{\omega_k d_{mn} / c},\tag{2.38}$$

which is a real-valued function.

### Visualization

To illustrate the spatial coherence, we measured the sound field of an arbitrary office room of size  $4m \times 5m$ , and a circular 6-element microphone array with a diameter of  $92.6mm$  [79]. We simulated a point sound source using a mobile phone, which plays broadband noise. By placing the sound source in the near-field of the microphones, i.e.  $r = 0.25m$ , we expect a directional coherence, i.e.

$$|\Gamma_{Z_m Z_n}|^2(k) = 1,\tag{2.39}$$

where we use the squared coherence to obtain a real-valued figure. By placing the sound source in the far-field of the microphones, i.e.  $r = 3m$ , we expect a diffuse coherence, i.e.

$$|\Gamma_{Z_m Z_n}|^2(k) = \frac{\sin^2(\omega_k d_{mn} / c)}{\omega_k^2 d_{mn}^2 / c^2}.\tag{2.40}$$

Figure 2.4 illustrates the squared coherence for the near-field and the far-field, using the first two microphones of the array. It is obtained using Eq. 2.9 and 2.33. Further, the theoretical result for the isotropic coherence from Eq. 2.38 is shown. It can be seen that the coherence for the directional case is close to one, indicating spatially correlated signals. Note that spatial correlation does not include temporal correlation, as the signals may still have different time lags. The coherence for the diffuse case is close to zero, indicating spatially uncorrelated signals. For low frequencies, the far-field condition is not met, as the coherence is close to one for both cases. This is due to the fact that the wavelength of low frequencies is large compared to the aperture of the microphone array. As a consequence, the IPDs between the microphone signals are small, and the spatial selectivity of the beamforming array is poor.

These observations are elementary properties of every sound field, and will be employed throughout this thesis in applications such as signal whitening, spatial decorrelation, and feature pre-processing.

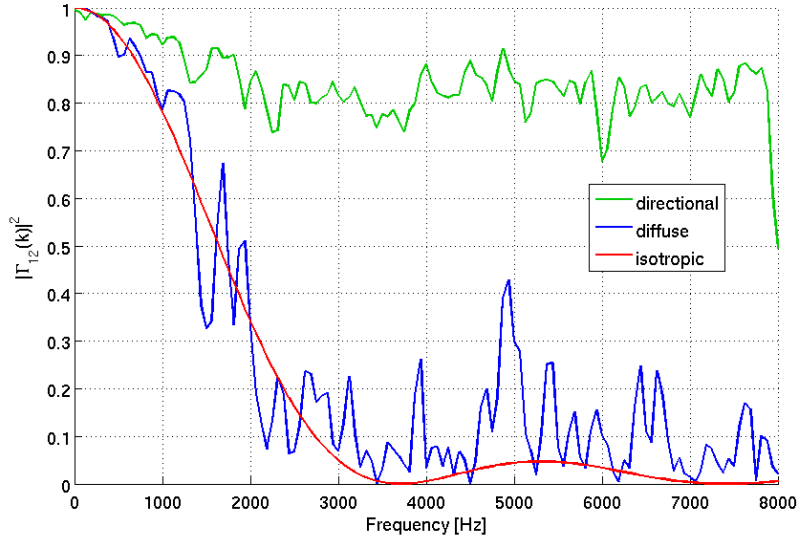


Figure 2.4: Squared spatial coherence for the near-field and the far-field between two microphones with a distance of  $d_{12} = 46.3\text{mm}$ .

## 2.4 Source Localization

As we have seen in Section 2.2, the desired speech signal is identified by the location of the speaker, relative to the microphone array. The acoustic path from the speech source to each microphone is defined by the ATFs, which are typically modeled by FIR filters. These filters contain all multi-path propagations and reverberations caused by the acoustic environment. The location of the speaker - i.e. the direction of the sound waves impinging at the array - is encoded as the group delays of these filters. Algorithms to estimate these delays are known as DOA estimators.

### 2.4.1 Steered Response Power Phase Transform

The most widely used DOA estimator is SRP-PHAT [60]. It uses a set of pre-defined time delays  $\tau_{m,q}$  and compares them against the time delays of the spatial PSD of the microphone observations  $\mathbf{Z}(k, l)$ . For a set of  $q = \{1 \dots Q\}$  of given speaker positions, and  $m = \{1 \dots M\}$  microphones,  $\tau_{m,q}$  is defined as

$$\tau_{m,q} = \frac{\sqrt{(x_m - x_q)^2 + (y_m - y_q)^2 + (z_m - z_q)^2}}{c}, \quad (2.41)$$

where  $x_m, y_m, z_m$  are the cartesian coordinates of the  $m^{\text{th}}$  microphone, and  $x_q, y_q, z_q$  are the cartesian coordinates of the  $q^{\text{th}}$  speaker position in the set. The SRP-PHAT is then defined as

$$p(k, q) = \frac{1}{M^2} \sum_{m=1}^M \sum_{n=1}^M \frac{\Phi_{Z_m Z_n}(k)}{|\Phi_{Z_m Z_n}(k)|} e^{-i\omega_k(\tau_{m,q} - \tau_{n,q})}, \quad (2.42)$$

where the term  $\Phi_{Z_m Z_n}(k)$  can be recognized as the  $(m, n)^{\text{th}}$  element of the spatial PSD matrix  $\Phi_{ZZ}(k)$  from Eq. 2.9. Maximizing over  $p(k, q)$  gives the time delays  $\tau_{m,\hat{q}}$  that best fit the

observations  $\mathbf{Z}(k, l)$ , i.e.

$$\hat{q} = \operatorname{argmax}_q \sum_{k=1}^K p(k, q). \quad (2.43)$$

The time delays  $\tau_{m, \hat{q}}$  can then be used to construct the steering vector

$$\mathbf{v}_S(k, \hat{q}) = [e^{-j\omega_k \tau_{1, \hat{q}}}, e^{-j\omega_k \tau_{2, \hat{q}}}, \dots, e^{-j\omega_k \tau_{M, \hat{q}}}]^T, \quad (2.44)$$

which may be used to construct the MVDR or GSC beamformer.

## 2.4.2 Direction-Dependent SNR

In [44], we proposed the DD-SNR, which is an alternative to the SRP-PHAT. It is based on the maximization of the SNR at the microphone signals. We formulate an analytic model of the sound field at the microphone array, which only depends on the ATFs from the sound source to the array. By approximating these ATFs with a monochromatic plane wave - i.e. a steering vector - we can infer the direction of the sound source by maximizing the SNR over a pre-defined set of source directions. Further, we assume that the spatial distribution of the background noise  $\mathbf{N}(k)$  is close to the ideal isotropic sound field. If the location of the noise source is in the far-field of the array, this assumption is always met [45], [80]. With this assumption, we can rewrite Eq. 2.11 to

$$\mathbf{\Phi}_{ZZ}(k) = \mathbf{A}(k)\mathbf{A}^H(k)\Phi_S(k) + \mathbf{\Gamma}_{NN}(k)\Phi_N(k), \quad (2.45)$$

where  $\mathbf{\Gamma}_{NN}(k)$  is the spatial coherence matrix of the ideal isotropic sound field, from Eq. 2.38. Its elements are given as

$$\Gamma_{N_m, N_n}(k) = \frac{\sin(\omega_k d_{m,n}/c)}{\omega_k d_{m,n}/c}, \quad (2.46)$$

which is equivalent to Eq. 2.38. Next, we define the  $(m, n)^{\text{th}}$  element of the coherence matrix  $\Gamma_{ZZ}$  of the microphone signals as

$$\Gamma_{Z_m Z_n}(k) = \frac{\Phi_{Z_m Z_n}(k)}{\sqrt{\Phi_{Z_m Z_m}(k)\Phi_{Z_n Z_n}(k)}}, \quad (2.47)$$

which is equivalent to Eq. 2.33. For an actual implementation, the spatial PSD matrix  $\mathbf{\Phi}_{ZZ}(k)$  may be obtained using recursive averaging, i.e.

$$\mathbf{\Phi}_{ZZ}(l, k) = \mathbf{\Phi}_{ZZ}(l-1, k)\alpha + (1-\alpha)\mathbf{Z}(l, k)\mathbf{Z}^H(l, k), \quad (2.48)$$

where  $0 \leq \alpha \leq 1$  is a smoothing parameter. If the array aperture is small, and the microphones are matched - i.e. their magnitude response is equal - we can assume that they receive the same signal energy. Hence, the diagonal elements of the PSD matrix  $\mathbf{\Phi}_{ZZ}$  are equal, i.e.

$$\Phi_{Z_m}(k) \approx \Phi_{Z_n}(k) \approx \Phi_S(k) + \Phi_N(k). \quad (2.49)$$

Note that the addition is valid, as the speech and noise signals are statistically independent. By inserting Eq. 2.49 into 2.47 and 2.45, we get

$$\mathbf{\Gamma}_{ZZ}(k) \left[ \Phi_S(k) + \Phi_N(k) \right] \approx \mathbf{A}(k) \mathbf{A}^H(k) \Phi_S(k) + \mathbf{\Gamma}_{NN}(k) \Phi_N(k). \quad (2.50)$$

Solving for the SNR  $\xi(k) = \frac{\Phi_S(k)}{\Phi_N(k)}$  leads to

$$\xi(k) \approx \text{Tr} \left\{ \left[ \mathbf{\Gamma}_{ZZ}(k) - \mathbf{A}(k) \mathbf{A}^H(k) \right]^{-1} \left[ \mathbf{\Gamma}_{NN}(k) - \mathbf{\Gamma}_{ZZ}(k) \right] \right\}. \quad (2.51)$$

In this expression, the coherence  $\mathbf{\Gamma}_{ZZ}(k)$  can be obtained using Eq. 2.47 and 2.48, and the coherence for the diffuse sound field  $\mathbf{\Gamma}_{NN}(k)$  is a constant, given by Eq. 2.46. Hence, the SNR  $\xi(k)$  only depends on the unknown ATFs  $\mathbf{A}(k)$ . By replacing the ATFs with the steering vector  $\mathbf{v}_S(k, q)$  from Eq. 2.44, we can evaluate the SNR against a pre-defined set of candidate directions  $q \in \{1 \dots Q\}$ . This leads to the DD-SNR, i.e.

$$\xi(k, q) \approx \text{Tr} \left\{ \left[ \mathbf{\Gamma}_{ZZ}(k) - \mathbf{v}_S(k, q) \mathbf{v}_S^H(k, q) \right]^{-1} \left[ \mathbf{\Gamma}_{NN}(k) - \mathbf{\Gamma}_{ZZ}(k) \right] \right\}. \quad (2.52)$$

Analogous to the SRP-PHAT, the time delays  $\tau_{m,q}$  for the steering vector  $\mathbf{v}_S(k, q)$  are calculated using Eq. 2.41. Maximizing the DD-SNR identifies the steering vector that best fits the observations at the microphones  $\mathbf{Z}(l, k)$ , i.e.

$$\hat{q} = \underset{q}{\text{argmax}} \sum_{k=1}^K \xi(k, q). \quad (2.53)$$

For further details on the DD-SNR, we refer the interested reader to Appendix A.2.

## 2.5 Postfiltering

Due to spatial correlations of the sound field, the achievable SNR at the output of a beamformer with  $M$  microphones is limited to approximately  $10 \log_{10}(M)$  dB for diffuse noise [1]. In practice, this value is significantly lower due to estimation errors in both the steering vector and the spatial noise PSD matrix. To increase the SNR at the output of the beamformer, a postfilter is used. In Section 2.2.2, Eq. 2.17 we have already defined the Wiener-optimal postfilter for the MVDR beamformer as

$$G = \frac{\Phi_S}{\Phi_S + [\mathbf{A}^H \Phi_{NN}^{-1} \mathbf{A}]^{-1}} = \frac{\Phi_S}{\Phi_S + \mathbf{W}^H \Phi_{NN} \mathbf{W}} = \frac{\Phi_S}{\Phi_S + \Phi_{\hat{N}}} = \frac{\xi}{1 + \xi}, \quad (2.54)$$

where  $\Phi_{\hat{N}}$  can be identified as the noise spectrum at the output of the beamformer. Note that  $\Phi_S$  is identical to the speech spectrum at the output of the beamformer, due to the *minimum distortion* constraint from the MVDR beamformer, i.e.  $\mathbf{A}^H \mathbf{W} \stackrel{!}{=} 1$ . Similar to SCSE, the noise spectrum  $\Phi_{\hat{N}}$  at the beamformer output has to be estimated. A widely used algorithm for this task is the IMCRA method [41], [81], [82].

### 2.5.1 Postfilter based on the GCC-PHAT

Typically, the postfilter  $G(l, k) \in [0, 1]$  is applied at the output of the beamformer, such that Eq. 2.8 expands to

$$Y(l, k) = \mathbf{W}^H(k) \mathbf{Z}(l, k) G(l, k). \quad (2.55)$$

Therefore,  $G(l, k)$  can be regarded as a real-valued gain mask, similar to SCSE. An intuitive postfilter is provided by the GCC-PHAT [60]. It is given as

$$G(l, k) = \frac{|\mathbf{Z}^H(l, k) \mathbf{v}_S(k)|^2}{\|\mathbf{Z}(l, k)\|_2^2 \cdot \|\mathbf{v}_S(k)\|_2^2}. \quad (2.56)$$

It can be seen that Eq. 2.56 exploits the cosine similarity between the magnitude-normalized microphone inputs  $\mathbf{Z}(l, k)$ , and the steering vector  $\mathbf{v}_S(k)$ . If the direction of the steering vector and  $\mathbf{Z}(l, k)$  match,  $G(l, k)$  is close to 1. For signals originating from other directions,  $G(l, k) < 1$ . However, we have already seen that the microphone signals are strongly correlated towards low frequencies in Section 2.3, Figure 2.4. This effect significantly reduces the performance of this postfilter [11], [12], [60], [74].

### 2.5.2 Postfilter based on the diffuse noise sound field

In [83], we proposed a postfilter based on *back-projection*: When extracting a single speaker from ambient noise, we typically assume the background noise to be in the far-field (diffuse), and the desired speaker to be in the near-field (directional) of the array. To direct a beamformer such as the MVDR or GSC towards the speaker, a steering vector as shown in Eq. 2.22 is required. As the steering vector consists of a set of time-delays  $\tau_1 \dots \tau_M$ , it can be thought of a monochromatic plane wave. Consequently, the enhanced signal at the output of the beamformer represents the directional component of the sound field. By back-projecting the beamformer output to the inputs, we subtract the directional component from the sound field, and only the diffuse component remains. This concept allows to formulate a postfilter, which further enhances the beamformer output.

Back-projection is achieved by multiplying the beamformer output  $Y(k)$  with the steering vector  $\mathbf{v}_S(k)$  (i.e. spatializing), and subtracting the product from the microphone signal  $\mathbf{Z}(l, k)$ . This leads to the *noise reference* signal  $\mathbf{Z}''(l, k)$ , i.e.

$$\begin{aligned} \mathbf{Z}'(l, k) &= \mathbf{v}_S(k) Y(k) = \mathbf{v}_S(k) \mathbf{W}^H(k) \mathbf{S}(k) + \mathbf{v}_S(k) \mathbf{W}^H(k) \mathbf{N}(k), \\ \mathbf{Z}''(l, k) &= \mathbf{Z}(l, k) - \mathbf{Z}'(l, k) = [\mathbf{I} - \mathbf{v}_S(k) \mathbf{W}^H(k)] \mathbf{N}(l, k). \end{aligned} \quad (2.57)$$

The spatial PSD matrices of these multi-channel signals are then given as

$$\begin{aligned} \Phi_{Z'Z'} &= \mathbf{v}_S \mathbf{W}^H \Phi_{SS} \mathbf{W} \mathbf{v}_S^H + \mathbf{v}_S \mathbf{W}^H \Phi_{NN} \mathbf{W} \mathbf{v}_S^H \\ &= \Phi_{S'S'} + \Phi_{N'N'} \\ &= \mathbf{v}_S \mathbf{v}_S^H \Phi_S + \mathbf{v}_S \mathbf{v}_S^H \Phi_{\hat{N}}, \end{aligned} \quad (2.58)$$

and

$$\begin{aligned}\Phi_{Z''Z''} &= [\mathbf{I} - \mathbf{v}_S \mathbf{W}^H] \Phi_{NN} [\mathbf{I} - \mathbf{W} \mathbf{v}_S^H] \\ &= \Phi_{N''N''},\end{aligned}\tag{2.59}$$

The SNR  $\xi(k)$  at the output of the beamformer from Eq. 2.54 can be expressed as

$$\xi(k) = \frac{\Phi_S}{\Phi_{\hat{N}}} = \frac{\text{Tr}\{\Phi_{S'S'}\}}{\text{Tr}\{\Phi_{N'N'}\}},\tag{2.60}$$

By inserting Eq. 2.58 and 2.59, we arrive at

$$\xi(k) = \frac{\text{Tr}\{\Phi_{Z'Z'}\}}{\text{Tr}\{\Phi_{Z''Z''}\}} \frac{\text{Tr}\{\Phi_{N''N''}\}}{\text{Tr}\{\Phi_{N'N'}\}} - 1,\tag{2.61}$$

Under the assumption of the ideal diffuse noise sound field, which we already defined in Eq. 2.46, we can approximate  $\Phi_{NN} \approx \Phi_N \mathbf{\Gamma}_{NN}$ , and simplify the ratio

$$\frac{\text{Tr}\{\Phi_{N''N''}\}}{\text{Tr}\{\Phi_{N'N'}\}} \approx \frac{\text{Tr}\{[\mathbf{I} - \mathbf{v}_S \mathbf{W}^H] \mathbf{\Gamma}_{NN} [\mathbf{I} - \mathbf{W} \mathbf{v}_S^H]\}}{\text{Tr}\{\mathbf{v}_S \mathbf{W}^H \mathbf{\Gamma}_{NN} \mathbf{W} \mathbf{v}_S^H\}},\tag{2.62}$$

Since we can directly measure  $\Phi_{Z'Z'}$  and  $\Phi_{Z''Z''}$  from the back-projected signals  $\mathbf{Z}'(l, k)$  and  $\mathbf{Z}''(l, k)$ , the SNR  $\xi(k)$  can be obtained from Eq. 2.61. For further details, see Appendix A.1.

## 2.6 Spatial Whitening

In Section 2.3, we have seen that the spatial selectivity of a beamforming array is poor for low frequencies. As the energy of most sounds is large at low frequencies, this effect will also have an impact on the performance of a statistical beamformer like the MVDR or GEV. In [53], [67], we proposed to decorrelate the microphone signals, using the properties of the isotropic sound field. This *spatial whitening* spreads both the IPDs and Inter-channel Time Differences (ITDs) of the microphone signals. As the whitening algorithm only depends on the array geometry, its parameters (i.e. the whitening matrix) can be calculated off-line. We demonstrate the effectiveness of spatial whitening by examining both the whitened beamformer and the whitened postfilter.

### 2.6.1 Effect on the beamformer

First, we define the steering vector with regard to the zenith angle  $\theta$  and the azimuth angle  $\phi$  of a sphere around the microphone array, i.e.

$$\mathbf{v}_S(k, \theta, \phi) = [e^{-i\omega_k \tau_{1,\theta,\phi}}, e^{-i\omega_k \tau_{2,\theta,\phi}}, \dots, e^{-i\omega_k \tau_{M,\theta,\phi}}]^T,\tag{2.63}$$



which we use to evaluate the *directivity pattern* of a simple delay-and-sum beamformer, i.e.

$$\Psi(k, \theta, \phi) = \frac{|\mathbf{Z}^H(k)\mathbf{v}_S(k, \theta, \phi)|^2}{\|\mathbf{Z}(k)\|_2^2 \cdot \|\mathbf{v}_S(k, \theta, \phi)\|_2^2}. \quad (2.64)$$

The directivity pattern  $\Psi(k, \theta, \phi)$  evaluates the normalized energy of a delay-and-sum beamformer with respect to the spherical angles  $\theta$  and  $\phi$  [11]. The input  $\mathbf{Z}(k)$  denotes a single plane wave impinging from an arbitrary direction  $\theta_Z$  and  $\phi_Z$ , simulating a single sound source. We decorrelate the microphone observations  $\mathbf{Z}(k)$  using Zero-phase Component Analysis (ZCA) whitening [84]. By using EVD, we can decompose the spatial coherence matrix of the ideal isotropic sound field from Eq. 2.38 into

$$\mathbf{\Gamma}_{NN}(k) = \mathbf{E}_\Gamma(k)\mathbf{D}_\Gamma(k)\mathbf{E}_\Gamma^H(k). \quad (2.65)$$

where  $\mathbf{E}_\Gamma$  and  $\mathbf{D}_\Gamma$  are  $M \times M$  sized Eigenvector and eigenvalue matrices of  $\mathbf{\Gamma}_{NN}(k)$ . The ZCA whitening matrix is then defined as

$$\mathbf{U}(k) = \mathbf{E}_\Gamma(k)\mathbf{D}_\Gamma^{-\frac{1}{2}}(k)\mathbf{E}_\Gamma^H(k). \quad (2.66)$$

To avoid a division by zero, the diagonal elements of  $\mathbf{D}_\Gamma$  are loaded with a small constant  $\epsilon = 10^{-3}$ . We prefer ZCA whitening over Principal Component Analysis (PCA) whitening, as the ZCA preserves the orientation of the distribution of the data [84]. Whitening of the individual time-frequency bins  $\mathbf{Z}(l, k)$  is achieved by using

$$\mathbf{Z}_U(l, k) = \mathbf{U}(k)\mathbf{Z}(l, k). \quad (2.67)$$

By whitening both  $\mathbf{Z}(k)$  and  $\mathbf{v}_S(k, \theta, \phi)$ , we arrive at

$$\Psi_U(k, \theta, \phi) = \frac{|\mathbf{Z}^H(k)\mathbf{U}^H(k) \cdot \mathbf{U}(k)\mathbf{v}_S(k, \theta, \phi)|^2}{\|\mathbf{U}(k)\mathbf{Z}(k)\|_2^2 \cdot \|\mathbf{U}(k)\mathbf{v}_S(k, \theta, \phi)\|_2^2}, \quad (2.68)$$

Figure 2.5 shows the directivity pattern for a signal  $\mathbf{Z}(k)$  originating from  $\theta_Z = \frac{\pi}{4}$  and  $\phi_Z = 0$ . Panel (a) shows the directivity  $\Psi(k, \theta, \phi)$  from Eq. 2.64, for  $\theta \in [-\pi, \pi]$  and  $\phi = 0$ . It can be seen that the directivity for low frequencies is poor. Panel (b) shows  $\Psi_U(k, \theta, \phi)$  from Eq. 2.68. It can be seen that the directivity for low frequencies is greatly increased.

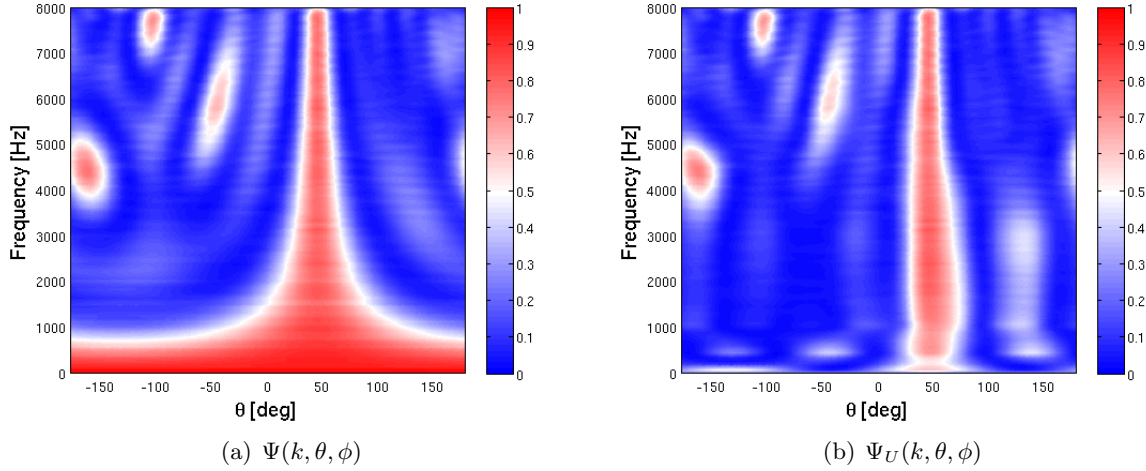


Figure 2.5: Directivity pattern for the delay-and-sum beamformer. (a)  $\Psi(k, \theta, \phi)$  from Eq. 2.56 for a single speaker at  $\theta_Z = 50^\circ$  and  $\phi_Z = 0^\circ$ . (b)  $\Psi_U(k, \theta, \phi)$  from Eq. 2.68 with whitening.

## 2.6.2 Effect on the Postfilter

Whitening the microphone signals also has a positive effect on the postfilter, which we demonstrate by using the postfilter based on the GCC-PHAT from Section 2.5. By whitening both the microphone signals  $\mathbf{Z}(l, k)$  and the steering vector  $\mathbf{v}_S(k)$  in Eq. 2.56, we obtain the whitened postfilter

$$G_U(l, k) = \frac{|\mathbf{Z}^H(l, k)\mathbf{U}^H(k) \cdot \mathbf{U}(k)\mathbf{v}_S(k)|^2}{\|\mathbf{U}(k)\mathbf{Z}(l, k)\|_2^2 \cdot \|\mathbf{U}(k)\mathbf{v}_S(k)\|_2^2}, \quad (2.69)$$

where  $\mathbf{U}(k)\mathbf{Z}(l, k)$  can be recognized as the whitened input mixture, and  $\mathbf{U}(k)\mathbf{v}_S(k)$  as whitened steering vector. Figure 2.6 demonstrates the effect of spatial whitening. Panel (a) shows  $G(l, k)$  from Eq. 2.56 for a single speaker and a matching DOA vector. It can be seen that the microphone signals are highly correlated at low frequencies. This can also be seen in Figure 2.4 in Section 2.3. Panel (b) shows  $G_U(l, k)$  from Eq. 2.69 with whitening. It can be seen that the separation performance is greatly improved for low frequencies, i.e. the microphone signals are decorrelated by the whitening matrix  $\mathbf{U}(k)$ . For further details, see Appendix A.7.

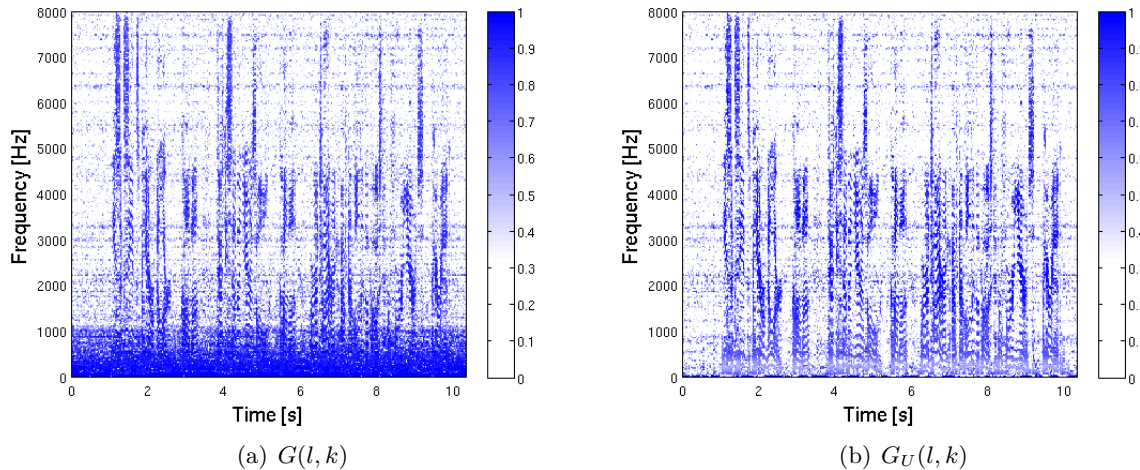


Figure 2.6: Effectiveness of spatial whitening at low frequencies. (a)  $G(l, k)$  from Eq. 2.56 for a single speaker. (b)  $G_U(l, k)$  from Eq. 2.69 with whitening.

## 2.7 Performance Measures

The performance of speech enhancement algorithms can be assessed in both speech quality and speech intelligibility. It is important to note that these two measures are not related, which has been shown 80 years ago with the first vocoders, invented at Bell Labs. A vocoder generates intelligible speech by using only a limited number of sine wave generators, completely ignoring any quality attributes. The intelligibility of speech can easily be quantified using a technical measure such as counting the number of correctly understood words in an uttered sentence, i.e. using the WER or the Speech Intelligibility Index (SII) as a metric. Other attributes such as the SNR or SDR may be used to quantify the amount of additive noise, or the presence of artifacts introduced by speech enhancement algorithms.

In contrast, speech quality is highly subjective in nature, which makes it difficult to evaluate reliably or in an objective fashion. The most reliable assessment of speech quality is given by a subjective listening test. Such a test requires a preferably large group of individual listeners, who are asked to rate the quality of speech by comparing a reference signal  $s(t)$  to an enhanced - and possibly degraded - speech signal  $y(t)$ . The listening test is standardized by the MUSHRA protocol [85]. Typically, a pre-determined scale such as Mean Opinion Score (MOS) is used for this purpose. However, with the advent of widespread speech communication applications and networks, subjective listening tests became impractical, as they are time consuming and labor intensive [5], [6], [86]. Therefore, the assessment of an automated, objective quality measure for speech is an active field of research for several decades. Defining speech quality in an objective and unambiguous fashion is a non-trivial task, as it is still unknown how to model speech quality in a deterministic manner [87]. Certain quantifiable factors which contribute to *perceived* speech quality are known, i.e. loudness, pitch, noise, reverberation, or bandwidth. However, indistinct attributes such as 'natural', 'scratchy', 'muffled' or 'timbre' are much harder to quantify. Further, psychoacoustic effects such as spectral and temporal masking, pitch perception, or sound localization play an important role in the assessment of speech quality. Also, the physiologic properties of the human auditory system cannot be ignored. Measurable properties such as the hearing threshold and bandwidth are modeled using Head-Related Transfer Functions (HRTFs) [1], [12]. The earliest psychoacoustic measures made an attempt to evaluate speech quality based on *discontinuity*, *noisiness*, and *coloration* of speech [88]–[90]. Algorithms such as the Speech Transmission Index (STI) [91] are able to assess simple nonlinear degradations such as clipping, or degradations introduced by the acoustic transmission path like a telephone line. More recent

measures model the subjective judgement of the average human listener, by predicting a MOS or percent score. Examples are the Perceptual Objective Listening Quality Analysis (POLQA) [92], Perceptual Model-Quality Assessment (PEMO-Q) [93], PESQ [94] or PEASS [95].

In this chapter, we will focus on the SNR, SDR, and WER as speech intelligibility measures, and the STOI, PESQ, and PEASS as objective speech quality measures.

### 2.7.1 SNR

As the most intuitive performance measure, the SNR is used as a means to express the ratio of desired to interfering signal energy. In the case of speech signals, this ratio is expressed in decibels, i.e.

$$SNR = 10\log_{10} \frac{\Phi_S}{\Phi_N} = 10\log_{10} \frac{\mathbb{E}\{|s(t)|^2\}}{\mathbb{E}\{|n(t)|^2\}}, \quad (2.70)$$

where  $s(t)$  and  $n(t)$  denote the desired and interfering signals, respectively. The SNR can also be used in frequency-domain representation, i.e.

$$SNR = 10\log_{10} \sum_{k=1}^K w(k) \frac{\Phi_S(k)}{\Phi_N(k)}, \quad (2.71)$$

where  $w(k)$  denotes a frequency-dependent weighting index. Due to its simplicity, it is often used as a design criterion for AEC or beamforming algorithms, e.g. the MVDR in Section 2.2.2. A relation to psychoacoustic measures such as the *articulation index* or the PESQ has been shown in [12]. Further, it is well known that the performance of ASR systems correlates directly with the SNR of the input speech signal.

### 2.7.2 SDR

An alternative to the SNR is given by the SDR [12]. It aims at measuring distortions originating from speech enhancement algorithms, i.e. residual echoes for AECs, or noise artifacts from both SCSE and MCSE algorithms. These artifacts are measured by comparing an enhanced signal  $y(t)$  with a clean reference signal  $s(t)$ . The SDR is defined as

$$SDR = 10\log_{10} \frac{\phi_S}{\phi_D} = 10\log_{10} \frac{\mathbb{E}\{|s(t)|^2\}}{\mathbb{E}\{|s(t) - y(t)|^2\}}, \quad (2.72)$$

where  $\phi_D$  denotes the energy of the distortions, i.e.  $s(t) - y(t)$ . It can also be formulated in frequency-domain, even though the result is dominated by mismatches in frequency bands where the energy of the reference signal  $s(t)$  is low. Further, the scale and sign of the enhanced signal  $y(t)$  have a significant impact on the SDR, as the slightest difference in amplitude results in a large mismatch. To address this issue, the Scale Independent Signal to Distortion Ratio (SI-SDR) has been formulated.

## SI-SDR

The SI-SDR weighs the reference signal  $s(t)$  before calculating the SDR [96], i.e.

$$\text{SI-SDR} = 10 \log_{10} \frac{\mathbb{E}\{|\alpha s(t)|^2\}}{\mathbb{E}\{|\alpha s(t) - y(t)|^2\}}. \quad (2.73)$$

The weighing factor  $\alpha$  maximizes the SDR by ensuring orthogonality between the reference signal  $s(t)$  and the residual  $s(t) - y(t)$ . It is given as

$$\alpha = \frac{\mathbb{E}\{s(t)y(t)\}}{\mathbb{E}\{|s(t)|^2\}}. \quad (2.74)$$

The SI-SDR is especially useful for speech enhancement or speech generation algorithms that do not focus on the signal magnitude, but rather on the signal statistics, such as temporal characteristics or formant structures. Examples are Text-To-Speech (TTS) systems or *non-linear* speech enhancement algorithms based on DNNs.

### 2.7.3 WER

In human-machine interfaces, algorithms for ASR or keyword spotting are used. The usability and acceptance of such systems depends greatly on the correct recognition and interpretation of the uttered commands [86]. Clearly, speech intelligibility plays a major role in this process, as speech degradation results in reduced recognition rates. Usually, the recognition rate is expressed as WER, which is a metric based on the Levenshtein distance [97]. It measures the similarity between two sentences, based on the correct recognition of individual words. In particular, the WER expresses the ratio of word recognition errors to the total number of words in a sentence, i.e.

$$\text{WER} = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}, \quad (2.75)$$

where  $\{S, D, I, C\}$  are the number of substitutions, deletions, insertions, and correct words, respectively. The total number of words in the sentence is given as  $N = S + D + C$ . Word substitutions account for erroneously identified words, insertions account for missing words, and deletions account for recognized words that are not part of the spoken utterance.

### 2.7.4 STOI

While the SNR, SNR, and WER are technical measures, the STOI is considered a psychoacoustic measure. It aims at evaluating the intelligibility of a speech signal which has been modified by a time-frequency weighing, i.e. spectral subtraction in SCSE. It is closely related to the STI, which measures the speech intelligibility of a signal after it has been degraded by a transmission path, such as a telephone line or a loudspeaker [5], [90], [98]. In contrast to many objective intelligibility measures, it does not evaluate the long-term statistics of entire speech signals, but rather local regions of 384ms in length. Therefore it is able to capture the fast variations in the temporal statistics of speech, which are affected by the great majority of both SCSE and MCSE algorithms [12]. The STOI measure operates on 15 one-third octave bands between 150Hz and 5kHz, where a STFT with a window length 50ms and 50% overlap is used. It compares the time-frequency bins of a clean reference signal  $s(t)$  and a degraded signal  $y(t)$ , whose intelligibility is to be evaluated. The time-frequency bins of both signals are normalized and weighted, before a

linear correlation coefficient is calculated to express the similarity of both signals in each band. The overall score is obtained by averaging over all one-third octave bands and all STFT time frames, and ranges from 0 to 100%.

### 2.7.5 PESQ

While psychoacoustic measures like the STOI are suitable for evaluating the distortions introduced by speech enhancement algorithms, PESQ explicitly addresses distortions encountered when speech is transmitted over telecommunication networks, i.e. clipping, packet loss, signal delays, jitter, additive noise, and codec compression artifacts. Even though these distortions may not be caused by speech enhancement algorithms directly, they cannot be ignored when evaluating the speech quality and intelligibility of the whole speech transmission system. To objectively quantify these types of distortions, PESQ was selected as the ITU-T recommendation P.862, thereby replacing the old Perceptual Speech Quality Measure (PSQM) standard [94]. The structure of the PESQ measure is shown in Figure 2.7. The reference signal  $s(t)$  and the degraded signal  $y(t)$  are equalized to the same level, and filtered to emulate the transmission bandwidth of a standard telephone headset, i.e. 300-3400Hz. Then, the signals are time-aligned to correct time delays introduced by the transmission network. Next, the signals are converted from the frequency-domain to the loudness domain using an auditory transform, i.e. the bark scale [12]. The loudness differences between the reference and the degraded signal are weighted to account for the distortions that may be perceived by a human listener, and averaged over time and frequency to predict an objective score closely related to the MOS.

With the advent of digital speech transmission and Voice over IP (VoIP) networks, PSQM quickly became a de-facto standard for automated assessment of speech quality and intelligibility of telephone systems. To accommodate to the rapid advances in available network bandwidth, a number of extensions have been proposed to PESQ, i.e. binaural listening through headphones, and a wideband frequency response which extends the standard telephone bandwidth to 50-7000Hz. Further, the predicted score is mapped by a logistic function to better fit the MOS. These extensions are summarized as *wideband* PESQ, and are documented in the ITU-T recommendation P.862.2 [5]. Figure 2.8 illustrates the relation between the predicted score for both the narrowband and wideband variants of PESQ and the perceptual MOS score.

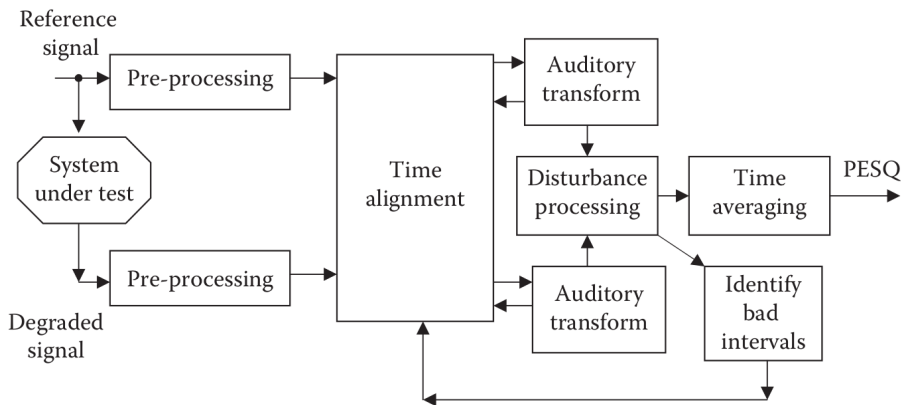


Figure 2.7: Block diagram of the PESQ algorithms [5].

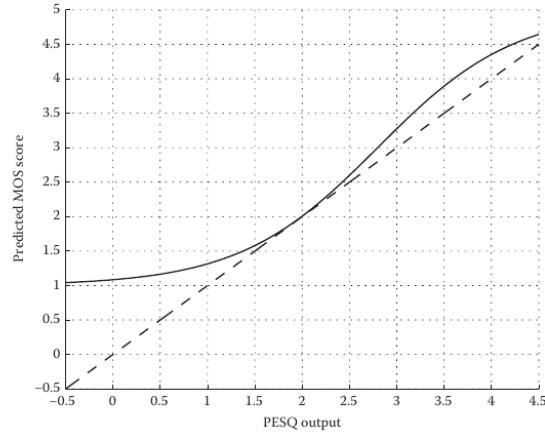


Figure 2.8: Output mapping between narrowband PESQ (dashed line), and wideband PESQ (solid line) to the subjective MOS score [5].

### 2.7.6 PEASS

To further improve on the assessment of perceptual speech quality, PEASS models the subjective judgement of human listeners using non-linear function approximators, i.e. NNs [95], [99]. Further, it addresses multi-channel speech enhancement methods such as beamforming. Therefore, PEASS can also be used to evaluate speech separation algorithms. It takes three multi-channel inputs: (i) the enhanced signal  $\hat{\mathbf{s}}(t)$ , (ii) the reference signal  $\mathbf{s}(t)$ , (iii) and the noise reference  $\mathbf{n}(t)$ . PEASS implements the following three-step procedure:

#### Decomposition of the input signals

First, the input signals are converted to the frequency-domain using the STFT. Next, the signals are converted from the frequency-domain to the loudness domain using an auditory transform, a gammatone filterbank [12]. This loudness information is then used to obtain three distortion components: (i) *target distortions*  $e^{\text{target}}$ , *additive interferences*  $e^{\text{interf}}$ , and *artifacts*  $e^{\text{artif}}$ , such that

$$\hat{s}_j(t) - s_j(t) = e_j^{\text{target}}(t) + e_j^{\text{interf}}(t) + e_j^{\text{artif}}(t), \quad (2.76)$$

where  $j$  denotes the channel index of the multi-channel input signals.

#### Assessment of the perceptual salience of each component

Next, the perceptual salience of each of the three components in Eq. 2.76 is expressed as the following four energy ratios: (i) A global ratio  $q_j^{\text{overall}}$ , which estimates the perceptual quality of the enhanced signal. (ii) A target-distortion ratio  $q_j^{\text{target}}$ , which measures the perceived distortions in relation to the reference signal. (iii) An interference ratio  $q_j^{\text{interf}}$ , which measures the perceived distortions in relation to the noise reference signal. (iv) An artifact ratio  $q_j^{\text{artif}}$ , measuring additional artifacts such as musical noise. The ratios are calculated using the PEMO-

Q algorithm [93], i.e.

$$\begin{aligned}
 q_j^{\text{overall}} &= \text{PEMO-Q}(\hat{s}_j(t), s_j(t)) \\
 q_j^{\text{target}} &= \text{PEMO-Q}(\hat{s}_j(t), \hat{s}_j(t) - e_j^{\text{target}}(t)) \\
 q_j^{\text{interf}} &= \text{PEMO-Q}(\hat{s}_j(t), \hat{s}_j(t) - e_j^{\text{interf}}(t)) \\
 q_j^{\text{artif}} &= \text{PEMO-Q}(\hat{s}_j(t), \hat{s}_j(t) - e_j^{\text{artif}}(t))
 \end{aligned} \tag{2.77}$$

### Non-linear mapping into four objective scores

The PEMO-Q saliences are then used as input features to a NN, which predicts the following four objective scores: (i) the Overall Perceptual Score (OPS), (ii) the Target Perceptual Score (TPS), (iii) the Interference Perceptual Score (IPS), and (iv) the Artifact Perceptual Score (APS). Each score ranges from 0 to 100 for improved human readability, where a larger number indicates a higher perceptual quality. To approximate the subjective judgement of human listeners, the NN is trained on 6400 subjective scores, which have been obtained by 23 human listeners for a variety of acoustic scenarios and background noises following the MUSHRA protocol [85]. Figure 2.9 illustrates the block diagram of the NN.

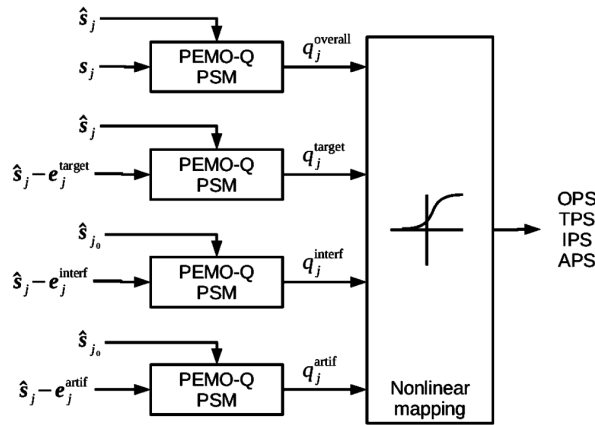


Figure 2.9: Block diagram of the NN used by the PEASS algorithm for the computation of the OPS, TPS, IPS and APS scores. [95]



## 3

## Mask-based Beamforming

## 3.1 Motivation

With the definition of the MVDR beamformer and the Wiener-optimal postfilter in Section 2.2.2 the theoretical SNR limit in terms of linear signal processing can be reached. However, directing the beamformer towards the desired source is still a challenging task, as the location of that source may be unknown, and the spatial PSD matrices of the individual sources are not observable. If the PSD matrices of the desired speaker and the unwanted noise are known, every statistical beamformer can be calculated without the need for adaptive algorithms such as the GSC beamformer. Based on the observation that human speech is sparse both in time and frequency, it is possible to estimate the spectrum of either the speech or the noise, by only evaluating the time-frequency bins of the input mixture  $\mathbf{Z}(l, k)$  which are dominated by the respective signal component. Hence, instead of the spatial PSD matrices, only the required activity patterns of the speech and the noise have to be estimated. Such an activity pattern is known as a *speech mask*.

This approach can be generalized to more than two independent sources in the input mixture, i.e. to multiple speakers for BSS: For  $C$  sources, the time-frequency bins of the input mixture  $\mathbf{Z}(l, k)$  are divided into  $C$  partitions, so that each partition only contains one source. Clearly, with an increasing number of sources, more overlaps occur, i.e. time-frequency bins which contain more than one source. However, as long as this percentage is low, this effect can be neglected [72], [78], [100]. For multi-channel signals, the location of each source is encoded in the ITDs and IPDs of the input, i.e. in the time delay or phase of the data. Therefore, stationary sources can be identified by their constant spatial statistics. The relation between the spatial statistics and the speech mask can be modeled using non-linear function approximators, where we distinguish two types: (i) Model-based approaches such as GMMs, where the Expectation Maximization (EM) algorithm is used to find the MAP estimates of the model parameters [65], [100], [101]. These methods usually rely on some prior knowledge about application-specific variables, i.e. the array geometry or the statistics of the noise. (ii) Data-driven approaches using DNNs [19], [20], [48]–[51], [72], where a DNN is used to infer the speech mask from the noisy microphone observations.

The CHiME3 challenge was won by a model-based approach, where a complex Gaussian Mixture Model (cGMM) is used to model the PSD matrices of the involved sound sources [65]. The model parameters are estimated with an EM algorithm, and the posterior probabilities are used as speech masks for the respective source components. However, a template for both the spatial speech and noise PSD matrices is required to initialize the EM algorithm [102]. While model-based approaches are often easy to implement, they have a number of other limitations, such as a fixed number of sources, or pre-determined signal statistics to describe the type and location of the involved signals, i.e. ambient background noise. Further, the PSD matrices are required to be constant over time by the GMM, which limits the model to stationary sources. Further limitations arise from both the frequency and source *permutation problem* [29], where it is not known which extracted signal belongs to which speaker. This problem can arise on an utterance level, where whole signal chunks are permuted, or on a frequency level, where

individual frequency bins are permuted [65].

All of these problems have already been solved by data-driven approaches using DNNs [53], [56], [78], [103]–[105]. They have the advantage of jointly estimating a mask for all frequencies, thereby exploiting both the spatial and frequency information embedded in the data. Further, several *end-to-end* solutions have been proposed to combine mask-based beamformers with ASR systems [104], [106], [107], allowing for joint training of both the mask estimator and the ASR front-end, allowing the ASR system to benefit from the multi-channel signal representation.

In this chapter, we introduce our contributions to mask-based beamforming, which includes our CHiME3 system in Appendix A.3, our CHiME4 system in Appendix A.4, and the *Eigenvector* beamformer in Appendix A.5. Further, we address multi-speaker separation and tracking in Appendix A.7.

## 3.2 Speech Masks

A speech mask provides the activity pattern of a desired source signal over the time-frequency bins of a given multi-channel microphone signal. The speech mask is represented a probability  $p(l, k) \in [0, 1]$ , which indicates whether the input  $Z(l, k)$  is part of the desired signal or not. This SPP [100] can be formulated as either Ideal Ratio Mask (IRM), Ideal Binary Mask (IBM) [23] or Cosine Similarity Mask (CSM) [51]. The IRM or *soft-mask* is defined as:

$$p_{\text{IRM}}(l, k) = \frac{\Phi_S(l, k)}{\Phi_S(l, k) + \Phi_N(l, k)} = \frac{\xi(l, k)}{1 + \xi(l, k)}, \quad (3.1)$$

where  $\Phi_S(l, k) = \sum_{m=1}^M |S(l, k, m)|^2$  denotes the instantaneous energy of the desired speech signal, and  $\Phi_N(l, k)$  is the instantaneous energy of the interfering noise signal(s). The ratio  $\xi(l, k) = \frac{\Phi_S(l, k)}{\Phi_N(l, k)}$  can be recognized as the instantaneous SNR, which is related to the Wiener-optimal postfilter in Section 2.2.2, Eq. 2.17. The IBM is defined using the indicator function

$$p_{\text{IBM}}(l, k) = \mathbb{1}(\Phi_S(l, k) > \Phi_N(l, k)), \quad (3.2)$$

where  $p_{\text{IBM}}(l, k)$  is assigned 1 if  $\Phi_S(l, k) > \Phi_N(l, k)$ , and 0 otherwise. For  $C > 2$  sources, the microphone signals are defined as

$$\mathbf{Z}(l, k) = \sum_{c=1}^C \mathbf{S}_c(l, k), \quad (3.3)$$

where we define  $S_1(l, k)$  as the desired source signal, and  $\sum_{c=2}^C S_c(l, k)$  as the interfering noise signal, without loss of generality. For both the IRM and IBM, the sum of all speech masks for all sources equates to 1, i.e.

$$\sum_{c=1}^C p_c(l, k) = 1. \quad (3.4)$$

An exception is given by the CSM [51], which we define as

$$p_{\text{CSM}}(l, k) = \frac{|\mathbf{Z}^H(l, k)\mathbf{U}^H(k) \cdot \mathbf{U}(k)\mathbf{S}(k)|^2}{\|\mathbf{U}(k)\mathbf{Z}(l, k)\|_2^2 \cdot \|\mathbf{U}(k)\mathbf{S}(k)\|_2^2}, \quad (3.5)$$

where  $\mathbf{U}(k)$  can be recognized as the whitening matrix from Eq. 2.67. The whitening matrix was derived from the ideal isotropic sound field, which assumes spatially isotropic noise. This assumption is often met in scenarios involving reverberation and many sources, as shown in [53], [58], [67]. The whitening is necessary to increase the spatial selectivity of the cosine similarity, as the signals  $\mathbf{Z}(k)$  and  $\mathbf{S}(k)$  are highly correlated towards low frequencies, as shown in Section 2.6, Figure 2.6. It can be seen from Eq. 3.1 that the IRM is solely defined by the signal energies, whereas the CSM from Eq. 3.5 is solely defined by the IPDs.

To illustrate the different speech masks, we used the 6-channel microphone array from [79], and recorded two speakers  $\mathbf{S}_1(l, k)$  and  $\mathbf{S}_2(l, k)$  at arbitrary positions in front of the array, and some background noise  $\mathbf{S}_3(l, k)$  from an office room. Figure 3.1 illustrates the mixture  $\mathbf{Z}(l, k)$ , and the three different speech masks, where the desired signal is defined as  $\mathbf{S}(l, k) = \mathbf{S}_1(l, k)$ , and the interfering noise is defined as  $\mathbf{N}(l, k) = \mathbf{S}_2(l, k) + \mathbf{S}_3(l, k)$ . From the pitch in panel (a) it can be seen that the input mixture contains a female and a male speaker. The pitch of the female speaker is significantly higher than the pitch of the male speaker. The IBM in panel (b) extracts the activity pattern of the female speaker using Eq. 3.2. The IRM and CSM in panels (c) and (d) appear to be very similar, except for a bit more noise in the CSM. However, the CSM only uses the spatial information embedded in the IPDs of the multi-channel input signal.

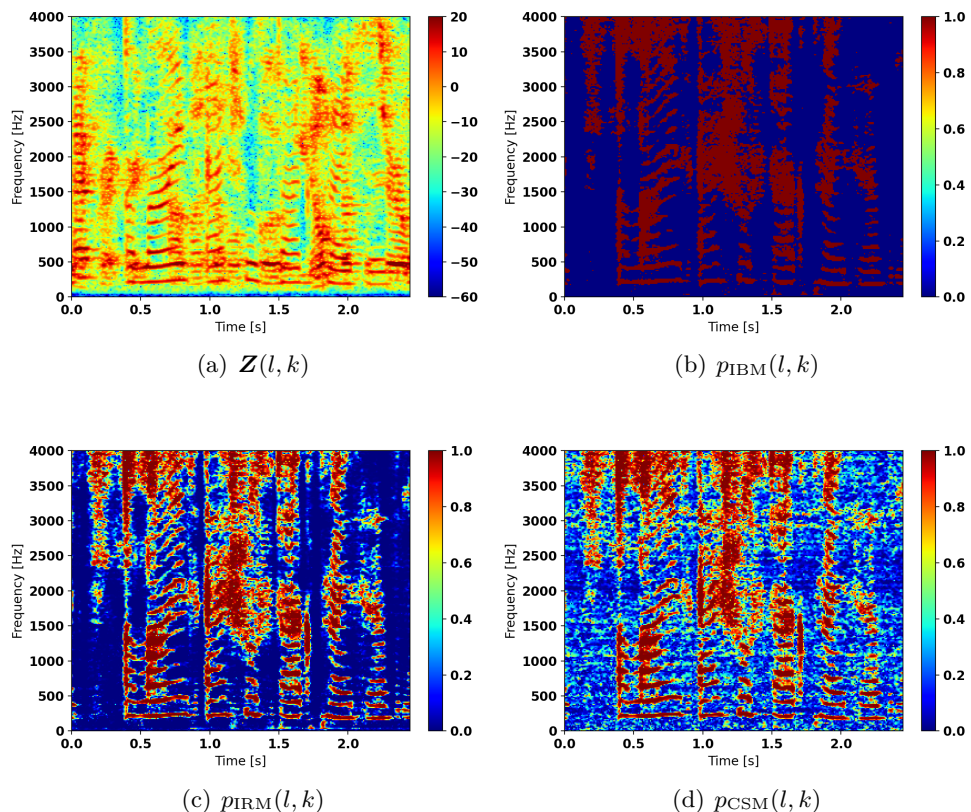


Figure 3.1: Speech masks. (a) First channel of the input mixture  $\mathbf{Z}(l, k)$ . (b) IBM. (c) IRM. (d) CSM.

### 3.2.1 PSD Matrix Estimation

By using a speech mask for a specific source  $c$ , we can approximate the spatial PSD matrix belonging to that source by partitioning the time-frequency bins of the input  $\mathbf{Z}(l, k)$ , i.e.

$$\hat{\Phi}_{SS,c}(l, k) = \frac{\sum_{t=l-T/2}^{l+T/2} \mathbf{Z}(t, k) \mathbf{Z}^H(t, k) p_c(t, k)}{\sum_{t=l-T/2}^{l+T/2} p_c(t, k)}, \quad (3.6)$$

where the window length  $T$  defines the number of frames during which we assume the spatial characteristics of  $\Phi_{SS,c}(l, k)$  to be stationary, i.e. the corresponding speaker is not moving.

This *block processing* allows to apply a statistical beamformer to a whole block of  $T$  frames at once. For moving sources, a trade-off has to be made: If  $T$  is set too small, the accuracy of the estimated PSD matrices might be poor. The matrices might even become singular if the gain mask is sparse. This may cause numerical problems with the MVDR or GEV beamformers, where matrix inversion or EVD is required. If the window length  $T$  is too large, the estimated PSD matrices might fail to adapt quickly enough to changes in the spatial characteristics of the potentially moving sources [58], [106]. By using recursive averaging, *real-time* operation becomes possible, i.e.

$$\hat{\Phi}_{SS,c}(l, k) = \hat{\Phi}_{SS,c}(l-1, k)[1 - p_c(l, k)] + \mathbf{Z}(l, k) \mathbf{Z}^H(l, k) p_c(l, k). \quad (3.7)$$

This formulation allows tracking the spatial characteristics in  $\hat{\Phi}_{SS,c}(l, k)$ . However, caution should be taken with the extreme values of the speech mask. If  $p_c(l, k)$  is very close to 1, the estimate  $\hat{\Phi}_{SS,c}(l, k)$  will also become singular. Further, Eq. 3.7 must be initialized with Eq. 3.6, to be usable in a real application. Similar approaches can be found in [65], [78].

### 3.2.2 Speech Mask Estimation

There is a wide variety of approaches to estimate a speech mask from noisy microphone observations, i.e. using cGMMs [65], Parametric Multichannel Wiener Filter (PMWF) [100], or GMMs [101]. Since the CHiME3 and CHiME4 challenges [50], [72], non-linear function approximators, i.e. NNs, are used to estimate a speech mask for a single speaker embedded in background noise [19], [20], [78], [103]–[105]. For this purpose, magnitude-spectra are used as feature vectors. However, the spatial information embedded in the IPDs of the microphone signals is neglected. Consequently, such models lack the ability to separate multiple speakers from a mixture. In this chapter, we introduce our *Eignnet* structure, which is able to extract multiple speakers from a mixture by exploiting the spatial information embedded in the Eigenvectors of the spatial PSD matrix of the noise microphone signals.

## 3.3 Neural Networks

In its simplest form, a NN is identical to a Multi-Layer Perceptron (MLP), which consists of hierarchically stacked Feed-Forward layers. Each layer applies an affine transformation to the input, followed by a non-linear activation function. The multilayer Feed-Forward architecture gives NNs the potential of being universal approximators [108]. Therefore, NNs are considered as *non-linear function approximators*, which are capable to learn abstract feature representations via back-propagation [109], genetic algorithms [110], or sampling [111].

### 3.3.1 Neural Network Layers

In this section, the most prominent building blocks of artificial NNs are briefly introduced.

#### Feed-Forward Layer

A Feed-Forward or Dense layer resembles a single layer of an MLP. It consists of several neurons, which output a weighted sum of its inputs, followed by a non-linear activation function [63]. A *fully connected* Dense layer operates on the input vector  $\mathbf{z} \in \mathbb{R}$ . Its output is defined as:

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{z} + \mathbf{b}), \quad (3.8)$$

where  $\mathbf{W} \in \mathbb{R}$  denotes a weight matrix, i.e. the *kernel*,  $\mathbf{b} \in \mathbb{R}$  is a bias vector, and  $\sigma(\cdot)$  is a non-linear activation function. Typically, a tanh or Rectified Linear Unit (ReLU) [112] activation are used.

#### Long Short-Term Memory layer

To model temporal correlations within a sequence of data points, Recurrent Neural Networks (RNNs) have been proposed [113]. A single RNN cell extends the Feed-Forward structure by adding a recurrent connection, where the output at time step  $t$  depends on the previous time step  $t - 1$ . A RNN cell is defined as

$$\mathbf{y}_t = \sigma(\mathbf{W}\mathbf{z}_t + \mathbf{U}\mathbf{y}_{t-1} + \mathbf{b}), \quad (3.9)$$

where  $\mathbf{U}$  denotes the recurrent weight matrix, and  $\mathbf{W}$  denotes the input weights. The RNN cell is trained using back-propagation through time, i.e. each time-step receives the gradient of the previous step. For long time sequences the gradient tends to diminish exponentially, thereby preventing the NN parameters from receiving further updates. To address this *vanishing gradient problem*, the LSTM cell has been proposed [114]. Further, LSTM cells store information in an internal cell state, which allows to model long-term dependencies within the data. An LSTM cell is defined as:

$$\mathbf{i}_t = \sigma_g(\mathbf{W}_i\mathbf{z}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i), \quad (3.10a)$$

$$\mathbf{f}_t = \sigma_g(\mathbf{W}_f\mathbf{z}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f), \quad (3.10b)$$

$$\mathbf{o}_t = \sigma_g(\mathbf{W}_o\mathbf{z}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o), \quad (3.10c)$$

$$\mathbf{a}_t = \sigma_a(\mathbf{W}_a\mathbf{z}_t + \mathbf{U}_a\mathbf{h}_{t-1} + \mathbf{b}_a), \quad (3.10d)$$

$$\mathbf{c}_t = \mathbf{a}_t \odot \mathbf{i}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1}, \quad (3.10e)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \sigma_h(\mathbf{c}_t), \quad (3.10f)$$

where  $\mathbf{i}_t$ ,  $\mathbf{f}_t$  and  $\mathbf{o}_t$  denote the *input*, *forget* and *output* gates, respectively. A nonlinear activation function  $\sigma_g(\cdot) \in [0, 1]$  is used to control the respective gate. Typically, a sigmoid function is chosen. To keep the cell state  $\mathbf{c}_t$  from overflowing, the magnitude of the gate activation must be smaller than 1, i.e.  $0 < \sigma_g(\cdot) < 1$ . Each gate depends on the input vector  $\mathbf{z}_t$ , the previous hidden state vector  $\mathbf{h}_{t-1}$ , the weight matrices  $\mathbf{W}$ ,  $\mathbf{U}$  and bias vectors  $\mathbf{b}$ . The cell input activation  $\mathbf{a}_t$  is controlled by the activation function  $\sigma_a(\cdot)$ , which is typically a tanh function. The cell state  $\mathbf{c}_t$  is updated using Eq. 3.10e. The output of the LSTM cell is provided by the hidden state  $\mathbf{h}_t$ , where another tanh function is typically used as activation  $\sigma_h(\cdot)$ .

### Convolutional Layer

A Convolutional Neural Network (CNN) layer is able to capture the spatial and temporal dependencies in multi-dimensional data such as images and sounds through the application of learnable filter kernels. Inspired by the visual cortex of the human brain, CNNs enabled monumental progress in the field of computer vision. In its simplest form, a CNN is defined as

$$\mathbf{y} = \sigma(\mathbf{W} \circledast \mathbf{z} + \mathbf{b}), \quad (3.11)$$

where  $\mathbf{W} \in \mathbb{R}$  are the filter weights,  $\mathbf{b} \in \mathbb{R}$  is a bias term, and  $\sigma(\cdot)$  is a non-linear activation function. Typically, a ReLU activation is used. Eq. 3.11 depicts a single filter channel of a convolutional layer. For multiple channels, multiple instances of Eq. 3.11 are executed in parallel.

### Back-propagation

NNs are trained using back-propagation [109]. Therefore, each function in the forward or *inference* path of a NN must be fully differentiable. The gradient of every trainable network parameter  $\theta_i \in \Theta$  is derived with respect to a cost or objective function  $J(\Theta)$ . Due to the hierarchical structure of a NN, the gradient follows the nested structure from the objective function back to the first layer of the network. Therefore, the chain rule [63] must be applied to calculate  $\frac{\partial J(\Theta)}{\partial \Theta}$ . For example, with a 2-layer NN, the objective is defined as

$$J(\Theta) = h_2(h_1(\Theta)), \quad (3.12)$$

Then, the gradient with respect to the parameters  $\Theta$  is derived using

$$\frac{\partial J(\Theta)}{\partial \Theta} = \frac{\partial J}{\partial h_1} \frac{\partial h_1}{\partial \Theta}. \quad (3.13)$$

The chain rule can be iteratively extended to pass error signals from the last layer of the NN back to the inputs. This *backward* path is computed using automatic differentiation by dedicated machine-learning frameworks. With the individual gradients  $\frac{\partial J}{\partial \theta_i}$ , the network parameters are updated using the delta rule, i.e.

$$\theta_{t+1} \leftarrow \theta_t - \mu \frac{\partial J}{\partial \theta_t}, \quad (3.14)$$

where  $t$  denotes the time step, and  $\theta_t \in \mathbb{R}$  are the network parameters. More sophisticated parameter update rules include stochastic gradient descent [115], or one of its extensions with momentum, i.e. ADAM [116].

## 3.4 Eigenvector-based Speech Mask Estimation

As discussed in Section 3.2, a speech mask is used to extract a single speaker embedded in background noise. Typically, a NN is used to estimate this speech mask from the spectral magnitude features of the noisy microphone observations [19], [20], [78], [103]–[105].

However, the spatial information embedded in the phase of the microphone signals is neglected. Consequently, only a single speech source can be extracted from a noisy mixture. By utilizing

the dominant Eigenvector of the spatial PSD matrix  $\Phi_{ZZ}(k)$  of the noisy microphone inputs, the loudest source in a noisy speech mixture can be extracted, i.e. the desired speaker. We refer to this approach as *Single-speaker Eigennet*. To separate multiple speakers based on their location, we utilize the normalized phase of the noisy speech mixture to track and isolate speakers within a pre-defined region of interest. We refer to this approach as *Multi-speaker Eigennet*.

### 3.4.1 Single-speaker Eigennet

In a typical beamforming application, we expect one or multiple speech sources embedded in ambient background noise such as random street noise or car engine noise. A distinct location can be assigned to each speaker, as they are an almost ideal point sound source, and typically close to the microphone array. On the other hand, most forms of background noise have no distinct origin, and are typically farther away from the microphone array. Based on the observations in Section 2.3, we can assume that the speakers exhibit a directional sound field (near-field), and the noise features a diffuse sound field (far-field).

The time-frequency bins of the STFT of such a noisy mixture have two remarkable properties: (i) Time-frequency bins that belong to a near-field source, i.e. a speaker, have a *stable* phase, if the speaker is stationary or slowly moving. (ii) Time-frequency bins occupied by a far-field source, i.e. the background noise, have an *unstable* phase, as there is no distinct origin. This allows separating the involved sources based on the stability of their respective phases. By performing EVD of the spatial PSD matrix  $\Phi_{ZZ}(k)$ , we get the phase of the *dominant* source in the microphone signals  $\mathbf{Z}(l, k)$ , i.e.

$$\Phi_{ZZ,U}(k) = \sum_{m=1}^M \lambda_{Z_m}(k) \mathbf{v}_{Z_m}(k) \mathbf{v}_{Z_m}^H(k), \quad (3.15)$$

where  $\lambda_{Z_m}(k)$  and  $\mathbf{v}_{Z_m}(k)$  are the Eigenvalues and Eigenvectors of the whitened PSD matrix  $\Phi_{ZZ,U}(k)$ , respectively. Note that  $\lambda_{Z_m}(k)$  corresponds to the signal power, and  $\mathbf{v}_{Z_m}(k)$  corresponds to the spatial information embedded in the whole signal, i.e. all time frames  $l = 1 \dots L$ . We denote  $m = 1$  as the *dominant Eigenvector*  $\mathbf{v}_{Z_1}(k)$  belonging to the largest eigenvalue  $\lambda_{Z_1}(k)$ . Clearly, the loudest signal component in the mixture  $\mathbf{Z}(l, k)$  will exhibit the largest eigenvalue. As shown in Section 2.6, the spatial selectivity of the phase in  $\mathbf{Z}(l, k)$  is greatly increased by using ZCA whitening. We use Eq. 2.67 to obtain the whitened time-frequency bins  $\mathbf{Z}_U(l, k) = \mathbf{U}(k)\mathbf{Z}(l, k)$ . With the definition of the spatial PSD matrix  $\Phi_{ZZ}(k)$  in Eq. 2.9, we calculate the whitened PSD matrix as

$$\Phi_{ZZ,U}(k) = \mathbf{U}(k)\Phi_{ZZ}(k)\mathbf{U}^H(k). \quad (3.16)$$

For a single speech source embedded in background noise, the dominant Eigenvector  $\mathbf{v}_{Z_1}(k)$  will either point towards the desired speech source, or towards a random position caused by the random phase of the diffuse background noise, depending on which of the two components is the loudest in the PSD matrix  $\Phi_{ZZ,U}(k)$ . Depending on the application, different approaches may be chosen to obtain this PSD matrix. In the case of the CHiME4-challenge [64], the desired speech source is not moving. Hence, whole utterances may be used to calculate the PSD matrix. If the speaker moves during an utterance, *block processing* may be used, where  $\Phi_{ZZ,U}(k)$  is calculated over a limited amount of time frames to track the speaker's location [48]. We use the cosine similarity between the dominant Eigenvector  $\mathbf{v}_{Z_1}(k)$  and the magnitude-normalized

time-frequency bins  $\mathbf{Z}_U(l, k)$  as speech mask  $p_{\text{evd}}(l, k)$ , i.e.

$$p_{\text{evd}}(l, k) = \frac{|\mathbf{Z}_U^H(l, k)\mathbf{v}_{Z_1}(k)|^2}{\|\mathbf{Z}_U(l, k)\|_2^2}, \quad (3.17)$$

where a value close to 1 indicates a high similarity, i.e. the direction of  $\mathbf{Z}_U(l, k)$  is the same as the Eigenvector  $\mathbf{v}_{Z_1}(k)$ . This means that the time-frequency bin  $\mathbf{Z}_U(l, k)$  has a high probability to belong to the desired source. A value close to zero indicates a low similarity, i.e. the direction of the two components are dissimilar, and the time-frequency bin  $\mathbf{Z}_U(l, k)$  has a low probability to belong to the desired source. Eq. 3.17 also provides an intuitive insight why more microphones increase the performance of this speech mask: The dimensionality of the Eigenvector  $\mathbf{v}_{Z_1}(k)$  is equal to the number of the microphones being used, i.e.  $M$ . The higher the dimensionality, the smaller the probability that the cosine similarity is large for random signals. However, there are three sources of errors for this speech mask: (i) Unwanted noise components may point towards the direction of the Eigenvector by chance. (ii) The noise may be louder than the desired speech signal at certain frequencies, resulting in an Eigenvector not pointing at the desired speech source. (iii) The noise may contain directional components from close-by sound sources like a second speaker.

We have no analytic means to correct these errors, but we can exploit the time-frequency structure of human speech to improve the speech mask. To do so, a NN is trained on  $p_{\text{evd}}(l, k)$ , where the IRM from Eq. 3.1 is used as label during training. In its simplest form, this NN consists of a bidirectional LSTM layer, and a Dense layer. It uses a sigmoid activation function to output the predicted speech mask  $p_{\text{est}}(l, k)$ . Figure 3.2 illustrates the architecture of the Eigennet. The predicted speech mask is used to obtain the spatial PSD matrices of the speech and noise components  $\Phi_{SS}(l, k)$  and  $\Phi_{NN}(l, k)$ , as shown in Eq. 3.6. With these PSDs, the weights  $\mathbf{W}(k)$  of the GEV or MVDR beamformer can be constructed. For further details, see Appendix A.7.

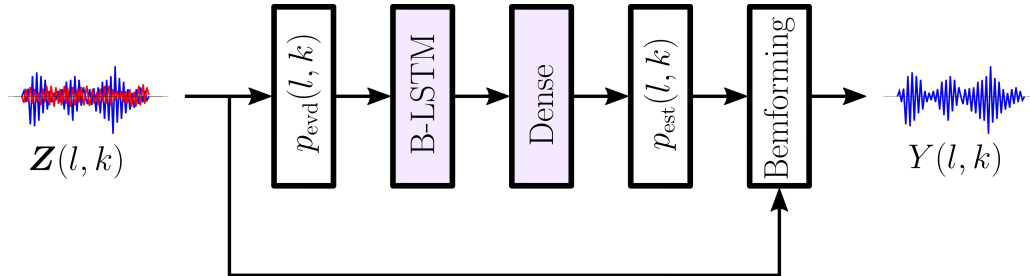


Figure 3.2: Block diagram of the Eigennet architecture [53].

### Subspace Steering

With the predicted speech mask  $p_{\text{est}}(l, k)$  and Eq. 3.6, we can estimate the PSD matrices of the desired and interfering signals. While the GEV beamformer can be constructed from these PSD matrices as shown in Section 2.2.4, the MVDR beamformer also requires a steering vector  $\mathbf{v}_S(k)$ , which provides a spatial focus of the speech source. In Section 2.4 we used DOA based steering vectors. However, it is also possible to extract the steering vector from signal subspace [42]. EVD of the speech PSD matrix  $\Phi_{SS}(l, k)$  yields:

$$\Phi_{SS}(k) = \mathbf{A}(k)\mathbf{A}^H(k)\Phi_S(k) = \sum_{m=1}^M \mathbf{v}_{S_m}(k)\mathbf{v}_{S_m}^H(k)\lambda_{S_m}(k), \quad (3.18)$$



where  $\lambda_{S_m}$  and  $\mathbf{v}_{S_m}$  are the Eigenvalues and Eigenvectors of  $\Phi_{SS}(l, k)$ . For a single sound source with a distinct location in the near-field, this matrix is of rank 1. Consequently,  $M - 1$  Eigenvalues are zero. We denote the only non-zero eigenvalue as  $\lambda_{S_1}$ , and the corresponding Eigenvector  $\mathbf{v}_{S_1}$  as dominant Eigenvector. Hence, Eq. 3.18 can be rewritten as

$$\Phi_{SS}(k) = \mathbf{v}_{S_1}(k)\mathbf{v}_{S_1}^H(k)\lambda_{S_1}(k). \quad (3.19)$$

Clearly, with  $\Phi_S(k)$  and  $\lambda_{S_1}(k)$  being scalars, the Eigenvector  $\mathbf{v}_{S_1}(k)$  must be identical to the ATF  $\mathbf{A}(k)$  up to an unknown scaling factor. Hence,  $\mathbf{v}_{S_1}(k)$  points towards the desired speaker in signal subspace. Unlike a DOA based steering vector,  $\mathbf{v}_{S_1}(k)$  is not limited to the ideal case of an anechoic environment. It can model an arbitrary FIR filter, including multi-path propagations and echoes of the target signal [11], [42], [43]. Therefore, subspace steering is preferred over DOA steering in real-world applications. However, depending on the application and the number of microphones being used, it may be favorable to use the GEV beamformer, as it has a similar performance and only requires EVD, whereas the MVDR beamformer with subspace steering requires EVD and matrix inversion.

### Eigennet Performance

To demonstrate the performance of the Eigennet architecture, we use a single utterance from the CHiME4 speech corpus [64]. The utterance F01\_22GC010X\_BUS is taken from the BUS subset, where 6-channel recordings have been obtained from a single speaker while riding on a bus. The interfering signals in this utterance include engine noise, ambient noise from other people talking, and loud directional noise from a screaming baby. Figure 3.3 shows the speech mask  $p_{\text{evd}}(l, k)$  obtained by Eq. 3.17 in panel (a). It can be seen that the speech mask contains the desired speaker, and the high-pitched cry from the baby during time 0-1s, and 6-8s. Due to the high volume of the scream, the Eigenvector  $\mathbf{v}_{Z_1}(k)$  points towards the baby rather than the desired speaker. Further, other people’s voice patterns can be recognized throughout the utterance, when comparing  $p_{\text{evd}}(l, k)$  with the ideal mask  $p_{\text{IRM}}(l, k)$  in panel (b). Due to the screaming baby, the IRM is missing some of the structure of the desired speaker. Panel (c) shows the predicted output of the NN,  $p_{\text{est}}(l, k)$ . It can be seen that the Eigennet ignored all the interferences and restored the structure of the desired speaker. Panel (d) shows the spectrogram of the first microphone signal  $Z(l, k, m = 1)$ , where the speech patterns of the desired speaker, the baby, and other passengers on the bus can clearly be identified. Panel (e) shows the enhanced signal  $Y(l, k)$  after beamforming with the GEV beamformer, and normalized using PAN (see Section 2.2.4). It can be seen that the interfering sources are suppressed by up to 30 dB.

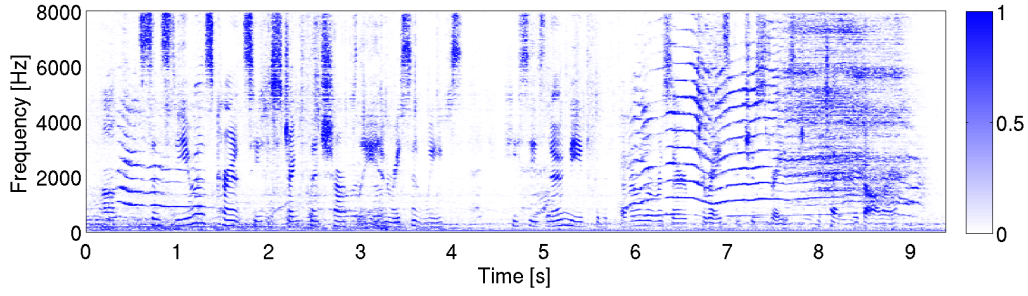
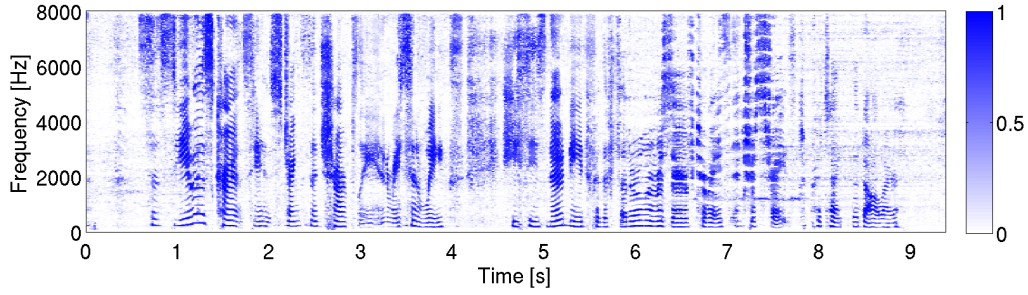
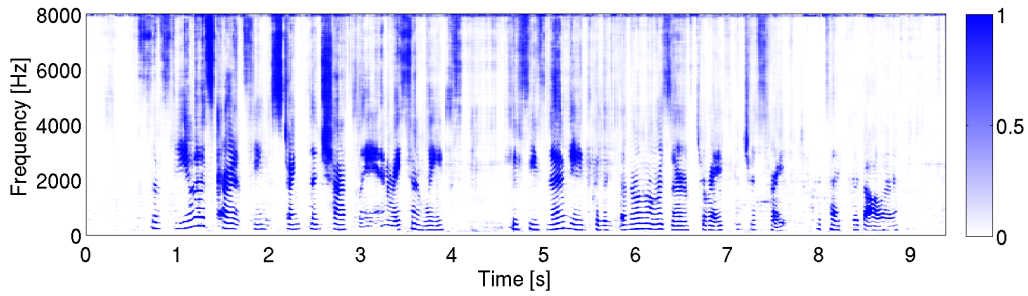
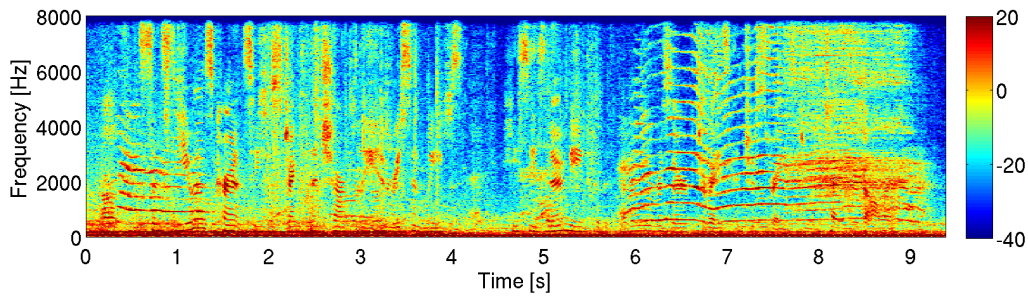
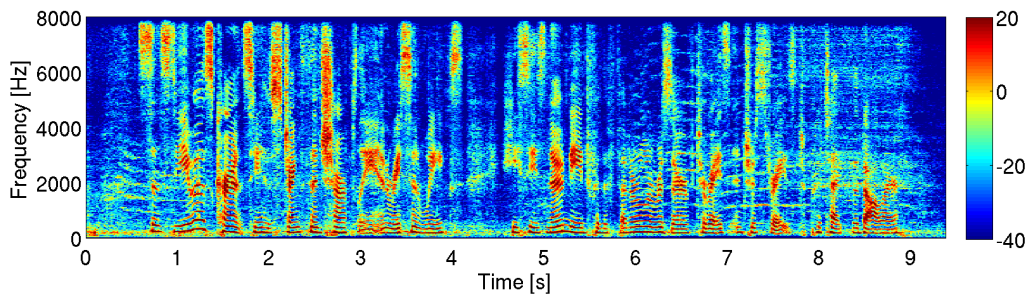
(a)  $p_{\text{evd}}(l, k)$ (b)  $p_{\text{IRM}}(l, k)$ (c)  $p_{\text{est}}(l, k)$ (d)  $Z(l, k, m = 1)$ (e)  $Y(l, k)$ 

Figure 3.3: (a) Cosine distance obtained by Eq. 3.17. (b) IIR mask obtained by Eq. 3.1. (c) Estimated speech mask using the Eigennet from [48]. (d) First channel of the noisy CHiME4 utterance F01\_22GC010X\_BUS. (e) Enhanced utterance obtained from the Eigennet and the GEV beamformer from Section 2.2.4.

### 3.4.2 Multi-speaker EigenNet

The EigenNet architecture can be extended to multiple speakers, as long as they are spatially separable. In order to do so, the NN requires additional information. Instead of  $p_{\text{evd}}(l, k)$ , we use both the spatial and magnitude information as a feature vector, i.e.

$$\mathbf{x}_{\text{EvsMag}}(l, k) = \mathbf{Z}_U(l, k)e^{-i\phi_1(l, k)}, \quad (3.20)$$

where  $\phi_1(l, k)$  denotes the phase of the first channel, i.e.  $\angle \mathbf{Z}_U(l, k)$ . This *phase normalization* is necessary to relate the complex phase to a reference microphone, rather than being randomly distributed due to the STFT. Note that this phase normalization is also done in the Single-speaker EigenNet: Each Eigenvector  $\mathbf{v}_Z(k)$  can be scaled by an arbitrary complex scalar  $\zeta$ . Typically, EVD algorithms chose  $\zeta = \frac{v_{Z,1}^*(k)}{|v_{Z,1}(k)|}$ , such that the phase of the first element (i.e. the first microphone) of the Eigenvector becomes zero.

From Eq. 3.20 it can be seen that  $\mathbf{x}_{\text{EvsMag}}(l, k)$  is complex-valued. Therefore, we stack its real and imaginary components to obtain real-valued features as inputs for the NN. Note that this NN is significantly larger than the single-speaker EigenNet, as it requires  $2M$  features per time-frequency bin, whereas the single-speaker EigenNet only requires one feature per time-frequency bin. With the spatial information fully available to the NN, it becomes possible to distinguish speakers based on their location. If the speaker is within a specific region of acceptance, the NN outputs ones for the respective time-frequency bins of the speech mask. Similarly, if the speaker is within a region of rejection, the NN outputs zeros for those time-frequency bins. This also allows for some speaker movement, as long as the speakers do not leave their designated regions. These regions are defined by choosing the training data accordingly, i.e. labeling the speech masks for speakers within the region of acceptance with ones. This information is sufficient to enable the NN to perform speaker separation without any knowledge of the microphone array or the acoustic environment. The NN is able to infer this data from the spatial information embedded within the training examples [53]. Figure 3.4 shows an example of how to achieve this selectivity with four speakers in an arbitrary arrangement and room. The region of interest or acceptance is colored in green, and the region of rejection is colored in red. The training data is selected in such a way that each speaker within the green region is a desired speech source  $\mathbf{S}(t, k)$ , and the sum of all speakers within the red region are the interfering noise sources  $\mathbf{N}(t, k)$ . The training labels, i.e. the IRM, can be obtained by Eq. 3.1. For further details, see Appendix A.7.

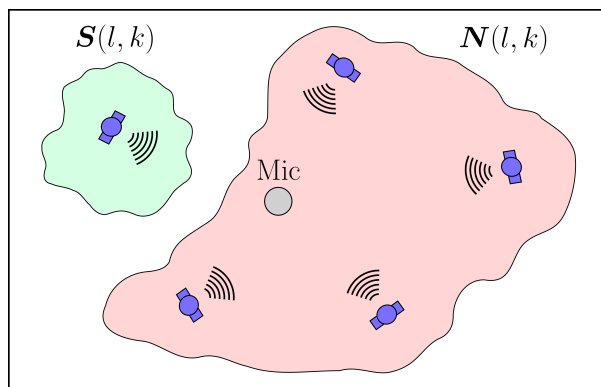


Figure 3.4: Floorplan of a rectangular room with a circular microphone array and a region of acceptance (green) and a region of rejection (red).

### 3.5 Conclusion

The Eigennet architecture allows estimating a speech mask from noisy microphone data by utilizing the spatial information embedded in the dominant Eigenvector of the whitened spatial PSD matrix  $\Phi_{ZZ,U}(k)$ . For a single speaker embedded in ambient diffuse noise, the stability of this Eigenvector is observed over the time frames of the STFT transformed microphone signals. A stationary, near-field speaker will have a stable Eigenvector, whereas diffuse noise will have a random Eigenvector. This behavior is exploited by a NN, which infers a speech mask from this data. This architecture allows to extract a single speaker from background noise.

For multiple speakers, the Eigennet utilizes the phase and the magnitude of the individual time-frequency bins of the STFT-transformed microphone signals, which encodes the location of the desired sound source. Speaker separation is achieved by defining regions (locations) of acceptance and rejection. The involved speakers may also be moving, and separation is possible as long as they do not leave their designated regions. However, there are three limitations to the Eigennet architecture: (i) In practice, it cannot be guaranteed that a moving speaker does not leave a pre-defined region. Therefore, the method is best applied to static speakers, i.e. the interior of a car. (ii) While a beamformer can separate speakers who occupy the same time-frequency bins, the speech mask cannot. Therefore, mask-based beamforming relies on the sparsity of human speech in both time and frequency. With multiple speakers, this assumption is harder to meet, resulting in a reduced quality of the beamformed output signal. (iii) Further, there is no real speaker tracking, as the *identity* of the target speaker is not considered. Each speaker within the region of acceptance is regarded as the target signal. If two speakers are in this region at the same time, both will be contained in the beamformed output signal. So far, mask-based beamforming solves two of the six problems stated in the introduction, i.e.

1. **Isolate a single speaker from background noise.** ✓
2. **Isolate a single speaker from a mixture of multiple speakers.** ✓
3. **Track moving speakers.**
4. **Isolate and dereverberate a speaker in the far-field.**
5. **Separate all speakers in a mixture of multiple speakers.**
6. **Assign an identity to an isolated speaker.**

Mask-based beamforming still requires a traditional beamformer such as the MVDR or GEV, which provides a static set of beamforming weights. In block-processing mode, a new speech mask and beamforming weights may be calculated for consecutive blocks of audio. However, speaker movements within that block, or shadowing by other speakers may also degrade the beamformed output signal. By stacking the real and imaginary parts of the input features  $Z_U(l, k)$ , certain properties of complex numbers such as rotation are lost. These properties may be useful to infer the location of the individual speakers, but have to be learned explicitly by the NN.

## 4

# Complex-valued Neural Beamforming

## 4.1 Motivation

Based on the limitations of the Eigennet structure, we abandon the concept of speech masks and statistical beamformers such as the MVDR or GEV in favor of a different approach. The flexibility of NNs allows estimating optimal beamforming weights directly from the noisy microphone observations. This concept has several benefits and offers new possibilities, i.e.

1. Block processing is no longer necessary, as a new set of beamforming weights is predicted for each time-frequency bin. This enables real-time speaker tracking and precise speaker extraction.
2. The computationally expensive matrix inversion or EVD of statistical beamformers is no longer necessary.
3. The beamforming weights can be optimized for individual time-frequency bins, rather than a whole block of time frames. This allows for increased suppression rates of unwanted signal components.
4. The design criteria of the beamformer are no longer limited to max-SNR or MVDR. Instead, any criteria can be formulated as cost function of the NN.

However, as the beamforming weights live in the domain of complex numbers, such a NN requires both complex-valued inputs and outputs, which in turn requires complex-valued gradients in the backward path of the NN. Further, the beamforming operation itself involves non-holomorphic functions like conjugation or absolute value, whose gradients do not exist. A widely adopted solution for this problem is to split complex-valued numbers into their *real* and *imaginary* parts, and treat them like real-valued numbers [26], [117]. Usually, this results in losing important properties like complex rotation or symmetry [67], [118]. While it can be argued that the universal approximation theorem enables a NN with a sufficient amount of parameters to learn these properties, it has also been shown that NNs without complex gradients require more parameters for the same task [119]. Using *Wirtinger Calculus*, it is possible to derive complex-valued gradients from non-holomorphic functions with respect to a real-valued variable [66], [120], [121].

## 4.2 Complex-valued Back-propagation

To use complex-valued functions with gradient descent optimization algorithms, we need to differentiate these functions. A given function  $f(z) = f(x + iy)$  is complex-differentiable (holomorphic), if its partial derivatives satisfy the Cauchy-Riemann equations, i.e.

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \text{and} \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}, \quad (4.1)$$

where  $f(x + iy) = u(x, y) + iv(x, y)$  is chosen as basis. Many complex-valued functions are not complex-differentiable (non-holomorphic), for example:  $f(z) = z^*$ , where  $u(x, y) = x$ , and  $v(x, y) = -y$ . For this function, the partial derivatives are not equal, i.e.

$$\frac{\partial u}{\partial x} = 1, \quad \text{and} \quad \frac{\partial v}{\partial y} = -1. \quad (4.2)$$

However, it is possible to choose a different basis, i.e.  $f(z) = f(z, z^*)$ . With this basis, the partial derivatives equate to

$$\frac{\partial f}{\partial z} = \frac{1}{2} \left( \frac{\partial f}{\partial x} - i \frac{\partial f}{\partial y} \right), \quad (4.3a)$$

$$\frac{\partial f}{\partial z^*} = \frac{1}{2} \left( \frac{\partial f}{\partial x} + i \frac{\partial f}{\partial y} \right), \quad (4.3b)$$

This definition is known as *Wirtinger Calculus* [66], [122]. With the basis given in Eq. 4.3, the following derivation rule applies: When calculating the derivative of a function  $\frac{\partial f(z, z^*)}{\partial z}$ , we can regard all instances of  $z^*$  as constants. Analogously, when calculating the derivative  $\frac{\partial f(z, z^*)}{\partial z^*}$ , we can regard all instances of  $z$  as constants [120]. If  $f(z, z^*)$  is the objective function of a neural network, its output must be real-valued, even if its inputs  $z \in \mathbb{C}$  [123]. Therefore, we can switch between 4.3a and 4.3b, i.e.

$$\frac{\partial f}{\partial z^*} = \left( \frac{\partial f}{\partial z} \right)^*. \quad (4.4)$$

### 4.2.1 Complex-valued Chain Rule

Analogous to real-valued functions, the complex-valued chain rule describes the relation of gradients for nested functions, i.e. when a function is applied to the output of another. Figure 4.1 illustrates a chain of two functions:

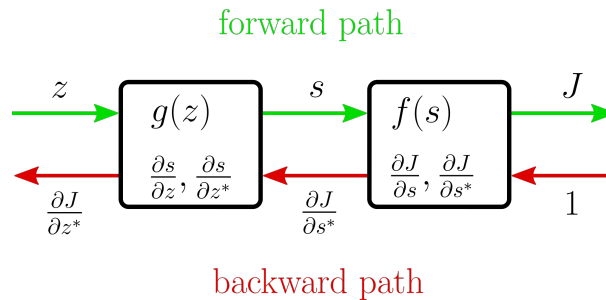


Figure 4.1: Visualization of the complex-valued chain rule: The forward path includes the complex-valued functions  $s = g(z)$  and  $J = f(s)$ . Each function has its intermediate derivatives, which contribute to the backward gradient.

In the example in Figure 4.1, the forward path consists of two complex-valued functions, i.e.

$$\begin{aligned} J &= f(s) = f(m + in), \\ s &= g(z) = g(x + iy), \end{aligned} \quad (4.5)$$

where  $z$  and  $s$  are complex-valued arguments, and  $J$  denotes a real-valued objective function. The chain rule for real-valued functions generalizes directly to the complex domain [124]. Therefore, we obtain:

$$\begin{aligned}\frac{\partial J}{\partial x} &= \frac{\partial J}{\partial m} \frac{\partial m}{\partial x} + \frac{\partial J}{\partial n} \frac{\partial n}{\partial x}, \\ \frac{\partial J}{\partial y} &= \frac{\partial J}{\partial m} \frac{\partial m}{\partial y} + \frac{\partial J}{\partial n} \frac{\partial n}{\partial y}.\end{aligned}\tag{4.6}$$

Inserting Eq. 4.6 into Eq. 4.3a leads to

$$\begin{aligned}\frac{\partial J}{\partial z^*} &= \frac{1}{2} \left( \frac{\partial J}{\partial m} \frac{\partial m}{\partial x} + \frac{\partial J}{\partial n} \frac{\partial n}{\partial x} + i \frac{\partial J}{\partial m} \frac{\partial m}{\partial y} + i \frac{\partial J}{\partial n} \frac{\partial n}{\partial y} \right), \\ &= \left( \frac{\partial J}{\partial s^*} \right)^* \frac{\partial s}{\partial z^*} + \frac{\partial J}{\partial s^*} \left( \frac{\partial s}{\partial z} \right)^*,\end{aligned}\tag{4.7}$$

with the partial derivatives [120]

$$\frac{\partial s}{\partial z^*} = \frac{1}{2} \left( \frac{\partial s}{\partial x} + i \frac{\partial s}{\partial y} \right) = \frac{1}{2} \left( \frac{\partial m}{\partial x} + i \frac{\partial n}{\partial x} + i \frac{\partial m}{\partial y} - \frac{\partial n}{\partial y} \right),\tag{4.8a}$$

$$\frac{\partial s}{\partial z} = \frac{1}{2} \left( \frac{\partial s}{\partial x} - i \frac{\partial s}{\partial y} \right) = \frac{1}{2} \left( \frac{\partial m}{\partial x} + i \frac{\partial n}{\partial x} - i \frac{\partial m}{\partial y} + \frac{\partial n}{\partial y} \right).\tag{4.8b}$$

$$\tag{4.8c}$$

Complex-valued back-propagation is achieved by iteratively applying the chain rule to each element of the neural network. Parameter optimization algorithms such as gradient descent or ADAM [116] generalize nicely to the complex domain, i.e.

$$\theta_{t+1} \leftarrow \theta_t - \mu \frac{\partial J}{\partial z^*},\tag{4.9}$$

where  $t$  denotes the time step, and  $\theta \in \mathbb{C}$  are the network parameters.

## 4.2.2 Numeric Gradients

Even though modern machine learning frameworks support complex-valued back-propagation, only a few non-holomorphic functions are implemented. To add custom functions, the gradients in Eq. 4.3a and 4.3b, and the analytic solution from Eq. 4.7 have to be implemented by hand. This can get difficult for complicated functions, therefore it is beneficial to test the implementation using a numeric approximation of the gradient for specific input values [125]. For a given function  $s = g(z) = g(x + iy)$ , the partial derivatives can be approximated using the finite differences

$$\begin{aligned}\frac{\partial s}{\partial x} &\approx \frac{g(x + \epsilon + iy) - g(x - \epsilon + iy)}{2\epsilon}, \\ \frac{\partial s}{\partial y} &\approx \frac{g(x + iy + i\epsilon) - g(x + iy - i\epsilon)}{2\epsilon},\end{aligned}\tag{4.10}$$

where  $\epsilon$  is a small, non-zero constant. Inserting Eq. 4.10 into Eq. 4.8a, 4.8b, and 4.7 gives

$$\frac{\partial J}{\partial z^*} \approx \frac{1}{2} \left( \frac{\partial J}{\partial s^*} \right)^* \left( \frac{\partial s}{\partial x} + i \frac{\partial s}{\partial y} \right) + \frac{1}{2} \frac{\partial J}{\partial s^*} \left( \frac{\partial s}{\partial x} - i \frac{\partial s}{\partial y} \right)^*. \quad (4.11)$$

This allows to perform numerical checks on the analytic gradient for a given gradient. Especially for complicated functions, Eq. 4.11 provides an essential tool to verify dedicated implementations.

### 4.2.3 Examples

In this section, we derive the analytic gradients of some relevant functions for beamforming. Further gradients of complex-valued functions can be found in [125].

#### Conjugate

For the conjugate function  $s = g(z) = z^*$ , the partial derivatives are

$$\begin{aligned} \frac{\partial s}{\partial z^*} &= 1, \\ \frac{\partial s}{\partial z} &= 0. \end{aligned} \quad (4.12)$$

Inserting into Eq. 4.7 leads to

$$\frac{\partial J}{\partial z^*} = \left( \frac{\partial J}{\partial s^*} \right)^* \frac{\partial s}{\partial z^*} + \frac{\partial J}{\partial s^*} \left( \frac{\partial s}{\partial z} \right)^* = \left( \frac{\partial J}{\partial s^*} \right)^* \cdot 1 + \frac{\partial J}{\partial s^*} \cdot 0 = \left( \frac{\partial J}{\partial s^*} \right)^*. \quad (4.13)$$

#### Squared Magnitude

For the squared magnitude function  $s = g(z) = |z|^2 = z \cdot z^*$ , the partial derivatives are

$$\begin{aligned} \frac{\partial s}{\partial z^*} &= z, \\ \frac{\partial s}{\partial z} &= z^*. \end{aligned} \quad (4.14)$$

Inserting into Eq. 4.7 leads to

$$\frac{\partial J}{\partial z^*} = \left( \frac{\partial J}{\partial s^*} \right)^* \cdot z + \frac{\partial J}{\partial s^*} \cdot \left( z^* \right)^* = 2 \cdot \operatorname{Re} \left\{ \frac{\partial J}{\partial s^*} \right\} z. \quad (4.15)$$



### Complex Tanh

We define the complex tanh function as  $s = g(z) = \tanh(|z|)\frac{z}{|z|}$ . The partial derivatives are

$$\begin{aligned}\frac{\partial s}{\partial z^*} &= \frac{\tanh(|z|)}{2|z|} + \frac{\operatorname{sech}^2(|z|)}{2}, \\ \frac{\partial s}{\partial z} &= \frac{z \operatorname{sech}^2(|z|)}{2z^*} - \frac{z^2 \tanh(|z|)}{2|z|^3}.\end{aligned}\tag{4.16}$$

Inserting into Eq. 4.7 leads to

$$\frac{\partial J}{\partial z^*} = \left(\frac{\partial J}{\partial s^*}\right)^* \left(\frac{\tanh(|z|)}{2|z|} + \frac{\operatorname{sech}^2(|z|)}{2}\right) + \frac{\partial J}{\partial s^*} \left(\frac{z \operatorname{sech}^2(|z|)}{2z^*} - \frac{z^2 \tanh(|z|)}{2|z|^3}\right)^*.\tag{4.17}$$

### Vector Magnitude Normalization

Let us consider the complex-valued vector  $\mathbf{z} = [z_1, \dots, z_N]^T$  with  $N$  elements. This vector is normalized using  $\mathbf{s} = g(\mathbf{z}) = \frac{\mathbf{z}}{|\mathbf{z}|}$ . The partial derivatives are given by

$$\begin{aligned}\frac{\partial \mathbf{s}}{\partial \mathbf{z}^*} &= -\frac{\mathbf{z}\mathbf{z}^T}{2|\mathbf{z}|^3}, \\ \frac{\partial \mathbf{s}}{\partial \mathbf{z}} &= \frac{\mathbf{I}}{|\mathbf{z}|} - \frac{\mathbf{z}\mathbf{z}^T}{2|\mathbf{z}|^3},\end{aligned}\tag{4.18}$$

where  $\mathbf{I}$  denotes the  $N \times N$  identity matrix. Inserting into Eq. 4.7 leads to

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{z}^*} &= \left(\frac{\partial J}{\partial \mathbf{s}^*}\right)^H \left(-\frac{\mathbf{z}\mathbf{z}^T}{2|\mathbf{z}|^3}\right) + \left(\frac{\partial J}{\partial \mathbf{s}^*}\right)^T \left(\frac{\mathbf{I}}{|\mathbf{z}|} - \frac{\mathbf{z}\mathbf{z}^T}{2|\mathbf{z}|^3}\right)^*, \\ &= \frac{\partial J}{\partial \mathbf{s}^*} \frac{1}{|\mathbf{z}|} - \frac{\mathbf{z}}{|\mathbf{z}|^3} \operatorname{Re}\left\{\mathbf{z}^H \frac{\partial J}{\partial \mathbf{s}^*}\right\}.\end{aligned}\tag{4.19}$$

### Vector Phase Normalization

We normalize the phase of the complex-valued vector  $\mathbf{z} = [z_1, \dots, z_N]^T$  to  $\mathbf{s} = g(\mathbf{z}) = \mathbf{z} \frac{z_1^*}{|z_1|}$ , where  $\frac{z_1^*}{|z_1|}$  denotes the phase factor of the first element of  $\mathbf{z}$ . To obtain the partial derivatives, we consider the first element  $s_1$  and the other  $N - 1$  elements  $\mathbf{s}_{2:N}$  separately, i.e.

$$\begin{aligned}s_1 &= z_1 \frac{z_1^*}{|z_1|} = |z_1|, \\ \mathbf{s}_{2:N} &= \mathbf{z}_{2:N} \frac{z_1^*}{|z_1|}.\end{aligned}\tag{4.20}$$

The partial derivatives for the first element  $z_1$  of the input are given by

$$\begin{aligned}\frac{\partial s_1}{\partial z_1^*} &= \frac{z_1}{2|z_1|}, \\ \frac{\partial s_1}{\partial z_1} &= \frac{z_1^*}{2|z_1|}.\end{aligned}\tag{4.21}$$

It can be seen from Eq. 4.20, that the first element also receives a gradient from  $\mathbf{s}_{2:N}$ . Its partial derivatives are given by

$$\begin{aligned}\frac{\partial \mathbf{s}_{2:N}}{\partial z_1^*} &= \frac{\mathbf{z}_{2:N}}{2|z_1|}, \\ \frac{\partial \mathbf{s}_{2:N}}{\partial z_1} &= -\frac{\mathbf{z}_{2:N}z_1^*}{2z_1|z_1|}.\end{aligned}\tag{4.22}$$

The partial derivatives of the other  $N - 1$  elements  $\mathbf{z}_{2:N}$  are given by

$$\begin{aligned}\frac{\partial \mathbf{s}_{2:N}}{\partial \mathbf{z}_{2:N}^*} &= 0, \\ \frac{\partial \mathbf{s}_{2:N}}{\partial \mathbf{z}_{2:N}} &= \frac{z_1^*}{|z_1|}.\end{aligned}\tag{4.23}$$

Inserting into Eq. 4.7 leads to

$$\frac{\partial J}{\partial z_1^*} = \left(\frac{\partial J}{\partial s_1^*}\right)^* \cdot \frac{z_1}{2|z_1|} + \frac{\partial J}{\partial s_1^*} \cdot \left(\frac{z_1^*}{2|z_1|}\right)^* + \left(\frac{\partial J}{\partial \mathbf{s}_{2:N}^*}\right)^* \cdot \frac{\mathbf{z}_{2:N}}{2|z_1|} + \frac{\partial J}{\partial \mathbf{s}_{2:N}^*} \cdot \left(\frac{\mathbf{z}_{2:N}z_1^*}{2z_1|z_1|}\right)^*,\tag{4.24}$$

and

$$\frac{\partial J}{\partial \mathbf{z}_{2:N}^*} = \left(\frac{\partial J}{\partial \mathbf{s}_{2:N}^*}\right)^* \cdot 0 + \frac{\partial J}{\partial \mathbf{s}_{2:N}^*} \cdot \left(\frac{z_1^*}{|z_1|}\right)^*.\tag{4.25}$$

## 4.3 Complex-valued Neural Networks

With the definition of complex back-propagation in Section 4.2 we can continue to define network elements such as activation functions and layers.

### 4.3.1 Complex-valued Activation Functions

In any neural network, the choice of a non-linearity as activation function is an important design parameter. Many activation functions are bounded (i.e. *tanh* or *sigmoid*), allowing the activations to saturate on a finite level. With complex-valued neural networks, the choice of an activation function is subject to certain restrictions: The Liouville Theorem states that any bounded holomorphic function must be a constant [66], [126]. Therefore, we can only use unbounded or non-holomorphic activation functions, where the latter requires *Wirtinger Calculus* from Section 4.2. Further caution must be taken when generalizing activation functions

from the real to the complex domain, where we have multiple options. For example, the *tanh* function can be generalized in multiple ways:

$$\sigma(z) = \tanh(z) = \frac{\sinh(z)}{\cosh(z)} = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \quad (4.26a)$$

$$\sigma(z) = \tanh(z) = \tanh(\operatorname{Re}\{z\}) + i \tanh(\operatorname{Im}\{z\}), \quad (4.26b)$$

$$\sigma(z) = \tanh(z) = \tanh(|z|) \frac{z}{|z|}, \quad (4.26c)$$

with  $z \in \mathbb{C}$ . In Eq. 4.26a we can see the straight-forward extension of the *tanh* function from the real-valued to the complex-valued domain [119]. Figure 4.2 shows the magnitude and phase response over  $z \in \mathbb{C}$ . It can be seen that this function has a value of 1.0 almost everywhere, and exhibits periodic poles with a period of  $i\pi$ . Further, its phase is discontinuous. Clearly, this function is not ideal for non-linear activation in a neural network. Eq. 4.26b shows a widely adopted scheme, where the *tanh* is calculated separately for the real and imaginary components of  $z$  [121]. Figure 4.3 shows the magnitude and phase response for this case. It can be seen that the magnitude does not saturate at 1.0, and the phase is nearly constant for each quadrant of the complex domain. This behavior is also not favorable in a non-linear activation function. Eq. 4.26c shows our approach from [67], where we only modify the magnitude with a real-valued *tanh*, and leave the phase intact. It can be seen from Figure 4.4 that the magnitude does saturate at 1.0, and that the phase is linear. Note that this representation closely resembles the behavior of a real-valued *tanh* function.

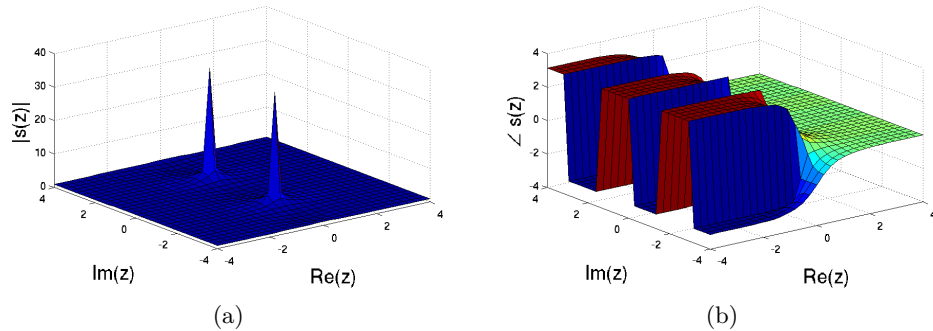


Figure 4.2: (a) Magnitude and (b) phase response of Eq. 4.26a.

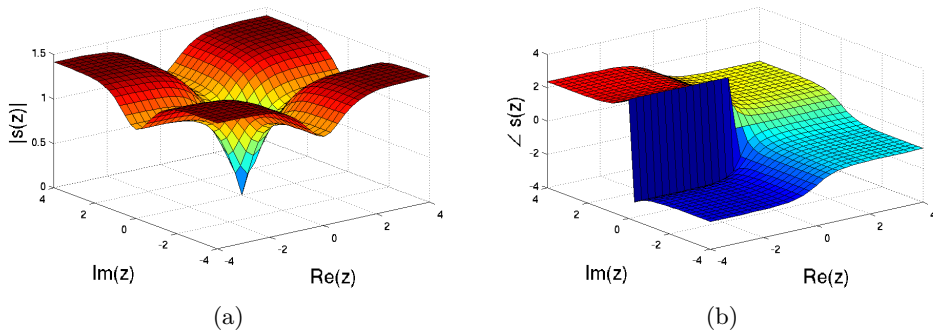


Figure 4.3: (a) Magnitude and (b) phase response of Eq. 4.26b.

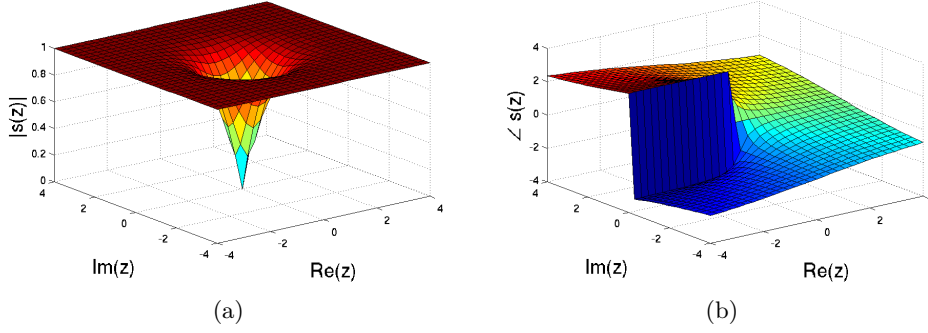


Figure 4.4: (a) Magnitude and (b) phase response of Eq. 4.26c.

Another complex-valued, non-linear activation function is given by the vector magnitude normalization from Section 4.2.3, i.e.

$$\sigma(\mathbf{z}) = \frac{\mathbf{z}}{|\mathbf{z}|}, \quad (4.27)$$

which is used in conjunction with the vector phase normalization, i.e.

$$\sigma(\mathbf{z}) = \mathbf{z} \frac{z_1^*}{|z_1|}, \quad (4.28)$$

to normalize both the magnitude and phase of the complex-valued beamforming weights  $\mathbf{W}(l, k)$ . Further activation functions are given by the complex-valued sigmoid function, i.e.

$$\sigma(z) = \frac{1}{1 + e^{-\text{Re}\{z\}}}, \quad (4.29)$$

and the complex-valued ReLU, i.e.

$$\sigma(z) = \frac{z + |\text{Re}\{z\}| + i|\text{Im}\{z\}|}{2}. \quad (4.30)$$

Figure 4.5 shows the magnitude and phase response of the complex-valued sigmoid from Eq. 4.29. It can be seen that the magnitude exhibits the shape of a sigmoid function for the real part of  $z$ , and the phase is zero, as this function only uses  $\text{Re}\{z\}$ . Figure 4.6 shows the magnitude and phase response of the complex-valued sigmoid from Eq. 4.30. It can be seen that the magnitude is zero for the quadrant where either  $\text{Re}\{z\}$  or  $\text{Im}\{z\}$  is negative. The phase is constant except for the quadrant where both  $\text{Re}\{z\}$  and  $\text{Im}\{z\}$  are positive.

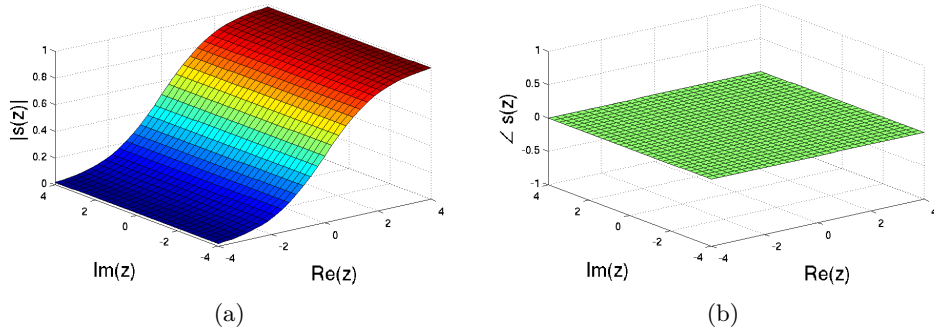


Figure 4.5: (a) Magnitude and (b) phase response of Eq. 4.29.

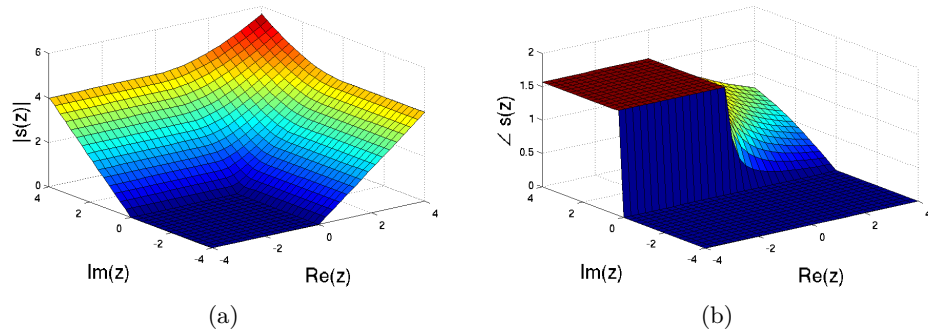


Figure 4.6: (a) Magnitude and (b) phase response of Eq. 4.30.

### 4.3.2 Complex-valued Neural Network Layers

In this section, we will extend the definition of the NN layers from Section 3.3 into the complex domain.

#### Complex-valued Feed-Forward Layer

The structure of a complex-valued Feed-Forward layer is identical to its real-valued counterpart from Eq. 3.8, i.e.

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{z} + \mathbf{b}), \quad (4.31)$$

where  $\mathbf{W} \in \mathbb{C}$  is the weight matrix, and  $\mathbf{b} \in \mathbb{C}$  is the bias vector. This generalizes well into the complex domain, as both the matrix-vector product and the addition are holomorphic functions. However, the activation function  $\sigma(\cdot)$  has to be replaced by one of the complex-valued activation functions from Section 4.3.1.

#### Complex-valued Long Short-Term Memory Layer

Analogously to the complex-valued Feed-Forward layer, the structure of the LSTM cell from Section 3.3 can be extended into the complex domain. However, all weights and variables are

$\in \mathbb{C}$ . This leads to the following definition of the complex-valued LSTM cell:

$$\mathbf{i}_t = \sigma_g(\mathbf{W}_i \mathbf{z}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \quad (4.32a)$$

$$\mathbf{f}_t = \sigma_g(\mathbf{W}_f \mathbf{z}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \quad (4.32b)$$

$$\mathbf{o}_t = \sigma_g(\mathbf{W}_o \mathbf{z}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \quad (4.32c)$$

$$\mathbf{a}_t = \sigma_a(\mathbf{W}_a \mathbf{z}_t + \mathbf{U}_a \mathbf{h}_{t-1} + \mathbf{b}_a), \quad (4.32d)$$

$$\mathbf{c}_t = \mathbf{a}_t \odot \mathbf{i}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1}, \quad (4.32e)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \sigma_h(\mathbf{c}_t), \quad (4.32f)$$

where  $\mathbf{i}_t$ ,  $\mathbf{f}_t$  and  $\mathbf{o}_t$  denote the *input*, *forget* and *output* gates, respectively. The input activation  $\mathbf{a}_t$ , the internal cell state  $\mathbf{c}_t$  and the output state  $\mathbf{h}_t$  are  $\in \mathbb{C}$ . Analogous to the real-valued LSTM cell, the magnitude of the gate activation must be smaller than 1, to prevent the cell state  $\mathbf{c}_t$  from overflowing, i.e.  $0 < \sigma_g(\cdot) < 1$ . An additional design choice needs to be made for the gate activation  $\sigma_g(\cdot)$ , as it can either have a real-valued or complex-valued output. If its output was complex-valued (i.e. with a non-zero phase), the phase of the internal cell state  $\mathbf{c}_t$  in Eq. 4.32e would change during each time step  $t$ , regardless of the weights  $\mathbf{W}$ ,  $\mathbf{U}$  or the bias  $\mathbf{b}$ . Consequently, the phase of the output  $\mathbf{h}_t$  would also change, as can be seen from Eq. 4.32f. This might lead to undesirable or even unstable behavior of the LSTM cell. To avoid this problem, we choose real-valued gate activation  $\sigma_g(\cdot)$ , i.e.

$$\sigma_g(z) = \frac{1}{1 + e^{-\text{Re}\{z\}}}, \quad (4.33)$$

The weight matrices  $\mathbf{W}$ ,  $\mathbf{U}$  and bias vectors  $\mathbf{b}$  are all  $\in \mathbb{C}$ . The cell state  $\mathbf{c}_t$  is updated using Eq. 4.32e. The complex-valued input and output activation functions are defined as

$$\sigma_a(z) = \sigma_h(z) = \tanh(|z|) \frac{z}{|z|}. \quad (4.34)$$

The output of the LSTM cell is provided by the hidden state  $\mathbf{h}_t$ . Due to the numerically demanding gradient calculation of these activation functions, complex-valued LSTM cells require longer execution times than real-valued LSTM cells. Further, the recurrent nature of these layers prevents an efficient GPU implementation.

### Complex-valued Convolutional Layer

Similar to the complex-valued Feed-Forward layer, the CNNs generalize well into the complex domain. It is defined as

$$\mathbf{y} = \sigma(\mathbf{W} \circledast \mathbf{z} + \mathbf{b}), \quad (4.35)$$

where  $\mathbf{W} \in \mathbb{C}$  are the filter weights,  $\mathbf{b} \in \mathbb{C}$  is a bias term, and  $\sigma(\cdot)$  is the complex-valued ReLU activation function, i.e.  $\sigma(z) = \frac{z + |\text{Re}\{z\}| + i|\text{Im}\{z\}|}{2}$ . As the convolution operator is holomorphic, it is possible to separate a complex-valued convolution into four real-valued convolutions [119], i.e.

$$\mathbf{W} \circledast \mathbf{z} = (\mathbf{A} \circledast \mathbf{x} - \mathbf{B} \circledast \mathbf{y}) + i(\mathbf{B} \circledast \mathbf{x} + \mathbf{A} \circledast \mathbf{y}), \quad (4.36)$$

with the complex-valued filter weights  $\mathbf{W} = \mathbf{A} + i\mathbf{B}$ , and the complex-valued inputs  $\mathbf{z} = \mathbf{x} + i\mathbf{y}$ .

Typically, efficient GPU implementations use the Fast Fourier Transform (FFT) to perform real-valued convolutions in frequency-domain [2]. Therefore, execution times are considerably faster than for LSTM cells.

## 4.4 CNBF Architecture

With the complex-valued neural network layers from Section 4.3.2, we can design a NN which accepts the complex-valued inputs  $\mathbf{Z}(l, k)$  and predicts complex-valued outputs  $\mathbf{W}(l, k)$ . We refer to this NN as CNBF. Unlike a statistical beamformer, which provides a single set of beamforming weights  $\mathbf{W}(k)$  for all  $L$  frames in a block of audio data, the CNBF provides individual beamforming weights  $\mathbf{W}(l, k)$  for each time frame  $l$ , i.e.

$$\begin{aligned} Y(l, k) &= \mathbf{W}^H(l, k)\mathbf{Z}(l, k) \\ &= \mathbf{W}^H(l, k)\mathbf{S}(l, k) + \mathbf{W}^H(l, k)\mathbf{N}(l, k) \\ &= \mathbf{W}^H(l, k)\phi_S(l, k)\mathbf{v}_S(l, k) + \mathbf{W}^H(l, k)\phi_N(l, k)\mathbf{v}_N(l, k), \end{aligned} \quad (4.37)$$

where  $\mathbf{v}_S(l, k)$  and  $\mathbf{v}_N(l, k)$  are the magnitude-normalized vectors  $\mathbf{S}(l, k)$  and  $\mathbf{N}(l, k)$ , respectively, i.e.

$$\begin{aligned} \mathbf{v}_S(l, k) &= \frac{\mathbf{S}(l, k)}{\|\mathbf{S}(l, k)\|_2}, \text{ and} \\ \mathbf{v}_N(l, k) &= \frac{\mathbf{N}(l, k)}{\|\mathbf{N}(l, k)\|_2}, \end{aligned} \quad (4.38)$$

and  $\phi_S(l, k)$  and  $\phi_N(l, k)$  are the magnitudes of  $\mathbf{S}(l, k)$  and  $\mathbf{N}(l, k)$ , respectively, i.e.

$$\begin{aligned} \phi_S(l, k) &= \|\mathbf{S}(l, k)\|_2, \text{ and} \\ \phi_N(l, k) &= \|\mathbf{N}(l, k)\|_2. \end{aligned} \quad (4.39)$$

Analogous to the statistical beamformers in Section 2.2, we have to define an optimization objective for the beamforming weights. However, there is no trade-off between signal distortions and achievable SNR as with the MVDR or GEV beamformers, because the CNBF provides individual weights  $\mathbf{W}(l, k)$  for each time-frequency bin. Therefore, we can define both a distortionless response towards the desired speech signal  $\mathbf{S}(l, k)$ , and a total suppression of the ambient noise  $\mathbf{N}(l, k)$  at the same time, i.e.

$$\mathbf{W}_{OPT}^H(l, k)\mathbf{v}_N(l, k) \stackrel{!}{=} 0, \text{ and} \quad (4.40a)$$

$$\mathbf{W}_{OPT}^H(l, k)\mathbf{v}_S(l, k) \stackrel{!}{=} 1. \quad (4.40b)$$

An intuitive solution to Eq. 4.40a is given by *null steering* [11], i.e.

$$\mathbf{W} = (\mathbf{I} - \mathbf{v}_N\mathbf{v}_N^H)\mathbf{v}_S, \quad (4.41)$$

where we omitted the time and frequency indices for brevity. From Eq. 4.38, it can easily be seen that  $\mathbf{v}_N^H\mathbf{v}_N \stackrel{!}{=} 1$ . Consequently, the constraint in Eq. 4.40a is fulfilled, i.w.  $\mathbf{W}^H\mathbf{v}_N = 0$ .

Inserting Eq. 4.41 into Eq. 4.40b gives

$$\mathbf{W}^H \mathbf{v}_S = \mathbf{v}_S (\mathbf{I} - \mathbf{v}_N \mathbf{v}_N^H) \mathbf{v}_S = 1 - |\mathbf{v}_N^H \mathbf{v}_S|^2. \quad (4.42)$$

The second constraint in Eq. 4.40b can be easily met by dividing Eq. 4.41 by 4.42, i.e.

$$\mathbf{W}_{OPT} = \frac{(\mathbf{I} - \mathbf{v}_N \mathbf{v}_N^H) \mathbf{v}_S}{1 - |\mathbf{v}_N^H \mathbf{v}_S|^2}. \quad (4.43)$$

Note that the theoretical SNR of this solution is infinite, as the noise signal is completely canceled as defined in Eq. 4.40a. However, the NN has no access to the normalized vectors  $\mathbf{v}_S(l, k)$  and  $\mathbf{v}_N(l, k)$ . Therefore, the optimal beamforming weights have to be inferred from the noisy inputs  $\mathbf{Z}(l, k)$ . By choosing an appropriate cost function, the NN will approximate  $\mathbf{W}_{OPT}(l, k)$ , i.e.

$$\mathcal{L} = - \sum_{l=1}^L \sum_{k=1}^K \Delta \text{SNR}(l, k), \quad (4.44)$$

where  $\Delta \text{SNR}(l, k)$  measures the improvement in SNR, achieved by the beamforming weights  $\mathbf{W}(l, k)$ , i.e.

$$\Delta \text{SNR}(l, k) = 10 \log_{10} \frac{|\mathbf{W}^H(l, k) \mathbf{S}(l, k)|^2}{|\mathbf{W}^H(l, k) \mathbf{N}(l, k)|^2} - 10 \log_{10} \frac{|\mathbf{S}(l, k)|_2^2}{|\mathbf{N}(l, k)|_2^2}. \quad (4.45)$$

It can be seen that Eq. 4.43 maximizes this cost function. However, the second term does not depend on the weights  $\mathbf{W}(l, k)$ , and is therefore a constant in the cost function. Its purpose is to remove the scaling of the gradient by the inputs  $\mathbf{S}(l, k)$  and  $\mathbf{N}(l, k)$ , thereby normalize the learning rate over the frequency bins  $k$ . Further, there remains a single degree of freedom for the predicted weight vectors  $\mathbf{W}(l, k)$ , as their magnitude cancels out in Eq. 4.45. We therefore choose to normalize the magnitude of the weights to 1.0, using

$$\mathbf{W}(l, k) \leftarrow \frac{\mathbf{W}(l, k)}{|\mathbf{W}(l, k)|_2}. \quad (4.46)$$

Analogous to mask-based beamforming from Chapter 3, we employ ZCA whitening from Eq. 2.67 to decorrelate the noisy microphone signals  $\mathbf{Z}(l, k)$ . With these building blocks, we can define the CNBF architecture. Figure 4.7 illustrates its block diagram. The complex-valued NN consists of two complex LSTM layers, and two complex Feed-Forward (Dense) layers. The inputs of the NN are the whitened microphone signals  $\mathbf{Z}_U(l, k)$ , and its outputs are the magnitude-normalized beamforming weights  $\mathbf{W}(l, k)$  from Eq. 4.46. The beamforming operation from Eq. 4.37 produces the enhanced signal  $Y(l, k)$ .



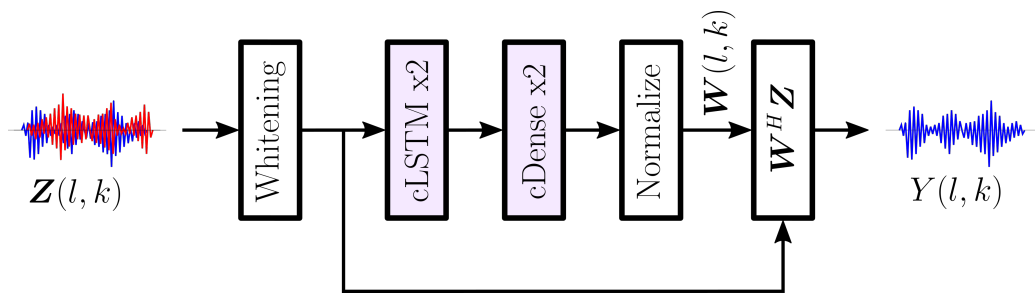


Figure 4.7: Block diagram of the CNBF architecture [67].

#### 4.4.1 Performance

We compare the CNBF to the Eigennet from Section 3.4.2 by simulating a *shoebox* model of a living room with various sound sources [45]. Figure 4.8 illustrates the acoustic setup with two static speakers  $S_1$  and  $S_2$ , a TV set  $S_3$ , and two moving speakers  $D_1$  and  $D_2$ . The dynamic paths  $D_1$  and  $D_2$  change randomly within their designated regions, illustrated by the paths on either side of the room. To simulate head movements of the static sources  $S_1$  and  $S_2$ , random position changes occur within a cube of  $20\text{cm}$  in size. We use a circular microphone array with  $M = 6$  microphones and a diameter of  $86\text{mm}$ , located next to the TV set. With this setup, we use the Image Source Method (ISM) [127] to spatialize monaural recordings from the WSJ0 speech database [128]. The WSJ0 contains 12776 utterances from 101 different speakers for training, and 5895 utterances from 18 different speakers for testing. For the dynamic paths, new Room Impulse Responses (RIRs) are generated for each 100ms of audio, where the speaker moves with  $1\text{m/s}$  along a pre-defined trajectory. Further, we generate diffuse background noise from YouTube [129], which is spatialized as described in [67].

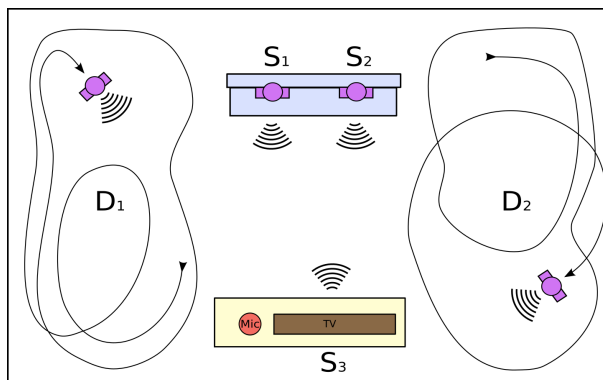


Figure 4.8: Shoebox model of a living room showing stationary sound sources  $S_1$  to  $S_3$ , and dynamic sound sources  $D_1$  and  $D_2$ . The microphone array is located next to the TV set [67].

Given the setup in Figure 4.8, we perform five experiments to compare the performance of the Eigennet from Chapter 3, and the CNBF. We compare four scores, i.e.: the  $\Delta\text{SNR}$  from Eq. 4.45, and the PESQ, STOI, and WER from Section 2.7. We use the Google Speech-to-Text API as ASR system [130]. No adaptation of the ASR has been performed. Note that the ASR framework reports a WER of 6.1% for the clean WSJ0 test set (`si_et_05`). Each experiment has a desired source  $S(l, k)$  and one or more interfering sources  $N(l, k)$ , as shown in Table 4.1. It can be seen that the CNBF outperforms the Eigennet in all scores and all experiments. This is due to the design goal of the CNBF from Eq. 4.45. It achieves a higher SNR, as it estimates optimal beamformer weights for each time-frequency bin in a max-SNR sense. This affects the other scores as well, leading to an improved overall performance. For further details, see Appendix A.8.

Method	Desired source	Interfering source(s)	$\Delta$ SNR	PESQ	STOI	WER
Eigennet	D <sub>1</sub>	D <sub>2</sub>	4.445 dB	1.514	0.834	46.2 %
	D <sub>1</sub>	diffuse	4.286 dB	1.576	0.837	32.9 %
	S <sub>1</sub>	diffuse	4.516 dB	1.751	0.866	18.6 %
	S <sub>1</sub>	S <sub>3</sub>	8.690 dB	1.439	0.811	45.7 %
	S <sub>2</sub>	D <sub>1</sub> , S <sub>3</sub>	7.011 dB	1.402	0.792	58.4 %
CNBF	D <sub>1</sub>	D <sub>2</sub>	6.156 dB	1.688	0.825	21.6 %
	D <sub>1</sub>	diffuse	8.736 dB	2.263	0.882	9.1 %
	S <sub>1</sub>	diffuse	9.558 dB	2.551	0.902	6.2 %
	S <sub>1</sub>	S <sub>3</sub>	10.306 dB	1.652	0.792	13.5 %
	S <sub>2</sub>	D <sub>1</sub> , S <sub>3</sub>	9.212 dB	1.441	0.758	33.8 %

Table 4.1: Performance comparison of the Eigennet and the CNBF methods.

Figure 4.9 demonstrates the performance of the CNBF system in terms of spectrograms of the involved signals of the fourth experiment, where  $S_1$  denotes the desired signal, and  $S_3$  is used as the interfering signal. Panel (a) shows the STFT of the first channel of the input mixture  $\mathbf{Z}(l, k)$ . Panel (b) shows the enhanced output  $Y_{\text{CNBF}}(l, k)$ , obtained by the CNBF. Panel (c) shows the enhanced output  $Y_{\text{MBF}}(l, k)$ , obtained by the Eigennet shown in Figure 3.2. And panel (d) shows the first channel of the desired target signal  $\mathbf{S}(l, k)$ . It can be seen that the enhanced signal from both the CNBF and the Eigennet remove the interfering speaker  $S_3$ . However, the SNR achieved by the CNBF is clearly higher. This is to be expected, as the CNBF estimates a set of beamforming weights for every time-frequency bin, whereas the Eigennet estimates a speech mask, which leads to a single set of beamforming weights being used for the entire utterance. Note that the optimal filter weights  $\mathbf{W}_{\text{OPT}}(l, k)$  from Eq. 4.43 would achieve perfect signal reconstruction, i.e. remove the interfering speaker completely.

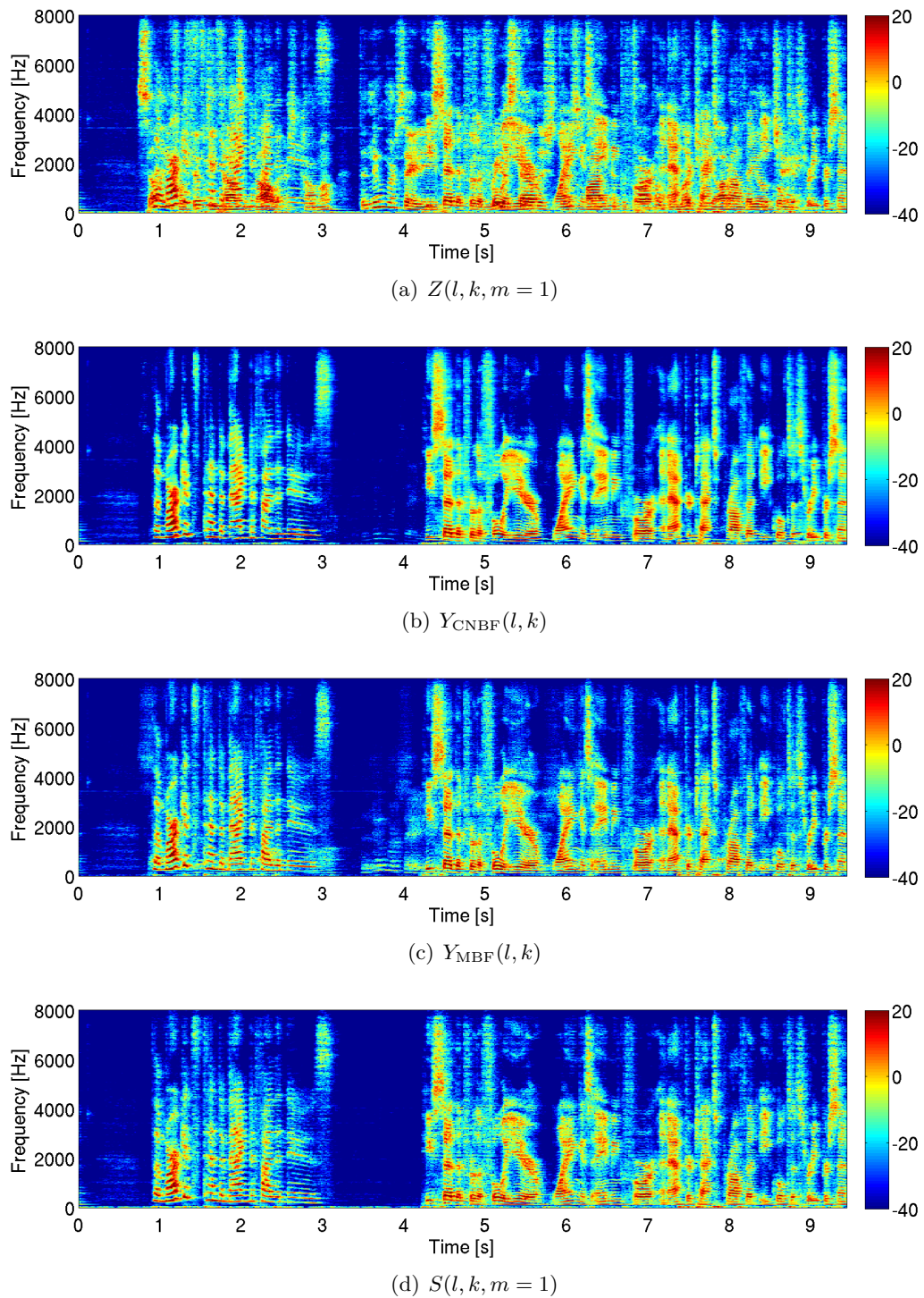


Figure 4.9: (a) First channel of the input mixture. (b) Enhanced output of the CNBF. (c) Enhanced output of the Eigennet. (d) First channel of the clean target speaker.

## 4.5 Conclusion

Unlike the Eigennet, the CNBF does not rely on a speech mask, or even on a traditional beamformer, to extract a single speaker from a mixture of multiple sound sources. By tapping into the full potential of complex-valued gradients, it is possible to outperform statistical beamformers such as the MVDR or GEV. The CNBF addresses many of the open issues of the Eigennet architecture, such as speaker tracking, fast adaption to moving speakers, or overlapping speakers in

the same time-frequency bins, which cannot be separated by a speech mask. So far, the CNBF solves three of the six problems stated in the introduction, i.e.

1. **Isolate a single speaker from background noise.** ✓
2. **Isolate a single speaker from a mixture of multiple speakers.** ✓
3. **Track moving speakers.** ✓
4. **Isolate and dereverberate a speaker in the far-field.**
5. **Separate all speakers in a mixture of multiple speakers.**
6. **Assign an identity to an isolated speaker.**

However, the issue of overlapping regions of interest, or overlapping paths of moving speakers remains unsolved. Furthermore, some practical problems arise from the usage of complex-valued gradients, i.e. Non-holomorphic functions are still not supported by major machine learning frameworks, and have therefore to be implemented by the user. Depending on the CUDA back-end and the GPU hardware being used, implementing highly optimized gradients can be a difficult task that requires an in-depth knowledge of cuDNN and the possible limitations and pitfalls of both the driver and the actual hardware. Consequently, the execution times for training complex-valued NNs are considerably larger than for their real-valued counterparts.

## 5

## Time-domain Neural Beamforming

## 5.1 Motivation

Beamforming in the frequency-domain allows for an efficient implementation of convolution operations as simple multiplications [2]. For example, the *filter and sum* beamformer in Eq. 2.8 reduces to an inner vector product, as the corresponding signals are represented in the STFT domain. With complex-valued NNs from Chapter 4, it is possible to back-propagate the complex-valued gradient of these functions, which are used in the CNBF architecture. We can further exploit the potential of complex-valued gradients by including functions such as the STFT to a NN. Both the FFT and inverse Fast Fourier Transform (iFFT) are optimized implementations of a complex-valued matrix-vector product. As such, they also have a complex-valued gradient that can be back-propagated through the NN.

Furthermore, the fixed STFT and inverse STFT operations may be replaced by learnable transformations, using network elements such as Feed-Forward or Convolutional layers. This allows for additional degrees of freedom and design choices, i.e.

1. The time-frequency representation of audio signals may not necessarily be an optimal representation for speech separation. A learnable transformation may be better suited for this task.
2. The FFT decouples the phase and magnitude information of a signal. Modeling the phase is a difficult task [14], [131]. Therefore, the majority of proposed SCSE methods only modify the magnitude of a signal.
3. The relevant spatial cues for MCSE are embedded the ITDs between the microphone channels. The FFT transforms these ITDs into IPDs, introducing issues such as spatial aliasing [11], [60]. A time-domain representation circumvents these problems.
4. The FFT requires an inherent trade-off between frequency resolution and window length, as the FFT matrix is square. This imposes restrictions on performance and real-time capabilities. A customized transformation does not have these restrictions and allows for very short latencies.
5. In time-domain, there is a greater variety of objective functions to choose from. We are not limited to the SNR, but may choose other performance measures, such as the SI-SDR, which enables additional tasks such as dereverberation.

Attracted by these possibilities, there are a number of time-domain SCSE algorithms, i.e. *Wave-U-Net* [26], *TasNet* [27] and *Conv-TasNet* [28]. Also, speech synthesizers like *WaveNet* have been successfully implemented in time-domain [25]. Further, mask-based beamformers using time-domain NNs to estimate a speech mask have been proposed, i.e. *Beam-TasNet* [68], *SpeakerBeam* [132], [133], *Neural Speech Separation* [134], *Multi-Channel Deep Clustering* [35], [135], and *Mask-based Convolutional Beamforming* [136]. However, all of these approaches still use a conventional frequency-domain beamformer such as the MVDR or GEV. Recently, true

end-to-end multi-channel speech separation - which is done entirely in time-domain - has been proposed in [137], [138] and [58]. In this chapter, we will compare time- and frequency-domain NNs in three different applications: (i) Neural beamforming, (ii) Dereverberation, (iii) Residual echo suppression.

## 5.2 Cross-domain Learning

For MCSE methods such as beamforming and BSS, the relevant information to separate sound sources such as speakers by their locations is encoded within the ITDs between the microphone channels, as demonstrated in Section 2.1. Let us ignore multi-path propagation for a moment, and look at the simplified acoustic model shown in Figure 2.1. Here,  $\tau_{s,m}$  denotes the time delay required by the sound waves to travel from the speaker  $X_S$  to the  $m^{\text{th}}$  microphone. Hence, the signal received by the microphone array can be expressed as

$$\begin{aligned} z(t, m) &= s(t + \tau_{s,m}) = s(t)\delta(t + \tau_{s,m}), \\ z(t, n) &= s(t + \tau_{s,n}) = s(t)\delta(t + \tau_{s,n}), \end{aligned} \quad (5.1)$$

where  $\delta(\cdot)$  denotes a Dirac pulse, and  $m$  and  $n$  are the indices of two different microphones. We obtain the ITD between those two microphones by calculating the temporal displacement of  $z(t, m)$  relative to  $z(t, n)$ . This is done using the cross-correlation, i.e.

$$\begin{aligned} R_{z_m, z_n}(t) &= \int_{-\infty}^{+\infty} z(\nu, m)z(t - \nu, n)d\nu, \\ &= \int_{-\infty}^{+\infty} s(\nu)s(t - \nu)\delta(t + \tau_{s,m} - \tau_{s,n})d\nu, \\ &= R_{ss}(t)\delta(t + \tau_{s,m} - \tau_{s,n}), \end{aligned} \quad (5.2)$$

where the ITD appears as Dirac pulse, i.e.  $\delta(t + \tau_{s,m} - \tau_{s,n})$ . Depending on the position of the microphones and the speakers, the ITDs will be different, which allows for a spatial separation of signals originating from different locations. Hence, a NN operating on the time-domain signals  $z(t)$  is able to infer the ITDs by performing a similar convolution operation. By transforming Eq. 5.2 into the frequency-domain, we obtain the IPD, i.e.

$$\begin{aligned} \Phi_{z_m, z_n}(f) &= \int_{-\infty}^{+\infty} R_{z_m, z_n}(t)e^{-i2\pi ft}dt, \\ &= \Phi_S(f)e^{-i2\pi f(\tau_{s,m} - \tau_{s,n})}. \end{aligned} \quad (5.3)$$

The IPD between the microphones is given by the phase, i.e.  $e^{-i2\pi f(\tau_{s,m} - \tau_{s,n})}$ . In theory, both the ITDs and IPDs contain the same spatial information. However, several restrictions apply in the frequency-domain: As sound waves travel with  $c = 343\text{m/s}$ , spatial under-sampling occurs for frequencies  $f > \frac{c}{2d_{mn}}$ , with  $d_{mn}$  being the distance between the two microphones. This under-sampling leads to aliasing effects known as *sidelobes* [11], [60]. Spatial aliasing causes ambiguous IPDs, as the phase wraps around every  $2\pi$  radians. Further, the window length of the STFT operation leads to additional problems: If the window is too short, the ITDs may not lie within the same window at all, causing aliased IPDs [2]. If the window is too large, the overall processing delay will rise, affecting the real-time capability of the application. These problems can be circumvented by replacing the STFT with a learnable transformation. A NN may be utilized to infer an optimal transformation, given only the time-domain audio data. We

refer to such an architecture as TDNBF.

Figure 5.1 illustrates a comparison of both the CNBF and TDNBF architectures. The signal flow at the top shows the CNBF using complex neural networks from Chapter 4, including the signal transformation to the frequency-domain using the STFT and inverse STFT stages. Both steps rely on the overlap-add method, which involves the multiplication with a constant window function to avoid aliasing [2], and the FFT and iFFT operations, respectively. The FFT and iFFT themselves are complex-valued matrix multiplications, with a square FFT matrix. Therefore, both the STFT and inverse STFT operations have a complex-valued gradient, and can be treated like any layer in a complex-valued NN.

The signal flow at the bottom of Figure 5.1 illustrates the pure time-domain approach of the TDNBF. It has a similar structure, but the frequency-domain has been replaced by a learnable, latent representation of the data. To transform to and from this latent representation, we use a Convolutional and a Deconvolutional layer, respectively. The Deconvolutional layer can be thought of as the inverse operation of a Convolutional layer, as it produces a time-series or multi-dimensional data from a latent representation, using the overlap-add method without a window function [139]. Note that it does not perform deconvolution in the Wiener-sense [140], it rather uses the same convolution operator as shown in Section 3.3.1. The length of the convolutional kernels determines the temporal context of each filter operation. The number of filters used in the convolutional layer determines the resolution in latent space, analogous to the number of frequency bins used by the STFT. Similarly, the stride determines the latency of the filter operation. In contrast to the STFT, all three of these parameters can be chosen independently.

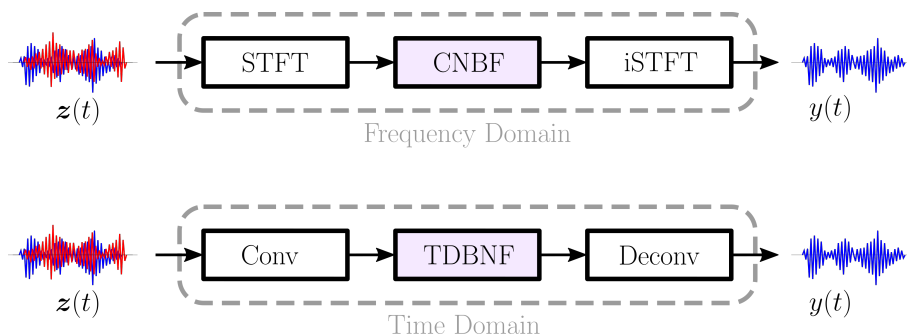


Figure 5.1: Cross-domain learning: Signal flow of the CNBF and TDNBF architectures.

In the following sections, we will discuss three applications in both time- and frequency-domain formulation, i.e. (i) Neural beamforming, using the TDNBF and the CNBF architectures. (ii) Dereverberation, using the TDNBF and Weighted Prediction Error (WPE) [141]. (iii) Residual echo suppression, using a gain mask similar to mask-based beamforming. As a prerequisite to these applications, we want to have realistic, multi-channel recordings of both stationary and moving speakers. Therefore, we use a multi-channel recording setup to obtain RIRs from real acoustic environments like offices and meeting rooms.

### 5.3 RIR Recording and Spatialization

In the last chapter, we used a simulated shoebox room [142] to spatialize monaural audio data for the CNBF experiments, i.e. we used simulated RIRs and convolved them with monaural audio data, to obtain multi-channel recordings. For the experiments in this chapter, we recorded realistic RIRs which were used to generate both stationary and moving speakers. The motivation to use recorded RIRs is rooted in the observation that the shoebox room fails to model the complexity of realistic environments, i.e. fully furnished office or living rooms with different materials and different absorption coefficients. This has already been observed in both the

CHiME3 and CHiME4 challenges [64]. Our recording setup consists of a 6-channel microphone array [79], and a 5W broadband loudspeaker (Visaton FR-58). To drive the loudspeaker from a Linux-based PC with ALSA [143], we use the PlayRec Python module [144], which simultaneously plays and records audio from a sound card. We use an exponential chirp with a duration of 5s, sweeping from 24 kHz down to 20 Hz as an excitation signal to deconvolve the RIRs [45], [145]. For the WSJ0 audio data, we only use a bandwidth of 8 kHz. The recording setup with the microphone array and the loudspeaker can be seen in Figure 5.2, panel (c).

### 5.3.1 Static RIRs

We use the above recording setup in 24 different, fully furnished office rooms with reverberation times  $RT_{60} \in [200 \dots 900]$  ms. The reverb of the room depends on multiple factors such as size, geometry, and absorption coefficient of the walls and the interior. We recorded 5 RIRs in each room, where the position of both the loudspeaker and the microphone array are chosen randomly, so that the distance between the two varies from  $1m \dots 3m$ . These 120 recordings are augmented to 720, by virtually rotating the array by  $6 \times 60^\circ$ , i.e. shifting the microphone channels. This allows to spatialize monaural data such as the WSJ0 [146] speech database with realistic RIRs, and superimpose multiple speakers, i.e.

$$\mathbf{z}(t) = \sum_{c=1}^C s_c(t) \circledast \mathbf{h}_c(t) \quad (5.4)$$

where  $\mathbf{h}_c(t)$  denotes the RIR for the  $c^{\text{th}}$  speaker, and  $C$  is the total number of speakers in the mixture. The vector  $\mathbf{h}_c(t)$  includes  $M$  RIRs from a specific location to the individual microphones. Each of these individual RIRs is convolved with the same monaural signal  $s_c(t)$ . Note that spatialization is done in frequency-domain, as the RIRs are very long. A similar concept can be found in the *wsj0-2mix* from [147], where a mixture of 2 speakers is created using WSJ0 data, but without reverberation or spatialization.

### 5.3.2 Dynamic RIRs

To simulate moving speakers, we use the above recording setup in a  $6m \times 7m$  office room with a reverberation times  $RT_{60}$  of about 450 ms. Here, we placed the microphone array on a table in the middle of the room, and record  $448 \times 4$  RIRs on a grid with a spacing of 20 cm around the microphone array. At each grid point, we measure four RIRs, where the loudspeaker is turned  $90^\circ$  during each measurement, so that the speaker faces each cardinal direction. Figure 5.2 shows the recording setup in panel (a). Panel (b) illustrates the floorplan with the 448 grid points, and four speaker positions similar to the setup from Figure 4.8. To simulate moving speakers, we let them follow the virtual trajectories shown in panel (b), with a speed of  $1m/s$ . To spatialize a speech signal  $s(t)$  with a dynamic RIR, we first spatialize the signal with the RIR on each grid point, i.e.

$$\mathbf{s}_{i,j,\theta}(t) = s(t) \circledast \mathbf{h}_{i,j,\theta}(t), \quad (5.5)$$

where  $i$  and  $j$  are the grid indices, and  $\theta$  is the looking direction. Next, we divide the speech signal into  $B$  blocks of 100 ms length and an overlap of 50 ms. For each block  $b$ , we select the nearest grid point  $i_b, j_b$  and facing direction  $\theta_b$  of the trajectory at time  $t_b$ . To assemble the overall signal  $\mathbf{s}(t)$ , we use the overlap-add method [2] and the window function  $\mathbf{w}(t)$ , as shown in Algorithm 1.



**Algorithm 1** Dynamic spatialization

---

```

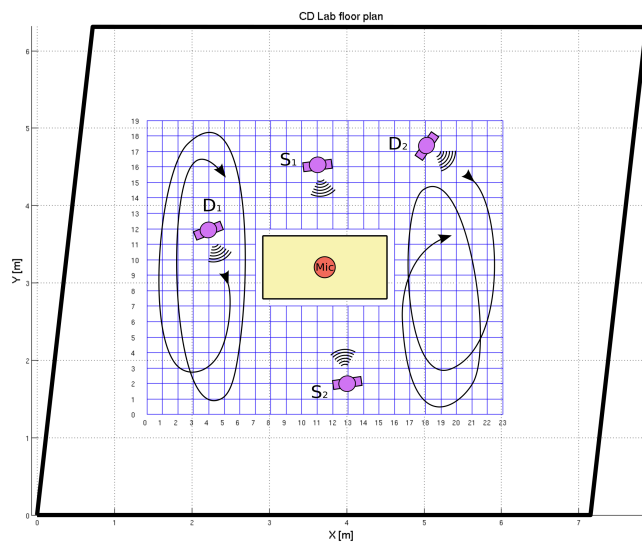
1:  $s(t) \leftarrow 0$ 
2: for  $b$  in  $B$  do
3:    $s(t) \leftarrow s(t) + w(t - t_b) s_{i_b, j_b, \theta_b}(t_b : t_{b+1})$ 
4: end for

```

---



(a)



(b)



(c)

Figure 5.2: (a) RIR recording setup for moving speakers in a 6m × 7m office room. (b) Floorplan with 448 grid points. (c) Microphone array and measurement loudspeaker.

## 5.4 TDNBF Architecture

The *filter-and-sum* beamformer from Eq. 2.8 can also be formulated in time-domain, i.e.

$$y(t) = \sum_{m=1}^M w(t, m) \otimes z(t, m), \quad (5.6)$$

where  $w(t, m)$  is the beamforming filter for the  $m^{\text{th}}$  microphone in time-domain. Therefore, we can use a NN with convolutional layers from Section 3.3.1 to learn these filters from the noisy microphone data. Consequently, the actual NN structure used for the TDNBF architecture resembles a beamforming operation similar to Eq. 2.8, as shown in Figure 5.3. Here, it can be seen that the convolutional layer transforms the multi-channel input signal  $\mathbf{z}(t)$  into a latent representation  $\mathbf{z}'(l)$ , where  $l$  denotes the  $l^{\text{th}}$  time frame, and each time frame has a width of  $h$  neurons, similar to  $k$  frequency bins of the STFT. The left branch of the NN consists of an LSTM layer, and two Feed-Forward (Dense) layers. The LSTM layer models the temporal correlations of the mixture of speech signals, and the first Feed-Forward layer provides a non-linear transformation using a *softplus* activation function, which is defined as

$$f(x) = \log(1 + e^x). \quad (5.7)$$

We chose a softplus activation instead of a tanh activation, to have both an unbounded activation and a nonlinearity. Alternatively, a ReLU activation can also be used. However, the ReLU outputs zeros for negative inputs, which effectively switches off the corresponding neurons. In contrast, the softplus activation provides a small gradient for negative inputs. The second Feed-Forward layer is a simple linear layer, i.e. a Dense layer with a linear activation function  $f(x) = x$ . It outputs the beamforming weights  $\mathbf{w}'(l)$  in latent space. Then, we perform the multiplication

$$\mathbf{y}'(l) = \mathbf{w}'(l) \odot \mathbf{z}'(l), \quad (5.8)$$

which produces the enhanced signal  $\mathbf{y}'(l)$  in latent space. Note that the linear layer produces unconstrained beamforming weights  $\mathbf{w}'(l)$ . Thereby, no constraints are imposed on the actual magnitude of the enhanced signal  $\mathbf{y}'(l)$ . Finally, the Deconvolutional layer converts the enhanced signal back to the time-domain using the overlap-add method.

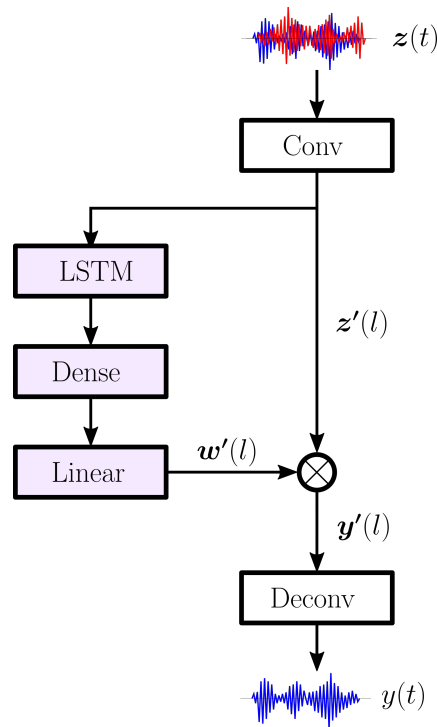


Figure 5.3: TDNBF architecture.

Cross-domain learning allows to formulate the cost function entirely in time-domain, as the gradient is back-propagated through the convolutional layers to the time-domain signal  $z(t)$ . Thereby, the training objective is not limited to technical measures such as the SNR or the MSE. Rather, we may choose a performance measure which is better suited for speech reconstruction and enhancement, i.e. the SDR from Section 2.7.2. In particular, we use the SI-SDR from Eq. 2.73, as the actual amplitude of the enhanced signal does not influence speech quality or intelligibility. For time-discrete signals, the cost function can be written as the negative SI-SDR, i.e.

$$\mathcal{L}_{\text{SI-SDR}} = -10 \log_{10} \left( \frac{\|\alpha r(t)\|_2^2}{\|\alpha r(t) - y(t)\|_2^2} \right), \quad (5.9)$$

where  $\alpha = \frac{y(t)^T r(t)}{r(t)^T r(t)}$ . The reference signal  $r(t)$  denotes the desired speech signal, which should be approximated by the enhanced TDNBF output  $y(t)$ . Note that this cost function can also be used with the CNBF architecture from Chapter 4, by back-propagating the gradient over the iFFT and FFT blocks to the time-domain input signals, as shown in Figure 5.1.

### 5.4.1 Performance

To demonstrate the performance of the TDNBF architecture, we use the recording setup from Section 5.3, where we generate two moving speakers  $D_1$  and  $D_2$ , and two static speakers  $S_1$  and  $S_2$ , as shown in Figure 5.2, panel (b). However, instead of the shoebox RIRs, we use the recorded ones from Section 5.3. As speech database, the WSJ0 [146] is used. We perform four experiments, where we compare the CNBF and TDNBF architectures, using the loss function defined in Eq. 5.9. To measure the performance in terms of speaker separation, we employ the SI-SDR and the WER from Section 2.7. We use the Google Speech-to-Text API as ASR system [130]. Note that Google-ASR reports a WER of 6.1% for the clean WSJ0 test set (si\_et\_05) [67]. Each experiment denotes the desired source  $S(t, k)$  and one or more interfering sources

$\mathbf{N}(t, k)$ , as shown in Table 5.1. It can be seen that both methods show similar performance in either score. Further, the experiments with static speakers show better scores than the ones with moving speakers. This is to be expected, as the NN only has to consider a static region of interest in the first case. It is also worth noting that the WER scores for the real RIRs are slightly worse than for the simulated ones, i.e. for the experiments in Table 4.1. A similar observation can be made when comparing the performance scores of the *real* and *simu* data sets of the CHiME3 and CHiME4 challenges [50], [72]. For further details, see Appendix A.10.

Method	Desired source	Interfering source(s)	SI-SDR	WER
CNBF	S <sub>1</sub>	S <sub>2</sub>	8.63 dB	12.9 %
	S <sub>1</sub>	D <sub>2</sub>	7.21 dB	26.4 %
	D <sub>1</sub>	D <sub>2</sub>	6.48 dB	30.2 %
	D <sub>1</sub>	S <sub>2</sub>	6.22 dB	29.0 %
TDNBF	S <sub>1</sub>	S <sub>2</sub>	9.21 dB	13.5 %
	S <sub>1</sub>	D <sub>2</sub>	7.69 dB	26.8 %
	D <sub>1</sub>	D <sub>2</sub>	6.56 dB	29.9 %
	D <sub>1</sub>	S <sub>2</sub>	6.39 dB	27.3 %

Table 5.1: Performance comparison of the CNBF and TDNBF methods with the SI-SDR objective.

## 5.5 Dereverberation

Algorithms such as speaker separation, speaker identification, or ASR deliver impressive results for speakers in the near-field of the microphone array [148]–[150]. However, in the far-field, these tasks are quite challenging. The spectrogram is smoothed out by reverberations and echoes of the acoustic environment, thereby severely degrading both speech intelligibility and quality [151], [152]. This type of signal degradation also occurs in rooms with a large reverberation time  $RT_{60}$ , i.e. office rooms or conference halls with hard floors and walls [153]. By dereverberating the speech signal, the intelligibility - and consequently the performance of speaker separation, speaker identification, and ASR algorithms - is greatly increased. Various deep learning based methods have been proposed for dereverberation [118], [154], [155], most of which are based on the WPE algorithm [141].

However, with the flexibility of cross-domain learning from Section 5.2, we can incorporate a dereverberation objective directly into the cost function of the TDNBF architecture. As we have seen in Section 2.3, sources in the far-field tend to behave like background noise, i.e. their spatial coherence is nearly isotropic. This makes it difficult to locate and track their exact position, as the phase noise increases with physical distance. This is a challenging condition for both the Eigennet and the CNBF. However, by splitting a reverberant RIR  $\mathbf{h}(t)$  into two components, we can model a signal in the far-field of the microphone array as

$$\begin{aligned} \mathbf{z}(t) &= s(t) \otimes \mathbf{h}(t), \\ &= s(t) \otimes \mathbf{h}_{\text{DIR}}(t) + s(t) \otimes \mathbf{h}_{\text{DIFF}}(t), \end{aligned} \tag{5.10}$$

where  $\mathbf{h}_{\text{DIR}}(t)$  denotes the portion of the RIR contributing to the direct line of sight between the speaker and the microphone array, and  $\mathbf{h}_{\text{DIFF}}(t)$  models all reverberations. Figure 5.4 illustrates the idea behind Eq. 5.10.

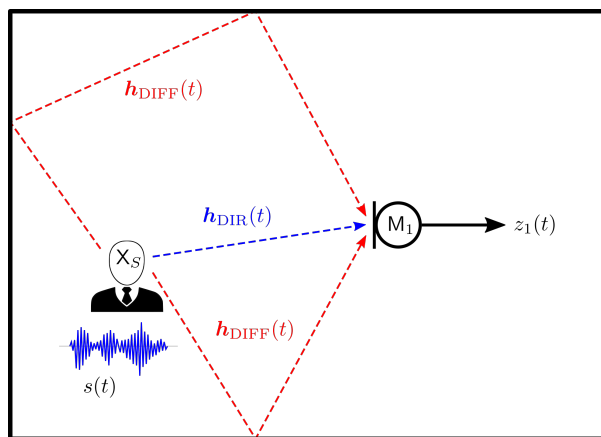


Figure 5.4: Separation of a reverberant RIR into its directional component  $\mathbf{h}_{\text{DIR}}(t)$  and its diffuse component  $\mathbf{h}_{\text{DIFF}}(t)$ .

Ideally, the direct line of sight results in  $\mathbf{h}_{\text{DIR}}(t)$  being reduced to a simple multi-channel time delay, analogous to a steering vector as discussed in Section 2.2, i.e.

$$h_{\text{DIR}}(t, m) = \delta(t - \tau_m), \quad (5.11)$$

where  $\delta(t - \tau_m)$  denotes a Dirac impulse at delay  $\tau_m$ , at the  $m^{\text{th}}$  microphone. This assumption only holds for unobstructed speakers, i.e. there must be no obstacles like corners or furniture between the speaker and the microphone array. If there is a direct line of sight, we regard it as the desired signal  $\mathbf{s}(t)$ , and all reflections are denoted as noise  $\mathbf{n}(t)$ , i.e.

$$\begin{aligned} \mathbf{s}(t) &= s(t) \otimes \mathbf{h}_{\text{DIR}}(t), \\ \mathbf{n}(t) &= s(t) \otimes \mathbf{h}_{\text{DIFF}}(t). \end{aligned} \quad (5.12)$$

Thereby, we can separate the clean speech  $\mathbf{s}(t)$ , and the reverberations  $\mathbf{n}(t)$ . Clearly, neither of these signals are directly observable. But we can define a monaural reference signal  $r(t)$ , using the directional component of the RIR from Eq. 5.11, i.e.

$$r(t) = s(t) \otimes \delta(t - \tau_1) = s(t - \tau_1), \quad (5.13)$$

where we chose the first microphone with the time delay  $\tau_1$  as an arbitrary reference. We insert this reference signal into the SI-SDR objective in Eq. 5.9. Hence, we can use the anechoic, monaural source signal  $s(t)$  as training target, up to an unknown time delay  $\tau_1$ . Given a direct line of sight between the speaker and the first microphone, this delay can be estimated from a known RIR  $h(t, m)$ , i.e.

$$\tau_1 = \underset{t}{\operatorname{argmax}} |h(t, m = 1)|, \quad (5.14)$$

which finds the time delay corresponding to the highest peak in the RIR. This delay belongs to the direct line of sight, as all other paths are longer and consequently receive more attenuation. Inserting the delay  $\tau_1$  into Eq. 5.13 aligns the reference signal  $r(t)$  with the desired speech signal of the first channel of the reverberated input signal  $\mathbf{z}(t)$ . By using this reference signal with the SI-SDR objective, the TDNBF will minimize the relative differences between the enhanced signal at the beamformer output  $y(t)$ , and the anechoic reference  $r(t)$ . Consequently, the input signal

$\mathbf{z}(t)$  will be dereverberated by the TDNBF. Note that this cost function can perform multiple tasks at once, i.e. if  $\mathbf{z}(t)$  contains other interfering sources, they will be removed as well. Consequently, the TDNBF can perform simultaneous speaker separation and dereverberation. The SI-SDR with the reference signal  $r(t)$  is given as

$$\mathcal{L}_{\text{SI-SDR}} = -10 \log_{10} \left( \frac{\|\alpha r(t)\|_2^2}{\|\alpha r(t) - y(t)\|_2^2} \right), \quad (5.15)$$

where  $\alpha = \frac{y(t)^T r(t)}{r(t)^T r(t)}$ . If there is only a little reverb, the reference  $r(t)$  may also be obtained by a close-talking microphone [64], although this approach requires very precise synchronization of the sample rates of the reference microphone and the microphone array.

For large-scale experiments, recording multi-channel data is impractical and expensive. Alternatively, monaural speech databases such as WSJ0 [146] may be spatialized with the RIR recordings from Section 5.3. By permuting the 720 RIRs with the 12776 utterances for training, more than 9 million training examples can be generated. For a single, monaural speech signal  $s(t)$ , spatialization is done using the RIR  $\mathbf{h}(t)$ , i.e.

$$\mathbf{s}(t) = s(t) \otimes \mathbf{h}(t). \quad (5.16)$$

With this setup, the required reference signal is given by  $r(t) = s(t - \tau_1)$ .

### 5.5.1 Performance

To demonstrate the dereverberation performance of the TDNBF architecture, we use the same NN as shown in Figure 5.3. Therefore, the only difference between beamforming and dereverberation is formulated in the training objective. Consequently, the frequency-domain formulation of the dereverberation NN is identical to the CNBF. As we have already compared the CNBF against the TDNBF in Section 5.4, we use the WPE algorithm to perform dereverberation in frequency-domain [141]. Figure 5.5 shows the dereverberation performance of the TDNBF architecture for a single speaker. Panel (a) shows the spectrogram of the first channel of the input  $z(t, m = 1)$ . Panel (b) shows the dereverberated speaker  $y(t)$  using the TDNBF. Panel (c) shows the dereverberated speaker  $y_{\text{WPE}}(t)$  using the WPE algorithm. And panel (d) shows the anechoic reference  $r(t)$ , which was used as training target for the SI-SDR in Eq. 5.9. It can be seen that both the TDNBF and WPE methods deliver similar results. However, the TDNBF operates entirely in time-domain, whereas the WPE algorithm operates in the frequency-domain. The TDNBF achieves a SI-SDR of 10.02 dB, and the WPE algorithm achieves a SI-SDR of 3.41 dB. This low score is due to the fact that the WPE algorithm is unsupervised and cannot align its output to the reference signal  $r(t)$ . The TDNBF architecture can be tuned to real-time operation by using causal Convolutions.



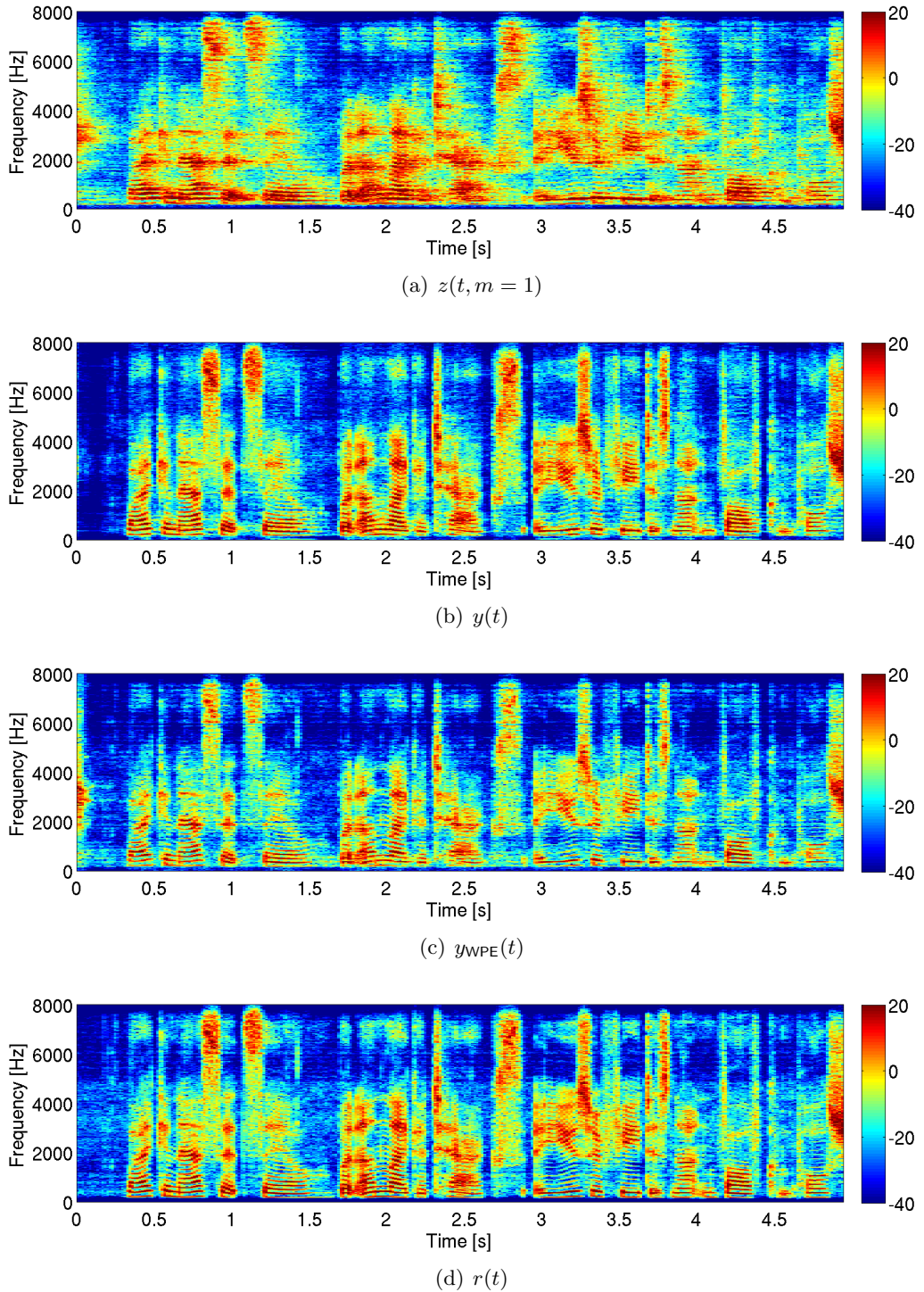


Figure 5.5: Dereverberation performance using the TDNBF and WPE approaches. (a) STFT of the first microphone of the input  $z(t, m = 1)$ . (b) Dereverberated speaker  $y(t)$  using the TDNBF. (c) Dereverberated speaker  $y_{WPE}(t)$  using the WPE algorithm [141]. (d) Anechoic reference  $r(t)$ .

## 5.6 Nonlinear Residual Echo Suppression

Another example for cross-domain learning is given by Nonlinear Residual Echo Suppression (NRES). Here, a speech mask is used to perform SCSE as a neural post-filter for an AEC. In particular, a NN is trained to suppress audible echo artifacts caused by non-linearities in the acoustic setup. Typically, an AEC models the acoustic path between a single loudspeaker and

microphone as a linear FIR filter. The AEC algorithm subtracts an estimate of the echo from the microphone signal, thereby enabling echo-free voice communication. However, the task of echo cancellation is complicated by non-linear distortions in the loudspeaker and the amplifier, and other sources of non-linearities such as structure-borne sounds [7], [45]. These distortions cannot be modeled by linear filters. Consequently, the performance of an AEC is limited in real-world applications, resulting in degraded speech quality and intelligibility. Figure 5.6 shows the block diagram of a NRES system, with the involved signals represented in frequency-domain. The loudspeaker is excited by the far-end signal  $X(l, k)$ , which is reverberated by the room acoustics, i.e. the acoustic echo. Together with the near-end speech signal  $S(l, k)$ , this echo is picked up by the microphone, i.e.

$$D(l, k) = H(k)X(l, k) + S(l, k), \quad (5.17)$$

where  $H(k)$  denotes the Echo Impulse Response (EIR), modeled as linear FIR filter. The linear AEC estimates the model  $Y(l, k)$ , which is subtracted from the microphone signal to obtain the residual signal, i.e.

$$E(l, k) = D(l, k) - Y(l, k). \quad (5.18)$$

Both the microphone signal and the residual echo are fed into the NRES, where a speech mask  $p(l, k) \in [0, 1]$  is calculated to obtain the enhanced output, i.e.

$$Z(l, k) = E(l, k)p(l, k). \quad (5.19)$$

In an ideal, linear setup, the FIR echo model matches the actual echo exactly, i.e.  $Y(l, k) = H(k)X(l, k)$ . However, in a real application, non-linearities in the echo path are neglected. Therefore, postfilter approaches such as *power filters* [156], [157] and *Volterra kernels* [158], [159] have been devised to model these non-linearities with varying degrees of success [160]–[163]. Similar to beamforming, NNs have outperformed traditional NRES postfilters dramatically [164]–[169].

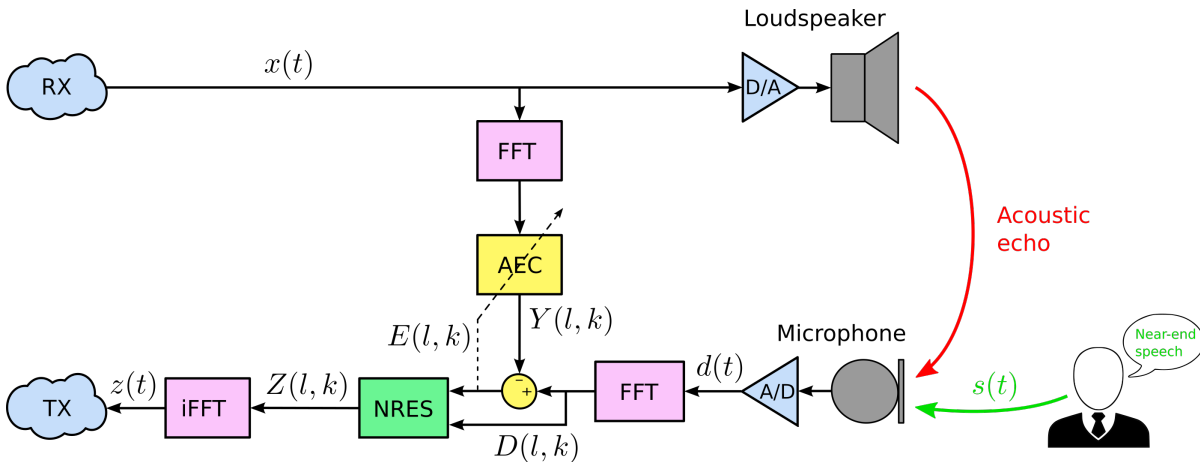


Figure 5.6: System model of an AEC with a NRES post-filter, with signals in the STFT domain.

In [170], we use a recurrent NN to estimate the real-valued speech mask  $p(l, k)$ , where the log-spectrograms of the microphone signal  $D(l, k)$  and the residual signal  $E(l, k)$  are used as feature vectors. The approach uses a small NN with one LSTM layer, and two Feed-Forward layers. Similar to the TDNBF, we formulate the training objective in time-domain. With the



application of AEC and NRES, there are two optimization goals: (i) During *single-talk* - i.e. when the near-end speaker  $S(l, k)$  is silent - we want to suppress as much of the echo as possible. (ii) During *double-talk* - i.e. when both the far-end  $X(l, k)$  and near-end speakers  $S(l, k)$  are overlapping - we want to maximize the intelligibility of the desired near-end speaker. To fulfill both constraints, we formulate a hybrid objective, i.e.

$$\mathcal{L}_{\text{SDR}} = 10 \log_{10} \frac{\sum_T |s(t)|^2}{\sum_T |s(t) - z(t)|^2}, \quad (5.20)$$

which maximizes the SDR during double-talk. We do not need the SI-SDR here, as the amplitude of the enhanced signal is determined by the speech mask, i.e.  $p(l, k) \in [0, 1]$ . Note that the SDR also suppresses most of the echo during single-talk. However, the human auditory system is very sensitive to even the faintest speech artifacts, which might not be suppressed in favor of the main objective of the SDR. Therefore, we explicitly maximize the Echo Return Loss Enhancement (ERLE) during single-talk, which is given by

$$\mathcal{L}_{\text{ERLE}} = 10 \log_{10} \frac{\sum_T |d(t)|^2}{\sum_T |z(t)|^2}. \quad (5.21)$$

The hybrid cost function is given as:

$$\mathcal{L}_{\text{NRES}} = -\mathcal{L}_{\text{ERLE}} - \lambda \mathcal{L}_{\text{SDR}}, \quad (5.22)$$

where the parameter  $\lambda$  allows to adjust the importance of either the ERLE or SDR constraint during training. By using the concept of cross-domain learning from Section 5.2, we can incorporate the gradient of the FFT and iFFT operations to optimize the time-domain cost function given in Eq. 5.22. Similar to the TDNBF, we can also formulate the NRES postfilter in time-domain. The required residual signal  $e(t)$  from the AEC is already available due to the zero-padding operations necessary in the block-partitioned, overlap-save AEC implementation from [171]. The architecture of the time-domain NRES-NN is shown in Figure 5.7. The two Convolutional layers at the top transform the input signals  $d(t)$  and  $e(t)$  into a latent space, where the speech mask  $\mathbf{p}'(l)$  is calculated. The speech mask is applied in latent space, i.e.

$$\mathbf{z}'(l) = \mathbf{e}'(l) \odot \mathbf{p}'(l). \quad (5.23)$$

The linear layer does not constrain the elements of the speech mask between 0 and 1. Instead, it allows for an unconstrained speech mask, which removes the signal components corresponding to the residual echo, which have been identified by the learnable transform in the Convolutional layers. The amplitude of the enhanced output signal  $z(t)$  is determined by the reference signal  $s(t)$  used in the SDR objective given in Eq. 5.20. Finally, the Deconvolution layer transforms the enhanced signal  $\mathbf{z}'(l)$  back to time-domain.

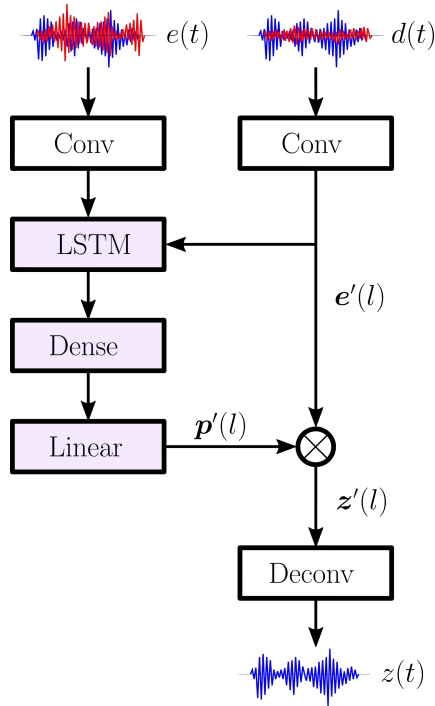


Figure 5.7: NRES architecture in time-domain.

### 5.6.1 Performance

To test the performance of the NRES system, we use the setup from [170], where 1.75 h of AEC recordings are obtained from a speakerphone in various hands-free talking scenarios. We train the NNs of both the frequency- and time-domain NRES on the same data. The echo model  $y(t)$  is obtained by the state-space block-partitioned AEC proposed in [172]. For each of the 30 s long training examples, we zero-initialize the AEC filter coefficients to train the NN during both the adaption phase and the stationary phase of the AEC. The ERLE of the AEC reaches its maximum after approximately 10 s at 19 dB. As a baseline, we compare the NRES postfilter to a state-of-the-art reference AEC implementation (Speex-DSP) [173]. Speex also uses a frequency-domain, block-based echo canceler [174], and a residual echo-suppressor. We configured the same echo-tail length of 512 ms. It can be seen that Speex slightly outperforms the baseline in all scores. However, the NRES postfilter yields a significant improvement in all scores in both its time- and frequency-domain implementations.

method	ERLE	SI-SDR	WER
AEC only	19.206	5.454	44.73%
Speex-DSP	21.726	6.716	25.16%
NRES-FD	60.447	14.543	12.56%
NRES-TD	58.864	15.128	13.01%

Table 5.2: ERLE, SI-SDR and WER scores for the time- and frequency-domain NRES postfilter, the reference system (Speex-DSP) and the AEC without a postfilter as a baseline.

Figure 5.8 demonstrates the performance of both the frequency- and time-domain NRES in terms of spectrograms. Panel (a) shows the STFT of the microphone signal  $d(t)$ . Panel (b) shows the enhanced output  $z_{FD}(t)$  of the frequency-domain NRES, while panel (c) shows the enhanced output of the  $z_{TD}(t)$  time-domain NRES, and panel (d) shows the ground truth of the near-end speaker  $s(t)$ . It can be seen that the ERLE of  $z_{FD}(t)$  is quite high during single-talk. However, small errors in the speech mask cause audible echo during double-talk, degrading

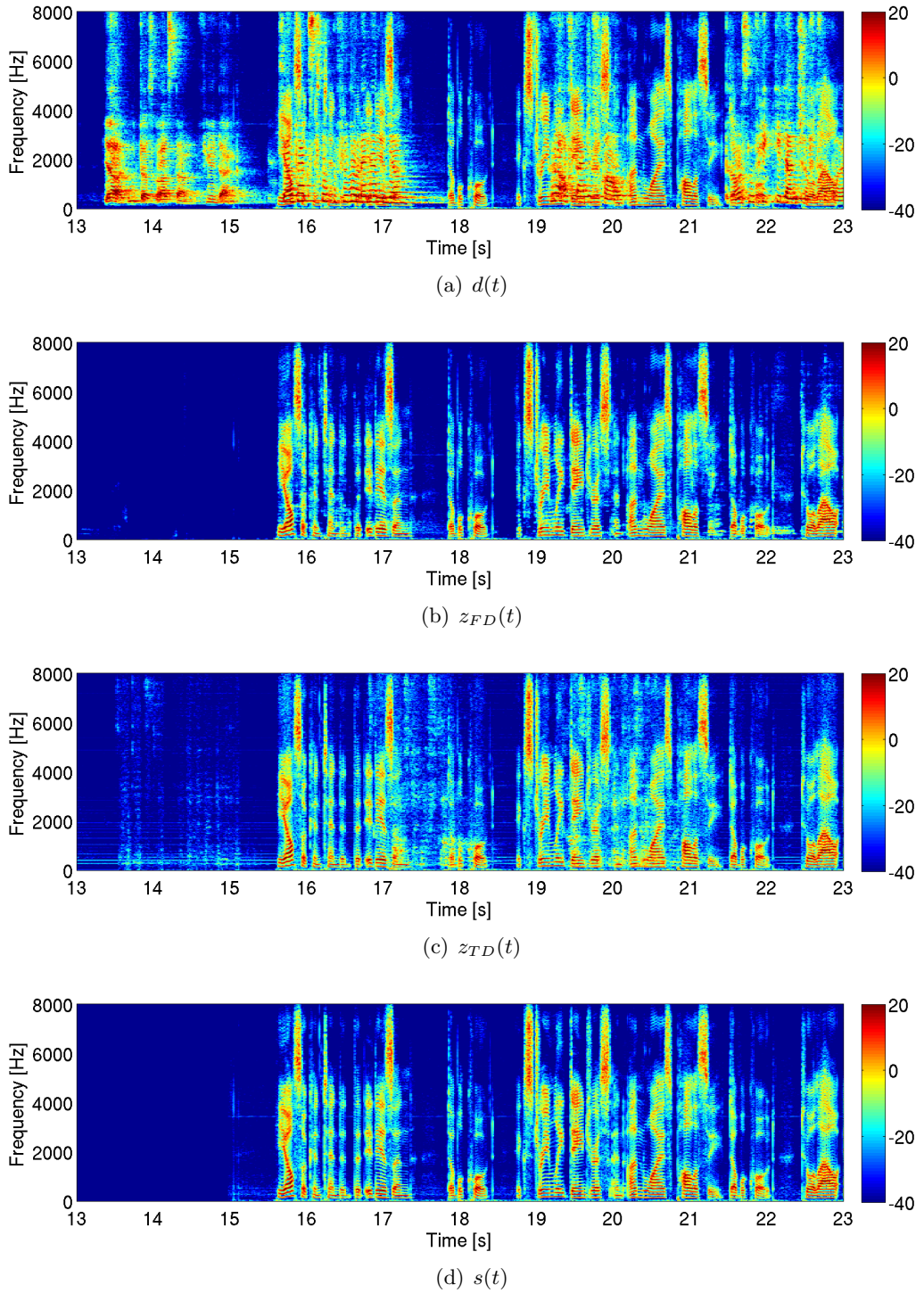


Figure 5.8: (a) Microphone signal  $d(d)$ . (b) Enhanced output of the frequency-domain NRES. (c) Enhanced output of the time-domain NRES. (d) Ground truth of the near-end speaker  $s(t)$ .

the overall signal quality. The ERLE of  $z_{TD}(t)$  is not as large, but the errors are more evenly distributed over the spectrogram, resulting in a higher perceived quality. For further details on the frequency-domain variant of the NRES, we refer the interested reader to Appendix A.9.

## 5.7 Conclusion

In this chapter, we introduced the concept of cross-domain learning, which uses the complex-valued gradient of the FFT and iFFT to allow for a cost function to be defined in time-domain. This has the advantage of using a performance measure that is better suited for speech quality, i.e. the SDR, instead of technical measures such as the SNR. This allows to include additional tasks into the objective of the NN, such as dereverberation. Further, we replaced the FFT by a learnable transformation, which generates a latent space using Convolutional neural networks. We demonstrated the versatility of cross-domain learning with three applications: Neural beamforming, dereverberation, and residual echo suppression. Each of these applications have been formulated in both time- and frequency-domain. While the time-domain systems deliver promising results, they did not significantly outperform their frequency-domain counterparts. This was also reported in [175]. However, the time-domain approaches offer increased flexibility in terms of system latency and real-time requirements. So far, the TDNBF solves four of the six problems stated in the introduction, i.e.

1. **Isolate a single speaker from background noise.** ✓
2. **Isolate a single speaker from a mixture of multiple speakers.** ✓
3. **Track moving speakers.** ✓
4. **Isolate and dereverberate a speaker in the far-field.** ✓
5. **Separate all speakers in a mixture of multiple speakers.**
6. **Assign an identity to an isolated speaker.**



# Blind Source Separation

## 6.1 Motivation

In the previous chapters, we have covered the evolution of neural beamforming, i.e. beamformers supplemented by NN. Starting with mask-based beamforming, we introduced complex-valued neural beamforming and time-domain neural beamforming. These concepts enable high-performance front-end systems for human-machine interfaces or teleconferencing systems. Especially with ASR systems, both speech intelligibility and speech quality play an important role to ensure a low WER in adverse acoustic environments. However, there are still three open aspects we have not covered so far:

1. *Blind source separation*: The speech separation methods discussed so far assume a known region of interest, where the desired source is required to be located at. However, in scenarios such as the infamous *cocktail party problem*, this assumption does not hold as we have no information about a particular speaker's location at any given moment in time. While many existing BSS algorithms are able to separate speakers at unknown locations, they are often limited to a pre-defined number of sources [26]–[28], [36], [68], [176]. In practice, the number of speakers in a mixture is often unknown.
2. *Speaker tracking*: Source separation methods have no means of knowing the identity of a specific speaker. Especially for moving speakers or meeting room scenarios, it may be relevant to focus on and isolate a specific speaker, to perform tasks such as ASR. To track a specific speaker, short blocks of audio have to be analyzed at a time. This implies a permutation problem at block level, requiring *speaker diarization* [57], [177], i.e. assigning a speaker identity to each block of extracted audio.
3. *Distant speaker separation*: In a close-talking application scenario, the desired speaker is located in the near-field of the microphone array. Hence, the RIR from the speaker to the microphone array is short, and there is not much coloration of the extracted speech signal. However, in many real-world scenarios reverberation and echoes cannot be ignored, which degrades speech separation, speaker recognition, and ASR performance [151]–[153].

In this chapter, we focus on these three aspects to construct an end-to-end speech separation system, which is capable to isolate an open number of speakers in the far-field, and which tags each speaker with a unique identification vector. We refer to this system as the Blind Source Separation and Dereverberation (BSSD) architecture. In contrast to the similar works mentioned above, the BSSD system adheres to application-driven constraints, such as a reverberant environment with an unknown number of speakers, low latency, and real-time processing using small blocks of audio at a time.

## 6.2 Speaker Localization

In the previous chapters, we used a pre-defined region of interest to isolate a specific speaker from a multi-speaker mixture. We used a NN to infer this location from a spatially selected training set, to separate desired from unwanted speaker locations. For separating all sources in a mixture, this approach cannot be used. Instead, we segment the 3D space around the microphone array into a finite set of possible speaker locations. In order to do so, we chose to segment the DOA using spherical coordinates. In particular, we use a spherical segmentation based on a Fibonacci spiral [178], which equally distributes a given number of points on the surface of a sphere. This design choice is rooted in the following rationale:

1. A spherical segmentation uses only two angles, i.e. azimuth and elevation, whereas a cubical segmentation requires three dimensions. This significantly reduces the number of segments.
2. Source localization is entirely based on the direction of the impinging sound waves, as shown in Section 2.3. Therefore, it can be treated as a 2D problem. While this introduces the theoretical possibility of one speaker shadowing another, in practice this hardly ever happens due to multi-path propagation of the sound waves [11], [60].
3. Source localization based on the direction of a sound source is robust and mostly independent of the room acoustics [45]. Therefore, it can be performed unsupervised using algorithms such as GCC-PHAT.
4. The designated source directions can be pre-calculated as a set of DOA vectors.

These DOA vectors are identical to the steering vectors used for the MVDR beamformer in Section 2.2.2. We refer to a set of such vectors as the DOA bases. Let us define a set of  $D$  unique DOA bases on a unit sphere around the microphone array, where each impinging sound wave is modeled as plane wave, i.e.

$$V(d, k, m) = e^{-i2\pi f_k \tau_{d,m}}, \quad (6.1)$$

where  $f_k$  is the frequency for index  $k$  and  $\tau_{d,m}$  is the time delay from a point on the sphere to the  $m^{\text{th}}$  microphone, i.e.

$$\tau_{d,m} = \frac{\sqrt{(x_m - x_d)^2 + (y_m - y_d)^2 + (z_m - z_d)^2}}{c}, \quad (6.2)$$

with the speed of sound  $c$ . The cartesian coordinates of the  $m^{\text{th}}$  microphone are denoted by  $\{x_m, y_m, z_m\}$ , and  $\{x_d, y_d, z_d\}$  are the coordinates of the  $d^{\text{th}}$  point on the sphere. By using a Fibonacci spiral [178], these points can be equally distributed on the surface of the sphere, i.e.

$$\begin{aligned} \phi_d &= g \cdot d, \\ \theta_d &= \arcsin \frac{d}{D-1}, \\ x_d &= \cos \theta_d \cos \phi_d, \\ y_d &= \cos \theta_d \sin \phi_d, \\ z_d &= \sin \theta_d, \end{aligned} \quad (6.3)$$

where  $g = \pi(3 - \sqrt{5})$  is known as the golden angle, and  $d = 1 \dots D$  is the DOA index. We use a circular microphone array with  $M$  microphones [79]. Note that the array is flat, and we

cannot distinguish between positive and negative  $z$  coordinates. It is therefore sufficient to only use half a sphere for the DOA bases. To assign a DOA vector to a given source, we utilize the GCC-PHAT [11], i.e.

$$d = \operatorname{argmax}_d \sum_{k=1}^K \frac{|\mathbf{H}^H(k) \cdot \mathbf{V}(d, k)|^2}{\|\mathbf{H}(k)\|_2^2}, \quad (6.4)$$

where  $\mathbf{H}(k)$  represents the RIR of that source, and  $d$  denotes the index of the DOA vector  $\mathbf{V}(d, k)$  which best matches the direction of the source position. Using Eq. 6.4, we can assign a DOA index  $d \in \{1 \dots D\}$  to each of the 720 recorded RIRs from Section 5.3. Figure 6.1 shows a histogram using  $D = 100$  DOA bases.

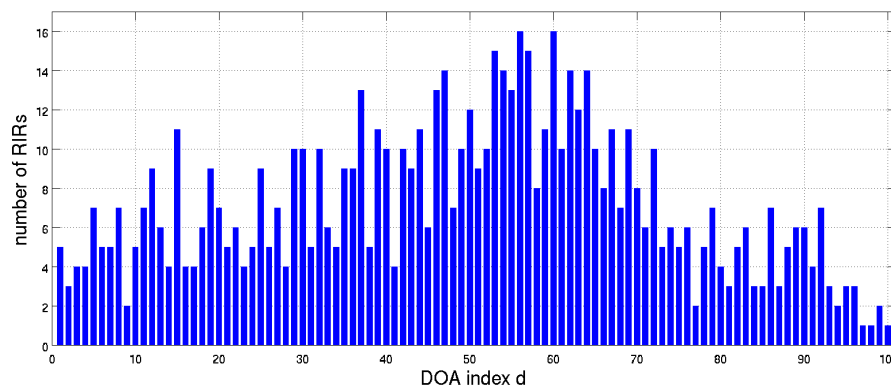


Figure 6.1: Histogram showing the number of RIRs from Section 5.3 assigned to a DOA index  $d$  using 6.4.

We again use the GCC-PHAT to estimate a speech presence probability for each time-frequency bin of a given speech signal  $\mathbf{Z}(l, k)$ , for each DOA position. By comparing each DOA vector  $\mathbf{V}(d)$  against the direction of the input vector  $\mathbf{Z}(l, k)$ , we obtain the speech presence probability  $\gamma \in [0, 1]$ , i.e.

$$\gamma(l, k, d) = \frac{|\mathbf{Z}^H(l, k) \cdot \mathbf{V}(d, k)|^2}{\|\mathbf{Z}(l, k)\|_2^2}. \quad (6.5)$$

To increase the spatial selectivity and sensitivity of the GCC-PHAT, we use ZCA whitening from Section 2.6. In particular, we rewrite Eq. 6.5 as

$$\gamma_U(l, k, d) = \frac{|\mathbf{Z}^H(l, k) \mathbf{U}^H(k) \cdot \mathbf{U}(k) \mathbf{V}(d, k)|^2}{\|\mathbf{U}(k) \mathbf{Z}(l, k)\|_2^2 \cdot \|\mathbf{U}(k) \mathbf{V}(d, k)\|_2^2}, \quad (6.6)$$

where  $\mathbf{U}(k) \mathbf{Z}(l, k)$  can be recognized as the whitened input mixture, and  $\mathbf{U}(k) \mathbf{V}(d, k)$  as whitened DOA vector. Note that  $\gamma_U(l, k, d)$  must not be confused with a speech mask, as it only evaluates the probability of the signal  $\mathbf{Z}(l, k)$  originating from direction  $\mathbf{V}(d, k)$ . Next, we weigh the GCC-PHAT using the signal energy, i.e.

$$\gamma_W(l, k, d) = \gamma_U(l, k, d) \sum_{m=1}^M |Z(l, k, m)|^2. \quad (6.7)$$

This maps the energy of each time-frequency bin of the input signal over each DOA location. To iteratively estimate the position of all speech sources in a given mixture of multiple speakers

$\mathbf{Z}(l, k)$ , we use the pseudo-code in Algorithm 2. First, we copy  $\gamma_W(l, k, d)$  from Eq. 6.7 into  $\gamma'_W(l, k, d)$ . Then, we initialize an empty list of DOA locations  $\mathcal{D}$ . During each iteration, we determine the index  $\hat{d}$  of the global maximum of  $\gamma'_W(l, k, d)$  over the set of  $D$  possible DOAs. Then, we subtract the weighted GCC-PHAT at that maximum  $\gamma_W(l, k, \hat{d})$  from all DOA locations in  $\gamma'_W(l, k, :)$ , which is used in the next iteration of the algorithm. This ensures that each speech source is only extracted once, as we subtract the signal energy towards the direction  $\hat{d}$  from all locations  $D$ . In essence, Algorithm 2 reorders the DOA indices into the list  $\mathcal{D}$ . However, it does not simply sort the indices by energy, as multiple DOA locations may share the energy from the same speaker, due to the limited spatial resolution of the microphone array. Also, the algorithm does not stop until all  $D$  positions are sorted. In the subsequent sections, we use the list of DOA indices  $\mathcal{D}$  to extract the correct number of speakers in a given mixture  $\mathbf{z}(t)$ .

---

**Algorithm 2** Source localization
 

---

```

1:  $\gamma'_W(l, k, d) \leftarrow \gamma_W(l, k, d)$ 
2:  $\mathcal{D} \leftarrow \square$ 
3: while  $\text{length}(\mathcal{D}) < D$  do
4:    $\hat{d} \leftarrow \underset{d}{\text{argmax}} \left( \sum_{l=1}^L \sum_{k=1}^K \gamma'_W(l, k, d) \right)$ 
5:    $\mathcal{D} \leftarrow \text{append}(\mathcal{D}, \hat{d})$ 
6:    $\gamma'_W(l, k, :) \leftarrow \max \left( \gamma'_W(l, k, :) - \gamma_W(l, k, \hat{d}), 0 \right)$ 
7: end while

```

---

### 6.2.1 Performance

To test the effectiveness of Algorithm 2, we first spatialize some arbitrary WSJ0 recordings using

$$\mathbf{Z}(l, k) = S_1(l, k)\mathbf{H}_1(k) + S_2(l, k)\mathbf{H}_2(k) + S_3(l, k)\mathbf{H}_3(k), \quad (6.8)$$

to generate a mixture  $\mathbf{Z}(k)$  containing  $C = 3$  speakers. For the RIRs  $\mathbf{H}_{\{1,2,3\}}(k)$ , we select three of the 720 RIR recordings from Section 5.3. For the STFT representation of the signals in Algorithm 2, we use a window length of 1024 samples and a stride of 256 samples, i.e. an overlap of 75%. During each iteration of Algorithm 2, we average the weighted speech presence probability  $\gamma'_W(l, k, d)$  over time and frequency, i.e.

$$\bar{\gamma}_W(d) = \frac{\sum_{l=1}^L \sum_{k=1}^K \gamma'_W(l, k, d)}{\sum_{l=1}^L \sum_{k=1}^K \sum_{m=1}^M |Z(l, k, m)|^2}. \quad (6.9)$$

By plotting  $\bar{\gamma}_W(d)$  on the surface of a sphere, we obtain a spatial visualization of the speech presence probability  $\gamma_U(l, k, d)$  from Eq. 6.6 for each possible DOA position  $d$ . Figure 6.2 illustrates  $\bar{\gamma}_W(d)$  in panel (a), (b) and (c) during the three iterations of Algorithm 2. The black dots indicate the equally distributed DOA positions  $d \in \{1 \dots D\}$ , obtained with the Fibonacci spiral from Eq. 6.3. The angular resolution averages to  $13.82^\circ$  between two neighboring points. The positions of the speakers are labeled with  $\mathbf{X}_{\{1,2,3\}}$ , where position  $\mathbf{X}_2$  and  $\mathbf{X}_3$  have been selected so that the respective speakers are standing right next to each other. The color gradient corresponds to  $\bar{\gamma}_W(d)$  from Eq. 6.9, where the  $D$  points have been interpolated to fill the hemisphere. The first iteration of Algorithm 2 is visualized in panel (a). It can be seen that  $\bar{\gamma}_W(d)$  indicates signal activity for all of the three locations  $\mathbf{X}_{\{1,2,3\}}$ . The maximum operator in Algorithm 2 selects the DOA index  $\hat{d}_1$ , which contributes the most energy to  $\gamma'_W(l, k, d)$ , i.e. position  $\mathbf{X}_1$ . Then, it subtracts  $\gamma_W(l, k, \hat{d}_1)$  from  $\gamma'_W(l, k, d)$ , and executes the second iteration.



From panels (b) and (c), it can be seen that Algorithm 2 successfully extracts the DOA indices for all three speaker positions, even though speaker 2 and 3 are standing right next to each other. In panel (c), a *sidelobe* [60] of the microphone array can be observed as the white patch at the far end of the hemisphere. This happens due to multi-path propagations of the sound waves in a realistic environment, i.e. with recorded RIRs.

Panel (d) shows the spectrogram of the first microphone of the input mixture, i.e.  $Z(l, k, m = 1)$ . Panels (e), (f) and (g) illustrate the GCC-PHAT  $\gamma_U(l, k, \hat{d}_{\{1,2,3\}})$  during each iteration of Algorithm 2. The GCC-PHAT is obtained by inserting the estimated DOA index  $\hat{d}$  into Eq. 6.6. Despite looking similar,  $\gamma_U(l, k, d)$  is fundamentally different to a speech mask from Chapter 3, as it measures the amount of energy originating from the spatial direction indicated by the DOA index  $d$ , while a speech mask captures all time-frequency bins belonging to a specific speaker. Panel (e) shows  $\gamma_U(l, k, \hat{d}_1)$ , which is obtained during the first iteration of Algorithm 2. It can be seen that there is a large amount of noise, as some time-frequency bins are shared amongst multiple speakers. Again, this is caused by multi-path propagations in the RIRs, which are different for each frequency. Therefore, reflections from speakers 2 and 3 are impinging from the direction  $\hat{d}_1$  at certain frequencies, which causes the noise. Further, from panels (f) and (g) it can be seen that  $\gamma_U(l, k, \hat{d}_2)$  and  $\gamma_U(l, k, \hat{d}_3)$  look very similar. This is due their almost identical position, and to the limited spatial resolution of the 6-channel microphone array [60]. However, for high frequencies, the differences are more pronounced. This happens because the wavelength of high frequencies is small compared to the array aperture, resulting in distinct phase differences.

The spatial resolution of the microphone array is roughly indicated by the size of the red region in panel (c), where only a single source is left in  $\tilde{\gamma}_W(d)$ . Note that the resolution is not equally distributed over the sphere, as it depends on the array geometry [60]. For a circular array, the resolution is highest for an azimuth of  $\theta = \pm 90^\circ$ , i.e. the poles of the sphere. Even though the spatial resolution of the microphone array is relatively low, and two speakers are standing right next to each other, Algorithm 2 is still able to separate all three speakers. It assigns three distinct DOA indices  $\hat{d}_{\{1,2,3\}}$  to the speakers, which is all we are interested in at this stage of the BSSD system.

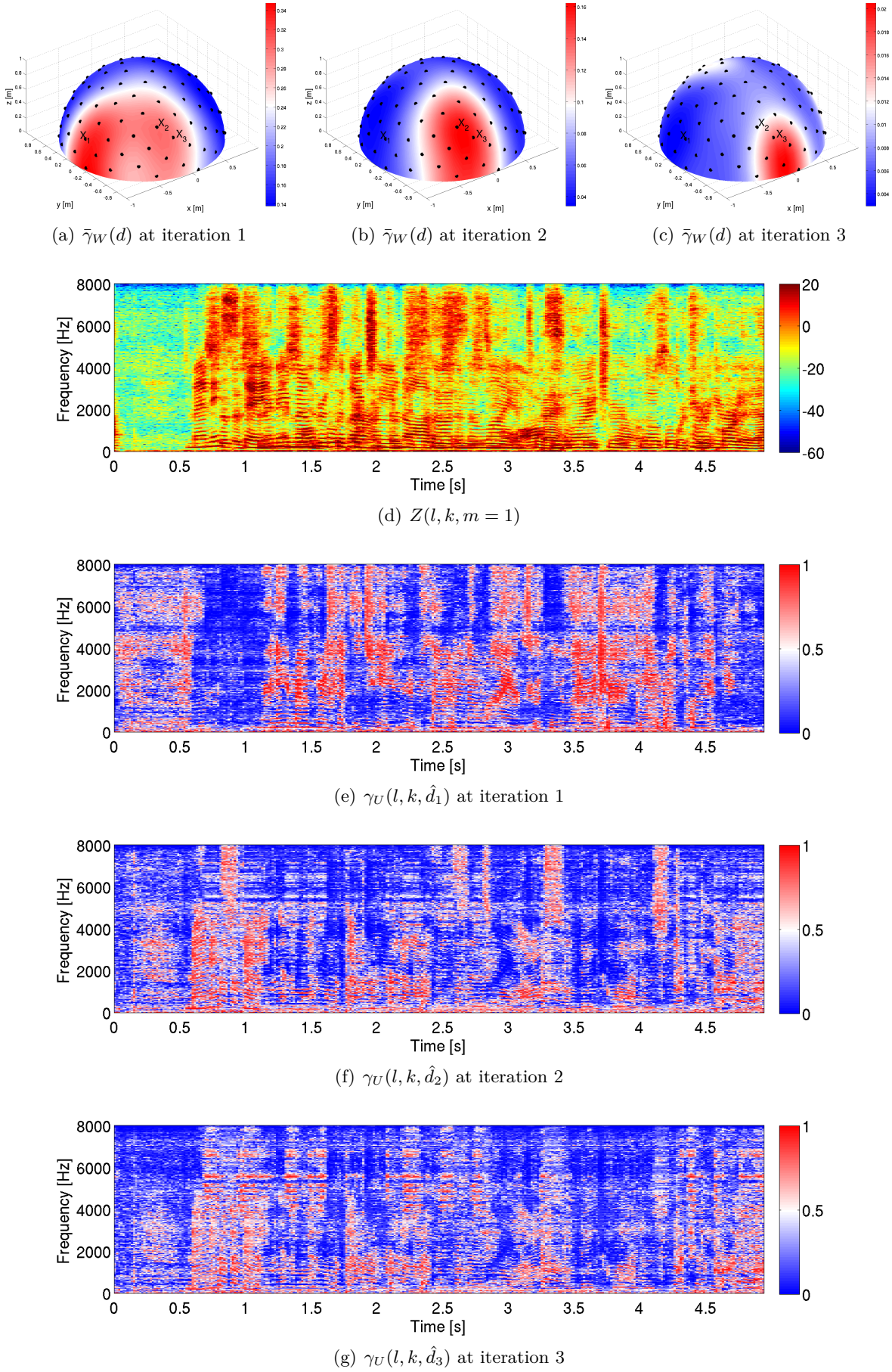


Figure 6.2: Speaker localization using Algorithm 2. (a)  $\bar{\gamma}_W(d)$  at iteration 1. (b)  $\bar{\gamma}_W(d)$  at iteration 2. (c)  $\bar{\gamma}_W(d)$  at iteration 3. (d) Spectrogram of the first channel of the input mixture  $Z(l, k, m = 1)$ . (e) Speech presence probability  $\gamma_U(l, k, \hat{d}_1)$ . (f)  $\gamma_U(l, k, \hat{d}_2)$ . (g)  $\gamma_U(l, k, \hat{d}_3)$ .

### 6.3 Selective Attention

In Section 6.2, we used a set of predefined DOA vectors to localize up to  $D$  independent sound sources. For each identified source, i.e. speaker, a DOA index is obtained from Algorithm 2. With this index, the direction of the source is given by the DOA vector  $\mathbf{V}(\hat{d}, k)$ . Now, we will use this vector to extract the speaker positioned at this location.

In Section 4.4 we introduced the CNBF which extracts a speaker at a certain region of interest, i.e. a pre-defined location. We can re-use this concept by modifying the input signal  $\mathbf{Z}(l, k)$ , so that the desired speaker at location  $\mathbf{V}(\hat{d}, k)$  is always in a virtual region of interest. In particular, we employ the DOA vector to modify the phase of the input signal, so that the IPDs are zero for the desired speaker, and non-zero for other signal components. This moves the desired speaker into the center of the microphone array, i.e.

$$\tilde{\mathbf{Z}}(l, k) = (\mathbf{U}(k)\mathbf{V}(\hat{d}, k))^* \odot (\mathbf{U}(k)\mathbf{Z}(l, k)), \quad (6.10)$$

where  $\mathbf{U}(k)$  denotes the whitening matrix from Section 2.6. When the multi-channel phase of  $\mathbf{Z}(l, k)$  is identical to the phase of the DOA vector  $\mathbf{V}(\hat{d}, k)$ , the IPDs of  $\tilde{\mathbf{Z}}(l, k)$  will be zero. Consequently, signal components located at different positions will have non-zero IPDs. We can train a NN such as the CNBF or TDNBF to extract a speaker whose IPDs are zero. Clearly, this concept neglects the case when two speakers are positioned in such a way that their relative direction towards the microphone array is identical, i.e. one speaker is shadowing the other. However, due to multi-path propagations in the RIRs, there will always be some minor differences in the IPDs for signals impinging from different locations, even when they are very close to one another, as shown in Figure 6.2. Due to these differences, Algorithm 2 is able to detect multiple source locations.

However, instead of modifying the phase of the input with the DOA vector  $\mathbf{V}(\hat{d}, k)$ , it is also possible to modify the input directly with a set of trainable NN weights, i.e.

$$\tilde{\mathbf{Z}}(l, k) = \mathbf{A}(\hat{d}, k)\mathbf{Z}(l, k), \quad (6.11)$$

where  $\mathbf{A}$  denotes a complex-valued tensor of shape  $D \times K \times M \times M$ . It allows to scale, shift and mix the  $M$  channels of the complex-valued inputs  $\mathbf{Z}(l, k)$  freely. Therefore, the tensor can learn the same transformation as in Eq. 6.10, and a dedicated whitening step is not required. However, to train all the weights in the tensor, examples for all  $D$  possible DOAs must be presented to the NN. To distinguish both approaches, we refer to Eq. 6.10 as *Analytic Adaption*, and to Eq. 6.11 as *Statistic Adaption*.

Both the analytic and statistic adaption can also be performed in time-domain, to be used with the TDNBF architecture. It can be seen that Eq. 6.11 represents a matrix-vector product in frequency-domain. Therefore, an identical operation can be formulated in time-domain using FIR filters, i.e.

$$\tilde{z}(t, m) = \sum_{i=1}^M z(t, i) \circledast a(\hat{d}, t_A, m, i), \quad (6.12)$$

where  $\mathbf{a}$  is a tensor of shape  $(D, T_A, M, M)$ , and  $T_A$  is the filter length of the learnable convolution kernels. This filter operation can be implemented using a single convolution layer. Similar to the frequency-domain, Eq. 6.12 synchronizes the ITDs of the input signal to be zero for signals originating from the direction of  $\mathbf{V}(\hat{d})$ , i.e. the desired speaker. For the analytic adaptation in time-domain, we refer the interested reader to Appendix A.10.

## 6.4 Speaker Identification

For the BSSD system to be useful in a real-world application such as meeting room or cocktail-party scenarios, it is essential to know the identity of the extracted speaker. This allows the algorithm to be executed in real-time applications, where small blocks of audio are processed at a time. With the identity of the speaker, it becomes possible to decide whether two blocks of extracted speech belong to the same speaker, or to different speakers. This process is known as *speaker diarization* [57], [177]. Ideally, a speaker identification algorithm is agnostic to the spoken text, and only relies on the speaker characteristics in the extracted signal. For this purpose, embedding vectors may be used to map utterances into a feature space where distances correspond to speaker similarity [179]. Typically, i-Vectors [148] or x-Vectors [149] are used for this task. Algorithms such as *Deep Speaker* [150] rely on contrastive loss or triplet loss to learn embeddings on a very large set of speakers [166], [180]–[182].

In the near-field of the microphone array, speaker identification algorithms relying on embedding vectors deliver impressive results [148]–[150]. However, speaker recognition in the far-field remains a challenging task, as the spectrogram is smoothed out by reverberations of the acoustic environment [151]–[153]. By including the dereverberation objective for the TDNBF architecture from Section 5.5, we obtain the isolated and dereverberated speaker  $y_{\hat{d}}(t)$  at the DOA position  $\mathbf{V}(\hat{d}, k)$ . We denote the identity of this speaker by the  $E$  dimensional embedding vector  $\mathbf{e}_{\hat{d}}$ . This embedding vector maps the utterance into a feature space where distances correspond to speaker similarity [179]. The embedding vector is obtained using a NN, which is trained on the log-power spectral density  $\log(|Y_{\hat{d}}(l, k)|^2)$  of the extracted speech signal. The structure of this NN is given in Figure 6.3.

As we want to identify an open set of speakers, we need to be able to compare two random utterances and determine whether they belong to the same speaker or not. Further, we want to use small batch sizes to train the whole BSSD network in an end-to-end fashion. Therefore we employ the triplet loss [180], which has been successfully used for speaker identification and diarization tasks [150], [166], [181], [182]. The triplet loss performs well on small batch sizes [183]. The aim of the triplet loss is to ensure that two utterances from the same speaker have their embeddings close together in the embedding space, and two examples from different speakers have their embeddings farther away by some margin  $\beta$ . This criterion causes the embeddings of the same speaker to form clusters, and these clusters are separated by the margin, i.e.

$$\mathcal{L}_{\text{TL}} = \sum_{B'^3} \left[ \|e_a - e_p\|_2 - \|e_a - e_n\|_2 + \beta \right]_+, \quad (6.13)$$

where the embedding  $e_a$  denotes an *anchor*,  $e_p$  is an embedding from the same speaker as the anchor (positive example), and  $e_n$  is an embedding from a different speaker (negative example). In a batch of  $B'$  utterances, there can be as much as  $B'^3$  triplets. It is therefore crucial to only select a subset of valid triplets, where the positive example is from the same speaker as the anchor, and the negative example belongs to a different speaker. Further, we only need to consider triplets where the loss  $\mathcal{L}_{\text{TL}}$  is actually greater than zero. To select relevant triplets, we utilize *Hard Triplet Mining* [184], where we select the hardest positive and negative example per anchor. In particular, we randomly select  $P$  utterances from  $B$  speakers, where we determine the largest distance  $\|e_a - e_p\|_2$  between an anchor and a positive example within the  $P$  utterances per speaker, and the smallest distance  $\|e_a - e_n\|_2$  between an anchor and a negative example from the  $P(B-1)$  remaining utterances. More formally, this procedure can be written as:

$$\mathcal{L}_{\text{TL-HTM}} = \frac{1}{B \cdot P} \sum_{i=1}^B \sum_{a=1}^P \left[ \beta + \max_{p=1 \dots P} (\|e_a^i - e_p^i\|_2) - \min_{\substack{j=1 \dots B \\ n=1 \dots P \\ i \neq j}} (\|e_a^i - e_n^j\|_2) \right]_+. \quad (6.14)$$

When the batch size  $B \cdot P$  is small, the embeddings may collapse into a single point during training [185]. To avoid this, we propose to minimize the cross-entropy between embeddings of different speakers as follows:

$$\mathcal{L}_{\text{TL-CE}} = \frac{-1}{(B^2 - B)P^2} \sum_{a=1}^B \sum_{\substack{n=1 \\ n \neq a}}^B \sum_{i=1}^P \sum_{j=1}^P \log\left(|(\tilde{\mathbf{e}}_a^i)^T \tilde{\mathbf{e}}_n^j|^2\right), \quad (6.15)$$

where  $\tilde{\mathbf{e}} = \frac{\mathbf{e}}{\|\mathbf{e}\|_2}$  is the magnitude-normalized embedding vector  $\mathbf{e}$ . This regularization ensures that the embeddings  $\mathbf{e}_a$  and  $\mathbf{e}_n$  will be different. The cost function for the speaker identification NN is then defined as:

$$\mathcal{L}_{\text{TL}} = \lambda_1 \mathcal{L}_{\text{TL-HTM}} + \lambda_2 \mathcal{L}_{\text{TL-CE}}, \quad (6.16)$$

where  $\lambda_1$  and  $\lambda_2$  are weights for the individual terms of the triplet loss.

### 6.4.1 Distance Measure

In order to determine whether two embeddings  $\mathbf{e}_1$  and  $\mathbf{e}_2$  belong to the same speaker, we use the euclidian distance from Eq. 6.13, i.e.  $\|\mathbf{e}_1 - \mathbf{e}_2\|_2$ . If the distance falls below a certain threshold  $\delta$ , we consider the two embeddings to belong to the same speaker. If it exceeds the threshold, the respective speakers are considered to be different. Hence, two types of errors exist: (i) A false positive is triggered when two embeddings from two different speakers are incorrectly classified as belonging to the same speaker, which is measured using the False Acceptance Rate (FAR), i.e.

$$\text{FAR}(\delta) = \frac{1}{(B^2 - B)P^2} \sum_{a=1}^B \sum_{\substack{n=1 \\ n \neq a}}^B \sum_{i=1}^P \sum_{j=1}^P \mathbb{1}\left(\|\mathbf{e}_a^i - \mathbf{e}_n^j\|_2 < \delta\right). \quad (6.17)$$

(ii) A false negative is triggered when two embeddings from the same speaker are classified as belonging to different speakers, which is measured using the False Reject Rate (FRR), i.e.

$$\text{FRR}(\delta) = \frac{1}{B(P^2 - P)} \sum_{\substack{a=1 \\ p=a}}^B \sum_{i=1}^P \sum_{\substack{j=1 \\ j \neq i}}^P \mathbb{1}\left(\|\mathbf{e}_a^i - \mathbf{e}_p^j\|_2 > \delta\right). \quad (6.18)$$

It can be seen that the FAR increases with the decision threshold  $\delta$ , and the FRR decreases. The value at which the FAR and FRR are equal, is known as the Equal Error Rate (EER). It is determined by:

$$\hat{\delta} = \underset{\delta}{\operatorname{argmin}}\left(|\text{FAR}(\delta) - \text{FRR}(\delta)|\right) \quad (6.19)$$

$$\text{EER} = \text{FAR}(\hat{\delta}) = \text{FRR}(\hat{\delta}),$$

where  $\hat{\delta}$  is considered as the optimal threshold belonging to the EER. We use this threshold to determine whether two embeddings  $\mathbf{e}_1$  and  $\mathbf{e}_2$  belong to the same speaker, by using the euclidian

distance as shown above, i.e.

$$p_{\hat{\delta}}(\mathbf{e}_1, \mathbf{e}_2) = \mathbb{1}(\|\mathbf{e}_1 - \mathbf{e}_2\|_2 < \hat{\delta}), \quad (6.20)$$

where a value of 1 indicates the same speaker, and a value of 0 indicates two different speakers.

## 6.5 BSSD Architecture

With the TDNBF beamformer defined in Section 5.4, the statistic adaption defined in Eq. 6.12, and the SI-SDR objective defined in Eq. 5.9, we can extract and dereverberate the speaker at the DOA  $\mathbf{V}(\hat{d}, k)$ . Further, by assigning a unique embedding vector  $\mathbf{e}_{\hat{d}}$  to each extracted speaker using the triplet loss from Eq. 6.16, we can identify each of the extracted speakers. In this last step, we construct a monolithic structure with all of these building blocks to train the BSSD network in an end-to-end fashion.

Figure 6.3 illustrates the structure of the BSSD architecture. The left branch performs beamforming and dereverberation, using the TDNBF beamformer from Section 5.4. The adaption layer at the top implements Eq. 6.12, followed by a convolutional layer, which transforms time-domain input  $\tilde{\mathbf{z}}(\mathbf{t})$  into a latent space  $\mathbf{z}'(l)$  with  $L$  frames and  $H$  filters. The filter kernels have a length of  $K_{BF}$  frames, and a stride of  $S_{BF}$ . The activation function of this Convolution layer is linear. The beamforming weights  $\mathbf{w}'(l)$  are predicted from the spatial information embedded in  $\mathbf{z}'(l, h)$ . This information is extracted by a bidirectional LSTM layer, a Feed-Forward (Dense) layer with a tanh activation, and a linear layer. The linear layer allows the NN to freely chose the amplitude and phase of the beamforming weights. The layer normalization in front of the bidirectional LSTM layer normalizes the mean and variance of the data to 0 and 1, respectively. This helps the NN to focus on the ITDs, instead of the magnitude of the speech signal. The enhanced output  $\mathbf{y}'_{\hat{d}}(l)$  is obtained by

$$\mathbf{y}'_{\hat{d}}(l) = \mathbf{w}'(l) \odot \mathbf{z}'(l), \quad (6.21)$$

where all variables are of shape  $L \times H$ . Finally, a Deconvolution layer with the same parameters as the Convolution layer produces the enhanced time-domain signal  $y_{\hat{d}}(t)$ . The right branch illustrates the structure of the speaker identification NN, which extracts the embedding vector  $\mathbf{e}_{\hat{d}}$ . It consists of a series of 6 convolutional layers with a filter length of  $K_{ID}$  frames, a stride of  $S_{ID}$ , and increasing dilation factors of (1,2,4,8,16,32) frames. These layers output a latent space of  $L \times E$  dimensional embeddings. Each convolutional layer uses a softplus activation function and layer normalization. A skip connection is added between every two convolutional layers. Then, the  $L$  time frames are averaged to obtain a single,  $E$  dimensional embedding for the whole utterance, using an averaging pooling layer. The linear layer at the end of the stack outputs the unconstrained embedding vector  $\mathbf{e}_{\hat{d}}$ . The overall cost function of the BSSD network is given by combining the triplet loss given in Eq. 6.16, and the dereverberation loss given in Eq. 5.15, i.e.

$$\mathcal{L}_{\text{BSSD}} = \mathcal{L}_{\text{SI-SDR}} + \lambda_1 \mathcal{L}_{\text{TL-HTM}} + \lambda_2 \mathcal{L}_{\text{TL-CE}}, \quad (6.22)$$

where  $\lambda_1$  and  $\lambda_2$  are weights for the components of the triplet loss.

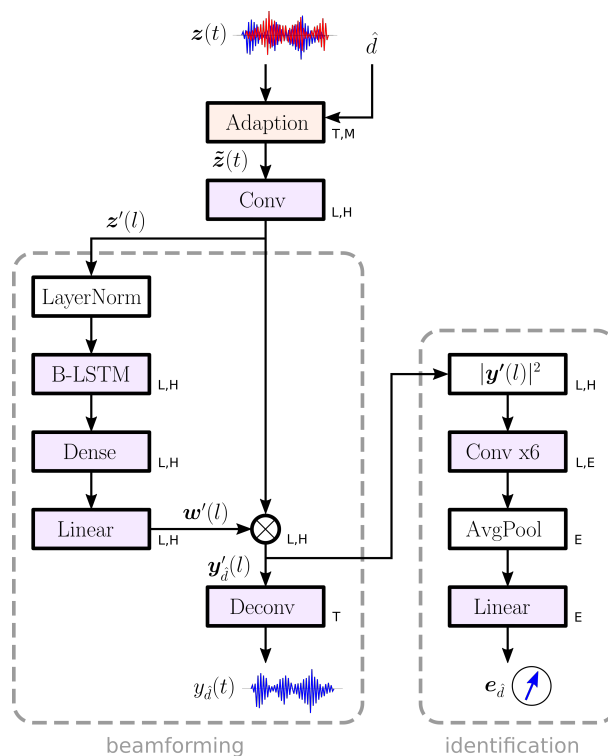


Figure 6.3: Architecture of the BSSD network. The left branch performs beamforming and dereverberation, and the right branch performs speaker identification. The symbols next to each layer denote the dimensionality of the respective output tensor.

### 6.5.1 Speaker Extraction

With the complete BSSD architecture shown in Figure 6.3, we can formulate an algorithm to extract the correct number of speakers from a given mixture  $z(t)$ . First, we initialize an empty list of extracted speech signals  $\mathcal{Y}$ , and an empty list of speaker embeddings  $\mathcal{E}$ . Next, we iterate over the list of DOA indices  $\mathcal{D}$ , which is obtained from Algorithm 2. Next, the input mixture  $z(t)$  and the current DOA index  $\hat{d}$  are used as input to the BSSD network, which predicts an isolated and dereverberated speech signal  $y_{\hat{d}}(t)$ , and an embedding vector  $e_{\hat{d}}$  to identify the speaker corresponding to that speech signal. Then, we use the distance measure  $p_{\hat{\delta}}(\mathcal{E}, e_{\hat{d}})$  from Eq. 6.20 to determine whether the newly found speaker embedding  $e_{\hat{d}}$  is already a member of the list  $\mathcal{E}$ . If  $p_{\hat{\delta}}$  returns 0, a new speaker has been found. The algorithm stops as soon as  $p_{\hat{\delta}}$  returns 1, i.e. a duplicate of an existing speaker has been found. This ensures that each speaker is only extracted once. A single speech signal may be "seen" from different directions, due to reflections and sidelobes in the RIRs [11]. Algorithm 3 shows the pseudo-code to illustrate the speaker extraction process.

**Algorithm 3** Speaker extraction

---

```

1:  $\mathcal{E} \leftarrow []$ 
2:  $\mathcal{Y} \leftarrow []$ 
3: for  $\hat{d}$  in  $\mathcal{D}$  do
4:    $y_{\hat{d}}(t), e_{\hat{d}} \leftarrow \text{BSSD}(z(t), \hat{d})$ 
5:   if  $p_{\hat{s}}(\mathcal{E}, e_{\hat{d}}) == 0$  then
6:      $\mathcal{Y} \leftarrow \text{append}(\mathcal{Y}, y_{\hat{d}}(t))$ 
7:      $\mathcal{E} \leftarrow \text{append}(\mathcal{E}, e_{\hat{d}})$ 
8:   else
9:     break
10:  end if
11: end for

```

---

## 6.5.2 Experiments

### Experimental Setup

We use the WSJ0 speech database which contains 12776 utterances from 101 different speakers for training, and 5895 utterances from 18 different speakers for testing. To generate mixtures with 1 to 4 speakers, we use Eq. 5.4, and the 720 recorded RIRs from Section 5.3. From the 720 RIRs available, 640 are used for training, and 80 for testing. All recordings use a sample rate of  $f_s = 16$  kHz. For the DOA vectors  $\mathbf{V}(d, k)$ , we use a set of  $D = 100$  bases, which are equally distributed on a sphere, as shown in Figure 6.2, panel (a). This provides sufficient spatial resolution to separate speakers standing right next to each other. By using Eq. 6.4, we assign a DOA index  $d$  to each of the 720 RIRs.

Next, we use Algorithm 2 to extract the corresponding DOA indices  $\hat{d}$  for each of the  $C$  speakers. Then, the *beamforming* branch of the BSSD network in Figure 6.3 isolates and dereverberates the speaker at the position corresponding to  $\hat{d}$  from the noisy input mixture  $z(t)$ . The NN uses a latent space of  $H = 500$  neurons to predict the beamforming weights  $\mathbf{w}'(l)$ . Then, the enhanced signal  $\mathbf{y}'_{\hat{d}_c}(l)$  is calculated with Eq. 6.21. Finally, the *identification* branch of the BSSD network predicts the speaker embedding  $e_{\hat{d}}$  from the enhanced signal  $\mathbf{y}'_{\hat{d}_c}(l)$  in latent space. Table 6.1 shows all hyper-parameters used for the BSSD network in Figure 6.3.

description	parameter	value
block length for 5s of audio sampled at $f_s = 16\text{kHz}$	$T$	80,000
number of microphones	$M$	6
number of DOA bases used for speaker localization	$D$	100
total number of frames in latent space	$L$	1600
speaker embedding dimension	$E$	100
number of Conv/Deconv filters in the beamforming branch	$H$	500
filter length of the Conv/Deconv layers in the beamforming branch	$K_{BF}$	200
stride of the of the Conv/Deconv layers in the beamforming branch	$S_{BF}$	50
filter length of the Adaption layer (see Eq. 6.12)	$T_A$	200
filter length of the Conv layers in the identification branch	$K_{ID}$	10
stride of the Conv layers in the identification branch	$S_{ID}$	1
number of different speakers in a single training batch	$B$	20
number of utterances per speaker in a single training batch	$P$	3

Table 6.1: Parameters of the BSSD network shown in Figure 6.3.



## Training

The BSSD network is trained on mixtures of  $C = 2$  sources, where each mixture  $\mathbf{z}(t) = \sum_{c=1}^C s_c(t) \otimes \mathbf{h}_c(t)$  is truncated to 5 s length. The RIR  $\mathbf{h}_c(t)$  is chosen randomly for each example, so that all possible DOA locations are trained in the statistic adaption layer in Eq. 6.12. We use a batch size of 60 mixtures from the 101 speakers of the WSJ0 training set. To enable efficient triplet mining with Eq. 6.14, we use  $P = 3$  different utterances from  $B = 20$  speakers for the first source  $s_{c=1}(t)$  of each mixture. The second source  $s_{c=2}(t)$  is chosen randomly from the remaining 100 speakers from the WSJ0 training set. We use the clean, anechoic first source as reference utterance, i.e.  $r(t) = s_{c=1}(t - \tau_1)$ . The ground truth DOA index  $\hat{d}$  is used to train the network. During testing, Algorithm 2 is used to obtain an estimate of the DOA index  $\hat{d}$ . We use  $\lambda_1 = 10^{-2}$  and  $\lambda_2 = 10^{-4}$  for the cost function in Eq. 6.16. This ensures that the *beamforming* branch is trained faster than the *identification* branch, as the latter depends on the former. As the combination of the different RIRs and WSJ0 utterances allows for millions of combinations, we randomly create new batches for training and validation for each epoch. ADAM is used as optimizer [116], with a learning rate of  $10^{-3}$ .

## Performance

Table 6.2 reports the SI-SDR, WER and EER scores. We use the Google Speech-to-Text API as ASR system [130]. In its current version, this ASR framework reports a WER of 5.6% for the anechoic WSJ0 test set (si\_et\_05). For  $C = 1$  speaker, the BSSDs models only performs dereverberation. It can be seen that the WER for this case is close to the anechoic ground truth, which indicates that this ASR system was explicitly trained on reverberated speech. Further, both the WER and EER scores are the lowest for one speaker. This is to be expected, as no interfering components of other speakers reduce the intelligibility of the single speaker. Consequently, the embeddings  $\mathbf{e}_{\hat{d}}$  exhibit the best identification performance. For more speakers, the scores degrade gradually. The SI-SDR and the WER drop noticeably faster than the EER.

$C$	SI-SDR	WER	EER
1	14.40 dB	5.72 %	2.89 %
2	9.33 dB	26.19 %	5.75 %
3	7.92 dB	42.32 %	7.28 %
4	6.84 dB	56.57 %	9.39 %

Table 6.2: Performance of the time-domain BSSD network on  $C = \{1, 2, 3, 4\}$  speakers.

Figure 6.5 shows the separation performance of the BSSD network for  $C = 3$  sources. We used the same speakers and the same spatial arrangement as in Figure 6.2, where speaker 2 and 3 are standing right next to each other. Panel (a) shows the spectrogram of the first channel of the input mixture  $\mathbf{z}(t, m = 1)$ . Panel (b) illustrates the spectrogram for the reference signal for the first speaker, i.e.  $r_1(t)$ . Panels (c), (d) and (e) show the spectrograms of the isolated and dereverberated sources  $y_{\{1,2,3\}}(t)$ , respectively. It can be seen that all three speakers are clearly separated from one another, and dereverberated, i.e. the smoothing of the spectrogram over time has been removed. The SI-SDR for the enhanced outputs  $y_{\{1,2,3\}}(t)$  is 9.36 dB, 8.44 dB and 7.81 dB, respectively. The third speaker has the lowest score, as there are small artifacts from the second speaker, which can be seen in panel (e).

Figure 6.4 illustrates the performance of the BSSD network during training. Panel (a) shows the loss  $\mathcal{L}_{\text{SI-SDR}}$  from Eq. 5.9 versus training epochs. It can be seen that the SI-SDR settles at around 10 dB after  $4 \cdot 10^4$  epochs. Panel (b) shows the EER from Eq. 6.19 versus training epochs. The EER drops to approximately 2% after  $4 \cdot 10^4$  epochs. As the speaker identification branch depends on the beamforming branch of the BSSD, the EER does not decrease until the

SI-SDR is at a certain level, i.e. speaker separation has to work before speaker identification can take place. Panel (c) shows the FAR and FRR from Eq. 6.17 and 6.18 versus the threshold  $\delta$ , after  $10^5$  training epochs. Both rates are equally low at  $\delta \approx 3.1$ , enabling the EER to be as low as 2%. For further experiments, as well as the frequency-domain formulation of the BSSD network, and the application as a block-online diarization system, the interested reader is referred to Appendix A.10.

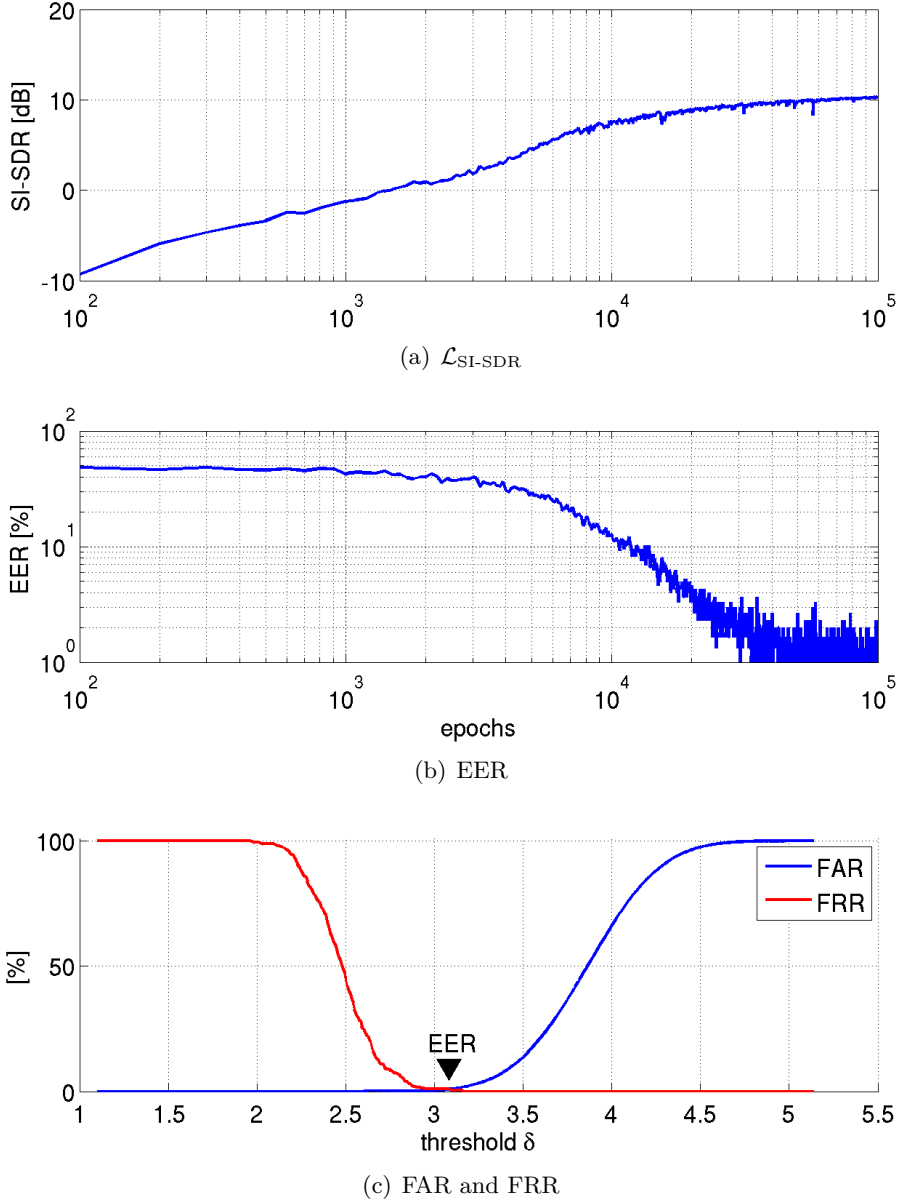


Figure 6.4: Performance of the BSSD network during training. (a) Loss  $\mathcal{L}_{\text{SI-SDR}}$  from Eq. 5.9 versus training epochs. (b) EER from Eq. 6.19 versus training epochs. (c) FAR and FRR from Eq. 6.17 and 6.18 versus the threshold  $\delta$ , after  $10^5$  training epochs.

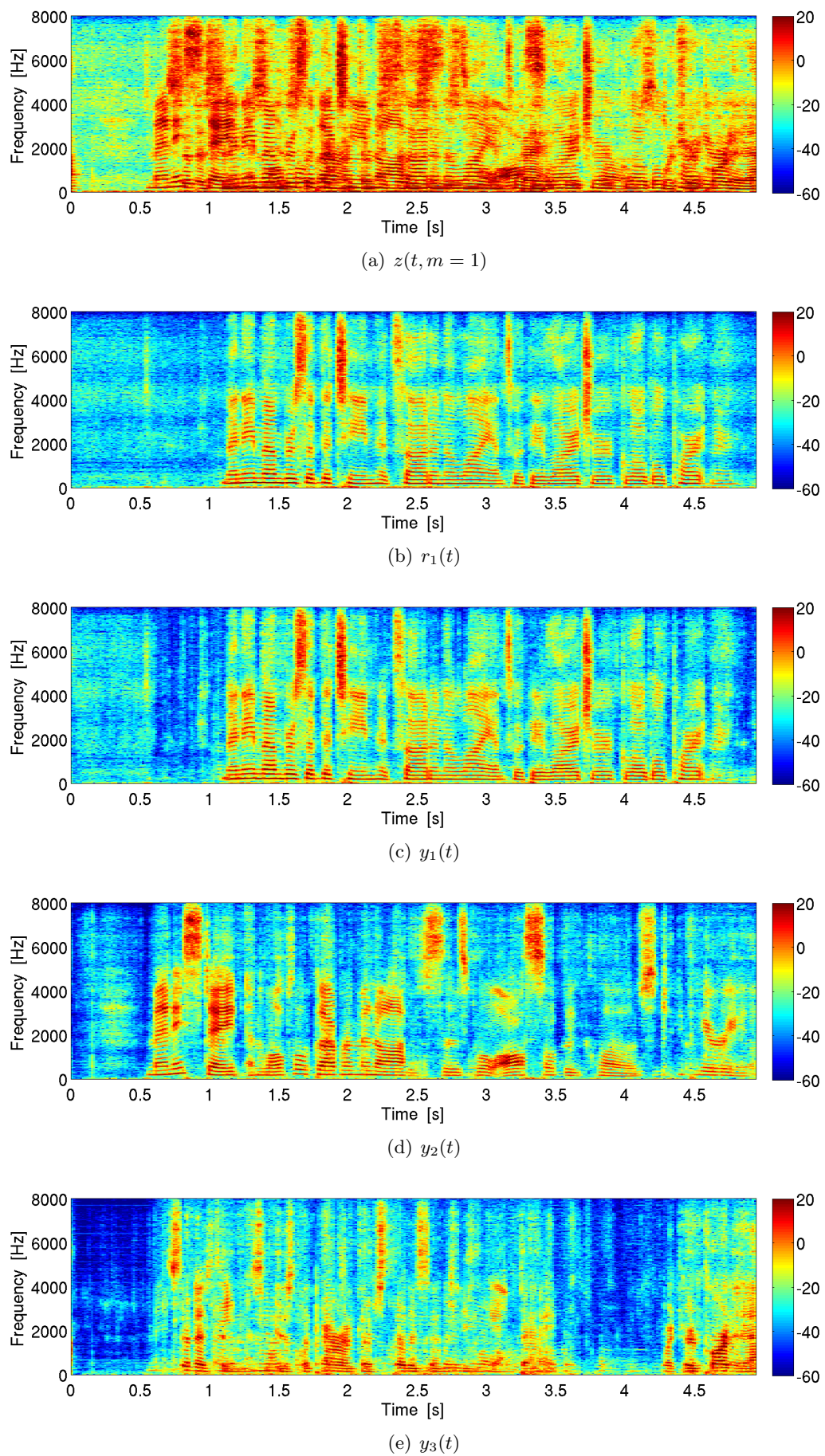


Figure 6.5: Isolated and dereverberated speakers of a mixture with  $C = 3$  speakers, using the BSSD network. (a) Spectrogram of the first microphone of the input  $z(t, m = 1)$ . (b) Reference signal  $r_1(t)$ . (c) Isolated source  $y_1(t)$ . (d)  $y_2(t)$ . (e)  $y_3(t)$ .

## 6.6 Conclusion

In this chapter, we introduced the BSSD network, which is able to separate, dereverberate, and diarize an unknown number of speakers from a given mixture of multiple, overlapping speakers. The BSSD architecture consists of three modules with dedicated tasks: (i) source localization, (ii) beamforming and dereverberation, (iii) and speaker identification. The source localization module operates unsupervised, i.e. by relying on a set of pre-defined DOA vectors, which are compared against the input signal to identify a set of candidate source positions. The beamforming and dereverberation module uses the TDNBF beamformer from Chapter 5, to extract and dereverberate a speech source at a given source position. The speaker identification module assigns an embedding vector to an extracted speech signal, which allows to re-identify a speaker along consecutively processed blocks of audio. The interaction of these three modules enables a monolithic, end-to-end training of the entire BSSD network. We have shown that this architecture is capable of separating up to four speakers, even when they are standing close to one another. By using block processing [53], it is possible to solve the cocktail party problem for both static and moving speakers [58]. Therefore, the BSSD architecture solves all of the six problems stated in the introduction, i.e.

1. **Isolate a single speaker from background noise.** ✓
2. **Isolate a single speaker from a mixture of multiple speakers.** ✓
3. **Track moving speakers.** ✓
4. **Isolate and dereverberate a speaker in the far-field.** ✓
5. **Separate all speakers in a mixture of multiple speakers.** ✓
6. **Assign an identity to an isolated speaker.** ✓

## 7

## Discussion and Outlook

In this thesis, we addressed six major problems in speech enhancement and speaker separation. We provided the mathematical foundation to solve all of these problems by utilizing the rich toolkit of classical signal processing and machine learning. We took inspiration from basic beamforming principles, as well as complex analysis and deep neural networks to tackle the cocktail party problem. Throughout this work, we provided solutions to four topics in the field of neural beamforming: (i) Mask-based beamforming, (ii) Complex-valued neural beamforming, (iii) Time-domain neural beamforming, (iv) and blind source separation. Each of these topics has its dedicated chapter, where we presented our contributions and insights. Here we summarize our most interesting findings and discuss how we advanced the field of speech enhancement and speaker separation.

- **Mask-based Beamforming:** Mask-based beamforming provides a convenient way to extract a single speaker from background noise. It has the distinct advantage of using a simple neural network to estimate a speech mask, and there is no need to change any of the well-established beamforming principles formulated in frequency-domain. However, the neural network does not utilize the phase information in the microphone signals, thereby neglecting spatial information. With the *Eigennet* beamformer, we provide this spatial information as an additional input to the neural network. We utilize the Eigenvector of the PSD matrix of the noisy input, which points towards the desired single speech source in signal subspace. Further, we formulated a multi-speaker variant of the *Eigennet*, which uses the normalized phase of the STFT representation of the noisy input signals. The phase normalization helps the neural network to detect phase changes, and thereby differences between multiple signal sources. Lastly, we introduced the concept of spatial whitening, which decorrelates the microphone inputs. Spatial whitening increases the spatial selectivity of any given microphone array, under the assumption of the ideal diffuse noise sound field. This benefits source localization algorithms, beamformers, postfilters, and the learning rate of neural networks, as was shown throughout this work.
- **Complex-valued Neural Beamforming:** While mask-based beamforming requires stationary speakers, the *Eigennet* allows for a limited degree of speaker movement, i.e. using block processing. With complex-valued neural beamforming, this limitation can be alleviated. In contrast to a statistical beamformer, our CNBF architecture is able to predict a set of individual beamforming weights for each time frame. This enables the neural network to quickly react to changes in the speaker’s position and movement. Further, the CNBF directly optimizes the max-SNR objective of the beamformer, instead of optimizing a speech mask. Consequently, it outperforms both traditional beamformers and mask-based approaches. As the prediction of beamforming weights requires complex-valued neural networks, we employed the Wirtinger calculus to create complex-valued recurrent network layers and non-holomorphic activation functions. As Wirtinger calculus is hardly supported by any major machine learning framework, we implemented both the forward and backward paths of numerous non-holomorphic functions required for beamforming.
- **Time-domain Neural Beamforming:** We introduced cross-domain learning as a natural extension to complex-valued neural beamforming. By including the gradient of both

the FFT and iFFT into a neural network, we can formulate a training objective purely in time-domain. This suggests to formulate the entire beamformer in time-domain, using neural networks to directly synthesize the desired output signal. This approach is completely detached from the physical representation of sound waves, or any beamforming algorithm. Instead, the neural network learns a latent representation that is optimized to solve a given problem. We demonstrated the versatility of this approach by conducting three experiments: (i) We formulated the TDNBF architecture, which performs the same tasks as the CNBF, but in time-domain instead of the frequency-domain. (ii) We performed dereverberation by using an anechoic reference signal in the training objective of the TDNBF, which provided competitive results compared to the WPE algorithm. (iii) Inspired by single-channel speech enhancement, we devised a postfilter to suppress non-linear residual echoes in an AEC application, which we termed NRES. Here, we also compared both time- and frequency-domain approaches.

- **Blind Source Separation:** Finally, we proposed a monolithic, all-in-one solution to perform multi-speaker separation, dereverberation and diarization using a single neural network, termed the BSSD architecture. As there are only little constraints on the number or the location of the involved speakers, this approach solves the cocktail party or meeting room problem. Speaker separation is achieved by using an analytic or statistic adaption layer, which virtually moves a speech source to the coordinate origin of the microphone array, from where it is extracted using a neural network in time-domain. This process only depends on a single DOA vector, which is identified by a specialized variant of the GCC-PHAT. Signal dereverberation is done by the dereverberation constraint formulated for the TDNBF, which simultaneously performs speaker separation and dereverberation. Finally, speaker diarization is built upon embedding vectors and the triplet loss, which we modified to cope with small batch sizes and short utterances.

While each of our contributions provided new findings and insights, they also gave rise to new questions and future research topics: With cross-domain learning, we can incorporate the SI-SDR objective into the neural network. While this objective is loosely related to signal intelligibility, there is no relation to signal quality. Measures such as PESQ or PEASS try to address this shortcoming. However, these measures cannot be used to optimize a neural network. It would be beneficial to have a subjective quality measure that provides a gradient, so that we can incorporate it into the training objective of a neural network. While Generative Adversarial Networks (GANs) provide a promising ansatz, there is yet a long way to go.

Non-linear residual echoes are challenging to model and hard to remove in a real-world AEC application. While the NRES provides a small and efficient neural network to remove the echo artifacts, the AEC itself is still formulated in frequency-domain, using a FIR filter with thousands of taps to model the echo tail. Our findings with the NRES experiment strongly suggest that a significantly shorter filter would be sufficient, if the gradient of the AEC itself was incorporated into the neural network. In frequency-domain, this can be done using Wirtinger calculus. In time-domain, this can be done using a latent representation similar to the TDNBF.

# A

## Appendix

### **A.1 Eusipco 2014**

*"A multi-channel postfilter based on the diffuse noise sound field"*, Lukas Pfeifenberger and Franz Pernkopf, 22nd European Signal Processing Conference (EUSIPCO), Lisbon, 2014

# A MULTI-CHANNEL POSTFILTER BASED ON THE DIFFUSE NOISE SOUND FIELD

Lukas Pfeifenberger<sup>1</sup> and Franz Pernkopf<sup>1</sup>

<sup>1</sup> Signal Processing and Speech Communication Laboratory  
Graz University of Technology, Graz, Austria

lukas.pfeifenberger@alumni.tugraz.at, pernkopf@tugraz.at

## ABSTRACT

In this paper, we present a multi-channel *Directional-to-Diffuse Postfilter* (DD-PF), relying on the assumption of a directional speech signal embedded in diffuse noise. Our postfilter uses the output of a superdirective beamformer like the *Generalized Sidelobe Canceller* (GSC), which is projected back to the microphone inputs to separate the sound field into its directional and diffuse components. From these components the SNR at the output of the beamformer can be derived without needing a *Voice Activity Detector* (VAD). The SNR is used to construct a noise cancelling Wiener filter. In our experiments, the developed algorithm outperforms two recent postfilters based on the *Transient Beam to Reference Ratio* (TBRR) and the *Multi-Channel Speech Presence Probability* (MCSSP).

**Index Terms**— beamforming, multi-channel postfilter, diffuse sound field

## 1. INTRODUCTION

Speech intelligibility is a paramount issue in modern telecommunication systems. In many applications, background noise is the primary source of speech degradation. While single-channel speech enhancement systems require an inherent trade-off between noise reduction and speech quality, multi-channel speech enhancement systems also exploit the spatial information of the sound field and, thereby achieve a better performance. For this purpose, superdirective beamformers like the *Generalized Sidelobe Canceller* (GSC) [1, 2] in conjunction with multi-channel postfilters have gained the most attraction over the last decade. In this paper we assume a diffuse noise sound field, which can be found in a wide range of applications, such as car interiors, subway stations or roadside emergency telephones [3]. Further, we assume that the speaker is located close to the array, resulting in strong directional components in the *Acoustic Transfer Functions* (ATFs). We therefore model the ATFs as simple time delays, which can be identified by estimating the *Direction of Arrival* (DOA) using one of the algorithms discussed in [4]. The assumption of a diffuse noise sound field has already been used in postfilter concepts like [3] and [5], where a Wiener

postfilter is derived from the speech and noise *Power Spectral Densities* (PSDs) at the beamformer output. However, many of these postfilters rely on a VAD and an accurate speech PSD estimate. A comprehensive overview of these methods is given in [6].

Our *Direct-to-Diffuse Postfilter* (DD-PF) algorithm estimates the SNR at the beamformer output by splitting the sound field at the microphones into its directional and diffuse components, using only the assumption of a diffuse noise field. This approach is inspired by the *Signal to Reverberant Ratio* (SRR) [7] and the *multi-channel SNR* in [8]. Other approaches to multi-channel postfilters are, for example: the *Transient Beam to Reference Ratio* (TBRR) [2], which relies on the ratio of transient energies in the beamformer output and in the output of the blocking matrix. These transient energies are determined using noise floor estimates in both the beamformer output and the blocking matrix outputs. The *Multi-Channel Speech Presence Probability* (MC-SPP) [8] algorithm can also be used without a beamformer, as it directly estimates the noise PSD matrix based on an a-priori speech presence probability and recursive averaging. In a similar approach given by [9], the SRR is mapped into a *speech absence probability* (SAP) used for recursive noise PSD estimation. Unlike these approaches, the performance of our postfilter only depends upon target leakage in the blocking matrix, and the diffuse noise field assumption.

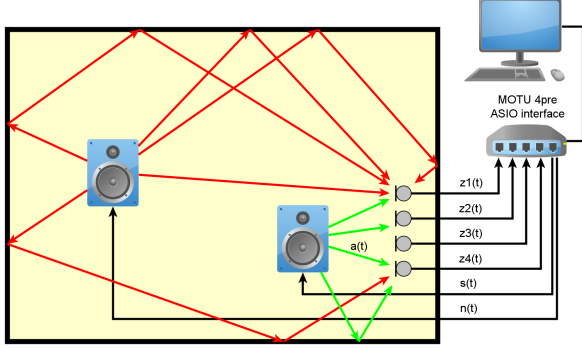
This paper is organized as follows: Section 2 verifies the assumptions about the sound fields. Section 3 introduces the signal model and the beamformer. The DD-PF is derived in Section 4. Section 5 presents the experimental setup and performance results, where our postfilter is compared with two other approaches: the TBRR [2] and the MC-SPP [8]. The performance and speech quality is evaluated by using the *Perceptual Evaluation Methods for Audio Source Separation* (PEASS) Toolkit [10, 11]. Section 6 concludes the paper.

## 2. VERIFICATION OF THE SOUND FIELDS

In our setup, we assume a hands-free telephone situated in a noisy environment. In such a scenario, the speaker is located much closer to the microphone array than the noise source(s). Hence, a mostly directional speaker sound field and a diffuse

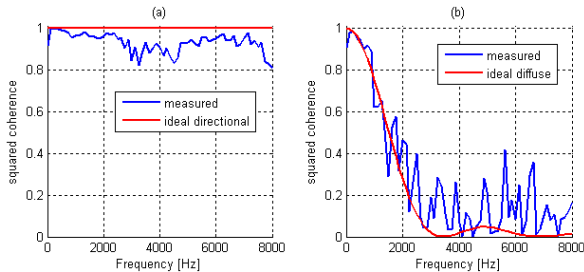


noise sound field is expected. To verify these assumptions, we placed  $M = 4$  microphones in a linear array with an inter-microphone distance of  $d = 5$  cm. The array is located in a  $5 \times 8$  m wide hall with a RT60  $\approx 550$  ms (see Figure 1).



**Fig. 1.** Setup with a linear array consisting of  $M = 4$  microphones at an inter-microphone distance of  $d = 5$  cm.

To simulate the speaker, a loudspeaker is placed at a distance of 0.5 m at a DOA of  $0^\circ$  in front of the array. For the diffuse background noise, a second loudspeaker is placed 5 m away from the array. Using the MLS technique for room impulse measurement [12], it can be verified that the speaker sound field has a strong directional component and the noise sound field is mainly diffuse. Figure 2 shows the squared coherence for both scenarios. This result is similar to [3].



**Fig. 2.** Measurement of the squared coherence using the first two microphones for the loudspeaker at the position of (a) 0.5 m and (b) 5 m.

### 3. SIGNAL MODEL

In Figure 1 we denote the ambient noise at the  $m^{\text{th}}$  microphone as  $n_m(t)$ , and the ATF from the speaker to the  $m^{\text{th}}$  microphone as  $a_m(t)$ . With these definitions, the signal model can be written as  $z_m(t) = a_m(t) * s(t) + n_m(t)$  in time-domain. In the fourier-domain  $Z_m(j\Omega) = A_m(j\Omega)S(j\Omega) + N_m(j\Omega)$ . Covering all  $M$  microphones, the signal model can be written in a more compact vector notation as

$$\mathbf{Z}(j\Omega) = \mathbf{A}(j\Omega)S(j\Omega) + \mathbf{N}(j\Omega). \quad (1)$$

While the proposed postfilter can be used in conjunction with any beamformer, we used the GSC for both its robustness and simplicity. It has been implemented as suggested in [2, 13, 14]. Its filter weights are given as  $\mathbf{W}(j\Omega) = \mathbf{F}(j\Omega) - \mathbf{H}(j\Omega)\mathbf{B}(j\Omega)$ , with the *delay and sum* beamformer  $\mathbf{F}(j\Omega)$ , the *blocking matrix*  $\mathbf{B}(j\Omega)$  and an *adaptive interference canceler*  $\mathbf{H}(j\Omega)$ .

Due to the mainly directional speaker sound field encountered in Section 2, we modeled the ATFs as simple time delays, i.e.  $\hat{A}_m(\Omega) = e^{jk d_m \sin \Theta}$ , where  $k = \frac{\omega}{c}$  is the wave number,  $d_m$  is the distance between the  $m^{\text{th}}$  microphone and an arbitrary reference point [1], and  $c$  is the speed of sound. Since the blocking matrix depends on the ATFs, target leakage might occur as a consequence of undermodeling, resulting in a degraded speech signal at the GSC output. However, we found the *signal blocking factor* [15] to be about 16dB in our experiments, which seems quite sufficient.

If the beamformer is steered towards the speech source, e.g.  $\hat{\mathbf{A}}(j\Omega) \approx \mathbf{A}(j\Omega)$ , all sounds originating from that direction are allowed to pass, since  $\mathbf{W}^H(j\Omega)\hat{\mathbf{A}}(j\Omega) \approx 1$ . This includes the speaker signal, and the portion of the noise impinging from that direction [1]. The beamformer output can therefore be written as

$$\begin{aligned} Y(j\Omega) &= \mathbf{W}^H(j\Omega)\mathbf{Z}(j\Omega) \\ &= \hat{S}(j\Omega) + \mathbf{W}^H(j\Omega)\mathbf{N}(j\Omega), \end{aligned} \quad (2)$$

where  $\hat{S}(j\Omega)$  is the estimate of the speech source, and  $\mathbf{W}^H(j\Omega)\mathbf{N}(j\Omega)$  is the noise component coming from the direction of the speaker.

### 4. MULTI-CHANNEL POSTFILTER

Our DD-PF algorithm estimates the SNR at the beamformer output without the need for a speech PSD estimate or a VAD. This is achieved by back-projecting the GSC output  $Y(j\Omega)$  to the microphone signals  $\mathbf{Z}(j\Omega)$  using the ATF model  $\hat{\mathbf{A}}(j\Omega)$ , we obtain

$$\begin{aligned} \hat{\mathbf{Z}}' &= \hat{\mathbf{A}}Y = \hat{\mathbf{A}}\hat{S} + \hat{\mathbf{A}}\mathbf{W}^H\mathbf{N}, \\ \hat{\mathbf{Z}}'' &= \mathbf{Z} - \hat{\mathbf{Z}}' \approx [\mathbf{I} - \hat{\mathbf{A}}\mathbf{W}^H]\mathbf{N}, \end{aligned} \quad (3)$$

assuming  $\hat{\mathbf{A}}\hat{S} = \mathbf{A}S$ . This assumption holds if the target leakage in the blocking matrix is low. The frequency argument  $j\Omega$  has been omitted for brevity. It can be easily seen that  $\hat{\mathbf{Z}}'$  denotes the directional signal components, and  $\hat{\mathbf{Z}}''$  the remaining diffuse components. Due to statistical independence of the speech and the noise signal, the spatial PSD matrices of  $\hat{\mathbf{Z}}'$  and  $\hat{\mathbf{Z}}''$  can be written as

$$\begin{aligned} \Phi_{\hat{\mathbf{Z}}'\hat{\mathbf{Z}}'} &= \hat{\mathbf{A}}\Phi_{\hat{S}\hat{S}}\hat{\mathbf{A}}^H + \hat{\mathbf{A}}\mathbf{W}^H\Phi_{\mathbf{N}\mathbf{N}}\mathbf{W}\hat{\mathbf{A}}^H \\ &= \Phi_{\hat{S}'\hat{S}'} + \Phi_{\hat{\mathbf{N}}'\hat{\mathbf{N}}'} \quad \text{and} \\ \Phi_{\hat{\mathbf{Z}}''\hat{\mathbf{Z}}''} &\approx [\mathbf{I} - \hat{\mathbf{A}}\mathbf{W}^H]\Phi_{\mathbf{N}\mathbf{N}}[\mathbf{I} - \mathbf{W}\hat{\mathbf{A}}^H] \\ &= \Phi_{\hat{\mathbf{N}}''\hat{\mathbf{N}}''}. \end{aligned} \quad (4)$$

In [8], a multi-channel SNR as generalization from the single-channel case was defined as  $\xi = \text{Tr}(\Phi_{\hat{S}'\hat{S}'}^{-1} \Phi_{\hat{S}'\hat{S}'})$ . Similarly, we evaluate only the power ratio of the main diagonals of these PSD matrices as

$$\xi = \frac{\text{Tr}(\Phi_{\hat{S}'\hat{S}'})}{\text{Tr}(\Phi_{\hat{N}'\hat{N}'}), \quad (5)$$

because both PSD matrices  $\Phi_{\hat{S}'\hat{S}'}$  and  $\Phi_{\hat{N}'\hat{N}'}$  represent purely directional sound fields. Additionally, we can circumvent the numerically ill-conditioned matrix inversion of  $\Phi_{\hat{N}'\hat{N}'}$ , caused by strong spatial correlations for low frequencies. However, we cannot measure  $\Phi_{\hat{S}'\hat{S}'}$  or  $\Phi_{\hat{N}'\hat{N}'}$  directly, but Eqn. (5) can be expressed as

$$\xi = \frac{\text{Tr}(\Phi_{\hat{Z}'\hat{Z}'}) \text{Tr}(\Phi_{\hat{N}''\hat{N}''})}{\text{Tr}(\Phi_{\hat{Z}''\hat{Z}''}) \text{Tr}(\Phi_{\hat{N}'\hat{N}'})} - 1. \quad (6)$$

By assuming an ideal spherical diffuse noise sound field at the microphones, the noise PSD matrix  $\Phi_{NN}$  can be written as

$$\Phi_{NN} = \Phi_{NN} \Gamma_{NN}, \quad (7)$$

where  $\Phi_{NN}$  denotes the unknown PSD of the noise source, and the elements of the spatial coherence matrix  $\Gamma_{NN}$  are defined as the coherence function [16] between the  $i^{\text{th}}$  and the  $j^{\text{th}}$  microphone, i.e.

$$\Gamma_{N_i N_j}(j\Omega) = \frac{\sin(kd_{ij})}{kd_{ij}}, \quad (8)$$

where  $d_{ij}$  is the distance between microphone  $i$  and  $j$ . Using Eqn. (7), the ratio  $\frac{\text{Tr}(\Phi_{\hat{N}''\hat{N}''})}{\text{Tr}(\Phi_{\hat{N}'\hat{N}'})}$  in Eqn. (6) is obtained by

$$\frac{\text{Tr}(\Phi_{\hat{N}''\hat{N}''})}{\text{Tr}(\Phi_{\hat{N}'\hat{N}'})} = \frac{\text{Tr}([I - \hat{A}\mathbf{W}^H] \Gamma_{NN} [I - \mathbf{W}\hat{A}^H])}{\text{Tr}(\hat{A}\mathbf{W}^H \Gamma_{NN} \mathbf{W}\hat{A}^H)}, \quad (9)$$

using the ATF model  $\hat{A}$  and the beamforming filter  $\mathbf{W}$ . The coherence matrix  $\Gamma_{NN}$  is a constant. The directional and diffuse component of the input signal,  $\hat{Z}'$  and  $\hat{Z}''$ , are estimated online using Eqn. (3). Their respective PSDs are found by recursive averaging, e.g.  $\Phi_{\hat{Z}'\hat{Z}',l} = \Phi_{\hat{Z}'\hat{Z}',l-1} \alpha + (1 - \alpha) \hat{Z}' \hat{Z}'^H$ , where  $l$  is the frame index. The SNR  $\xi$  is obtained by using Eqn. (6). This SNR is then used to construct a Wiener filter. We used the *Optimally-Modified Log-Spectral Amplitude Estimator* (OM-LSA) algorithm [17], which is often found in noise cancelling applications.

## 5. EXPERIMENTS

### 5.1. Directivity Pattern

The proposed postfilter depends only on the current beamformer state defined by  $\hat{A}$  and  $\mathbf{W}$ . Therefore, the postfilter can easily be incorporated into the overall *Directivity Pattern*

of the beamformer. The procedure described in [18] is used to simulate the theoretical directivity pattern with a two element array with  $d = 5$  cm. The beamformer is fixed to look towards  $0^\circ$ . In comparison to the beampattern of the GSC without a postfilter [4], Figure 3 demonstrates the improved directivity especially for low frequencies.

To measure the real directivity pattern for comparison, we used the room from Figure 1, and the array mounted on a turntable. Figure 4 shows the measured beampattern for two microphones. Especially for low frequencies, it is sharper than the theoretical result. A cause for this effect could be minor gain differences in the microphones, which are not modeled by the simplified ATFs. However, it can be seen that signals impinging from outside  $\pm 20^\circ$  are completely suppressed. Increasing the number of microphones up to four did not change the directivity pattern significantly.

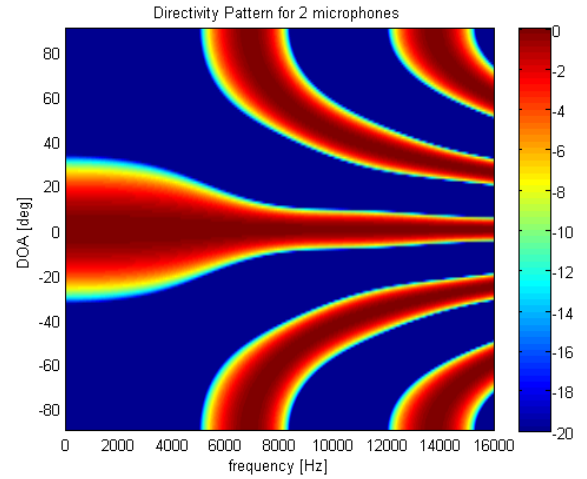


Fig. 3. Simulated directivity pattern for a two microphone beamformer with an aperture of  $d = 5$  cm.

### 5.2. Experimental Setup

To test the speech quality of our MCSE system against a significant amount of speech data, the TIMIT [19], KCORS [20], and (KCOSS) [21] speech corpora have been used. The speech signals have been replayed with the loudspeaker at the 0.5 m position (see Figure 1). For the noise data, recordings from various sources, e.g. traffic noise, industry parks, subway stations and the NOIZEUS database have been replayed with the loudspeaker at the 5 m position. In total, about 60 minutes of test material has been generated. For comparison, we also implemented two other postfilter approaches – the MC-SPP approach and the TBRR. For the GSC beamformer we used a sparse blocking matrix  $\mathbf{B}(j\Omega)$ , which has the same performance as a dense eigenspace blocking matrix [22]. Its main benefit is the linear growth of computational complexity

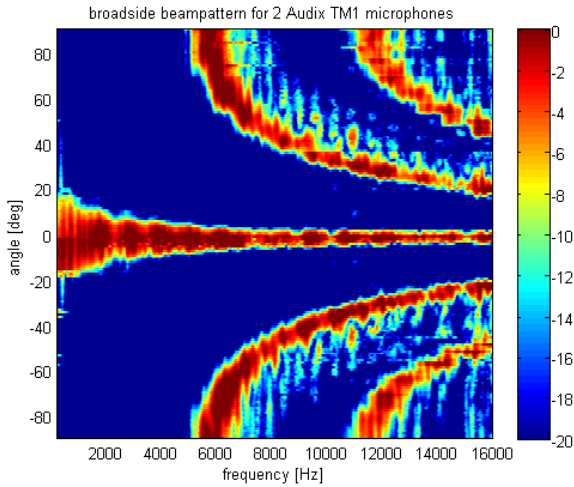


Fig. 4. Measured directivity pattern for 2 Audix-TM1 microphones placed  $d = 5$  cm apart.

with the number of microphones. All GSC filters are implemented as non-causal FIR filters, to allow both positive and negative time delays [1]. The sampling frequency is  $f_s = 16$  kHz and the SFFT length is 16ms, where we used a hanning window and 50% overlapping frames.

The PEASS Toolkit [10, 11] is used to evaluate the performance of the algorithms in terms of performance and perceptual speech quality. While PEASS might not be intended specifically for speech enhancement tasks, we found it represents the perceived speech quality much better than for example PESQ. PEASS delivers four scores: The *Target Perceptual Score* (TPS) measures the perceptual quality of the desired speech signal contained in the postfilter output. The *Interference Perceptual Score* (IPS) measures the influence of the residual noise components in the beamformer output. The *Artifact Perceptual Score* (APS) measures the influence of artifacts like musical noise generated by the algorithm. And the *Overall Perceptual Score* (OPS) provides a global measure of the perceptual quality of the enhanced output. Each score ranges from 0 to 100 and large values indicate better performance.

### 5.3. Results

Each algorithm is tested with a signal-to-interference ratio (SIR) ranging from -20 dB to +20 dB in 5 dB steps. Figure 5 shows the performance of the postfilters in terms of the PEASS measures. The OPS score of the TBRR and the MC-SPP postfilters are more or less equal. However, the TBRR performs better than the MC-SPP for the IPS and TPS score, and the APS score indicates that the TBRR introduces the most artifacts. The MC-SPP algorithm has the lowest IPS score, as it relies on the inversion of the spatial noise PSD

matrix which is numerically unstable at low frequencies due to high signal correlations. The TBRR algorithm has the lowest APS score, as it relies on recursive noise floor estimation [23,24]. Depending on the instationarity of the noise, this technique is known to introduce musical artifacts. The speech quality of the proposed DD-PF does not depend on spatial speech PSD estimation or a VAD, but only on the estimate of the directional and the diffuse sound components  $\Phi_{\hat{z}'\hat{z}'}$  and  $\Phi_{\hat{z}''\hat{z}''}$ . Their accuracy is determined by the shape of the assumed noise field and the target leakage in the blocking matrix. In our experiments, target leakage was quite low, and the noise sound field was nearly diffuse. Therefore, we achieved both a good speech quality and a good noise suppression at the same time, even for low frequencies. This can be seen by the OPS and TPS score.

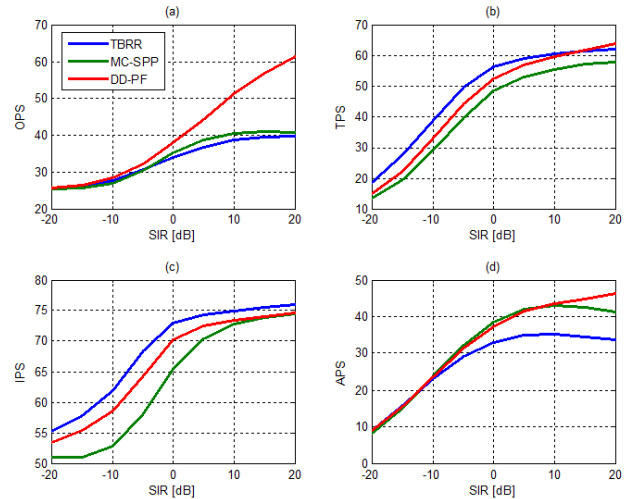


Fig. 5. Comparison of postfilters using PEASS measures; (a) OPS, (b) TPS, (c) IPS and (d) APS.

## 6. CONCLUSIONS

In this paper, we introduced the *Directional-to-Diffuse Postfilter* (DD-PF), which splits the sound field at the microphones into its directional and diffuse components to derive the SNR at the output of the beamformer, from which a noise reduction Wiener filter is derived. Unlike similar approaches, the algorithm does not depend on spatial speech PSD estimation or a VAD, but only on target leakage in the beamformer and the diffuse noise field assumption. In our experiments, these conditions have been sufficiently met. The achieved directivity pattern is selective even at low frequencies and the speech quality is significantly higher compared to the TBRR and MC-SPP approaches.

## REFERENCES

- [1] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*, Springer, Berlin–Heidelberg–New York, 2008.
- [2] S. Gannot and I. Cohen, “Speech enhancement based on the general transfer function GSC and postfiltering,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 6, Nov. 2004.
- [3] I. A. McCowan and H. Bourlard, “Microphone array post-filter based on noise field coherence,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 11, no. 6, Nov. 2003.
- [4] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*, Springer, Berlin–Heidelberg–New York, 2008.
- [5] N. Ito, N. Ono, E. Vincent, and S. Sagayama, “Designing the wiener post-filter for diffuse noise suppression using imaginary parts of inter-channel cross spectra,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2818–2821, Nov. 2010.
- [6] T. Wolff and M. Buck, “A generalized view on microphone array postfilters,” *International Workshop on Acoustic Signal Enhancement*, Sept. 2010.
- [7] O. Thiergart, G. Del Galdo, and E. A. P. Habets, “Signal-to-reverberant ratio estimation based on the complex spatial coherence between omnidirectional microphones,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 309–312, 2012.
- [8] Mehrez Souden, Jingdong Chen, Jacob Benesty, and Sofiene Affes, “An integrated solution for online multichannel noise tracking and reduction,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, Sept. 2011.
- [9] M. Taseska and E. A.P. Habets, “MMSE-based blind source extraction in diffuse noise fields using a complex coherence-based a priori sap estimator,” *International Workshop on Acoustic Signal Enhancement*, Sept. 2012.
- [10] E. Vincent, “Improved perceptual metrics for the evaluation of audio source separation,” *10th Int. Conf. on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pp. 430–437, 2012.
- [11] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, “Subjective and objective quality assessment of audio source separation,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, Sept. 2011.
- [12] G.B. Stan, J.J. Embrechts, and D. Archambeau, “Comparison of different impulse response measurement techniques,” Tech. Rep., Sound and Image Department, University of Liege, Belgium, 2002.
- [13] I. Cohen, “Analysis of two-channel generalized side-lobe canceller (GSC) with post-filtering,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, Nov. 2003.
- [14] I. Cohen, “Multichannel post-filtering in nonstationary noise environments,” *IEEE Transactions on Signal Processing*, vol. 52, no. 5, May 2004.
- [15] R. Talmon, I. Cohen, and S. Gannot, “Relative transfer function identification using convolutive transfer function approximation,” *IEEE Transactions on audio, speech, and language processing*, vol. 17, no. 4, May 2009.
- [16] H. Kuttruff, *Room Acoustics*, Spoon Press, London–New York, 5th edition, 2009.
- [17] I. Cohen, “Optimal speech enhancement under signal presence uncertainty using log-spectral amplitude estimator,” *IEEE Signal Processing Letters*, vol. 9, no. 4, Apr. 2002.
- [18] P. Vary and R. Martin, *Digital Speech Transmission*, Wiley, West Sussex, 2006.
- [19] “Timit acoustic-phonetic continuous speech corpus,” Website, Available online at <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1>; visited on January 16th 2013.
- [20] “The kiel corpus of read speech vol. 1,” Website, Available online at <http://www.ipds.uni-kiel.de/forschung/kielcorpus.en.html>; visited on January 16th 2013.
- [21] “The kiel corpus of spontaneous speech vol. 1-3,” Website, Available online at <http://www.ipds.uni-kiel.de/forschung/kielcorpus.en.html>; visited on January 16th 2013.
- [22] M. G. Shmulik, S. Gannot, and I. Cohen, “A sparse blocking matrix for multiple constraints GSC beamformer,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2012.
- [23] I. Cohen, “Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, Sept. 2003.
- [24] R. Martin, “Noise power spectral density estimation based on optimal smoothing and minimum statistics,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, July 2001.

## A.2 InterSpeech 2014

*"Blind source extraction based on a direction-dependent a-priori SNR"*, Lukas Pfeifenberger and Franz Pernkopf, Interspeech 2014 - 15th Annual Conference of the International Speech Communication Association (InterSpeech), Singapore, 2014



# Blind source extraction based on a direction-dependent a-priori SNR

Lukas Pfeifenberger<sup>1</sup>, Franz Pernkopf<sup>1</sup>

<sup>1</sup> Signal Processing and Speech Communication Laboratory  
Graz University of Technology, Graz, Austria

lukas.pfeifenberger@alumni.tugraz.at, pernkopf@tugraz.at

## Abstract

In many hands-free applications, we encounter a speaker located in the near-field embedded in diffuse far-field noise. In this paper, we contribute an algorithm to estimate the speech and noise power spectral density (PSD) based on a *direction-dependent SNR* (DD-SNR). The only prior knowledge needed is a model of the diffuse noise sound field. The enhanced speech signal is obtained by a parametric multi-channel Wiener filter (PMWF), which is constructed without any speech presence or absence probabilities, or smoothing in frequency. We achieve high speech quality and sufficient noise reduction by iteratively improving the speech PSD estimate using the output of the PMWF. The performance of our algorithm is demonstrated by using the PESQ and PEASS measures.

**Index Terms:** multi-channel speech enhancement, blind source extraction, noise PSD estimation

## 1. Introduction

Speech intelligibility is a paramount issue in many telecommunication devices. Especially in hands-free applications, background noise is the primary source of speech degradation. Great efforts have been made over the last decades to reduce this ambient noise. While single-channel speech enhancement algorithms require an inherent trade-off between noise reduction and speech quality, multi-channel algorithms also exploit the spatial information of the sound field and thereby achieve better results.

Recent algorithms try to estimate the noise *power spectral density* (PSD) from either the output of a beamformer, or directly from the microphone signals. Noise reduction is then achieved by using a single- or multi-channel Wiener filter. In [1], a *transient beam to reference ratio* (TBRR) is used to derive a speech absence probability, which is used to control a recursive noise floor averaging estimator. In [2], a *multi-channel speech presence probability* (MC-SPP) is derived as a generalization of the classical single-channel *a posteriori* speech presence probability. The MC-SPP is used as a soft-decision rule to estimate the spatial correlation matrix of the noise signal. A similar approach can be found in [3], where a *direct to diffuse ratio* (DDR) is used instead of the MC-SPP.

In this paper we do not use a speech absence or presence probability, but estimate the noise PSD from differential signals which cancel out the speaker. First, we extend the signal-to-reverberant ratio proposed in [4] to the multi-channel case, in order to obtain an a-priori *direction-dependent SNR* (DD-SNR). This SNR is then used to estimate the *acoustic transfer functions* (ATFs) from the speaker to the microphones. The ATFs include reflections like acoustic echoes, and are therefore hard to estimate in general. However, in our case of a near-field speaker the ATFs mainly consist of a constant delay and unity gain. The ATFs are used to align the microphone inputs so that the speech

signal is either constructively or destructively added, thereby allowing us to estimate the speech and noise PSDs. Finally, a *parametric multi-channel Wiener filter* (PMWF) is employed to obtain the enhanced speech signal. We show how the PMWF can be used in a second iteration to achieve considerable noise reduction and maintain a high speech quality at the same time. The noise reduction performance and speech quality of our approach is evaluated by using the PESQ score and the *Perceptual Evaluation Methods for Audio Source Separation* (PEASS) Toolkit [5, 6].

This paper is organized as follows: Section 2 introduces the signal model and the involved sound field. Section 3 considers the estimation of the DD-SNR, and in Section 4 the speech and noise PSDs are estimated. In Section 5, we formulate the PMWF and show how to use its output in a second iteration. Section 6 summarizes the entire algorithm for clarity. Section 7 evaluates our approach using the PESQ and PEASS scores, and compares it against the TBRR and MC-SPP algorithms. Section 8 concludes the paper.

## 2. Problem formulation

In our setup, we assume the desired speech source to be in the near-field, and the interfering noise source to be located in the far-field of a linear microphone array of  $M$  sensors, with an inter-microphone distance of  $d = 5\text{cm}$ . Diffuse noise and a near-field speaker (i.e.,  $0.5\text{m}$  speaker distance) are found in many real-world scenarios like car interiors, subway stations or roadside emergency telephones [7].

In the frequency domain, we define the signal at the  $i^{\text{th}}$  microphone as  $Z_i(k, l) = A_i(k, l)S(k, l) + N_i(k, l)$ , using the wave number  $k = \frac{2\pi f}{c}$  and the frame index  $l$ , where  $f$  and  $c$  denote the frequency and the speed of sound, respectively. The unknown speech signal is denoted by  $S(k, l)$ , and  $N_i(k, l)$  expresses the ambient noise signal at the  $i^{\text{th}}$  channel.  $A_i(k, l)$  is the unknown ATF from the speaker to the  $i^{\text{th}}$  microphone. Covering all  $M$  microphones, the signal model can be written in compact vector notation as

$$\mathbf{Z}(k, l) = \mathbf{A}(k)S(k, l) + \mathbf{N}(k, l). \quad (1)$$

The spatial correlation matrix [8] for all microphone signals is defined as expectation of  $\mathbf{Z}(k, l)\mathbf{Z}^H(k, l)$ , i.e.:

$$\Phi_{\mathbf{Z}\mathbf{Z}}(k) \triangleq E\{\mathbf{Z}(k, l)\mathbf{Z}^H(k, l)\}. \quad (2)$$

Usually,  $\Phi_{\mathbf{Z}\mathbf{Z}}(k)$  can be estimated by recursive averaging using  $\Phi_{\mathbf{Z}\mathbf{Z}}(k, l) = \Phi_{\mathbf{Z}\mathbf{Z}}(k, l-1)\alpha + (1-\alpha)\mathbf{Z}(k, l)\mathbf{Z}^H(k, l)$ . By assuming uncorrelated speech and noise signals, Eqn. (2)



can also be stated as

$$\begin{aligned}\Phi_{ZZ}(k, l) &= \Phi_{SS}(k, l) + \Phi_{NN}(k, l) \\ &= \mathbf{A}(k)\mathbf{A}^H(k)\Phi_S(k, l) + \Gamma_{NN}(k)\Phi_N(k, l),\end{aligned}\quad (3)$$

where  $\Phi_S(k, l)$  and  $\Phi_N(k, l)$  denote the PSDs of the unknown speech and noise sources, and  $\Gamma_{NN}(k)$  is the spatial coherence matrix of the diffuse or isotropic noise sound field. It can be thought of as the summation of infinitely many plane waves impinging from all directions at equal strength [9]. Its elements are given as  $\Gamma_{N_i N_j}(k) = \frac{\sin(kd_{ij})}{kd_{ij}}$ , where  $d_{ij}$  is the distance between the  $i^{\text{th}}$  and the  $j^{\text{th}}$  microphone. With this setup, our aim is to estimate the speech source  $S(k, l)$ . However, we do not intend to perform blind dereverberation, it is sufficient to estimate the speech signal at the first (the reference) microphone  $A_1(k)S(k, l)$ .

### 3. Direction-dependent a-priori SNR

By simplifying the ATFs to single monochromatic plane waves [9], it becomes possible to detect the presence of the speech signal in the mixed sound field  $\Phi_{ZZ}(k)$  without prior knowledge [4], i.e.:

$$A_i(k, l) \approx \tilde{A}_i(k, l) = e^{jk d_i \sin \Theta}, \quad (4)$$

where  $\Theta$  is the impinging angle of the sound wave towards the array, and  $d_i$  is the distance between the  $i^{\text{th}}$  microphone and an arbitrary reference point. This reference point is chosen to be the first microphone [10], so that  $\tilde{A}_1(k, l) \triangleq 1$ . Using this simplified model, the SNR between the speech and noise PSDs  $\xi_\Theta(k, l) = \frac{\Phi_S(k, l)}{\Phi_N(k, l)}$  can be estimated from Eqn. (3). We will use this *direction-dependent a-priori SNR* (DD-SNR) as a sensitive and robust voice activity detector. It can be derived using the spatial coherence matrix for all  $M$  microphone signals:

$$\Gamma_{ZZ}(k, l) \triangleq \mathbf{E}(k, l)\Phi_{ZZ}(k, l)\mathbf{E}^H(k, l), \quad (5)$$

with  $\mathbf{E}(k, l) =$

$$\text{diag}\left(\frac{1}{\sqrt{\Phi_{Z_1 Z_1}(k, l)}}, \frac{1}{\sqrt{\Phi_{Z_2 Z_2}(k, l)}}, \dots, \frac{1}{\sqrt{\Phi_{Z_M Z_M}(k, l)}}\right),$$

where  $\text{diag}(\cdot)$  denotes a diagonal matrix [8]. The PSDs  $\Phi_{Z_i Z_i}(k, l)$  are the main diagonal elements of the spatial correlation matrix  $\Phi_{ZZ}(k, l)$ . Especially with small microphone array apertures, we can assume equal signal energies among all microphones. Therefore  $\mathbf{E}(k, l) \approx \frac{1}{\sqrt{\Phi_S(k, l) + \Phi_N(k, l)}} \mathbf{I}_{M \times M}$ , and Eqn. (5) becomes

$$\Gamma_{ZZ}(k, l) = \Phi_{ZZ}(k, l) \frac{1}{\Phi_S(k, l) + \Phi_N(k, l)}. \quad (6)$$

Substituting Eqn. (6) into (3) gives the DD-SNR:

$$\begin{aligned}\xi_\Theta(k, l) &= \text{Tr}([\Gamma_{ZZ}(k, l) - \tilde{\mathbf{A}}(k)\tilde{\mathbf{A}}^H(k)]^{-1} \\ &\quad \cdot [\Gamma_{NN}(k) - \Gamma_{ZZ}(k, l)]),\end{aligned}\quad (7)$$

which is similar to [3]. If the direction of arrival  $\Theta$  is not known a-priori, it can be globally detected by searching over a small set of possible angles using  $\Theta_{OPT}(l) = \arg \max_{\Theta} \frac{1}{K} \sum_{k=0}^{K-1} \xi_\Theta(k, l)$ . In [3], a similar measure to the DD-SNR is used to derive a noise reduction Wiener filter. However, we found  $\xi_\Theta(k, l)$  to be too inaccurate especially for low frequencies, because in practice the ATFs won't be pure time delays, and the signal energies at the microphones won't be

equal due to gain tolerances. But we can use  $\xi_\Theta(k, l)$  to improve the model of the ATFs from simple plane waves to multi-path propagations, i.e. acoustic echos. With a good estimate  $\hat{\mathbf{A}}(k) \approx \mathbf{A}(k)$  we can align the microphone signals to either constructively or destructively add the speech components, which are used to derive the speech PSD  $\Phi_S(k, l)$  and the noise PSD  $\Phi_N(k, l)$ . From the mixture model in Eqn. (1), it can be seen that  $A_i(k)$  is generally unobservable, since it is embedded in additive noise. However, by inserting  $\xi_\Theta(k, l)$  into Eqn. (3) and using  $\hat{A}_1(k, l) \triangleq 1$  for the reference microphone we can construct the following estimator:

$$\begin{aligned}\hat{A}_i(k, l + 1) &= \hat{A}_i(k, l)\alpha_1(k, l) + (1 - \alpha_1(k, l)) \\ &\quad \cdot \left[ \frac{1 + \xi_\Theta(k, l)}{\xi_\Theta(k, l)} \frac{\Phi_{Z_i Z_1}(k, l)}{\Phi_{Z_1 Z_1}(k, l)} - \frac{\Gamma_{N_i N_1}(k)}{\xi_\Theta(k, l)} \right].\end{aligned}\quad (8)$$

To ensure this algorithm only adapts on frequency bins containing speech, we use the DD-SNR as voice activity detector:

$$\alpha_1(k, l) = \begin{cases} \alpha, & \text{if } \xi_\Theta(k, l) > \xi_0 \text{ and } \frac{1}{K} \sum_{k=0}^{K-1} \xi_\Theta(k, l) > \xi_0 \\ 0, & \text{otherwise.} \end{cases}\quad (9)$$

Clearly, estimating the ATFs only works if the represented filters have a finite impulse response which can be modeled within the duration of a FFT frame  $l$ . We chose 32ms as frame length, which is sufficient to model the acoustic path from the speaker's location to the microphone array within the speaker distance of 0.5m. Reasonable results are obtained by setting the threshold  $\xi_0$  to 0dB.

### 4. Estimation of the speech and noise PSD

We estimate the speech and noise PSDs by using a summation signal and a differential signal, which are both obtained from the ATF estimate  $\hat{\mathbf{A}}(k, l)$ . The summation signal is given by

$$\mathbf{Y}(k, l) = \mathbf{F}^H(k, l)\mathbf{Z}(k, l), \quad (10)$$

which constructively adds the speech components in  $\mathbf{Z}(k, l)$ . This is done via the matrix  $\mathbf{F}(k, l) = \frac{\hat{\mathbf{A}}(k, l)}{\|\hat{\mathbf{A}}(k, l)\|_2^2}$  [11, 12]. The differential signal is obtained with

$$\mathbf{U}(k, l) = \mathbf{B}^H(k, l)\mathbf{Z}(k, l) \approx \mathbf{B}^H(k, l)\mathbf{N}(k, l), \quad (11)$$

which destructively adds the speech components, such that  $\mathbf{B}^H(k, l)\hat{\mathbf{A}}(k, l) \triangleq \mathbf{0}$ . In effect, the speech signal is canceled out.  $\mathbf{B}(k, l)$  can be identified as a blocking matrix [11, 13] which forms a spatial zero towards the speech source [10]. A very straightforward blocking matrix is given by  $\mathbf{B}(k, l) = \mathbf{I}_{M \times M} - \hat{\mathbf{A}}(k, l)\mathbf{F}^H(k, l)$ . We used a more efficient sparse blocking matrix, which is discussed in detail in [14]. The spatial correlation matrix of the noise reference is given by

$$\begin{aligned}\Phi_{UU}(k, l) &\triangleq E\{\mathbf{U}(k, l)\mathbf{U}^H(k, l)\} \\ &\approx \mathbf{B}^H(k, l)\Gamma_{NN}(k)\mathbf{B}(k, l)\Phi_N(k, l),\end{aligned}\quad (12)$$

using Eqn. (3) and the assumptions above. An estimate of the noise PSD  $\Phi_N(k, l)$  is then obtained by:

$$\hat{\Phi}_N(k, l) = \frac{\text{Tr}(\Phi_{UU}(k, l))}{\text{Tr}(\mathbf{B}^H(k, l)\Gamma_{NN}(k)\mathbf{B}(k, l))}. \quad (13)$$

In a similar fashion, the PSD of the summation signal is given by

$$\begin{aligned}\Phi_{YY}(k, l) &\triangleq E\{\mathbf{Y}(k, l)\mathbf{Y}^H(k, l)\} \\ &= \mathbf{F}^H \mathbf{A} \mathbf{A}^H \mathbf{F} \Phi_S + \mathbf{F}^H \Gamma_{NN} \mathbf{F} \Phi_N,\end{aligned}\quad (14)$$

where we omitted the frequency and frame indices for brevity. Solving for  $\hat{\Phi}_S(k, l)$  gives an estimate of the speech PSD

$$\hat{\Phi}_S(k, l) = \max \left( \Phi_{YY} - \mathbf{F}^H \mathbf{\Gamma}_{NN} \mathbf{F} \hat{\Phi}_N, 0 \right), \quad (15)$$

with  $\mathbf{F}^H \mathbf{A} \mathbf{A}^H \mathbf{F} \approx 1$ . Following the signal model in Eqn. (3), the estimated spatial correlation matrices for the noise and speech signals are obtained by:

$$\begin{aligned} \hat{\Phi}_{NN}(k, l) &= \mathbf{\Gamma}_{NN}(k) \hat{\Phi}_N(k, l) \\ \hat{\Phi}_{SS}(k, l) &= \hat{\mathbf{A}}(k, l) \hat{\mathbf{A}}^H(k, l) \hat{\Phi}_S(k, l). \end{aligned} \quad (16)$$

Many algorithms do not estimate the speech PSD  $\hat{\Phi}_S(k, l)$  separately, since  $\hat{\Phi}_{SS}(k, l) = \hat{\Phi}_{ZZ}(k, l) - \hat{\Phi}_{NN}(k, l)$ . However, in practice this will cause over-subtraction in the PMWF. As a consequence, musical artifacts may appear in the output signal.

## 5. Parametric multi-channel Wiener filter

Having an estimate of both the noise and speech PSD matrices, a parametric multi-channel noise reduction Wiener filter [8] can be formulated as:

$$\mathbf{h}_{PMWF}(k, l) = \frac{\hat{\Phi}_{NN}^{-1}(k, l) \hat{\Phi}_{SS}(k, l) \mathbf{F}(k, l)}{\zeta(k, l) + \mu(k, l)}, \quad (17)$$

where  $\zeta(k, l) = \text{Tr} \left( \hat{\Phi}_{NN}^{-1}(k, l) \hat{\Phi}_{SS}(k, l) \right)$  can be identified as the multi-channel SNR [8]. Heuristically, we chose the trade-off parameter to be  $\mu(k, l) = \frac{1}{\zeta(k, l)}$ . For low  $\mu(k, l)$ , the PMWF is close to the MVDR filter [10, 15]. For high  $\mu(k, l)$  there is a sufficient amount of noise reduction. Finally, a MMSE estimate of the clean speech signal at the first microphone is obtained by  $X(k, l) = \mathbf{h}_{PMWF}^H(k, l) \mathbf{Z}(k, l)$ . Clearly, estimation errors in the ATFs  $\hat{\mathbf{A}}(k, l)$ , and the possible oversimplification of the noise sound field  $\mathbf{\Gamma}_{NN}(k)$  will degrade the overall performance. We found this degradation to be mainly caused by residual noise, and not by missing speech components. By inserting Eqn. (16) into (17), and using  $\mathbf{F}(k, l) = \frac{\hat{\mathbf{A}}(k, l)}{\|\hat{\mathbf{A}}(k, l)\|_2^2}$  it can be seen that:

$$\mathbf{h}_{PMWF}(k, l) = \frac{\mathbf{\Gamma}_{NN}^{-1}(k) \hat{\mathbf{A}}(k, l)}{\hat{\mathbf{A}}^H(k, l) \mathbf{\Gamma}_{NN}^{-1}(k) \hat{\mathbf{A}}(k, l)} \frac{\zeta(k, l)}{\zeta(k, l) + \mu(k, l)}. \quad (18)$$

This result can be identified as the MVDR filter [10] multiplied by a SNR-dependent gain function. Given that the MVDR filter does not distort signals defined by the ATFs, the PMWF output  $X(k, l)$  contains the same speech components as the summation signal  $Y(k, l)$  [10, 11]. We can greatly enhance the noise reduction performance by updating the speech PSD  $\hat{\Phi}_S(k, l)$  from Eqn. (15). For this update,  $\Phi_{XX}(k, l) \triangleq E\{X(k, l)X^*(k, l)\}$  is used instead of  $\Phi_{YY}(k, l)$ , so that Eqn. (15) turns into:

$$\hat{\Phi}'_S(k, l) = \max \left( \Phi_{XX} - \mathbf{F}^H \mathbf{\Gamma}_{NN} \mathbf{F} \hat{\Phi}_N, 0 \right). \quad (19)$$

The updated speech PSD is then used to iterate Eqn. (16) and (17) a second time, which removes almost all residual noise components and preserves the speech components identified in the first run.

## 6. Summary of the DD-SNR algorithm

The determination of the DD-SNR algorithm consists of three main parts: The calculation of the DD-SNR, the estimation of the ATFs, and the calculation of the PMWF. It can be summarized as follows:

1. Calculate the spatial coherence matrix  $\mathbf{\Gamma}_{ZZ}(k, l)$  using Eqn. (5) and (2).
2. Define a range for  $\Theta$  and maximize the DD-SNR  $\xi_{\Theta}(k, l)$  using Eqn. (4) and (7), and  $\Theta_{OPT}(l) = \arg \max_{\Theta} \frac{1}{K} \sum_{k=0}^K \xi_{\Theta}(k, l)$ .
3. Recursively update the ATFs  $\hat{\mathbf{A}}(k, l)$  using Eqn. (8).
4. Calculate the speech and noise PSDs using Eqn. (15) and (13).
5. Use the PMWF in Eqn. (17) to obtain the speech estimate  $X(k, l) = \mathbf{h}_{PMWF}^H(k, l) \mathbf{Z}(k, l)$ .
6. Update the speech PSD  $\hat{\Phi}'_S(k, l)$  using Eqn. (19), and iterate Eqn. (16) and (17) a second time to obtain the final result.

## 7. Performance evaluation

### 7.1. Experimental Setup

We used 2 microphones with a distance of  $d = 5\text{cm}$  in an approximately  $8 \times 5\text{m}$  wide room with a  $\text{RT60} \approx 550\text{ms}$  to record multi-channel speech and noise tracks. The speaker source was located  $0.5\text{m}$  from the array, and the noise source  $5\text{m}$ . This setup produced sufficiently exact direct and diffuse sound fields. The speech and noise tracks have been mixed together with a signal-to-interference ratio (SIR) ranging from  $-20\text{dB}$  to  $+20\text{dB}$ . To test the algorithm against a significant amount of speech data, the TIMIT [16], KCORS [17] and KCOSS [18] speech corpora are employed. For the noise data, recordings from various sources, i.e. traffic noise, industry parks, subway stations and the NOIZEUS database have been used. In total, 60 minutes of test material has been generated.

For comparison to other approaches, we implemented the aforementioned TBRR [1] and the MC-SPP [2]. To get the theoretical maximum performance, the ground truth (i.e., the true speech and noise correlation matrices  $\Phi_{SS}(k)$  and  $\Phi_{NN}(k)$ ) has been used to construct a PMWF in an additional implementation. A sampling frequency of  $f_s = 16\text{kHz}$  and a FFT size of 512 bins with 50% overlapping hanning windows is used for each algorithm. To evaluate the performance of the algorithms in terms of perceptual speech quality, the PESQ score and the PEASS toolkit [5, 6] are used. The latter explicitly aims at the psycho-acoustically motivated quality assessment of audio source separation algorithms. It delivers four scores: The *Target Perceptual Score* (TPS) measures the perceptual quality of the desired speech signal contained in the enhanced output. The *Interference Perceptual Score* (IPS) measures the influence of the residual noise components. The *Artifact Perceptual Score* (APS) measures the influence of artificial artifacts like musical noise, and the *Overall Perceptual Score* (OPS) provides a global measure of the perceptual quality. Each score ranges from 0 to 100 and large values indicate better performance.

### 7.2. Performance results

In Figure 1, a comparison in terms of the PESQ score is given. The ground truth marks the theoretical limit for the algorithms. It can be seen that the DD-SNR algorithm provides a significant improvement over the TBRR and MC-SPP algorithms, especially when the second iteration is included. For a SIR of  $+5\text{dB}$ , the DD-SNR achieves an improvement of 1.6 in mean opinion score (MOS) over MC-SPP.

Figure 2 shows the PEASS scores of the algorithms. The OPS score for the TBRR and the MCSPP are very similar. A



reason might be that both rely on the same Gaussian model for the speech presence probability. Our DD-SNR approach relies on the ATFs and the noise sound field model, and outperforms the other algorithms especially for high SIRs. The TBRR and DD-SNR show the highest TPS and IPS scores, which indicates a higher speech intelligibility and a higher amount of noise reduction. However, the TBRR seems to introduce the most artifacts, as the APS score indicates. A possible reason might be that this algorithm uses recursive noise floor averaging, which is known to introduce artifacts for instationary noises.

For demonstration, the spectrograms of the signal at the first microphone  $z_1(t)$ , the output of the DD-SNR algorithm  $x(t)$ , and the SNR  $\zeta(k, l)$  after the first and second iteration are shown in Figure 3 through 6. In this experiment, 15s of KCOSS speech data have been mixed with instationary city traffic noise with a SIR of 0dB, using the setup described above. The benefit of the second iteration of the DD-SNR algorithm can be seen by comparing Figure 5 and 6: The second iteration removes almost all residual noise while preserving the speech components.

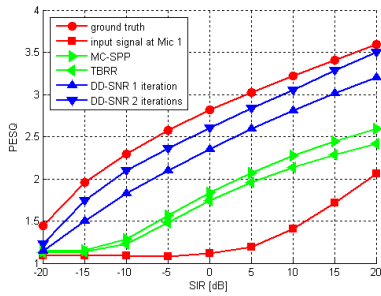


Figure 1: Comparison of the algorithms in terms of PESQ, using the MOS scale.

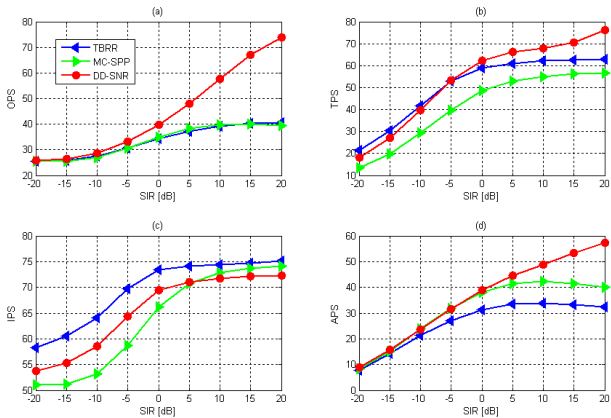


Figure 2: Comparison of the algorithms using PEASS measures; (a) OPS, (b) TPS, (c) IPS and (d) APS.

## 8. Conclusions

We proposed a multi-channel speech enhancement algorithm that blindly estimates the speech and noise PSDs based on a diffuse noise sound field and the DD-SNR. A PMWF is used to obtain a MMSE estimate of the desired clean speech signal. The overall performance is greatly increased by improving the estimate of the speech PSD using the output of the PMWF in a sec-

ond iteration. We demonstrated that the algorithm outperforms similar approaches using the PESQ and PEASS measures.

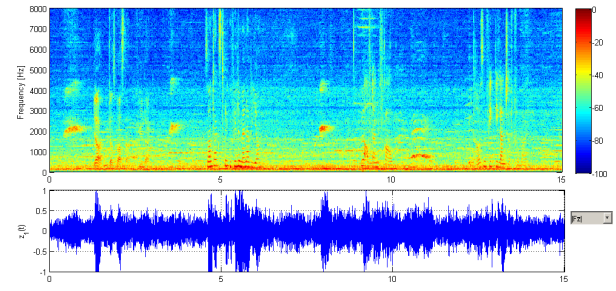


Figure 3: Received signal  $z_1(t)$  at the first microphone, containing non-stationary city traffic noise with a SIR of 0dB in a setup with  $M = 2$  microphones and  $d_{12} = 5$ cm.

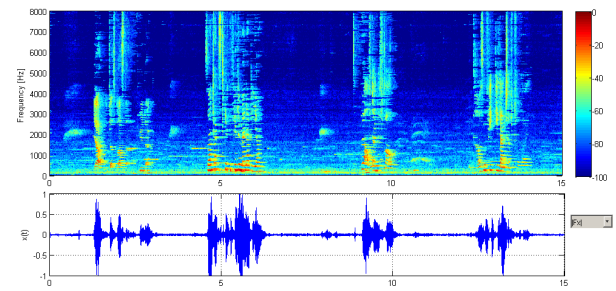


Figure 4: Output  $x(t)$  of the DD-SNR algorithm after the second iteration. The gain in SNR is limited to 30dB.

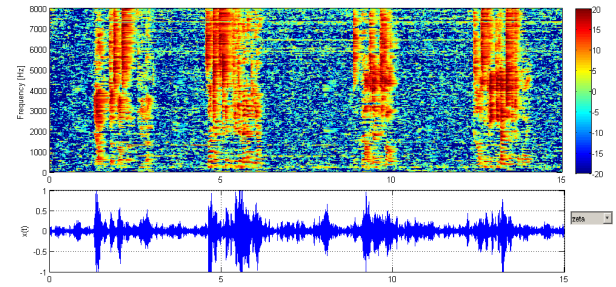


Figure 5: SNR  $\zeta(k, l)$  and output  $x(t)$  of the DD-SNR algorithm after the first iteration.

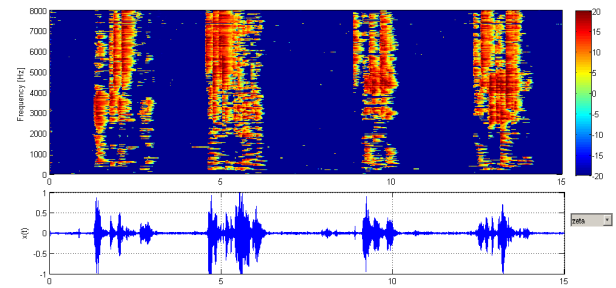


Figure 6: SNR  $\zeta(k, l)$  and output  $x(t)$  of the DD-SNR algorithm after the second iteration.

## 9. References

- [1] S. Gannot and I. Cohen, "Speech enhancement based on the general transfer function GSC and postfiltering," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 6, Nov. 2004.
- [2] M. Souden, J. Chen, J. Benesty, and S. Affes, "An integrated solution for online multichannel noise tracking and reduction," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, Sep. 2011.
- [3] M. Taseska and E. A. Habets, "MMSE-based blind source extraction in diffuse noise fields using a complex coherence-based a priori sap estimator," *International Workshop on Acoustic Signal Enhancement*, Sep. 2012.
- [4] O. Thiergart, G. D. Galdo, and E. A. P. Habets, "Signal-to-reverberant ratio estimation based on the complex spatial coherence between omnidirectional microphones," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 309–312, 2012.
- [5] E. Vincent, "Improved perceptual metrics for the evaluation of audio source separation," *10th Int. Conf. on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pp. 430–437, 2012.
- [6] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, Sep. 2011.
- [7] I. A. McCowan and H. Bourlard, "Microphone array post-filter based on noise field coherence," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 11, no. 6, Nov. 2003.
- [8] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. Berlin–Heidelberg–New York: Springer, 2006.
- [9] H. Kuttruff, *Room Acoustics*, 5th ed. London–New York: Spoon Press, 2009.
- [10] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Berlin–Heidelberg–New York: Springer, 2008.
- [11] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Berlin–Heidelberg–New York: Springer, 2008.
- [12] J. Benesty, J. Chen, Y. Huang, and J. Dmochowski, "On microphone-array beamforming from a mimo acoustic signal processing perspective," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, Mar. 2007.
- [13] I. Cohen, "Analysis of two-channel generalized sidelobe canceller (GSC) with post-filtering," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, Nov. 2003.
- [14] M. G. Shmulik, S. Gannot, and I. Cohen, "A sparse blocking matrix for multiple constraints GSC beamformer," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2012.
- [15] P. Vary and R. Martin, *Digital Speech Transmission*. West Sussex: Wiley, 2006.
- [16] "Timit acoustic-phonetic continuous speech corpus," Website, available online at <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1>; visited on January 16th 2013.
- [17] "The kiel corpus of read speech vol. 1," Website, available online at <http://www.ipds.uni-kiel.de/forschung/kielcorpus.en.html>; visited on January 16th 2013.
- [18] "The kiel corpus of spontaneous speech vol. 1-3," Website, available online at <http://www.ipds.uni-kiel.de/forschung/kielcorpus.en.html>; visited on January 16th 2013.

### **A.3 IEEE/ASRU 2015**

*"Multi-channel speech processing architectures for noise robust speech recognition: 3<sup>rd</sup> CHiME challenge results"*, Lukas Pfeifenberger, Tobias Schrank, Matthias Zöhrer, Martin Hagmüller and Franz Pernkopf, IEEE Automatic Speech Recognition and Understanding Workshop, 2015

# MULTI-CHANNEL SPEECH PROCESSING ARCHITECTURES FOR NOISE ROBUST SPEECH RECOGNITION: 3<sup>RD</sup> CHiME CHALLENGE RESULTS

Lukas Pfeifenberger, Tobias Schrank, Matthias Zöhrer, Martin Hagmüller, Franz Pernkopf

Signal Processing and Speech Communication Laboratory  
Graz University of Technology, Graz, Austria

lukas.pfeifenberger@alumni.tugraz.at,

{tobias.schrank,matthias.zoehrer,hagmueller,pernkopf}@tugraz.at

## ABSTRACT

Recognizing speech under noisy condition is an ill-posed problem. The CHiME3 challenge targets robust speech recognition in realistic environments such as street, bus, cafe and pedestrian areas. We study variants of beamformers used for pre-processing multi-channel speech recordings. In particular, we investigate three variants of generalized sidelobe canceller (GSC) beamformers, i.e. GSC with sparse blocking matrix (BM), GSC with adaptive BM (ABM), and GSC with minimum variance distortionless response (MVDR) and ABM. Furthermore, we apply several postfilters to further enhance the speech signal. We introduce MaxPower postfilters and deep neural postfilters (DPFs). DPFs outperformed our baseline systems significantly when measuring the overall perceptual score (OPS) and the perceptual evaluation of speech quality (PESQ). In particular DPFs achieved an average relative improvement of 17.54% OPS points and 18.28% in PESQ, when compared to the CHiME3 baseline. DPFs also achieved the best WER when combined with an ASR engine on simulated development and evaluation data, i.e. 8.98% and 10.82% WER. The proposed MaxPower beamformer achieved the best overall WER on CHiME3 real development and evaluation data, i.e. 14.23% and 22.12%, respectively.

**Index Terms**— multi-channel speech processing, deep postfilter, automatic speech recognition

## 1. INTRODUCTION

Background noise is the primary source of performance degradation in speech recognition systems. While the capabilities of single-channel speech pre-processing are limited, multi-channel systems exploit the spatial information of the sound field and usually achieve better speech recognition results. Adaptive beamforming is a widely used technique for multi-channel pre-processing of speech as alternative to blind source separation approaches. For a sufficient amount of noise reduction, beamformers are generally used in conjunction with a postfilter.

The aim of the 3<sup>rd</sup> CHiME challenge is to develop a multi-channel speech recognition system [1], where we encounter multi-channel recordings of a speaker located in the near-field, embedded in mostly far-field noise. The setup covers different speakers, noise environments, and real-world problems like microphone failure, clipping, and other recording glitches.

In this paper, we present a multi-channel speech enhancement system which tries to cope with these conditions: First, we detect recording glitches using the prediction error of an auto-regressive model. Then, we estimate the position of the speaker relative to the microphone array using our *direction-dependent signal-to-noise ratio* (DD-SNR) algorithm [2], which also provides a sufficiently accurate *voice activity detection* (VAD). The speaker position is used to obtain a steering vector for a *generalized sidelobe canceller* (GSC) beamformer, which we implemented in three different variants. We also present two novelties here: Firstly we introduce a *MaxPower* postfilter (PF), leading to the best speech recognition result on CHiME3 real data. Secondly we present deep neural PFs – deep neural networks attached to beamformers, improving the overall perceptual quality (OPS) of the target speech significantly and also outperforming baseline systems on simulated data. This front-end, i.e. the three beamformer variants and different PFs, are empirically evaluated using the PESQ and the OPS measures [3].

In the back-end, we use two speech recognition systems based on the Kaldi toolkit [4]. The first is a GMM system which makes extensive use of feature transformations as this was shown to provide good results for distant talk speech recognition [5]. The second is a DNN system that employs pre-training with restricted Boltzmann machines, cross entropy training and state-level minimum Bayes risk training [1]. Our best model, i.e. MaxPower PF with a GMM backend, reduces word error rate (WER) from 37.61% for the baseline enhancement system to 22.12% (41% relative improvement) on the real evaluation set.

The outline of the paper is as follows: In Section 2 we introduce the architecture of the proposed system. Section 3 de-

tails the multi-channel speech processing approaches including the proposed beamformers. PFs are introduced in Section 4 while the PESQ and PEASS scores of the front-end are summarized in Section 6.1. The ASR system is presented in Section 5 and the results are discussed in Section 6.2. Section 7 concludes the paper.

## 2. SYSTEM OVERVIEW

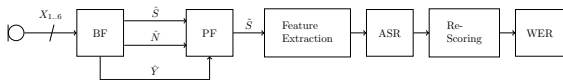


Fig. 1. System overview.

Figure 1 shows the setup of the components of the proposed ASR system. Speech estimate  $\hat{S}$ , the noise estimate  $\hat{N}$  and the beamformer output  $\hat{Y}$  are fed into a postfilter predicting an enhanced speech estimate  $\tilde{S}$ . After feature extraction the signal is fed into the ASR. Next, Language model re-scoring is applied and then the final word error rate (WER) is calculated.

## 3. MULTI-CHANNEL SPEECH PROCESSING

The input signal vector  $\mathbf{X}$  of the 6 microphone channels is written as

$$\mathbf{X}(k, l) = \mathbf{A}(k, l)S(k, l) + \mathbf{N}(k, l), \quad (1)$$

where  $S$  is the speech signal,  $N$  is the noise part of the 6-channel input signal in frequency-domain,  $k$  and  $l$  denote the frequency bin and time frame, respectively, and  $\mathbf{A}(k, l)$  denotes the *acoustic transfer function* (ATF) from the true speaker position to each microphone. In this challenge, additional information is supplied by the *noise context*, a short section of noise-only signal before each utterance. The noise context for each utterance is referenced in annotations provided by the challenge organizers. This allows to estimate the spatial noise correlation matrix  $\Phi_{NN}$ , which is given as

$$\Phi_{NN}(k, l) \triangleq E\{\mathbf{N}^H(k, l)\mathbf{N}(k, l)\}, \quad (2)$$

where  $E\{\cdot\}$  denotes the expectation operation and  $\{\cdot\}^H$  the Hermitian transpose.

We found that the noise context contains speech in some utterances, which would cause speech cancellation in a beamformer. We therefore decided to adaptively estimate  $\Phi_{NN}$  by using VAD.

### 3.1. Failed Channel Detection

The above signal model requires signals which strictly adhere to the linear time-invariant theory. Clearly, errors such

as recording glitches, amplitude variations, time shifts or total signal loss must be detected before multi-channel speech enhancement such as beamforming. In particular, we noticed that especially channel 4 and 5 exhibit rather complex recording glitches in about 15% of all isolated recordings. To address these problems, a mere energy threshold may not suffice. We therefore employed an auto-regressive linear predictive coding (LPC) on each channel  $c$  in time-domain [6, 7], and used the predictor error  $e(t)$  as criterion whether a channel is considered as failed, i.e.

$$e(t) = x_c(t) - \sum_{m=1}^M x_c(t-m)a(m), \quad (3)$$

where  $a(m)$  are LPC coefficients and  $M = 100$ . A channel  $x_c(t)$  is considered as failed if the power of its predictor error  $e(t)$  lies outside the  $\pm 10dB$  corridor around the median of the energy of the predictor errors of all channels. If a failed channel is detected this channel is not used for further processing.

### 3.2. Direction Of Arrival Estimation

For successful beamforming an accurate *direction of arrival* (DOA) estimation is required. Therefore, the *steered response power phase transform* (SRP-PHAT) [8] algorithm has been already provided for this purpose. But it lacks a proper VAD estimate, which might also be useful for estimating the spatial noise correlation matrix  $\Phi_{NN}$  during speech pauses. For this purpose, we used our DD-SNR algorithm [2], which provides a direction-dependent a-priori SNR  $\xi_\tau(k, l)$  under the assumption of an ideal, spherical noise sound field, i.e.

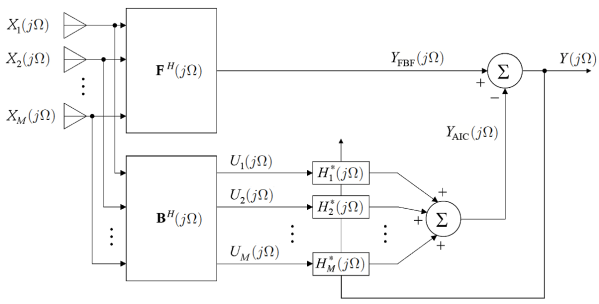
$$\xi_\tau(k, l) = \text{Tr}([\Gamma_{\mathbf{X}\mathbf{X}}(k, l) - \mathbf{A}_\tau(k, l)\mathbf{A}_\tau^H(k, l)]^{-1} \cdot [\Gamma_{\mathbf{N}\mathbf{N}}(k) - \Gamma_{\mathbf{X}\mathbf{X}}(k, l)]), \quad (4)$$

where the DD-SNR  $\xi$  is also used as VAD,  $\tau$  is the relative *time difference of arrival* (TDOA) between all microphone pairs,  $\mathbf{A}_\tau$  the corresponding ATFs,  $\Gamma_{\mathbf{X}\mathbf{X}}$  and  $\Gamma_{\mathbf{N}\mathbf{N}}$  are the spatial coherence matrices [2] for the multi-channel signals  $\mathbf{X}$  and noise-only components  $N$ . The interested reader is referred to [2] for more details.

The optimal TDOA  $\tau$  also maximizes  $\xi_\tau$ . It can be detected for each time frame  $l$  by searching over a small set of possible delays using  $\tau_{OPT}(l) = \arg \max_{\tau} \frac{1}{K} \sum_{k=0}^K \xi_\tau(k, l)$ . We quantize  $\tau$  into 13 equally spaced segments which is sufficient for each microphone pair and the given aperture.

### 3.3. Beamforming

After evaluating a wide variety of beamforming and multi-channel speech enhancement algorithms [9–13], we decided to use the *general sidelobe canceller* (GSC) [14]. The main



**Fig. 2.** Block diagram of the generalized sidelobe canceller.

reasons are its observed empirical performance and robustness for the given problem.

The entire beamformer can be expressed as

$$\mathbf{W}(k, l) = \mathbf{F}(k, l) - \mathbf{H}(k, l)\mathbf{B}(k, l) \quad (5)$$

using the *fixed beamformer* (FBF)  $\mathbf{F}$ , the *adaptive interference canceler* (AIC)  $\mathbf{H}$ , and the *blocking matrix* (BM)  $\mathbf{B}$ . In particular, we implemented the following three GSC variants detailed in the following sub-sections. Details can be found in [2, 15].

### 3.3.1. GSC with sparse BM

This variant is the standard GSC, as depicted in Figure 2. The FBF is given as  $\mathbf{F}(k, l) = \frac{\mathbf{A}(k, l)}{\mathbf{A}^H(k, l)\mathbf{A}(k, l)}$ . The BM is defined as [16]

$$\mathbf{B}(k, l) = \begin{bmatrix} -\frac{A_2^*(k, l)}{A_1^*(k, l)} & -\frac{A_3^*(k, l)}{A_1^*(k, l)} & \dots & -\frac{A_M^*(k, l)}{A_1^*(k, l)} \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}, \quad (6)$$

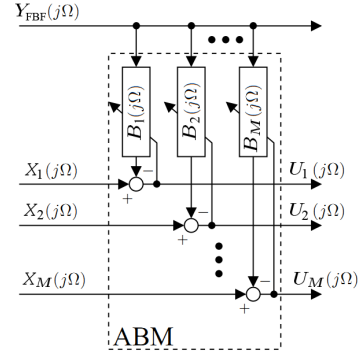
with  $M = 6$  channels, and channel 1 as *reference microphone*. The asterisk in (6) denotes the conjugate complex coefficient. We used the channel with the highest signal energy as reference in our implementations. The AIC  $\mathbf{H}$  is a non-causal adaptive filter.

### 3.3.2. GSC with adaptive Blocking Matrix (ABM)

This variant features an adaptive BM presented in Figure 3. The columns of the ABM are designed as non-causal adaptive filters and the coefficients are determined via the *normalized least mean squares* (NLMS) approach [17].

### 3.3.3. GSC with MVDR and ABM

It is possible to estimate the spatial noise correlation matrix  $\Phi_{NN}$  during speech pauses using the DD-SNR from Section 3.2 as VAD. Hence, the GSC may be replaced with the



**Fig. 3.** Block diagram of the adaptive blocking matrix.

*minimum variance distortionless response* (MVDR) solution [18, 19] given as:

$$\mathbf{F}(k, l) = \frac{\Phi_{NN}^{-1}(k, l)\mathbf{A}(k, l)}{\mathbf{A}^H(k, l)\Phi_{NN}^{-1}(k, l)\mathbf{A}(k, l)}. \quad (7)$$

This has already been provided in the baseline enhancement system, however, the estimate  $\Phi_{NN}$  may be inaccurate, therefore we only replaced the FBF in Figure 2 with the MVDR solution. This allows for additional noise removal by the ABM and AIC.

## 4. POSTFILTERING

### 4.1. MaxPower postfilter

Our first postfilter is based on the GSC with MVDR and ABM. Similar to [15], the beamformer output  $Y(k, l)$  is back-projected to the microphones using the ATFs  $\mathbf{A}(k, l)$ . This way, the microphone inputs  $\mathbf{X}$  can be split into their speech and noise components  $\hat{\mathbf{S}}$  and  $\hat{\mathbf{N}}$ :

$$\begin{aligned} \hat{\mathbf{S}}(k, l) &= \mathbf{A}(k, l)Y(k, l) \\ \hat{\mathbf{N}}(k, l) &= \mathbf{X}(k, l) - \mathbf{A}(k, l)Y(k, l) \end{aligned} \quad (8)$$

The final output of this method is chosen to be the maximum energy of  $|\hat{\mathbf{S}}(k, l)|^2$  for each frequency bin  $k$  and time frame  $l$ . As the phases of  $\hat{\mathbf{S}}(k, l)$  do not match, there would be no reconstruction back into time domain. To circumvent this limitation, each channel in  $\hat{\mathbf{S}}(k, l)$  has been aligned to the geometric origin of the setup.

### 4.2. Multi-Channel postfilter

As second postfilter we used our parametric multi-channel Wiener filter (PMWF) proposed in [2]. With the noise PSD matrix  $\Phi_{NN}$  being already available, estimating the residual noise power in the beamformer becomes straightforward.

With the beamforming filter  $\mathbf{W}$ , the residual noise power in the beamformer output is given as

$$\Phi_{Y_N Y_N}(k, l) \triangleq E\{\mathbf{W}^H(k, l)\Phi_{NN}(k, l)\mathbf{W}(k, l)\}. \quad (9)$$

Together with the overall output power of the beamformer

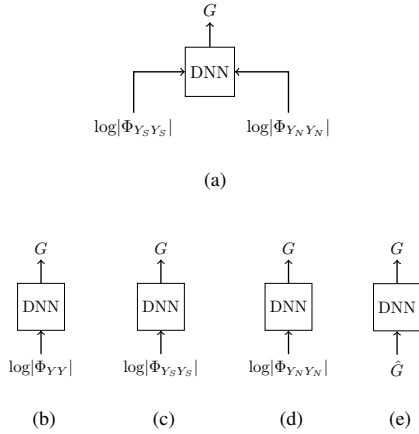
$$\Phi_{YY}(k, l) \triangleq E\{\mathbf{W}^H(k, l)\Phi_{XX}(k, l)\mathbf{W}(k, l)\} \quad (10)$$

the real-valued gain mask  $G$  is obtained as

$$G(k, l) = \frac{\zeta(k, l)}{1 + \zeta(k, l)}, \quad (11)$$

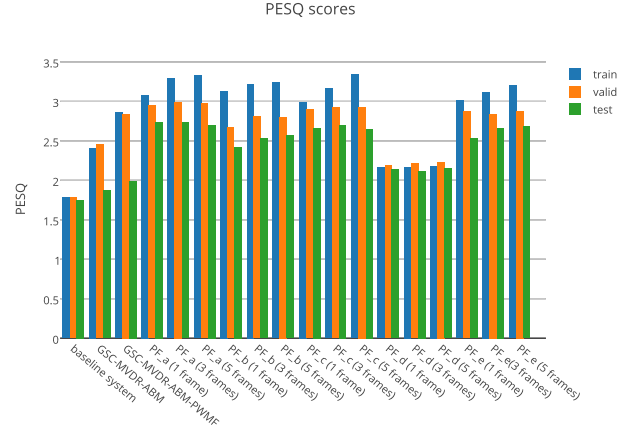
where  $\zeta(k, l) = \frac{\Phi_{YY}(k, l)}{\Phi_{Y_N Y_N}(k, l)} - 1$  can be identified as the output SNR. Further smoothing over time may be achieved using a spectral subtraction algorithm like the mean-square error log-spectral amplitude estimator [20].

### 4.3. Deep neural postfilter



**Fig. 4.** Variants of deep postfilter models. A neural network maps the beamformed speech  $\Phi_{Y_S Y_S}$ , noise  $\Phi_{Y_N Y_N}$  or estimated gain mask  $\hat{G}$  to the optimal gain mask  $G$ . The first column shows the different combinations of various beamformer components (a-d), respectively.

In [21–24] deep neural networks (DNNs) were applied to single channel source separation, improving the overall quality of speech in terms of PESQ and OPS scores. In order to analyze the enhancement capabilities of DNNs for multi-channel inputs, we introduce deep postfilter models: In particular, we use DNNs to map beamformed log-spectrogram outputs to the optimal gain mask  $G$  estimated from the close talking microphone (channel 0). Figure 4 shows variants of these postfilters using different beamformer components. In particular, model (a) uses concatenated beamformed speech log-spectrograms  $\Phi_{Y_S Y_S}$  and noise log-spectrograms  $\Phi_{Y_N Y_N}$



**Fig. 5.** PESQ scores of deep postfilter models (a-f).

as input.  $\Phi_{Y_N Y_N}$  is estimated as in (9).  $\Phi_{Y_S Y_S}$  can be calculated directly as  $\Phi_{Y_S Y_S}(k, l) = \Phi_{YY}(k, l) - \Phi_{Y_N Y_N}(k, l)$ . In case of the models (b-e)  $\Phi_{YY}$ ,  $\Phi_{Y_S Y_S}$ ,  $\Phi_{Y_N Y_N}$ , or the estimated gain mask  $\hat{G}$  were fed into the network. After training, mask estimates are applied to the output signal of the beamformer obtaining enhanced speech  $\tilde{S}$  and noise estimates.

We trained 3 layer multi-layer perceptrons [25] with rectifier activation functions using a context window of 1, 3, 5 frames and a MSE criteria on a subset of the CHiME 3 database. In particular we selected 400 utterances, 50 validation utterances and 50 test utterances from the simulated training corpus. Figure 5 and Figure 6 show the PESQ and OPS scores [3] of the postfilter (PF) models (a-e), respectively. For objective evaluation the estimated speech was compared to the output of the GSC with MVDR and ABM (with/without PMWF postfilter) and the baseline system. The best deep postfilter, i.e. PF variant a (PF<sub>a</sub>), achieved an OPS score of 71.97, a validation score of 54.03 and a test OPS of 50.83. It outperforms the beamformed signal GSC-MVDR-ABM (with/without PMWF postfilter) as well as the provided CHiME 3 baseline system. Therefore, we further investigate this approach when applied to ASR.

## 5. ASR

Both ASR systems employed in this paper are based on the baseline system provided by the 3<sup>rd</sup> CHiME challenge [1]. The GMM system uses mel frequency cepstral coefficients (MFCC) as features which are input to a series of feature-space transformations. The features are in this order transformed by applying linear discriminant analysis, maximum likelihood linear transformation and feature-space maximum likelihood linear regression. In addition, inter-speaker differences are compensated for by doing speaker-adaptive training. This pipeline proved to be highly competitive in



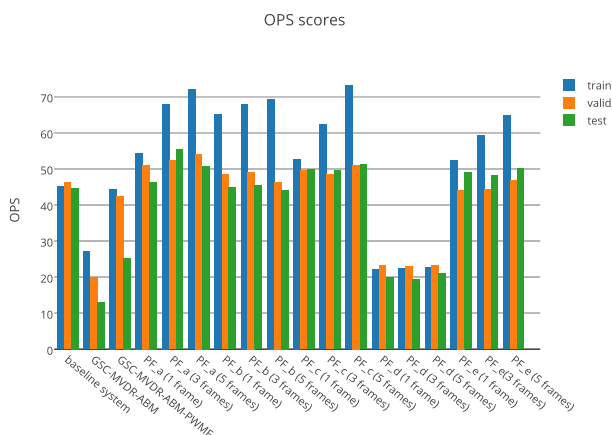


Fig. 6. OPS scores of deep postfilter models (a-f).

the CHiME 2 challenge [5]. The DNN system employs 40-dimensional filterbank features and is pre-trained using restricted Boltzmann machines with 6 hidden layers. The actual training stage of the DNN uses 4 hidden layers and also does cross entropy training. Finally, sequence discriminative training is performed using a state-level minimum Bayes risk criterion.

In the following sections, we describe the changes we made to the baseline system. These are to be found in the frontend and in the postprocessing stage.

### 5.1. Feature extraction

In contrast to the baseline which uses MFCC features, we additionally employ power-normalised cepstral coefficients (PNCC) [26]. For these features, we use a Hamming window with a window duration of 25 ms and a step size of 10 ms. Parallel to MFCCs, we extract 13 features and collect deltas and delta-deltas of these.

### 5.2. Rescoring

The postprocessing step features n-best list language model rescoring. For this, we collect the 36 best hypotheses for each utterance and reweight them with a class-based recurrent neural network language model (RNN-LM) [27]. The RNN-LM is trained on the official training data only and is configured to use a class size of 50.

## 6. RESULTS AND DISCUSSION

The data of the challenge and the recording setup is described in detail in [1]. The data is a collection of two sets of recordings: real data and simulated data. The first are speech recordings made in noisy environments. The second

are clean recordings mixed with noise that has been recorded in the same noisy environments. The real recordings were made using 6 microphones custom-fitted to a tablet hand-held device. The recordings with this device were conducted in four different environments: on a bus (BUS), in a café (CAF), in a pedestrian area (PED), and at a street junction (STR). For real data, there is an additional channel recorded with a head-mounted close-talking microphone. This channel, however, may not be used directly for obtaining ASR results but is only to be used in training.

### 6.1. Preprocessing results

To evaluate our three beamformers, we used PESQ and OPS scores. Evaluation is performed against the close-talking microphone channel for the real data set, and against the WSJ corpus for the simulated data set. Tables 1 and 2 show the scores for our four beamformers, and the baseline enhancement system for comparison. Again the GSC-MVDR with ABM and deep postfilter ( $PF_a$ ) outperforms the other beamformers in terms of OPS and PESQ scores. In particular the proposed system achieved an average relative improvement of 17.54% in OPS and 18.28% in PESQ compared to the baseline enhancement system.

	set	train	dev	eval
Baseline enhancement system	simu	2.00	1.64	1.72
	real	1.59	1.42	1.50
GSC with sparse BM, and PMWF	simu	2.15	1.73	1.81
	real	1.51	1.37	1.35
GSC with ABM, and PMWF	simu	1.53	1.49	1.52
	real	1.36	1.30	1.36
GSC with MVDR and ABM	simu	2.05	1.60	1.73
	real	1.60	1.45	<b>1.73</b>
GSC with MVDR and ABM, and $PF_a$	simu	<b>2.55</b>	<b>2.17</b>	<b>2.28</b>
	real	<b>1.73</b>	<b>1.56</b>	1.56
GSC with MVDR and ABM, and MaxPower PF	simu	1.98	1.69	1.63
	real	1.51	1.39	1.44

Table 1. PESQ scores for our beamformers with PFs and the baseline.

	set	train	dev	eval
Baseline enhancement system	simu	54.80	44.22	47.31
	real	44.66	40.98	31.48
GSC with sparse BM, and PMWF	simu	59.64	46.99	46.77
	real	38.69	33.05	29.04
GSC with ABM, and PMWF	simu	48.61	43.84	43.71
	real	43.16	42.81	<b>38.02</b>
GSC with MVDR and ABM	simu	52.4	45.87	47.18
	real	48.26	45.87	37.93
GSC with MVDR and ABM, and $PF_a$	simu	<b>63.94</b>	<b>53.83</b>	<b>54.53</b>
	real	<b>48.69</b>	<b>46.54</b>	37.72
GSC with MVDR and ABM, and MaxPower PF	simu	56.08	44.82	44.48
	real	47.18	44.90	36.96

Table 2. OPS scores for our beamformers with PFs and the baseline.



## 6.2. ASR results

Table 3 shows ASR results for the preprocessing methods presented in this paper. MaxPower outperforms all other proposed methods on the real development data and the real evaluation data (14.53% WER and 22.14% WER, respectively), whereas  $PF_a$  achieved the best ASR scores on simulated data, i.e. 8.98% and 10.82% on development and evaluation, respectively. When comparing MFCCs and PNCCs, on average, PNCCs lead to an improvement of 6.04% WER on the real evaluation set. Improvements vary, however, depending on noise environment and preprocessing. After language model rescoring, the scores for the real development set and the real evaluation set decrease slightly to 14.23% WER and 22.12% WER, respectively (see Table 4).

Due to time constraints, our results for the DNN-based ASR system are limited to MaxPower which achieves best results among GMM-based systems. While considerable improvements are gained for the system using MFCCs (−3.02% WER on real evaluation set), DNNs lead to increased WER for the system using PNCCs (+2.03% WER on real evaluation set).

	features	development		evaluation	
		real	simu	real	simu
Baseline	MFCC	20.38	9.72	37.61	11.10
GSC sparse BM	MFCC	26.14	10.39	44.01	12.75
GSC ABM	MFCC	15.66	20.15	36.39	79.05
+ MVDR	MFCC	16.78	10.16	27.45	11.47
+ $PF_a$	MFCC	17.93	<b>8.98</b>	27.72	<b>10.82</b>
+ MaxPower	MFCC	15.70	10.77	25.22	14.86
+ DNN	FBANK	14.54	9.52	22.20	15.67
Baseline	PNCC	18.99	11.14	31.57	12.15
GSC sparse BM	PNCC	22.32	11.17	36.98	13.87
GSC ABM	PNCC	15.60	21.96	34.02	77.47
+ MVDR	PNCC	16.34	11.01	24.55	12.69
+ $PF_a$	PNCC	16.77	10.64	25.58	12.37
+ MaxPower	PNCC	<b>14.53</b>	12.05	<b>22.14</b>	15.08
+ DNN	FBANK	15.79	10.42	24.17	16.72

**Table 3.** ASR results for our beamformers and the baseline enhancement system.

environment	development		evaluation	
	real	simulated	real	simulated
BUS	16.17	10.52	29.00	12.46
CAF	13.78	13.97	24.04	15.61
PED	11.73	9.53	19.75	14.81
STR	15.26	13.38	15.69	16.64
AVG	14.23	11.85	22.12	14.88

**Table 4.** Detailed results for single best system, MaxPower using PNCC features and RNN language model rescoring.

## 7. CONCLUSION

We presented a comparison of different beamformers and postfilters applied to the CHiME3 speech database. We studied three variants of GSC beamformers, i.e. GSC with sparse blocking matrix (BM), GSC with adaptive BM (ABM), and GSC with minimum variance distortionless response (MVDR) and ABM. In addition we investigated three postfilters (PF), a MaxPower PF, a parametric multi-channel Wiener filter, and a deep neural PF. The proposed ASR systems use either MFCC or PNCC features calculated from the pre-processed signals which are fed into GMM or DNN-based systems. Finally n-best list re-scoring, using a recurrent neural network (RNN) language model, was applied.

We evaluated the overall perceptual score (OPS), and perceptual evaluation of speech quality (PESQ) of the proposed beamformers and postfilters. Deep neural postfilters using an GSC-MVDR-ABM beamformer outperformed other BF systems significantly, achieving an average relative improvement of 17.54% in OPS and 18.28% in PESQ compared to the baseline system. However, improvements in OPS were not reflected in the ASR performance on the real data set, although the best scores were achieved on the simulated data. The GSC-MVDR-ABM beamformer followed by the MaxPower postfilter and GMM ASR achieved the best WER on real data. This configuration obtained a 22.14% WER and a 22.12% WER on the real evaluation set, with or without re-scoring, respectively.

## 8. ACKNOWLEDGEMENTS

This work was supported by the Austrian Science Fund (FWF) under the project number P27803-N15, and the K-Project ASD. The K-Project ASD is funded in the context of COMET Competence Centers for Excellent Technologies by BMVIT, BMWFJ, Styrian Business Promotion Agency (SFG), the Province of Styria – Government of Styria and the Technology Agency of the City of Vienna (ZIT). The program COMET is conducted by Austrian Research Promotion Agency (FFG). Furthermore, we acknowledge NVIDIA for providing GPU computing resources.

## 9. REFERENCES

- [1] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, “The third ‘CHiME’ speech separation and recognition challenge: Dataset, task and baselines,” in *IEEE 2015 Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015, submitted.
- [2] L. Pfeifenberger and F. Pernkopf, “Blind source extraction based on a direction-dependent a-priori SNR,” in *Interspeech*, 2014.

- [3] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, 2011.
- [4] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. 2011, IEEE Signal Processing Society.
- [5] Y. Tachioka, S. Watanabe, J. Le Roux, and J. R. Hershey, "Discriminative methods for noise robust speech recognition: A chime challenge benchmark," in *Proceedings of the 2nd International Workshop on Machine Listening in Multisource Environments (CHiME)*, 2013, pp. 19–24.
- [6] T. D. Rossing, *Springer Handbook of Acoustics*, Springer, Berlin–Heidelberg–New York, 2007.
- [7] P. Vary and R. Martin, *Digital Speech Transmission*, Wiley, West Sussex, 2006.
- [8] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*, Springer, Berlin–Heidelberg–New York, 2008.
- [9] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, 2007.
- [10] R. Talmon, I. Cohen, and S. Gannot, "Relative transfer function identification using convolutive transfer function approximation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, 2009.
- [11] W. Herbordt and W. Kellermann, "Analysis of blocking matrices for generalized sidelobe cancellers for non-stationary broadband signals," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 2002.
- [12] E. Warsitz, A. Krueger, and R. Haeb-Umbach, "Speech enhancement with a new generalized eigenvector blocking matrix for application in a generalized sidelobe canceller," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 73–76, 2008.
- [13] M. Souden, J. Chen, J. Benesty, and S. Affes, "An integrated solution for online multichannel noise tracking and reduction," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, 2011.
- [14] O. Hoshuyama, A. Sugiyama, and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, 1999.
- [15] L. Pfeifenberger and F. Pernkopf, "A multi-channel postfilter based on the diffuse noise sound field," in *European Association for Signal Processing Conference*, 2014.
- [16] M. G. Shmulik, S. Gannot, and I. Cohen, "A sparse blocking matrix for multiple constraints GSC beamformer," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012.
- [17] J. Li, Q. Fu, and Y. Yan, "An approach of adaptive blocking matrix based on frequency domain independent component analysis in generalized sidelobe canceller," *IEEE 10th International Conference on Signal Processing*, pp. 231–234, 2010.
- [18] K. Lae-Hoon, M. Hasegawa-Johnson, and S. Koeng-Mo, "Generalized optimal multi-microphone speech enhancement using sequential minimum variance distortionless response(MVDR) beamforming and postfiltering," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, 2006.
- [19] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*, Springer, Berlin–Heidelberg–New York, 2008.
- [20] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 2, 1985.
- [21] M. Zöhrer and F. Pernkopf, "Representation models in single channel source separation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2015.
- [22] M. Zöhrer and F. Pernkopf, "Single channel source separation with general stochastic networks," in *Inter-speech*, 2014.
- [23] M. Zöhrer, R. Peharz, and F. Pernkopf, "Representation learning for single-channel source separation and bandwidth extension," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 2015, accepted.
- [24] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

- [25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” in *Neurocomputing: Foundations of Research*, James A. Anderson and Edward Rosenfeld, Eds., pp. 673–695. MIT Press, Cambridge, MA, USA, 1988.
- [26] C. Kim and R. M. Stern, “Power-normalized cepstral coefficients (PNCC) for robust speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 4101–4104.
- [27] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, “Recurrent neural network based language model,” in *INTERSPEECH*, 2010.

## A.4 CHiME4 2016

*"Deep beamforming and data augmentation for robust speech recognition: Results of the 4<sup>th</sup> CHiME challenge"*, Tobias Schrank, Lukas Pfeifenberger and Matthias Zöhrer, Johannes Stahl, Pejman Mowlaei, Franz Pernkopf, 4th International Workshop on Speech Processing in Everyday Environments, San Francisco, 2016

# Deep Beamforming and Data Augmentation for Robust Speech Recognition: Results of the 4<sup>th</sup> CHiME Challenge

Tobias Schrank, Lukas Pfeifenberger, Matthias Zöhrer, Johannes Stahl, Pejman Mowlae, Franz Pernkopf

Signal Processing and Speech Communication Laboratory  
Graz University of Technology, Graz, Austria

lukas.pfeifenberger@alumni.tugraz.at,

{tobias.schrank,matthias.zoehrer,johannes.stahl,pejman.mowlae,pernkopf}@tugraz.at

## Abstract

Robust automatic speech recognition in adverse environments is a challenging task. We address the 4<sup>th</sup> CHiME challenge multi-channel tracks by proposing a deep eigenvector beamformer as front-end. To train the acoustic models, we propose to supplement the beamformed data by the noisy audio streams of the individual microphones provided in the real set. Furthermore, we perform data augmentation by modulating the amplitude and time-scale of the audio. Our proposed system achieves a word error rate of 4.22% on the real development and 8.98% on the real evaluation data for 6-channels and 6.45% and 13.69% for 2-channels, respectively.

## 1. Background

This report describes our proposed ASR system for the 6- and 2-channel task of the 4<sup>th</sup> CHiME challenge. The proposed modifications of the baseline system are:

- As multi-channel front-end we employ an optimal multi-channel Wiener filter, which consists of an eigenvector GSC beamformer and a single-channel postfilter. Both components depend on a speech presence probability mask, which we learn using a deep neural network (DNN).
- In addition to the beamformed signals we use noisy multi-channel real data to train the acoustic model of the ASR, i.e. we perform *multi-channel* training.
- We perform data augmentation by modulating the signal amplitude (volume perturbation) and time-scale modifications (speed perturbation).
- We perform sequential language model rescoring using (gated) RNNs.
- We combine multiple systems with a lattice-based approach which uses minimum Bayes risk decoding.

A detailed introduction of the individual components and relevant literature are provided in the next section.

## 2. Robust Multi-Channel ASR System

Figure 1 shows the block diagram of the proposed multi-channel ASR system including the data augmentation and multi-channel training of the recognizer. Each processing step is detailed in the following sections.

This work was supported by the LEAD project, the Austrian Science Fund (FWF) under the project number P25244-N15 and P27803-N15 and the K-Project ASD. Furthermore, we acknowledge NVIDIA for providing GPU computing resources.

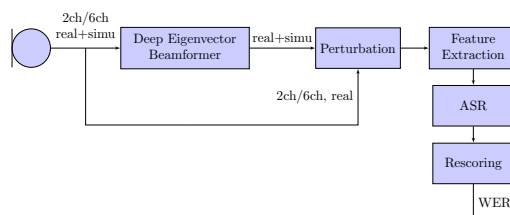


Figure 1: System overview.

### 2.1. Deep Eigenvector Beamformer

As multi-channel speech enhancement front-end we employ a *deep eigenvector beamformer*, which consists of a generalized sidelobe canceller (GSC) beamformer [1–5], followed by a single-channel postfilter. The GSC consists of a steering vector  $\mathbf{F}$ , a blocking matrix  $\mathbf{B}$ , and an adaptive interference canceller, such that:  $\mathbf{W} = \mathbf{F} - \mathbf{B}\mathbf{H}_{AIC}$ . The GSC block diagram is given in figure 2. The steering vector  $\mathbf{F}$  has to model the *acoustic transfer functions* (ATFs) from the speaker to the microphones [6]. Usually this is done by a *direction of arrival* (DOA) estimation. However, this method does not include the complex propagation paths present in the CHiME4 data, which is why we use the dominant eigenvector of the speech PSD matrix  $\hat{\Phi}_{SS}$  as steering vector  $\mathbf{F}$ , such that the beamformer is directed towards the speech source in signal subspace. This allows the beamformer to account for early echoes and reverberation of the speaker signal [6–8]. Hence, we refer to this beamformer as *eigenvector GSC* (EV-GSC).

Using the steering vector  $\mathbf{F}$ , the blocking matrix is given as  $\mathbf{B} = \mathbf{I} - \mathbf{F}\mathbf{F}^H$ . The adaptive interference canceller  $\mathbf{H}_{AIC}$  is learned using an adaptive NLMS filter [9]. The single-channel postfilter consists of a real-valued gain mask  $G = \frac{\xi}{1+\xi}$ , which is obtained from the SNR  $\xi$  at the beamformer output. It is given as  $\xi = \frac{\mathbf{W}^H \hat{\Phi}_{SS} \mathbf{W}}{\mathbf{W}^H \hat{\Phi}_{NN} \mathbf{W}}$ . The SNR depends on both the speech and noise PSD matrices, which are estimated using a time and frequency dependent *speech presence probability*  $p_{SPP}$ .

We use a DNN to learn  $p_{SPP}$  from the dominant eigenvector of the PSD matrix of the noisy inputs. As we are operating in the frequency domain, each frequency bin  $k$  is assigned to a kernel as shown in Figure 3. The feature vector  $\mathbf{x}_k$  for each kernel consists of the cosine distance between the eigenvectors of 5 consecutive frames. This introduces some context-

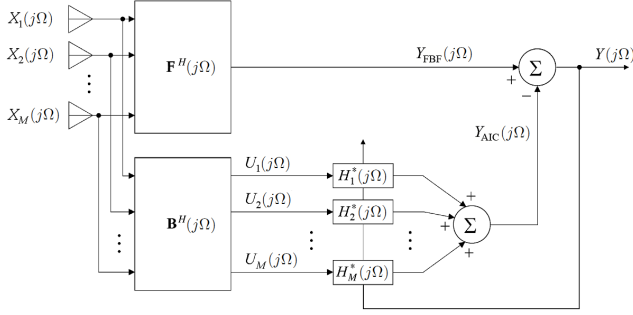


Figure 2: GSC beamformer

sensitivity into our model. The DNN of each kernel uses a hybrid model with a generative and a discriminative component [10]. The generative component consists of two autoencoder layers, which perform unsupervised clustering of the input data  $\mathbf{x}_k$ . The autoencoder kernels operate independently for each frequency bin. We used 20 neurons in the first layer, and 10 neurons in the second layer. The discriminative component consists of a regression layer which fuses the activations of all autoencoder kernels, in order to exploit information which is distributed across the frequency. The regression layer predicts the  $K$  output labels  $p_{SPP}(\mathbf{x}_k)$ . Figure 3 illustrates the kernelized DNN used in our system.

For more details on the EV-GSC beamformer and the kernelized DNN, we refer the reader to [11]. We use the same architecture for the 2ch and 6ch track, as the training data is the same for both tracks.

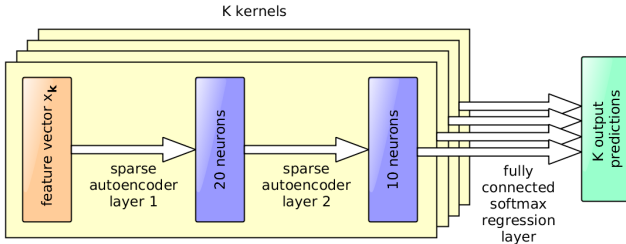


Figure 3: Kernelized DNN to estimate the speech presence probability  $p_{SPP}$

## 2.2. ASR

The ASR system employs a hybrid DNN architecture built with the Kaldi toolkit [12]. We do not only use the beamformed data for training but add the noisy channels of the real data (except for channel 2 which faces backwards). With this *multi-channel training (MC)* we can both compensate for the small amount of training data and make the acoustic model less sensitive to noise that might be left over in the evaluation data. In the evaluation stage we still use only the beamformed signals.

The GMM system uses 13 MFCCs and their deltas and delta-deltas. The DNN uses 40 fMLLR features extracted from this GMM system. For the DNN the data is augmented with speed-perturbed copies of the original data. Additionally, the data is volume-perturbed for greater robustness (*pert*). The DNN is then generatively pre-trained using restricted Boltzmann machines. The DNN has 6 hidden layers and is trained

with a state-level minimum Bayes risk (sMBR) criterion. The results which have been obtained in this way are then rescored with a Kneser-Ney smoothed 5-gram model (*5-gram*), a recurrent neural network language model (RNNLM) and a gated RNNLM (*GRNNLM*). The two RNNLMs consist of a single hidden layer with 300 and 500 neural units, respectively.

We perform system combination by first combining the lattices of the system with perturbed training data (*pert*), the system with multi-channel training (*MC*) and the system that uses both (*MC + pert*). We then decode the resulting lattices with an sMBR criterion.

## 3. Experimental Evaluation

Table 1 shows the results of our systems for the 6- and 2-channel tasks of the 4<sup>th</sup> CHiME challenge. For each data set the best score for a single system and for system combination is in bold-face. Due to time constraints we report only those results for the 2-channel task which uses the system architecture that we have found to be optimal for the 6-channel task ( $S_C$ ). Therefore the following comparison focuses on the 6-channel task.

On average over the test sets, our proposed EV-GSC beamformer of  $S_2$  performs 2% WER better than the baseline *beamformIt* beamformer of  $S_1$ , i.e. 7.95% WER vs. 9.98% WER for the RNNLM-rescored DNN. However, this performance improvement is the least pronounced for the real evaluation data. Data augmentation through speed perturbation and volume perturbation (*pert*) of  $S_3$  results in an improvement of .74% WER on average, i.e. 7.20% WER vs. 7.95% WER. Multi-channel (MC) training of  $S_4$  leads to an improvement of 0.80% WER on average, i.e. 7.15% WER vs. 7.95% WER. Both multi-channel training and amplitude and time-scale perturbation (MC+*pert*) of  $S_5$  results in an improvement of 1.19% WER on average, i.e. 6.75% WER vs. 7.95% WER. Further rescoreing with the gated RNNLM leads to a small improvement of 0.04% WER. The best results for 6-channels are achieved by a combination of systems  $S_3$ ,  $S_4$ , and  $S_5$  as  $S_6$ . In particular, we obtain a WER of 8.98% and 7.02% on the real and simulated test set, respectively.

Table 2 shows the individual results for each environment of our best system for the 6- and 2-channel track. For both systems, performance on the real evaluation data set is considerably worse for BUS than for any other environment.

## 4. References

- [1] O. Hoshuyama, A. Sugiyama, and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, Oct. 1999.
- [2] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Berlin-Heidelberg-New York: Springer, 2008.
- [3] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Transactions on Signal Processing*, vol. 49, no. 8, Aug. 2001.
- [4] W. Herbordt and W. Kellermann, "Analysis of blocking matrices for generalized sidelobe cancellers for non-stationary broadband signals," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May 2002.
- [5] S. Gannot and I. Cohen, "Speech enhancement based on the general transfer function GSC and postfiltering," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 6, Nov. 2004.
- [6] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Berlin-Heidelberg-New York: Springer, 2008.

Table 1: Average WER (%) for the tested systems.

Track	System				Dev		Test	
	Tag	ASR	Data	BF	real	simu	real	simu
2ch	S <sub>A</sub>	GMM	–	EV-GSC	14.16	15.13	26.33	24.12
	S <sub>B</sub>	GMM	MC	EV-GSC	13.41	15.36	23.46	23.49
	S <sub>C</sub>	DNN	MC + pert	EV-GSC	9.38	11.33	17.92	18.10
		+sMBR			9.24	10.91	17.16	17.46
		+5-gram			7.63	9.60	15.29	15.81
+RNNLM		6.66			8.54	14.02	14.46	
+GRNNLM	<b>6.45</b>	<b>8.29</b>			<b>13.69</b>	<b>14.33</b>		
6ch	S1	GMM		beamformit	12.78	14.87	23.13	23.06
		DNN			9.59	11.08	17.89	17.48
		+sMBR			8.58	9.90	15.88	16.01
		+5-gram			7.02	8.43	14.12	13.94
		+RNNLM			6.45	7.83	12.79	12.86
	S2	GMM		EV-GSC	11.21	11.92	23.41	16.13
		DNN			8.32	8.32	17.36	11.75
		+sMBR			7.37	7.52	15.55	10.83
		+5-gram			6.01	6.14	14.05	9.35
		+RNNLM			5.14	5.48	12.60	8.56
	S3	DNN	pert	EV-GSC	7.82	7.96	16.13	11.01
		+sMBR			6.83	6.86	14.34	10.16
		+5-gram			5.66	5.76	12.78	8.70
		+RNNLM			4.71	5.13	11.53	7.44
		+GRNNLM			4.74	<b>5.05</b>	11.45	<b>7.34</b>
	S4	GMM	MC	EV-GSC	11.05	11.77	19.65	15.93
		DNN			8.15	7.94	14.38	11.37
		+sMBR			7.30	7.49	13.38	10.56
		+5-gram			5.82	6.17	11.55	9.51
		+RNNLM			4.96	5.27	10.23	8.14
	S5	DNN	MC + pert	EV-GSC	7.65	8.03	13.53	10.89
		+sMBR			6.81	7.24	12.50	10.01
		+5-gram			5.53	6.08	10.94	8.57
		+RNNLM			<b>4.65</b>	5.35	9.63	7.38
+GRNNLM		4.66			5.28	<b>9.54</b>	7.38	
S6	combination		EV-GSC	<b>4.22</b>	<b>4.73</b>	<b>8.98</b>	<b>7.02</b>	

Table 2: WER (%) per environment for the best system.

Track	Envir.	Dev		Test	
		real	simu	real	simu
2ch	BUS	8.35	7.24	19.46	9.28
	CAF	5.78	10.80	13.41	16.92
	PED	4.23	5.86	12.07	15.00
	STR	7.45	9.25	9.81	16.12
6ch	BUS	5.25	3.79	13.72	4.20
	CAF	3.98	5.99	7.12	7.73
	PED	2.79	3.58	7.31	8.29
	STR	4.85	5.56	7.79	7.86

- [7] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. Berlin–Heidelberg–New York: Springer, 2006.
- [8] E. Warsitz and R. Haeb-Umbach, “Blind acoustic beamforming based on generalized eigenvalue decomposition,” *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, Jul. 2007.
- [9] P. Vary and R. Martin, *Digital Speech Transmission*. West Sussex: Wiley, 2006.
- [10] M. Zöhrer, R. Peharz, and F. Pernkopf, “Representation learning for single-channel source separation and bandwidth extension,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2398–2409, 2015.
- [11] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, “Dnn-based speech mask estimation for eigenvector beamforming,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017, submitted.
- [12] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The kaldi speech recognition toolkit,” in *ASRU*, 2011.

## A.5 IEEE/ICASSP 2017

*"DNN-based speech mask estimation for eigenvector beamforming"*, Lukas Pfeifenberger, Matthias Zöhner and Franz Pernkopf, The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, 2017



# DNN-BASED SPEECH MASK ESTIMATION FOR EIGENVECTOR BEAMFORMING

Lukas Pfeifenberger<sup>1</sup>, Matthias Zöhrer<sup>1</sup>, Franz Pernkopf<sup>1</sup>

<sup>1</sup> Signal Processing and Speech Communication Laboratory  
Graz University of Technology, Graz, Austria

lukas.pfeifenberger@alumni.tugraz.at  
{matthias.zoehrer,pernkopf}@tugraz.at

## ABSTRACT

In this paper, we present an optimal multi-channel Wiener filter, which consists of an eigenvector beamformer and a single-channel postfilter. We show that both components solely depend on a speech presence probability, which we learn using a deep neural network, consisting of a deep autoencoder and a softmax regression layer. To prevent the DNN from learning specific speaker and noise types, we do not use the signal energy as input feature, but rather the cosine distance between the dominant eigenvectors of consecutive frames of the power spectral density of the noisy speech signal. We compare our system against the BeamformIt toolkit, and state-of-the-art approaches such as the front-end of the best system of the CHiME3 challenge. We show that our system yields superior results, both in terms of perceptual speech quality and classification error.

**Index Terms**— multi-channel speech enhancement, eigenvector beamforming, speech mask estimation

## 1. INTRODUCTION

In recent years, conventional single channel speech enhancement methods have been outperformed by data-driven approaches. *deep neural networks* (DNNs) have been employed to discriminatively learn a gain mask for separation of the speech and noise components in a noisy speech signal [1–5].

For multi-channel speech enhancement, acoustic beamforming still outperforms single-channel methods due to the underlying physical model that can be exploited [6]. However, DNNs have proven to be useful for learning a postfilter subsequent to a beamformer [7]. The *generalized sidelobe canceller* (GSC) is one of the most popular beamformer designs. It requires an estimate of either the *direction of arrival* (DOA) or the *acoustic transfer function* (ATF) from the speech source to the microphones, which is then used as steering vector [6]. For DOA estimation, the geometry of the microphone array has to be known, while ATF estimation requires knowledge of the statistics of the speech signal. More advanced beamforming techniques require an estimate of the *power spectral density* (PSD) matrix of the noise signal [8].

In this paper, we first show that the speech presence probability mask for estimating the speech and noise statistics is sufficient to construct an optimal multi-channel Wiener filter, consisting of an *eigenvector GSC* (EV-GSC) and a single-channel postfilter. Recently, various works have been presented on how to obtain the speech presence probability using neural networks, e.g. [1, 3, 9]. Most methods rely on the energy of the noisy speech signals, and therefore are highly dependent on the array geometry and the statistics of the speech and noise presented in the training data. We aim

to use a more general approach, which requires as little assumptions about the signals as possible: We only assume that the speaker is *moving slowly*, and that the noise is *non-stationary*. We empirically observed that the eigenvectors of the PSD matrix of the noisy speech signals provide a good measure for speaker activity, independent of signal energy and array geometry. Based on this observation, we estimate the speech presence probability mask using a simple DNN structure consisting of a deep autoencoder with a softmax regression layer. The deep autoencoder learns a sparse representation of the eigenvectors of the PSD matrix of the noisy speech signals for each frequency bin. The softmax regression layer discriminatively maps this representation to the speech presence probability mask. We empirically compare our multi-channel speech enhancement system to three state-of-the-art approaches: The BeamformIt-toolkit [10], a GSC with steering vector estimation and an *adaptive blocking matrix* (ABM) [7], and the front-end of the best CHiME3 system [11], which uses a complex Gaussian mixture model (CGMM-EM) to estimate the speech and noise statistics.

This paper is structured as follows: After the introduction of the system model in Section 2 we show the importance of the speech presence probability for constructing an optimal multi-channel Wiener filter in Section 3. In Section 4 the estimation of the speech presence probability is presented. In Section 5 we evaluate our model on CHiME4 data. Section 6 concludes the paper.

## 2. SYSTEM MODEL

We use the CHiME4 setup [10], which provides multi-channel recordings of a single speaker embedded into ambient noise. The recordings have been made with  $M = 6$  microphones mounted to a tablet computer. Both real and simulated data is provided, as well as a ground truth (i.e. speaker separated from noise). This allows to evaluate the performance of our system based on the true speech signal. According to this scenario, the signal model is given as

$$\mathbf{Z}(k, l) = \mathbf{S}(k, l) + \mathbf{N}(k, l), \quad (1)$$

where  $\mathbf{Z}(k, l)$  denotes the  $M$ -channel recordings in the frequency domain, stacked to a  $M \times 1$  vector at frequency bin  $k = 1, \dots, K$  and time frame  $l$ .  $\mathbf{S}(k, l)$  and  $\mathbf{N}(k, l)$  denote the separated multi-channel speech and noise components.<sup>1</sup> For uncorrelated speech and noise signals, the PSD matrix of the input is given as

$$\Phi_{ZZ} = \Phi_{SS} + \Phi_{NN}. \quad (2)$$

<sup>1</sup>For enhanced readability, the frequency and time frame indices will be omitted except where necessary.

Since  $\Phi_{SS}$  contains a single speech source, it can be decomposed into the speech PSD  $\Phi_S$  and the *acoustic transfer functions* (ATFs)  $\mathbf{A}$  from the speaker to the microphones [12], i.e.

$$\Phi_{SS} = \mathbf{A}\mathbf{A}^H\Phi_S. \quad (3)$$

### 3. MULTI-CHANNEL SPEECH ENHANCEMENT

The MSE-optimal multi-channel Wiener filter for estimating the single speaker from the inputs  $\mathbf{Z}(k, l)$  is given as [13, 14]

$$\begin{aligned} \mathbf{W}_{OPT} &= \Phi_{ZZ}^{-1}\Phi_{ZS} \\ &= [\mathbf{A}\mathbf{A}^H\Phi_S + \Phi_{NN}]^{-1}\Phi_S\mathbf{A} \\ &= \underbrace{\Phi_{NN}^{-1}\mathbf{A}}_{\mathbf{W}_{MVDR}} \cdot \underbrace{\Phi_S}_{G=\frac{\xi}{1+\xi}} [\mathbf{A}^H\Phi_{NN}^{-1}\mathbf{A}]^{-1}, \end{aligned} \quad (4)$$

where  $\Phi_{ZS}$  is the cross-PSD of  $\mathbf{Z}(k, l)$  and  $\mathbf{S}(k, l)$ , and the vector  $\mathbf{W}_{MVDR}$  can be recognized as the MVDR beamformer.  $G$  depicts a real-valued, single-channel gain mask. From (4),  $\xi$  is given as

$$\xi = \Phi_S\mathbf{A}^H\Phi_{NN}^{-1}\mathbf{A} \quad (5)$$

which can be recognized as the SNR at the beamformer output, i.e.

$$\xi = \frac{\mathbf{W}_{MVDR}^H\Phi_{SS}\mathbf{W}_{MVDR}}{\mathbf{W}_{MVDR}^H\Phi_{NN}\mathbf{W}_{MVDR}}. \quad (6)$$

#### 3.1. Eigenvector Beamforming

In real-world applications, both the ATFs  $\mathbf{A}$  and the noise PSD matrix  $\Phi_{NN}$  are hard to estimate. The latter might even be ill-conditioned and therefore not invertible. As a consequence, the MVDR beamformer in (4) is difficult to be implemented. Instead, the GSC is widely used [6, 15–18]. The GSC consists of a *steering vector*  $\mathbf{F}$ , a *blocking matrix*  $\mathbf{B}$ , and an *adaptive interference canceller*  $\mathbf{H}_{AIC}$ , i.e.

$$\mathbf{W}_{MVDR} \approx \mathbf{W}_{GSC} = \mathbf{F} - \mathbf{B}\mathbf{H}_{AIC}. \quad (7)$$

While the GSC avoids the inversion of  $\Phi_{NN}$ , the steering vector  $\mathbf{F}$  is still a crucial component, as it directs the beamformer into the direction of the desired speech signal. Obviously, the optimal steering vector would be the ATFs  $\mathbf{A}$ , but they are unknown and hard to estimate in reverberant environments [6]. Eigenvalue decomposition of (3) yields

$$\Phi_{SS} = \mathbf{v}_S\mathbf{v}_S^H\lambda_S = \mathbf{A}\mathbf{A}^H\Phi_S, \quad (8)$$

where  $\lambda_S$  and  $\mathbf{v}_S$  are the principal eigenvalue<sup>2</sup> and eigenvector of  $\Phi_{SS}$ , respectively. It can be seen that  $\mathbf{v}_S$  points towards the speech source. The eigenvector includes reverberations and early echoes of the target signal, hence it qualifies as a substitute for the unknown ATFs  $\mathbf{A}$ , and can be used as steering vector  $\mathbf{F}$ . This concept is known as *eigenvector* or *subspace* beamforming [12, 19] where

$$\mathbf{F} := \mathbf{v}_S. \quad (9)$$

However,  $\Phi_{SS}$  cannot be directly observed, but for the purpose of eigenvector decomposition it can be approximated using

$$\hat{\Phi}_{SS}(k, l) = \frac{\sum_{t=1}^T \mathbf{Z}(k, t)\mathbf{Z}^H(k, t)p_{SPP}(k, t)}{\sum_{t=1}^T p_{SPP}(k, t)}, \quad (10)$$

<sup>2</sup>Note that  $\Phi_{SS}$  is of rank 1 for a single speaker, see (3).

where  $p_{SPP}$  is the *speech presence probability* ( $0 \leq p_{SPP} \leq 1$ ), and  $T$  is a number of frames during which the dominant eigenvector  $\mathbf{v}_S$  is assumed to be constant, i.e. the speaker is not moving. Intuitively, using (9), a blocking matrix which satisfies  $\mathbf{B}^H\mathbf{A} \stackrel{\dagger}{=} \mathbf{0}_{1 \times M}$  is then given by

$$\mathbf{B} = \mathbf{I} - \mathbf{F}\mathbf{F}^H = \mathbf{I} - \mathbf{v}_S\mathbf{v}_S^H, \quad (11)$$

where  $\mathbf{I}$  is the  $M \times M$  identity matrix. A similar concept is also used in [20]. The adaptive interference canceller  $\mathbf{H}_{AIC}$  is usually implemented using an adaptive normalized least squares (NLMS) filter [21]. Adaption of this filter has to be stopped while the speaker is active, otherwise target cancellation occurs. Usually this is done using voice activity detection (VAD). However, we used a state-space model [22] to adapt  $\mathbf{H}_{AIC}$ , which does not require a VAD.

Note that the steering vector  $\mathbf{F}$  and the blocking matrix  $\mathbf{B}$  depend on the dominant eigenvector  $\mathbf{v}_S$ , hence we refer to this beamformer as *eigenvector GSC* (EV-GSC). Furthermore,  $\mathbf{v}_S$  depends on the speech presence probability  $p_{SPP}$ , see (10). Therefore, the performance of the beamformer depends on an accurate estimate of  $p_{SPP}$ .

#### 3.2. Optimal Postfilter

Analogously to (10), the noise PSD matrix  $\Phi_{NN}(k, l)$  can be approximated as

$$\hat{\Phi}_{NN}(k, l) = \frac{\sum_{t=1}^T \mathbf{Z}(k, t)\mathbf{Z}^H(k, t)(1 - p_{SPP}(k, t))}{\sum_{t=1}^T (1 - p_{SPP}(k, t))}. \quad (12)$$

Using (6), the SNR at the beamformer output is

$$\xi = \frac{\mathbf{W}_{GSC}^H\hat{\Phi}_{SS}\mathbf{W}_{GSC}}{\mathbf{W}_{GSC}^H\hat{\Phi}_{NN}\mathbf{W}_{GSC}} \quad (13)$$

and the postfilter from (4) is given as  $G = \frac{\xi}{1+\xi}$ . Similar as for the beamformer, the postfilter solely depends on the speech presence probability  $p_{SPP}$ .

### 4. SPEECH MASK ESTIMATION

As demonstrated above, the speech presence probability  $p_{SPP}$  is sufficient to construct an optimal multi-channel Wiener filter consisting of our EV-GSC and a postfilter. Therefore, the estimation of  $p_{SPP}$  is the key component of our multi-channel speech enhancement system. There are a number of concepts for estimating a speech mask from noisy data, like parameter estimation using a CGMM [11], or neural networks operating on spectrogram data [1, 3, 9]. Usually, these methods use the signal energy or PSDs as feature vectors, and are therefore highly dependent on the array geometry and statistics of the speech and noise presented in the training data.

However, in a scenario like CHiME4, no reliable assumptions can be made about the signal statistics. The speaker position is unknown, and the background noise is non-stationary and can contain all sorts of sounds from passing-by cars, transient bursts from pneumatic bus doors to human speech. The number of usable microphones can also change, due to microphone failures. Further the array geometry might be unknown, like for the 2 channel track in CHiME4 [10]. Also, the microphones may not be matched. In such situations, the signal power alone is no reliable indicator for speech presence. We observed that the eigenvectors of the PSD matrix  $\Phi_{ZZ}$

of the noisy inputs provide a good measure for speaker activity, independent of signal energy and array geometry. We only assume that the speaker is *slowly moving*, and that the noise is *non-stationary*. Eigenvalue decomposition of  $\Phi_{ZZ}$  gives

$$\Phi_{ZZ} = \sum_{m=1}^M \lambda_{Z,m} \mathbf{v}_{Z,m} \mathbf{v}_{Z,m}^H, \quad (14)$$

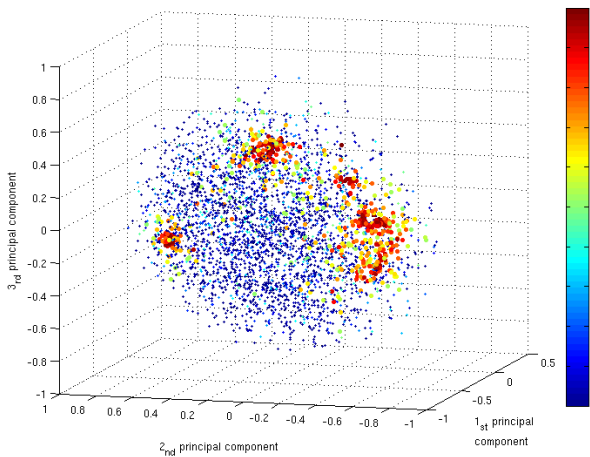
where  $\lambda_{Z,m}$  and  $\mathbf{v}_{Z,m}$  are the eigenvalues and eigenvectors of  $\Phi_{ZZ}$ . We denote  $m = 1$  as the dominant eigenvector  $\mathbf{v}_{Z,1}$ . Note that  $\lambda_{Z,m}$  corresponds to the signal power, and  $\mathbf{v}_{Z,m}$  corresponds to the spatial information embedded in the signal.

#### 4.1. Visualization of Eigenvectors

For  $M = 6$  channels, the complex-valued eigenvectors  $\mathbf{v}_{Z,1}(k, l)$  lie on the surface of a 11-dimensional unit sphere<sup>3</sup>. In Figure 1 we show  $\mathbf{v}_{Z,1}(k, l)$  for 10,000 consecutive frames  $l$  from the ‘embedded’ street recordings. The selected frequency bin  $k$  corresponds to  $\approx 2650\text{Hz}$ . The dots are colored according to  $p_{SPP}(k, l)$ , which has been calculated from the PSDs of the speech and noise ground truth available for the simulated data of CHiME4, i.e.

$$p_{SPP, \text{true}} = \frac{\text{Tr}\{\Phi_{SS}\}}{\text{Tr}\{\Phi_{SS} + \Phi_{NN}\}}. \quad (15)$$

Using PCA to visualize the first three principal components of  $\mathbf{v}_{Z,1}$  reveals an interesting structure. It can be seen that the dominant eigenvectors form local clusters if speech is present (red dots). During speech absence they are uniformly distributed over the sphere (blue dots). This clustering indicates that the speaker is indeed *slowly moving*, which will be exploited to estimate  $p_{SPP}$ .



**Fig. 1.** 3D projection of  $\mathbf{v}_{Z,1}$  for a single frequency bin over time. The dots are colored according to  $p_{SPP, \text{true}}$ .

#### 4.2. Kernelized DNN

We use a DNN to learn  $p_{SPP}$  from the dominant eigenvector  $\mathbf{v}_{Z,1}$  of the PSD matrix  $\Phi_{ZZ}$  of the noisy inputs. As we are operating in the frequency domain, a separate kernel for each frequency bin

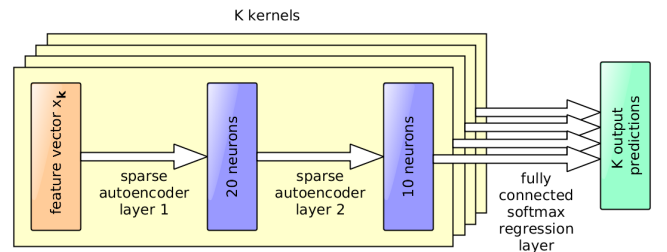
<sup>3</sup>An  $m$ -dimensional complex eigenvector has  $2m - 1$  non-redundant real-valued dimensions, as the eigenvector can be scaled by an arbitrary complex constant so that one dimension collapses to zero.

$k$  is required. To introduce some context-sensitivity into our model, we do not use  $\mathbf{v}_{Z,1}(k, l)$  directly as feature vector, but calculate the cosine distance<sup>4</sup>  $x_{i,k}$  between the current eigenvector  $\mathbf{v}_{Z,1}(k, l)$  at time frame  $l$  and the  $i^{\text{th}}$  most recent frame, i.e.

$$x_{i,k} = \text{Re}\left\{ \mathbf{v}_{Z,1}(k, l)^H \mathbf{v}_{Z,1}(k, l - i) \right\}. \quad (16)$$

This enables the DNN to exploit the temporal information embedded in the signal.  $x_{i,k}$  is stacked to produce a feature vector  $\mathbf{x}_k$  per kernel  $k$ , so that a feature vector covering  $\Delta$  consecutive frames consists of  $\mathbf{x}_k = [x_{1,k}, x_{2,k}, \dots, x_{\Delta,k}]$ . Note that (16) effectively eliminates the number of microphones from the feature vector. Hence, we can apply the same DNN structure to a wide range of multi-channel speech enhancement tasks.

The DNN of each kernel uses a hybrid model with a generative and a discriminative component [2]. The generative component consists of two autoencoder layers, which perform unsupervised clustering of the input data  $\mathbf{x}_k$ . The autoencoder kernels operate independently for each frequency bin. We varied the number of hidden layers and the number of neurons per layer in our experiments, and heuristically determined that 2 hidden layers comprising 20 and 10 neurons are a good compromise between clustering performance and computational complexity. The discriminative component consists of a regression layer which fuses the activations of all autoencoder kernels, in order to exploit information which is distributed across the frequency. The regression layer predicts the  $K$  output labels  $p_{SPP}(\mathbf{x}_k)$ . Figure 2 illustrates the kernelized DNN used in our system. Note that we could also use a (bidirectional) long short term memory (B-LSTM), but our kernelized DNN has the advantage of an efficient implementation, and it is easy to train.



**Fig. 2.** Kernelized DNN with feature vector  $\mathbf{x}_k$  and output predictions  $p_{SPP}(\mathbf{x}_k)$ .

#### 4.3. DNN training

We use greedy layer-wise pretraining for the autoencoder kernels [23], and discriminative fine-tuning for the softmax-layer using the true label  $p_{SPP, \text{true}}$  from (15). Optimization is done using stochastic gradient descent with ADAM [24]. The autoencoder uses the KL-divergence and weight decay to enforce a sparse representation of the inputs  $\mathbf{x}_k$ . The softmax layer uses the cross entropy between the true and predicted speech presence probability as cost function.

## 5. RESULTS

We trained our kernelized DNN using the 6-channel training data of the CHiME4 corpus [10], for which the ground truth  $p_{SPP, \text{true}}$  is

<sup>4</sup>Note that the eigenvector is already normalized to 1, i.e.  $\|\mathbf{v}_{Z,1}(k, l)\|_2^2 = 1$

available. The training set comprises 1600 real and 7138 simulated utterances. Then we applied the DNN to the entire 2 and 6-channel corpus consisting of 14658 utterances, which translates roughly into 28 hours of audio data. The DNN outputs the speech presence probability  $p_{SPP}$ , which we use to construct the EV-GSC beamformer and postfilter as described in Section 3. For more details on the CHiME4 data the interested reader is referred to [10].

### 5.1. Speech Mask Accuracy

Figure 3 shows the performance of the DNN for a single utterance from the evaluation set (M04.420C020M.CAF). Panel (a) shows  $x_{i=1,k}$  from the feature vector for the DNN, (b) shows the true label calculated with (15), and (c) shows the prediction for  $p_{SPP}$  obtained from the softmax regression layer.

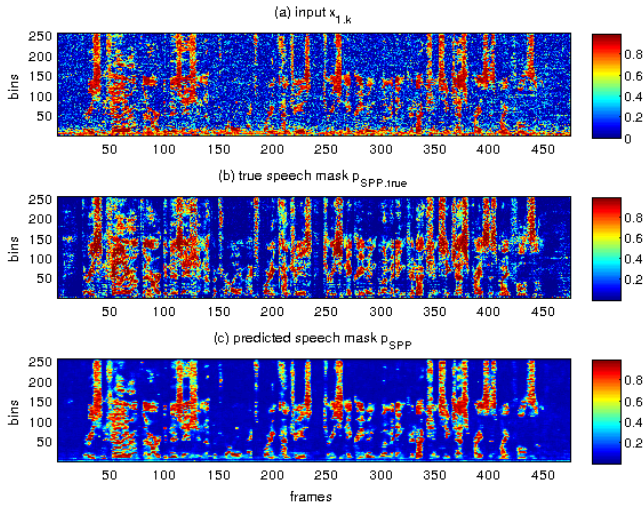


Fig. 3. Speech presence probability mask prediction.

We observe that  $x_{i=1,k}$  already shows a high similarity to the true speech presence probability, except for low frequencies and some noise. Due to the fully connected softmax layer, the noise can be almost completely removed, and the prediction accuracy is also good for low frequencies. Table 1 reports the prediction error for  $p_{SPP}$ <sup>5</sup> for the 2 and 6 channel data of CHiME4 [10], and various feature vector lengths  $\Delta$ . Using a feature vector with more than 5 consecutive frames gives no significant performance improvement, hence we select  $\Delta = 5$  to be a reasonable trade-off between accuracy and computational complexity.

Scenario	Train	Dev	Eval
2ch, $\Delta = 3$	15.46	15.85	16.58
2ch, $\Delta = 5$	15.08	15.61	16.17
2ch, $\Delta = 7$	14.89	15.32	16.02
6ch, $\Delta = 3$	11.16	11.69	12.24
6ch, $\Delta = 5$	10.74	11.41	11.85
6ch, $\Delta = 7$	10.55	11.28	11.74

Table 1. Prediction error for  $p_{SPP}$  in %.

<sup>5</sup>The prediction error is the average over all time-frequency bins of  $|p_{SPP} - p_{SPP,true}|$ .

### 5.2. Perceptual Speech Quality

With the predicted speech mask  $p_{SPP}$ , we construct the EV-GSC beamformer from Section 3.1. We use the *Perceptual Evaluation methods for Audio Source Separation* (PEASS) Toolkit [25, 26] to evaluate the performance of our multi-channel speech enhancement system, and report the *Overall Perceptual Score* (OPS) and PESQ [27] values. Tables 2 and 3 give a comparison of our system (EV-GSC) against the CHiME4-baseline enhancement system using the BeamformIt-toolkit [10], our GSC with steering vector estimation and an *adaptive blocking matrix* (ABM) [7], and the front-end of the best CHiME3 system [11], which uses a complex gaussian mixture model (CGMM-EM) to estimate the speech and noise PSD matrices. The model parameters are estimated with an EM algorithm, and the posterior probability is used as speech presence probability.

It can be seen that our approach (EV-GSC) outperforms the CHiME4 baseline, the GSC with ABM, and the CGMM-EM systems in terms of PESQ and OPS on the simulated (simu) and real (real) data set for 6-channels (6ch). Even in the 2-channel case (2ch) we obtain competitive results. In this case, the two channels are randomly selected from the 6-channels, i.e. the array geometry changes randomly.

Method	Data set	Train	Dev	Eval
CHiME4 baseline (BeamformIt), 5ch [10]	simu	1.35	1.31	1.26
	real	1.35	1.28	1.37
GSC with ABM and postfilter, 6ch [7]	simu	1.98	1.69	1.63
	real	1.51	1.39	1.44
CGMM-EM with MVDR and postfilter, 6ch [11]	simu	1.79	1.59	1.51
	real	1.53	1.41	1.44
<b>EV-GSC and postfilter, 6ch, <math>\Delta = 5</math></b>	simu	<b>2.04</b>	<b>1.89</b>	<b>1.86</b>
	real	<b>1.72</b>	<b>1.74</b>	<b>1.63</b>
<b>EV-GSC and postfilter, 2ch, <math>\Delta = 5</math></b>	simu	1.68	1.61	1.58
	real	1.55	1.43	1.54

Table 2. PESQ scores.

Method	Data set	Train	Dev	Eval
CHiME4 baseline (BeamformIt), 5ch [10]	simu	33.11	34.73	31.46
	real	29.97	36.45	36.74
GSC with ABM and postfilter, 6ch [7]	simu	56.08	44.82	44.48
	real	47.18	44.90	36.96
CGMM-EM with MVDR and postfilter, 6ch [11]	simu	52.15	43.02	40.59
	real	44.95	41.89	36.87
<b>EV-GSC and postfilter, 6ch, <math>\Delta = 5</math></b>	simu	<b>59.09</b>	<b>48.32</b>	<b>48.64</b>
	real	<b>52.34</b>	<b>46.09</b>	<b>44.16</b>
<b>EV-GSC and postfilter, 2ch, <math>\Delta = 5</math></b>	simu	47.32	40.97	40.75
	real	43.43	42.83	39.82

Table 3. OPS scores.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we have shown the importance of the speech presence probability mask, which is used to construct an optimal multi-channel Wiener filter followed by a single-channel postfilter. Further, we presented a kernelized DNN to estimate this speech presence probability mask. To prevent the DNN from learning specific speaker and noise types, we used the cosine distance between the dominant eigenvectors of consecutive frames of the PSD of the noisy speech as input feature. Finally, we compared our system against three state-of-the-art approaches, and evaluate the perceptual speech quality and classification error. Future work includes an in-depth evaluation of the DNN being used and performance comparison against B-LSTMs. Furthermore, the relationship between the eigenvectors and the speech presence probability mask is investigated.

## 7. REFERENCES

- [1] Y. Wang and D. Wang, "Cocktail party processing via structured prediction," in *Neural Information Processing Systems (NIPS)*, 2012, pp. 224–232.
- [2] M. Zöhrer, R. Peharz, and F. Pernkopf, "Representation learning for single-channel source separation and bandwidth extension," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2398–2409, 2015.
- [3] F. Weninger, J. Le Roux, J. R. Hershey, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. GlobalSIP Symposium on Machine Learning Applications in Speech Processing*, Dec. 2014.
- [4] L. Deng, M.L. Seltzer, D. Yu, A. Acero, A. Mohamed, and G.E. Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *Interspeech*, 2010, pp. 1692–1695.
- [5] M. Zöhrer and F. Pernkopf, "Single channel source separation with general stochastic networks," in *Interspeech*, 2014.
- [6] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*, Springer, Berlin–Heidelberg–New York, 2008.
- [7] L. Pfeifenberger, T. Schrank, M. Zöhrer, M. Hagmüller, and F. Pernkopf, "Multi-channel speech processing architectures for noise robust speech recognition: 3rd chime challenge results," in *Proc. IEEE ASRU*, 2015.
- [8] L. Pfeifenberger and F. Pernkopf, "Blind source extraction based on a direction-dependent a-priori SNR," in *Interspeech*, May 2014.
- [9] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *2016 IEEE ICASSP*, 2016.
- [10] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe, "The third 'chime' speech separation and recognition challenge: Dataset, task and baselines," in *IEEE 2015 Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.
- [11] Higuchi T., Ito N., Yoshioka T., and Nakatani T., "Robust mvdr beamforming using time-frequency masks for online/offline asr in noise," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 5210–5214, Mar. 2016.
- [12] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*, Springer, Berlin–Heidelberg–New York, 2008.
- [13] P. Vary and R. Martin, *Digital Speech Transmission*, Wiley, West Sussex, 2006.
- [14] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*, Springer, Berlin–Heidelberg–New York, 2006.
- [15] O. Hoshuyama, A. Sugiyama, and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, Oct. 1999.
- [16] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Transactions on Signal Processing*, vol. 49, no. 8, Aug. 2001.
- [17] W. Herbordt and W. Kellermann, "Analysis of blocking matrices for generalized sidelobe cancellers for non-stationary broadband signals," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May 2002.
- [18] S. Gannot and I. Cohen, "Speech enhancement based on the general transfer function GSC and postfiltering," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 6, Nov. 2004.
- [19] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, July 2007.
- [20] E. Warsitz, A. Krueger, and R. Haeb-Umbach, "Speech enhancement with a new generalized eigenvector blocking matrix for application in a generalized sidelobe canceller," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 73–76, May 2008.
- [21] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, New Jersey, 4th edition, 2002.
- [22] F. Kuech, E. Mabande, and G. Enzner, "State-space architecture of the partitioned-block-based acoustic echo controller," *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014.
- [23] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Neural Information Processing Systems (NIPS)*, 2007.
- [24] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations, San Diego, 2015*, July 2015.
- [25] E. Vincent, "Improved perceptual metrics for the evaluation of audio source separation," *10th Int. Conf. on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pp. 430–437, 2012.
- [26] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, Sept. 2011.
- [27] "ITU-T recommendation P.862. Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," 2000.

## **A.6 InterSpeech 2017**

*"Eigenvector-based speech mask estimation using logistic regression"*, Lukas Pfeifenberger, Matthias Zöhrer and Franz Pernkopf, International Conference on Spoken Language Processing (InterSpeech), Stockholm, 2017





# Eigenvector-based Speech Mask Estimation using Logistic Regression

Lukas Pfeifenberger<sup>1,2</sup>, Matthias Zöhrer<sup>1</sup>, Franz Pernkopf<sup>1</sup>

<sup>1</sup> Signal Processing and Speech Communication Laboratory  
Graz University of Technology, Graz, Austria

<sup>2</sup> Ognios GmbH Salzburg, Austria

lukas.pfeifenberger@alumni.tugraz.at  
{matthias.zoehrer,pernkopf}@tugraz.at

## Abstract

In this paper, we use a logistic regression to learn a speech mask from the dominant eigenvector of the *Power Spectral Density* (PSD) matrix of a multi-channel speech signal corrupted by ambient noise. We employ this speech mask to construct the *Generalized Eigenvalue* (GEV) beamformer and a Wiener post-filter. Further, we extend the beamformer to compensate for speech distortions. We do not make any assumptions about the array geometry or the characteristics of the speech and noise sources. Those parameters are learned from training data. Our assumptions are that the speaker may move slowly in the near-field of the array, and that the noise is in the far-field. We compare our speech enhancement system against recent contributions using the CHiME4 corpus. We show that our approach yields superior results, both in terms of perceptual speech quality and speech mask estimation error.

**Index Terms:** Multi-channel speech enhancement, broadband beamforming, speech mask estimation

## 1. Introduction

In many beamforming structures, a *steering vector* is required to provide a spatial focus towards the location of the speaker. A simple and robust method is to obtain the steering vector from a *Direction Of Arrival* (DOA) estimate. Many algorithms have been devised for that purpose, i.e. PHAT, MUSIC [1], or DD-SNR [2]. However, the DOA cannot model reverberation or multi-path propagation caused by the enclosure, i.e. office rooms or car interiors. This may result in target leakage, which limits beamforming performance. More advanced approaches aim at estimating the acoustic path from the speech source to each microphone, which is known as *Acoustic Transfer Function* (ATF). Approximations are done using *Relative Transfer Functions* (RTFs) [3,4]. The RTFs relate the ATFs with respect to a reference point, and can be modeled by shorter FIR filters [5]. Recent contributions use a spectral gain mask to distinguish between speech and noise signals, which is then used to estimate their respective PSD matrices. Such a *speech mask* may be obtained using model-based clustering approaches [6–8], or data-driven regression [9–12] based on various types of *neural networks* (NN). While clustering approaches require some prior knowledge like the array geometry or the statistics of the noise, NNs are able to learn the speech mask from training data, without additional information. Moreover, NNs have the distinct advantage of jointly estimating a speech mask for all frequencies, which proved to be superior in recent multi-channel speech enhancement and recognition tasks [13, 14].

In this paper, we extend our work in [12], where we used several NN architectures to estimate the speech mask using eigenvector features. As the largest NN requires over a million

weights to be trained, the aim is to significantly reduce model complexity, while maintaining performance. We introduce a different approach to estimate the speech mask compared to [8] and [9]. Instead of energy-related features, our NN utilizes the dominant eigenvector of the PSD matrix of the microphone signals as feature vector. Therefore, the spatial information hidden in the multi-channel data is exploited. The predicted speech mask is then used to split the PSD matrix of the microphone signals into its speech and noise components, where we use the dominant eigenvector of the speech PSD as steering vector for the beamformer. We show that the cosine similarity between dominant eigenvectors of consecutive PSD matrices of the microphone signals is sufficient to predict the speech mask. By using the cosine similarity, we obtain a feature which is independent of both the signal energy and the microphone array geometry. Our assumptions are that the speaker is in the near-field of the array and that the non-stationary noise is in the far-field. The speaker may move slowly, resulting in slowly varying ATFs. These relaxed conditions are found in many telephony applications, i.e. hands-free calling kits, voice chats on mobile devices, or roadside emergency telephones.

This paper is structured as follows: After the introduction of the system model in Section 2, we demonstrate our extension to the GEV beamformer for reducing speech distortions in Section 3. In Section 4 we show the importance of the speech presence probability for constructing the beamformer and a postfilter. In Section 5 the estimation of the speech presence probability using logistic regression is presented. In Section 6 we present our results. Section 7 concludes the paper.

## 2. System Model

We assume a single speech source embedded in ambient noise. The array consists of  $M$  microphones, arranged into an arbitrary array geometry. There may be multiple noise sources, and their spatial and temporal characteristics may be unknown. Our speech enhancement system is shown in Figure 1. We define

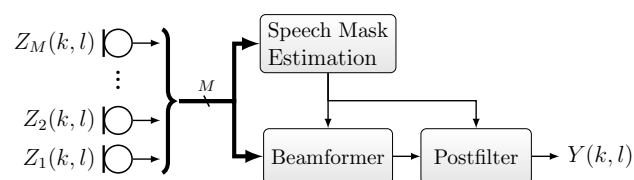


Figure 1: System overview, showing the microphone signals  $Z_m(k, l)$  and the beamformer+postfilter output  $Y(k, l)$  in frequency domain.

the signal at the  $m^{\text{th}}$  microphone in the STFT domain as

$$Z_m(k, l) = S(k, l)A_m(k, l) + N_m(k, l), \quad (1)$$

where the frequency bin  $k = 1, \dots, K$  and the time frame is denoted by  $l$ . The ATF of the speaker  $S(k, l)$  to the  $m^{\text{th}}$  microphone is denoted by  $A_m(k, l)$ , and the noise received at that microphone is denoted by  $N_m(k, l)$ . By stacking all  $M$  signals to a  $M \times 1$  vector, the signal model can be written as:

$$\mathbf{Z}(k, l) = S(k, l)\mathbf{A}(k, l) + \mathbf{N}(k, l). \quad (2)$$

For enhanced readability, the frequency and time frame indices will be omitted except where necessary. The PSD matrix for all microphone signals  $\mathbf{Z}(k, l)$  is obtained using recursive averaging, i.e.:  $\Phi_{ZZ}(k, l) = \Phi_{ZZ}(k, l-1)\alpha + (1-\alpha)\mathbf{Z}(k, l)\mathbf{Z}^H(k, l)$ , where  $0 \leq \alpha \leq 1$  is a smoothing parameter [15]. For uncorrelated speech and noise signals, this PSD matrix can be split into its speech and noise components

$$\Phi_{ZZ}(k, l) = \Phi_{SS}(k, l) + \Phi_{NN}(k, l). \quad (3)$$

For a single speaker,  $\Phi_{SS}(k, l)$  will be of rank 1 and can therefore be decomposed into

$$\Phi_{SS}(k, l) = \mathbf{A}(k, l)\mathbf{A}^H(k, l)\Phi_S(k, l). \quad (4)$$

Note that the magnitude of the ATFs can be modeled by  $\Phi_S(k, l)$  in (4), hence we define  $\|\mathbf{A}\|^2 = 1$ .

### 3. Multi-Channel Speech Enhancement

We use a *broadband* beamformer [16] for multi-channel speech enhancement. The beamformer output is given by

$$Y(k, l) = \mathbf{W}^H(k, l)\mathbf{Z}(k, l)G(k, l), \quad (5)$$

with the filter weights  $\mathbf{W}(k, l)$ , and the single-channel Wiener postfilter  $G(k, l)$ . Following the definition of the system model in (2), the optimal filter weights are given by the MSE-optimal multi-channel Wiener filter  $\mathbf{W}_{OPT}$  [15, 17], i.e.

$$\mathbf{W}_{OPT} = \underbrace{\frac{\Phi_{NN}^{-1}\mathbf{A}}{\mathbf{A}^H\Phi_{NN}^{-1}\mathbf{A}}}_{\mathbf{W}_{MVDR}} \cdot \underbrace{\frac{\Phi_S}{\Phi_S + [\mathbf{A}^H\Phi_{NN}^{-1}\mathbf{A}]^{-1}}}_{G = \frac{\xi}{1+\xi}}. \quad (6)$$

The filter  $\mathbf{W}_{MVDR}$  can be recognized as the MVDR beamformer [18, 19]. The postfilter  $G = \frac{\xi}{1+\xi}$  depicts a real-valued gain mask, which is applied at the beamformer output. It can be seen from (4) and (6), that  $\xi$  is given as the multi-channel SNR [17]

$$\xi = \Phi_S\mathbf{A}^H\Phi_{NN}^{-1}\mathbf{A} = \text{Tr}\{\Phi_{NN}^{-1}\Phi_{SS}\}. \quad (7)$$

#### 3.1. GEV Beamformer

While it is possible to select from a broad range of broadband beamformers such as the MVDR or the GSC, we use the GEV for its superior performance in earlier experiments [12]. The GEV beamformer, constrains the filter weights  $\mathbf{W}(k, l)$  to maximize the SNR  $\xi(k, l)$  at the beamformer output [20, 21], i.e.

$$\mathbf{W}_{GEV} = \arg \max_{\mathbf{w}} \xi. \quad (8)$$

The solution to (8) is given by the following generalized eigenvalue problem

$$\Phi_{SS}\mathbf{W}_{GEV} = \xi\Phi_{NN}\mathbf{W}_{GEV}, \quad (9)$$

which is solved by

$$\mathbf{W}_{GEV} = \zeta\Phi_{NN}^{-1}\mathbf{A}, \quad (10)$$

where  $\zeta$  is an arbitrary complex scalar. Comparing the GEV to the MVDR beamformer, it can be immediately seen that they only differ by a complex constant  $C$

$$\mathbf{W}_{MVDR} = \frac{\Phi_{NN}^{-1}\mathbf{A}}{\mathbf{A}^H\Phi_{NN}^{-1}\mathbf{A}} = C\mathbf{W}_{GEV} \quad (11)$$

However, this difference causes target speech distortions in the GEV, i.e.  $\mathbf{W}_{GEV}^H(k, l)\mathbf{A}(k, l) \neq 1$ . To compensate for these distortions, we derive an expression for  $C$  as follows:

Assuming normalized ATFs  $\|\mathbf{A}\|^2 = 1$ , we can rearrange (10) into  $\zeta = \mathbf{A}^H\Phi_{NN}\mathbf{W}_{GEV}$  and express the complex constant  $C$  by

$$C_{PAN} = \frac{\mathbf{W}_{GEV}^H\Phi_{NN}\mathbf{A}}{\mathbf{W}_{GEV}^H\Phi_{NN}\mathbf{W}_{GEV}}, \quad (12)$$

which we refer to as *Phase Aware Normalization* (PAN). Note that [20] proposes the *Blind Analytical Normalization* (BAN) and the *Blind Statistical Normalization* (BSN) compensation methods to estimate the absolute value of  $C$ , i.e.:

$$C_{BAN} = \frac{\sqrt{\mathbf{W}_{GEV}^H\Phi_{NN}\Phi_{NN}\mathbf{W}_{GEV}}}{\mathbf{W}_{GEV}^H\Phi_{NN}\mathbf{W}_{GEV}}. \text{ In fact, it can be easily verified that the magnitudes of the BAN and PAN compensation factors are identical. Inserting (12) into (11) gives the GEV-PAN beamformer:}$$

$$C_{PAN}\mathbf{W}_{GEV} = \frac{\mathbf{W}_{GEV}\mathbf{W}_{GEV}^H\Phi_{NN}\mathbf{A}}{\mathbf{W}_{GEV}^H\Phi_{NN}\mathbf{W}_{GEV}} = \mathbf{P}\mathbf{A} \quad (13)$$

with the projection matrix  $\mathbf{P}$  [22]. The expression  $\mathbf{B} = \mathbf{I} - \mathbf{P}$  can be identified as blocking matrix [21]. In theory, the compensation factor  $C_{PAN}$  turns the GEV into the MVDR beamformer. However, as the former avoids the inversion of  $\Phi_{NN}$  when using (9), it is numerically more stable and achieves better PESQ and OPS scores [12].

#### 3.2. Steering Vector Estimation

From (13) it can be seen that the GEV-PAN beamformer requires the ATFs  $\mathbf{A}(k, l)$ . As they are unknown in practice and hard to estimate in reverberant environments [18], we use a steering vector  $\mathbf{F}(k, l)$ , which provides a spatial focus of the speech source. Under reverberation-free conditions the steering vector may be modeled as simple time delays using DOA estimation [1]. However, in realistic environments this approach will result in speech loss at the beamformer output. We therefore advocate a steering vector in signal subspace [23]. Eigenvalue decomposition (EVD) of the speech PSD matrix gives

$$\Phi_{SS} = \mathbf{A}\mathbf{A}^H\Phi_S = \mathbf{v}_{S_1}\mathbf{v}_{S_1}^H\lambda_{S_1}, \quad (14)$$

where  $\lambda_{S_1}$  and  $\mathbf{v}_{S_1}$  are the single eigenvalue and eigenvector of  $\Phi_{SS}(k, l)$ , as this matrix is of rank 1 for a single speaker. Rearranging (14) leads to:

$$\mathbf{A} = \frac{\lambda_{S_1}}{\mathbf{A}^H\mathbf{v}_{S_1}\Phi_S}\mathbf{v}_{S_1} = \zeta_{S_1}\mathbf{v}_{S_1}, \quad (15)$$

where  $\zeta_{S_1}$  can be recognized as another complex scalar. Therefore, the dominant eigenvector of  $\Phi_{SS}$  is a scaled version of the true ATF  $\mathbf{A}(k, l)$ , including multi-path propagations and early echoes of the target signal [1, 20, 23]. Assuming  $\|\mathbf{A}\|^2 = 1$ , the dominant eigenvector  $\mathbf{v}_{S_1}$  is equal to the ATF.



## 4. Speech Mask Estimation

In the last section, we have shown that the GEV-PAN beamformer and the steering vector require an estimate of both the speech and noise PSD matrices.

### 4.1. PSD matrix approximation

By using an oracle speech mask  $0 \leq p_{\text{SPP}}(k, l) \leq 1$ , which represents the probability for each time-frequency bin to contain speech,  $\Phi_{\text{SS}}(k, l)$  can be approximated with

$$\hat{\Phi}_{\text{SS}}(k, l) = \frac{\sum_{t=l-T/2}^{l+T/2} \mathbf{Z}(k, t) \mathbf{Z}^H(k, t) p_{\text{SPP}}(k, t)}{\sum_{t=l-T/2}^{l+T/2} p_{\text{SPP}}(k, t)}, \quad (16)$$

where  $T$  is a number of frames during which we assume the spatial characteristics of  $\Phi_{\text{SS}}(k, l)$  to be stationary, i.e. the speaker is not moving [24]. Note that the energy of the speech signal may change during the time period  $T$ , but this does not affect (9), and hence the performance of the GEV beamformer. Analogously to (16), the noise PSD matrix  $\Phi_{\text{NN}}(k, l)$  can be approximated using the complementary probability  $1 - p_{\text{SPP}}(k, t)$ .

### 4.2. Speech Presence Probability

As shown above, the estimation of  $p_{\text{SPP}}(k, l)$  is the key component in our speech enhancement system. Using (7), we define the optimal speech presence probability as

$$p_{\text{SPP, opt}}(k, l) = \frac{\xi(k, l)}{1 + \xi(k, l)} = G(k, l). \quad (17)$$

Note that the optimal speech presence probability is equal to the Wiener postfilter given in (6).

Eigenvalue decomposition of the noisy speech PSD matrix gives  $\Phi_{\text{ZZ}} = \sum_{m=1}^M \lambda_{Z_m} \mathbf{v}_{Z_m} \mathbf{v}_{Z_m}^H$ , where  $\lambda_{Z_m}$  and  $\mathbf{v}_{Z_m}$  are its eigenvalues and eigenvectors, respectively. We observed that the dominant eigenvector  $\mathbf{v}_{Z_1}(k, l)$  is related to  $p_{\text{SPP, opt}}(k, l)$  [24]. It was shown that eigenvectors containing speech tend to form local clusters, while noisy eigenvectors are distributed randomly over a multi-dimensional unit sphere. Hence, the dominant eigenvector  $\mathbf{v}_{Z_1}(k, l)$  provides a reliable measure for speaker activity. Using the cosine similarity<sup>1</sup> between two neighboring eigenvectors

$$x_{\Delta}(k, l) = |\mathbf{v}_{Z_1}(k, l)^H \mathbf{v}_{Z_1}(k, l - \Delta)|, \quad (18)$$

we obtain a scalar  $x_{\Delta}(k, l)$ , independent of the number of microphones being used. To observe a difference between two neighboring eigenvectors, the matrix  $\Phi_{\text{ZZ}}$  has to be updated with a sufficiently small time constant. During speaker activity,  $x_{\Delta}(k, l)$  is close to one, and close to zero otherwise. Note that this feature is also independent of signal energy and array geometry. Figure 2 shows  $x_{\Delta}(k, l)$  for a single utterance of the CHiME4 corpus. It can be seen that  $x_{\Delta}(k, l)$  already has some similarity with the optimal speech mask  $p_{\text{SPP, opt}}(k, l)$ , shown in Figure 3a. At low frequencies, the separation capability of this feature is poor, as the wavelength of the signal is large compared to the aperture of a typical microphone array. This information has to be inferred from other frequency components.

<sup>1</sup>Note that the eigenvectors are already normalized to 1, i.e.  $\|\mathbf{v}_{Z_1}(k, l)\|_2^2 = 1$ .

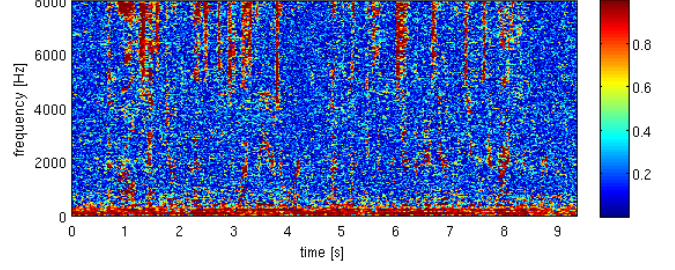


Figure 2: Cosine similarity  $x_{\Delta=1}(k, l)$  for a single utterance of the CHiME4 corpus.

## 5. Logistic Regression

In contrast to the NNs based on LSTMs and MLPs used in [12], we aim for a resource-efficient regression model to estimate the speech mask. As our model operates in the time-frequency domain, we derive a feature vector for each frequency bin  $k$  and time frame  $l$ . In particular, we stack  $n_{\Delta}$  cosine distances  $x_{\Delta}(k, l)$  to add some context to the feature vector

$$\mathbf{x}_{\text{evd}}(k, l) = [x_{\Delta=1}(k, l), \dots, x_{\Delta=n_{\Delta}}(k, l)]^T, \quad (19)$$

where we consider the eigenvectors in the vicinity  $n_{\Delta}$  of the current time frame  $l$  containing the most relevant information. We refer to (19) as *eigenvector-delta*. For each frequency bin  $k$  we obtain an estimate

$$\tilde{p}_{\text{SPP}}(k, l) = \frac{e^{\tilde{a}(k, l)}}{\sum_{i=1}^2 e^{\tilde{a}(k, i)}}, \quad (20)$$

using the activation

$$\tilde{a}(k, i) = \tilde{\mathbf{W}}(k, i) \mathbf{x}_{\text{evd}}(k, l) + \tilde{b}(k, i), \quad (21)$$

where  $\tilde{\mathbf{W}}$  and  $\tilde{b}$  denote the weights and bias values of the logistic regression, respectively. Note that  $\tilde{p}_{\text{SPP}}(k, l)$  is calculated independently for each frequency bin. To exploit the broadband nature of human speech, we employ a second logistic regression which calculates a refined estimate  $\hat{p}_{\text{SPP}}(k, l) = \frac{e^{\hat{a}(k, l)}}{\sum_{i=1}^2 e^{\hat{a}(k, i)}}$ . The activation  $\hat{a}(k, i)$  uses the estimate  $\tilde{p}_{\text{SPP}}(k, l)$  of the neighboring  $k - k_{\Delta} \dots k + k_{\Delta}$  frequency bins, i.e.

$$\hat{a}(k, i) = \sum_{j=k-k_{\Delta}}^{k+k_{\Delta}} \hat{\mathbf{W}}(k, j, i) \tilde{p}_{\text{SPP}}(j, l) + \hat{b}(k, i). \quad (22)$$

This architecture is capable to learn the basic structure of human speech. As the eigenvector-features do not contain any information about signal energy, *speaker-dependent* features are ignored.

## 6. Results

### 6.1. Experimental Setup

To evaluate our model, we use the CHiME4 corpus [13], which provides 2 and 6-channel recordings of a close-talking speaker corrupted by four different types of ambient noise. The database provides a training set (tr05), a validation set (dt05) and a test set (et05). We use all utterances (real and simu) from each set. The ground truth (i.e. the separated speech and noise signals) is available for all recordings, which we use to calculate the true speech mask  $p_{\text{SPP, opt}}(k, l)$  with (7) and (17). Once

trained, the logistic regression provides a prediction  $\hat{p}_{SPP}(k, l)$  for each utterance, which is required to calculate  $\hat{\Phi}_{SS}(k, l)$  and  $\hat{\Phi}_{NN}(k, l)$  with (16). The averaging window length is set to  $T = 250ms$ . We use a STFT window length of 32ms and an overlap of 50% to process the data. The speech and noise PSD estimates are then used to construct the GEV-PAN beamformer. Following (17), we use  $G(k, l) = \hat{p}_{SPP}(k, l)$  for the postfilter.

## 6.2. Speech Mask Accuracy

Figure 3 shows the performance of the logistic regression models by visualizing the optimal and predicted speech masks for a single utterance from the test set. Table 1 reports the predic-

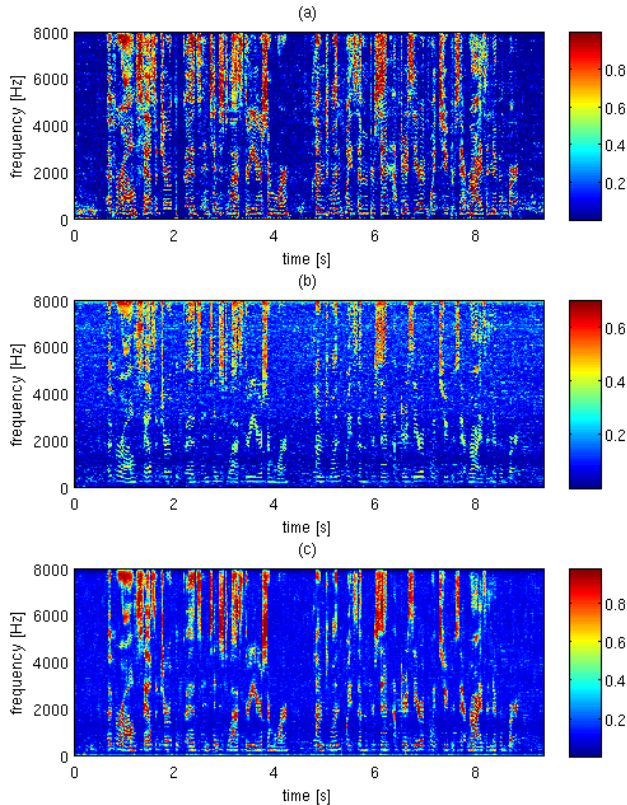


Figure 3: Speech presence probability for a single utterance from the CHiME4 test set. (a) ground truth  $p_{SPP,opt}(k, l)$ , (b) coarse prediction  $\hat{p}_{SPP}(k, l)$ , (c) refined prediction  $\hat{p}_{SPP}(k, l)$ .

tion error  $\mathcal{L} = \frac{100}{KL} \sum_{k=1}^K \sum_{l=1}^L |\hat{p}_{SPP}(k, l) - p_{SPP,opt}(k, l)|$  in % for the logistic regression, labeled as `evd_logreg`, and four alternative NN models which we used in [12]. They are labeled as `ev_lstm`, `evd_lstm`, `evd_mlp` and `psd_lstm`, and use multiple hidden layers, each containing  $n_h$  neurons. The last column shows the number of parameters to be trained for each model. It can be seen that the logistic regression uses two orders of magnitude fewer weights while achieving comparable results.

## 6.3. Perceptual Speech Quality

Using the predicted speech mask  $\hat{p}_{SPP}(k, l)$ , we construct the GEV-PAN beamformer for both the 2 and 6-channel data. To evaluate the performance of the resulting speech signal  $Y(k, l)$  in terms of perceptual speech quality, we use the *Perceptual Evaluation methods for Audio Source Separation* (PEASS)

Table 1: Prediction error for  $\hat{p}_{SPP}(k, l)$  in % for the 6 channel data. Results of proposed methods are bold face.

architecture	$n_\Delta$	$n_h$	train	valid	test	# of weights
ev_lstm, 6ch	-	20,10	1.889	2.685	3.003	1457704
evd_lstm, 6ch	7	20,10	2.184	2.183	2.520	1252104
evd_mlp, 6ch	7	20,10	2.349	2.285	2.825	156513
psd_lstm, 6ch	-	20,10	2.711	3.415	3.489	1210984
<b>evd_logreg, <math>\hat{p}_{SPP}</math>, 6ch</b>	3	-	<b>3.980</b>	<b>3.894</b>	<b>5.700</b>	<b>1542</b>
<b>evd_logreg, <math>\tilde{p}_{SPP}</math>, 6ch</b>	3	-	<b>2.767</b>	<b>2.671</b>	<b>3.598</b>	<b>12336</b>
ev_lstm, 2ch	-	10,5	3.919	4.265	4.377	1128744
evd_lstm, 2ch	7	10,5	3.566	3.495	3.992	1252104
evd_mlp, 2ch	7	10,5	3.695	3.613	4.778	156513
psd_lstm, 2ch	-	10,5	4.620	5.082	4.902	1046504
<b>evd_logreg, <math>\hat{p}_{SPP}</math>, 2ch</b>	3	-	<b>6.575</b>	<b>6.493</b>	<b>7.497</b>	<b>1542</b>
<b>evd_logreg, <math>\tilde{p}_{SPP}</math>, 2ch</b>	3	-	<b>4.382</b>	<b>4.282</b>	<b>5.901</b>	<b>13364</b>

toolkit [25], and report the *Overall Perceptual Score* (OPS) and PESQ [26] values. The ground truth required for these scores is obtained using the  $p_{SPP,opt}(k, l)$  and the GEV-PAN. Table 2 reports the PESQ and OPS scores of the logistic regression and the other models used in Table 1. Further, we also report the scores of the CHiME4-baseline enhancement system, i.e. the BeamformIt!-toolkit [13], and the front-end of the best CHiME3 system [8], which uses CGMM priors and the EM algorithm to estimate the speech mask. It can be seen that the performance of the much smaller logistic regression architecture (`evd_logreg`) is comparable to the NN models, even for the 2-channel track. For this track, 2 out of the 6 microphones have been chosen randomly, so that the array geometry is unknown for each utterance. In summary, all our eigenvector-based speech mask estimation models show an improvement over the BeamformIt! baseline and the CGMM-EM system.

Table 2: Performance comparison of 6- and 2-channel data, against the BeamformIt! and CGMM-EM systems.

architecture	$n_\Delta$	$n_h$	PESQ [MOS]			OPS [%]		
			train	valid	test	train	valid	test
ev_lstm, 6ch	-	20,10	2.443	2.007	1.891	72	58	51
evd_lstm, 6ch	7	20,10	2.226	1.969	1.874	67	59	52
evd_mlp, 6ch	7	20,10	2.197	1.944	1.829	67	59	52
psd_lstm, 6ch	-	20,10	1.977	1.758	1.724	63	54	49
<b>evd_logreg, <math>\hat{p}_{SPP}</math>, 6ch</b>	3	-	<b>1.830</b>	<b>1.676</b>	<b>1.551</b>	<b>57</b>	<b>52</b>	<b>46</b>
<b>evd_logreg, <math>\tilde{p}_{SPP}</math>, 6ch</b>	3	-	<b>2.071</b>	<b>1.862</b>	<b>1.704</b>	<b>63</b>	<b>57</b>	<b>50</b>
ev_lstm, 2ch	-	10,5	1.965	1.706	1.725	51	44	45
evd_lstm, 2ch	7	10,5	2.090	1.850	1.869	48	43	43
evd_mlp, 2ch	7	10,5	2.042	1.818	1.820	46	42	42
psd_lstm, 2ch	-	10,5	1.867	1.669	1.703	44	40	41
<b>evd_logreg, <math>\hat{p}_{SPP}</math>, 2ch</b>	3	-	<b>1.696</b>	<b>1.578</b>	<b>1.579</b>	<b>35</b>	<b>32</b>	<b>34</b>
<b>evd_logreg, <math>\tilde{p}_{SPP}</math>, 2ch</b>	3	-	<b>1.940</b>	<b>1.754</b>	<b>1.741</b>	<b>43</b>	<b>39</b>	<b>40</b>
BeamformIt!, 5ch	-	-	1.350	1.292	1.326	31	36	35
CGMM-EM, 6ch	-	-	1.635	1.483	1.468	48	42	38

## 7. Conclusion

In this paper, we proposed a resource-efficient linear regression architecture for speech mask estimation as alternative to NNs. Our system uses the dominant eigenvector of the PSD of the microphone signals as feature vector. We compared our results against the most recent model-based and data-driven approaches using the CHiME4 corpus. We have shown that our model yields good results, both in terms of perceptual speech quality and speech mask prediction error, while using two orders of magnitude fewer parameters than comparable NN models. Unlike existing approaches, our system does not require any information about the array geometry or the characteristics of the speech and noise sources. Our assumptions are that the speaker moves slowly and is located in the near-field of the array, while the non-stationary noise is in the far-field.

## 8. References


- [1] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Berlin–Heidelberg–New York: Springer, 2008.
- [2] L. Pfeifenberger and F. Pernkopf, “Blind source extraction based on a direction-dependent a-priori SNR,” in *Interspeech*, May 2014.
- [3] R. Talmon, I. Cohen, and S. Gannot, “Relative transfer function identification using convolutive transfer function approximation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, May 2009.
- [4] I. Cohen, “Relative transfer function identification using speech signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, Sep. 2004.
- [5] S. Gannot, D. Burshtein, and E. Weinstein, “Signal enhancement using beamforming and nonstationarity with applications to speech,” *IEEE Transactions on Signal Processing*, vol. 49, no. 8, Aug. 2001.
- [6] M. Souden, J. Chen, J. Benesty, and S. Affes, “An integrated solution for online multichannel noise tracking and reduction,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, Sep. 2011.
- [7] —, “Gaussian model-based multichannel speech presence probability,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 5, Jul. 2010.
- [8] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, “Robust MVDR beamforming using time-frequency masks for online/offline asr in noise,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 5210–5214, Mar. 2016.
- [9] J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Mar. 2016.
- [10] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, “BLSTM supported GEV beamformer front-end for the 3rd chime challenge,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 444–451.
- [11] H. Erdogan, J. Hershey, S. Watanabe, M. Mandel, and J. L. Roux, “Improved MVDR beamforming using single-channel mask prediction networks,” in *Interspeech*, Sep. 2016.
- [12] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, “Eigenvector-based speech mask estimation for multi-channel speech enhancement,” Signal Processing and Speech Communication Laboratory, Technical University of Graz, Austria, Tech. Rep., Dec. 2016.
- [13] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, “The third ‘chime’ speech separation and recognition challenge: Dataset, task and baselines,” in *IEEE 2015 Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.
- [14] T. Schrank, L. Pfeifenberger, M. Zöhrer, J. Stahl, P. Mowlae, and F. Pernkopf, “Deep beamforming and data augmentation for robust speech recognition: Results of the 4th chime challenge,” in *Proc. of the 4th Intl. Workshop on Speech Processing in Everyday Environments (CHiME 2016)*, 2016.
- [15] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. Berlin–Heidelberg–New York: Springer, 2006.
- [16] M. Brandstein and D. Ward, *Microphone Arrays*. Berlin–Heidelberg–New York: Springer, 2001.
- [17] M. Souden, J. Benesty, and S. Affes, “On optimal frequency-domain multichannel linear filtering for noise reduction,” *IEEE Trans. Audio, Speech, Lang. Process.*, pp. 260–275, Feb. 2010.
- [18] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Berlin–Heidelberg–New York: Springer, 2008.
- [19] B. D. V. Veen and K. M. Buckley, “Beamforming: a versatile approach to spatial filtering,” *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, no. 5, pp. 4–24, Apr. 1988.
- [20] E. Warsitz and R. Haeb-Umbach, “Blind acoustic beamforming based on generalized eigenvalue decomposition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, Jul. 2007.
- [21] E. Warsitz, A. Krueger, and R. Haeb-Umbach, “Speech enhancement with a new generalized eigenvector blocking matrix for application in a generalized sidelobe canceller,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 73–76, May 2008.
- [22] L. Pfeifenberger and F. Pernkopf, “A multi-channel postfilter based on the diffuse noise sound field,” in *European Association for Signal Processing Conference 2014*, Jun. 2014.
- [23] M. G. Shmulik, S. Gannot, and I. Cohen, “Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, Aug. 2009.
- [24] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, “DNN-based speech mask estimation for eigenvector beamforming,” *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017.
- [25] E. Vincent, “Improved perceptual metrics for the evaluation of audio source separation,” *Int. Conf. on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pp. 430–437, 2012.
- [26] “ITU-T recommendation P.862. Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs,” 2000.

## A.7 IEEE/ACM 2019

*"Eigenvector-Based speech mask estimation for multi-channel speech enhancement"*, Lukas Pfeifer-berger, Matthias Zöhrer and Franz Pernkopf, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2019



# Eigenvector-Based Speech Mask Estimation for Multi-Channel Speech Enhancement

Lukas Pfeifenberger , Matthias Zöhrer , and Franz Pernkopf , *Senior Member, IEEE*

**Abstract**—We present the *Eigennet* architecture for estimating a gain mask from noisy, multi-channel microphone observations. While existing mask estimators use magnitude features, our system also exploits the spatial information embedded in the phase of the data. The mask is used to obtain the *Minimum Variance Distortionless Response* (MVDR) and *Generalized Eigenvalue* (GEV) beamformers. We also derive the *Phase Aware Normalization* (PAN) postfilter, which corrects both magnitude and phase distortions caused by the GEV. Further, we demonstrate the properties of our eigenvector features, and compare their performance with three state-of-the-art reference systems. We report their performance in terms of SNR improvement and *Word Error Rate* (WER) using Google and Kaldi Speech-to-Text API. Experiments are performed on the WSJ0 and CHiME4 corpora, where competitive performance in both WER and SNR is achieved.

**Index Terms**—Multi-channel speech enhancement, Eigenvector beamforming, speech mask estimation, Neural Networks.

## I. INTRODUCTION

SPEECH enhancement is of paramount significance in many telecommunication applications. Especially in hands-free scenarios, background noise is the primary source of speech degradation. Consequently, improving speech intelligibility and quality has been an active field of research for many decades. As computing platforms keep getting cheaper and faster, the focus has shifted from single-channel speech enhancement (SCSE) to multi-channel speech enhancement (MCSE) methods. While SCSE relies exclusively on the spectral characteristics of the speech and noise signals, MCSE allows the additional use of spatial information.

MCSE methods can be divided into blind source separation (BSS) and beamforming approaches. While BSS aims at separating all involved sources in the mixture, beamforming enhances only a set of desired sources while treating all others as interference. A beamformer is comprised of a set of spatio-temporal filters, which are convolved with each of the microphone signals prior to summation. If those filters are designed with the objective to extract a broadband signal like speech, it is considered

as a *broadband beamformer*. Common beamforming structures are the *Minimum Variance Distortionless Response* (MVDR) beamformer [1], and its *Generalized Sidelobe Canceller* (GSC) formulation [2]. Both aim at minimizing the signal power at the beamformer output, with the constraint to avoid distorting the target signal. For its simplicity and robustness, the GSC is widely used [3]–[6]. Another variant is the *Generalized Eigenvalue* (GEV) beamformer [7], which trades minimal speech distortions for maximum SNR at the beamformer output. While those distortions can be controlled using either the *Blind Analytical Normalization* (BAN) or *Blind Statistical Normalization* (BSN) postfilters, we introduce the *Phase Aware Normalization* (PAN) postfilter, which also accounts for phase distortions.

Beamformers like the MVDR or GSC require a *steering vector* to direct the beamformer towards the desired signal, i.e. the speaker. This direction can be estimated using *Direction Of Arrival* (DOA) algorithms like PHAT, MUSIC [8], or DD-SNR [9]. However, the acoustic path from the speaker to the microphones is comprised of multiple reflections, caused by reverberations of the acoustic environment. This path is known as *Acoustic Transfer Function* (ATF) [10], [11]. As the DOA only models the direct path, *target leakage* limits the beamforming performance [12]. As an alternative to DOA estimation, a fixed set of beamforming kernels may be learned in time-domain directly from the noisy data, i.e. [13], [14].

With recent trends towards *mask based beamforming*, the steering vector is no longer required, as the gain mask identifies the time-frequency bins that contain the desired signal. This mask is used to obtain the spatial *Power Spectral Density* (PSD) matrices of the desired and interfering sound sources. The PSD matrix of the desired speech signal contains the ATF of the speaker in its principal eigenvector [6]. Hence, mask-based beamforming proved to be superior to DOA-based approaches [15]–[20].

There are a number of concepts for estimating a gain mask from noisy data. Since the CHiME3 and CHiME4-challenges [21], magnitude features are widely used to train a *Neural Network* (NN) [22]–[27]. However, the spatial information embedded in the phase of the microphone signals is neglected. Consequently, such models lack the ability to separate multiple speakers from a mixture. Alternatively, approaches like [28]–[30] also incorporate the phase information into their systems. The CHiME3 challenge was won by [29], where a *Complex Gaussian Mixture Model* (CGMM) is used to model the PSD matrices of the involved sound sources. The model parameters are estimated with an EM algorithm, and the

Manuscript received April 26, 2019; revised August 21, 2019; accepted September 4, 2019. Date of publication September 16, 2019; date of current version October 2, 2019. This work was supported in part by the Austrian Science Fund (FWF) under Grant I2706-N31. The work of L. Pfeifenberger was supported by Ognios GmbH. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lei Xie. (*Corresponding author: Franz Pernkopf.*)

The authors are with the Intelligent Systems Group, Signal Processing and Speech Communication Laboratory, Graz University of Technology, 8010 Graz, Austria (e-mail: lpfeifen@gmail.com; matthias.zoehrer@tugraz.at; pernkopf@tugraz.at).

Digital Object Identifier 10.1109/TASLP.2019.2941592

2329-9290 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See [http://www.ieee.org/publications\\_standards/publications/rights/index.html](http://www.ieee.org/publications_standards/publications/rights/index.html) for more information.

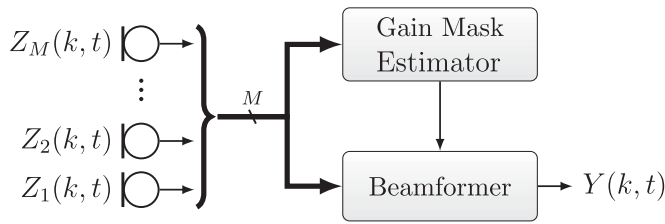


Fig. 1. System overview, showing the microphone signals  $Z_m(k, t)$  and the beamformer output  $Y(k, t)$  in frequency domain.

posterior probabilities are used as gain masks for the respective source components. To avoid the *source permutation problem*, a template for each PSD matrix is required to initialize the EM algorithm [31]. Further, the PSD matrices are assumed to be constant over time. Therefore, this model is limited to stationary sources.

In this paper, we propose the *EigenNet* architecture, which exploits both the magnitude and spatial information contained in the eigenvectors of the PSD matrix of the noisy speech. This facilitates a NN which is able to track and separate multiple sources. We demonstrate the properties of three different eigenvector-based features, i.e. spatial features (Evs), combined spatial and magnitude features (EvsMag), and a hybrid variant (Evd). Similar to existing phase-aware features like IPD [32] or GCC [33], the eigenvectors contain the phase difference between all microphone pairs, and hence encode the location of the sources in the observed signal. Further, we compare those features against a NN-based mask estimator using magnitude features, similar to [29], [34], a CGMM-EM based method [29], and the *BeamformIt* toolkit [35] as a baseline.

We use two speech corpora to evaluate those systems, CHiME4 [21], and WSJ0 [36]. While the former provides 6 and 2 channel recordings, the latter consists of monaural recordings, which we use in conjunction with simulated room acoustics to obtain 6 channel data of multiple, moving speakers [37]. We report the performance of all experiments in terms of SNR improvement and *Word Error Rate* (WER) using the Google Speech-to-Text API [38] and the Kaldi-ASR toolkit [39].

## II. SYSTEM MODEL

We assume a number of  $C$  independent sound sources being picked up by a microphone array which consists of  $M$  microphones. The array may be arranged into an arbitrary geometry, and the sound sources may be non-stationary, i.e. moving speakers. The spatial and temporal characteristics of the individual sources are unknown. We arbitrarily denote the first source  $c = 1$  to be the desired source of interest, and regard the other sources as unwanted interference. Our speech enhancement system encompasses a NN to estimate gain masks from the noisy microphone observations, and a *broadband beamformer* to isolate the desired signal. Fig. (1) provides a system overview. The signal arriving at the microphones is composed of an additive mixture of  $C$  independent sound sources. In the *short-time Fourier transform* (STFT) domain, the time-frequency bins of all  $M$  microphones can be stacked into a single  $M \times 1$

vector, i.e.

$$\mathbf{Z}(k, t) = \sum_{c=1}^C \mathbf{X}_c(k, t), \quad (1)$$

where

$$\mathbf{Z}(k, t) = [Z_1(k, t), \dots, Z_M(k, t)]^T. \quad (2)$$

The vector  $\mathbf{X}_c(k, t)$  represents the  $c^{\text{th}}$  sound source at frequency bin  $k = 1, \dots, K$  and time frame  $t$ . For enhanced readability, the frequency and time frame indices will be omitted where the context is clear. Each sound source is composed of a monaural recording  $X_c(k, t)$  convolved with an *Acoustic Transfer Function* (ATF) denoted by  $\mathbf{A}_c(k, t)$ , i.e.

$$\mathbf{X}_c(k, t) = \mathbf{A}_c(k, t)X_c(k, t). \quad (3)$$

The ATFs model the acoustic path from a sound source to the microphones, including all reverberations and reflections caused by the room acoustics [40]. If the source  $X_c(k, t)$  is located in the near field of the array, the ATFs can be modeled by a *finite impulse response* (FIR) filter [12]. Depending on the movement of the speaker, this filter may change over time, i.e. being non-stationary. Note that the STFT window needs to be sufficiently long to model the multiplicative filter operation  $\mathbf{A}_c(k, t)X_c(k, t)$  without aliasing. Using Eq. (1), we define the signal of interest to be the first source, i.e.

$$\mathbf{S}(k, t) = \mathbf{X}_1(k, t), \quad (4)$$

and the interfering signal as the sum of all other sources, i.e.

$$\mathbf{N}(k, t) = \sum_{c=2}^C \mathbf{X}_c(k, t). \quad (5)$$

Consequently, the spatial PSD matrix for the desired signal is given as [41]

$$\Phi_{SS}(k) = \mathbb{E}\{\mathbf{S}(k, t)\mathbf{S}^H(k, t)\}, \quad (6)$$

and for the interfering signal

$$\Phi_{NN}(k) = \mathbb{E}\{\mathbf{N}(k, t)\mathbf{N}^H(k, t)\}. \quad (7)$$

## III. BEAMFORMING

To isolate the desired source  $\mathbf{S}(k, t)$  and attenuate the interfering sources  $\mathbf{N}(k, t)$  at the same time, we use a *filter and sum* beamformer [42]. Each microphone signal  $Z_m(k, t)$  is filtered with the beamforming weights  $W_m(k, t)$ , prior to summing all outputs. The beamforming operation can be written as inner vector product

$$Y(k, t) = \mathbf{W}^H(k)\mathbf{Z}(k, t), \quad (8)$$

with the weight vector

$$\mathbf{W}(k) = [W_1(k), \dots, W_M(k)]^T. \quad (9)$$

### A. MVDR Beamformer

The well-known MVDR beamformer  $\mathbf{W}_{MVDR}$  [6], [12] minimizes the signal energy at the output of the beamformer,

while maintaining an undistorted response with respect to the steering vector  $\mathbf{v}_S(k)$ , i.e.

$$\mathbf{W}_{MVDR} = \frac{\Phi_{NN}^{-1} \mathbf{v}_S}{\mathbf{v}_S^H \Phi_{NN}^{-1} \mathbf{v}_S}. \quad (10)$$

The steering vector guides the beamformer towards the direction of the desired signal of interest. Clearly, the ATFs of that source would be the ideal steering vector. However, they are hard to estimate from noisy speech signals [12], [40]. Therefore, they are often replaced by simple time delays, i.e.

$$\mathbf{v}_S(k) := [e^{-j\omega_k \tau_1}, e^{-j\omega_k \tau_2}, \dots, e^{-j\omega_k \tau_M}]^T, \quad (11)$$

where  $\omega_k = 2\pi \frac{k}{2K} f_s$  is the discrete frequency variable, and  $\tau_m$  denotes the time delay from the desired source to the  $m^{\text{th}}$  microphone [3], [10], [11]. The time delays correspond to the direction of the source. This direction can be obtained using one of the *Direction Of Arrival* (DOA) estimation algorithms covered in [3], [8]–[10]. However, this approach does not account for reverberations and multi-path propagations. Therefore it is of limited use in real-world applications.

If we assume the PSD matrix of the desired source to be known, the steering vector may be obtained in signal subspace [6]. Eigenvalue decomposition (EVD) of the PSD matrix  $\Phi_{SS}(k)$  yields:

$$\Phi_{SS}(k) = \sum_{m=1}^M \mathbf{v}_{S_m} \mathbf{v}_{S_m}^H \lambda_{S_m}, \quad (12)$$

where  $\lambda_{S_m}$  and  $\mathbf{v}_{S_m}$  are the eigenvalues and eigenvectors of  $\Phi_{SS}$ , respectively. We denote the eigenvector belonging to the largest eigenvalue as steering vector, i.e.

$$\mathbf{v}_S(k) := \mathbf{v}_{S_1}(k) \quad (13)$$

### B. GEV Beamformer

A widely used alternative to the MVDR beamformer is the GEV beamformer [7], [43], which determines the filter weights  $\mathbf{W}$  to maximize the SNR  $\xi$  at the beamformer output, i.e.

$$\mathbf{W}_{GEV} = \arg \max_{\mathbf{W}} \xi, \quad (14)$$

where

$$\xi = \frac{\mathbf{W}^H \Phi_{SS} \mathbf{W}}{\mathbf{W}^H \Phi_{NN} \mathbf{W}} \quad (15)$$

is the SNR at the output of the beamformer. (14) can be rewritten as a generalized eigenvalue problem [43]:

$$\Phi_{NN}^{-1} \Phi_{SS} \mathbf{W} = \xi \mathbf{W}. \quad (16)$$

A solution for Eq. (16) is given by

$$\mathbf{W}_{GEV} = \zeta \Phi_{NN}^{-1} \mathbf{v}_S, \quad (17)$$

where  $\zeta$  is an arbitrary complex scalar. The beamforming filter  $\mathbf{W}_{GEV}$  will not have a distortionless response, i.e.  $\mathbf{v}_S^H \mathbf{W}_{GEV} \neq 1$ . Therefore, the *Blind Analytical Normalization* (BAN) compensation factor has been proposed in [7]. It is

given as

$$G_{BAN} = \frac{\sqrt{\mathbf{W}_{GEV}^H \Phi_{NN} \Phi_{NN} \mathbf{W}_{GEV}}}{\mathbf{W}_{GEV}^H \Phi_{NN} \mathbf{W}_{GEV}}. \quad (18)$$

While  $G_{BAN}$  normalizes the magnitude response of the GEV beamformer, it does not account for phase distortions. We therefore propose the *Phase Aware Normalization* (PAN) factor: By comparing Eq. (10) and (17), it can be seen that the MVDR and GEV beamformers are identical up to another complex scalar  $G_{PAN}$ :

$$\mathbf{W}_{MVDR} = G_{PAN} \mathbf{W}_{GEV} = \frac{\Phi_{NN}^{-1} \mathbf{v}_S}{\mathbf{v}_S^H \Phi_{NN}^{-1} \mathbf{v}_S}. \quad (19)$$

With Eq. (17), (19) can be rewritten as

$$G_{PAN} = \frac{\zeta^*}{\mathbf{W}_{GEV}^H \Phi_{NN} \mathbf{W}_{GEV}}. \quad (20)$$

Using  $\|\mathbf{v}_S\|_2^2 \stackrel{!}{=} 1$ , Eq. (17) can be rearranged to  $\zeta = \mathbf{v}_S^H \Phi_{NN} \mathbf{W}_{GEV}$ , which can be inserted into Eq. (20) to give the *Phase Aware Normalization* factor:

$$G_{PAN} = \frac{\mathbf{W}_{GEV}^H \Phi_{NN} \mathbf{v}_S}{\mathbf{W}_{GEV}^H \Phi_{NN} \mathbf{W}_{GEV}}. \quad (21)$$

From Eq. (19) it can be seen that the *Phase Aware Normalization* factor will turn the GEV beamformer into the MVDR beamformer. However, the GEV avoids the inversion of the noise PSD matrix  $\Phi_{NN}$  by using the Schur decomposition to solve Eq. (16). This leads to an improved numerical stability, as shown in [24].

### C. PSD Matrix Estimation

Let us assume that we have access to an oracle *gain mask*  $p(k, t)$ , which represents the probability  $0 \leq p(k, t) \leq 1$  for each time-frequency bin whether it contains the desired signal or not. In this case, the spatial PSD matrix  $\Phi_{SS}(k)$  can be approximated using

$$\hat{\Phi}_{SS}(k, t) = \frac{1}{L} \sum_{l=t-L/2}^{t+L/2} \mathbf{Z}(k, l) \mathbf{Z}^H(k, l) p(k, l). \quad (22)$$

Analogously, the estimate  $\hat{\Phi}_{NN}(k, t)$  can be found using another gain mask for the interfering signal. Note that the window length  $L$  defines the number of time frames used for estimating the PSD matrices. This *block processing* allows to apply the MVDR or GEV beamformer to a whole block of  $L$  frames at once. For moving sources, a tradeoff has to be made; If  $L$  is set too small, the accuracy of the estimated PSD matrices might be poor. The matrices might even become singular if the gain mask is sparse. When using the MVDR beamformer, this will lead to numerical problems as it requires to invert the noise PSD matrix. If  $L$  is too large, the estimated PSD matrices might fail to adapt quickly enough to changes in the spatial characteristics of the moving sources. An alternative to block processing is given by



recursive estimation, i.e.

$$\begin{aligned} \hat{\Phi}_{SS}(k, t) &= \hat{\Phi}_{SS}(k, t-1)[1 - p(k, t)] \\ &+ \mathbf{Z}(k, t)\mathbf{Z}^H(k, t)p(k, t). \end{aligned} \quad (23)$$

This *online processing* [44] allows to apply the MVDR or GEV beamformer to each time frame  $t$ . When using this method, the PSDs have to be initialized using Eq. (22).

#### IV. EIGENNET ARCHITECTURES

As demonstrated above, the gain mask is sufficient to estimate the speech and noise PSDs, from which we can construct the MVDR and the GEV beamformers. Therefore, the estimation of  $p(k, t)$  is the key component of our multi-channel speech enhancement system.

The vast majority of mask estimators use magnitude spectrograms as features to train a NN [22]–[25]. However, the spatial information embedded in the phase of the complex-valued inputs  $\mathbf{Z}(k, t)$  is neglected. Consequently, such models lack the ability to separate and track multiple speakers from a mixture. To overcome those limitations, we propose to use eigenvector features. We observed that the principal eigenvector of the spatial PSD matrix  $\Phi_{ZZ}(k) = \mathbb{E}\{\mathbf{Z}(k, t)\mathbf{Z}^H(k, t)\}$  will point towards the dominant signal source in subspace. As speech signals are sparse, the principal eigenvector will form a cluster for each stationary source. Dynamic sources can be tracked using recurrent neural network structures.

##### A. Feature Preprocessing

For low frequencies, the aperture of the microphone array is small compared to the wavelength. As a consequence, the spatial resolution is poor. Therefore, we propose to decorrelate the noisy inputs  $\mathbf{Z}(k, t)$  using *Principal Component Analysis* (PCA) whitening, i.e.

$$\bar{\mathbf{Z}}(k, t) = \mathbf{U}_{PCA}(k)\mathbf{Z}(k, t), \quad (24)$$

with the whitening matrix

$$\mathbf{U}_{PCA}(k) = \mathbf{D}^{-\frac{1}{2}}\mathbf{E}^H, \quad (25)$$

where  $\mathbf{E}$  and  $\mathbf{D}$  are the eigenvector and eigenvalue matrices of the spatial coherence matrix of the ideal isotropic sound field  $\Gamma(k)$  [40]. Its elements are given as  $\Gamma_{i,j}(k) = \frac{\sin(\omega_k d_{i,j}/c)}{\omega_k d_{i,j}/c}$ , where  $\omega_k = 2\pi k f_s$ ,  $d_{i,j}$  is the distance between the  $i^{\text{th}}$  and  $j^{\text{th}}$  microphone. Further,  $f_s$  is the sample rate, and  $c$  is the speed of sound at room temperature. Note that whitening also decorrelates the feature dimensions, leading to faster convergence during training the NN [45]. To obtain eigenvector features, we calculate the principal eigenvector  $\mathbf{v}_{\bar{\mathbf{Z}},1}(k)$  of the spatial PSD matrix  $\Phi_{\bar{\mathbf{Z}}\bar{\mathbf{Z}}}(k) = \mathbb{E}\{\bar{\mathbf{Z}}(k, t)\bar{\mathbf{Z}}^H(k, t)\}$ . In practice,  $\Phi_{\bar{\mathbf{Z}}\bar{\mathbf{Z}}}(k)$  will be a long-term estimate over an entire utterance. For a short-term estimate, we normalize the whitened features to 1, which gives an *instantaneous eigenvector*, i.e.

$$\mathbf{v}_{\bar{\mathbf{Z}},i}(k, t) = \frac{\bar{\mathbf{Z}}(k, t)}{\|\bar{\mathbf{Z}}(k, t)\|_2} e^{-j\phi_1(k, t)}, \quad (26)$$

TABLE I  
FEATURE VARIANTS

Feature vector	Eigennet model
$\mathbf{x}_{\text{EvsMag}}(k, t) = \bar{\mathbf{Z}}(k, t)e^{-j\phi_1(k, t)}$	large
$\mathbf{x}_{\text{Evs}}(k, t) = \mathbf{v}_{\bar{\mathbf{Z}},i}(k, t)$	large
$x_{\text{Mag}}(k, t) = \ \bar{\mathbf{Z}}(k, t)\ _2$	small
$x_{\text{Evd}}(k, t) =  \cos \Theta(k, t) ^2$	small

where  $e^{-j\phi_1(k, t)}$  normalizes the phase with respect to the first microphone, which enables the NN to distinguish phase differences between multiple sources. The phase of the first microphone is denoted by  $\phi_1(k, t)$ . To detect a single, dominant source in the observed signal, we express the cosine similarity between  $\mathbf{v}_{\bar{\mathbf{Z}},i}(k, t)$  and the principal eigenvector  $\mathbf{v}_{\bar{\mathbf{Z}},1}(k)$  of the spatial PSD  $\Phi_{\bar{\mathbf{Z}}\bar{\mathbf{Z}}}(k)$  as

$$\cos \Theta(k, t) = \mathbf{v}_{\bar{\mathbf{Z}},i}^H(k, t)\mathbf{v}_{\bar{\mathbf{Z}},1}(k). \quad (27)$$

Using these quantities, we construct four different features, as shown in Table I. The features have the following properties:

1) *EvsMag Features*: The whitened signal  $\bar{\mathbf{Z}}(k, t)$  contains both spatial and magnitude information, hence it can be used to separate multiple sources based on their location and temporal characteristics. Again, we normalize its phase with respect to the first microphone. As it consists of  $M$  complex numbers per time-frequency bin, we stack its real and imaginary components into the  $2M$  sized feature vector  $\mathbf{x}_{\text{EvsMag}}(k, t)$ .

2) *Evs Features*: To test the performance of the spatial information alone, we use the normalized features  $\mathbf{v}_{\bar{\mathbf{Z}},i}(k, t)$ . Again, we stack its real and imaginary components into the  $2M$  sized feature vector  $\mathbf{x}_{\text{Evs}}(k, t)$ .

3) *Mag Features*: To test the performance of the magnitude information alone, we calculate the vector magnitude using  $x_{\text{Mag}}(k, t) = \|\bar{\mathbf{Z}}(k, t)\|_2$ . This feature has only 1 element per time-frequency bin.

4) *Evd Features*: For scenarios with a single dominant source, we use the squared cosine similarity between the Evs features and the principal eigenvector  $\mathbf{v}_{\bar{\mathbf{Z}},1}(k)$ , i.e.  $x_{\text{Evd}}(k, t) = |\cos \Theta(k, t)|^2$ . This feature has only 1 element per time-frequency bin.

##### B. Eigennet Models

We use *Multi-Layer Perceptron* (MLP) and *Bi-directional Long Short-Term Memory* (BLSTM) units as building blocks of our NN. Those components have been used with great success in recent mask-based beamforming tasks [16], [18], [24], [46].

1) *Small Model*: The small model is used for feature vectors with 1 element per time-frequency bin, i.e.  $x_{\text{Mag}}$  and  $x_{\text{Evd}}$ . As the NN operates on whole STFT frames, it processes  $K$  feature vectors at a time. Fig. 2 shows the two layers of the small model. The first layer is a BLSTM unit, which consists of two separate LSTM units, each with  $K$  neurons. While the first LSTM processes the data in forward direction (i.e. one time frame after another), the second LSTM operates in backward direction. The output of both LSTMs is then concatenated to an intermediate vector with  $2K$  elements. The second layer



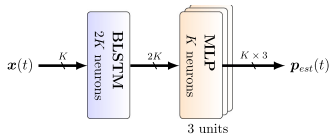


Fig. 2. Small model.

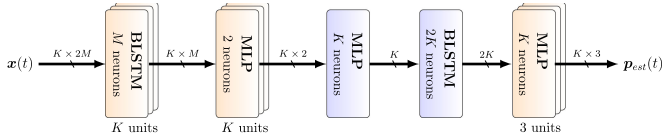


Fig. 3. Large model.

consists of three separate, fully-connected MLPs. The first MLP estimates the mask for the desired source, the second estimates the mask for the interfering sources, and the third estimates the mask for time-frequency bins which are not assigned to the other two classes. The activation function at the topmost layer is a softmax, so that the sum of each of the three masks is 1 for each time-frequency bin, i.e.  $\sum_{i=1}^3 p_{est}(k, t, i) \stackrel{!}{=} 1$ .

2) *Large Model*: The large model is used with the  $\mathbf{x}_{Evs}(k, t)$  and  $\mathbf{x}_{EvsMag}(k, t)$  features. It encompasses five layers; the first three reduce the feature vector size from  $2M$  elements per time-frequency bin down to 1. Note that those layers have very few weights, as they consist of  $K$  independent units with  $M$  neurons each, or fewer. The upper two layers are identical to the small model. Fig. 3 shows the details of the large model.

### C. Labels

Using the separated sources from the signal model in Eq. (4) and (5), optimal binary masks can be obtained. We use three masks, one for the desired signal, i.e.

$$p_{opt}(k, l, 1) = \|\mathbf{S}(k, t)\|_2 > \max(\|\mathbf{N}(k, t)\|_2, \epsilon(k)), \quad (28)$$

one for the interfering signals, i.e.

$$p_{opt}(k, l, 2) = \|\mathbf{N}(k, t)\|_2 > \max(\|\mathbf{S}(k, t)\|_2, \epsilon(k)), \quad (29)$$

and one for weak signal components, which do not contribute to any of the PSD matrices:

$$p_{opt}(k, l, 3) = 1 - p(k, l, 1) - p(k, l, 2). \quad (30)$$

The constant  $\epsilon(k)$  controls the amount of energy required for the signal to be assigned to the desired or interfering class label. As the sum of all three masks is always 1 for each time-frequency bin, we can use the *cross-entropy* as loss function [45] to train the Eigennet models.

## V. EXPERIMENTS USING WSJ0

### A. Experimental Setup

To demonstrate the performance of the different Eigennet models and feature variants, we simulate a typical living room scenario with two static speakers  $S_1$  and  $S_2$ , two moving speakers  $D_1$  and  $D_2$ , and an isotropic background noise source  $I$ . Fig. 4 illustrates the floorplan of the setup. The red circle denotes

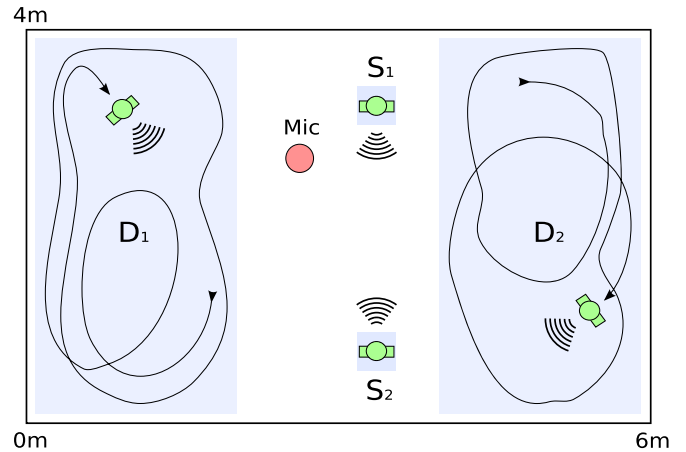


Fig. 4. Shoebox model of a living room showing stationary sound sources  $S_1$  and  $S_2$ , and dynamic sound sources  $D_1$  and  $D_2$ . The microphone array is indicated by the red circle (Mic).

TABLE II  
EXPERIMENTAL SCENARIOS

Scenario name	Desired source	Interfering source(s)
$S_1$ -I	static1	isotropic
R-I	random	isotropic
$D_1$ -I	dynamic1	isotropic
$S_1$ - $S_2$ I	static1	static2 + isotropic
$D_1$ - $D_2$ I	dynamic1	dynamic2 + isotropic

a circular microphone array with  $M = 6$  microphones and a diameter of 86mm. To simulate head movements of the static speakers  $S_1$  and  $S_2$ , random position changes occur within a cube of 20cm in size. The dynamic paths  $D_1$  and  $D_2$  move randomly within a region of 2m by 4m on either side of the microphone array, with a speed of  $0.5 \frac{m}{s}$ .

Using this environment, we define five scenarios given in Table II:

1) *>Static1 vs. Isotropic* ( $S_1$ -I): This scenario simulates a stationary speaker against isotropic background noise. Note that the head movements will cause a varying phase especially at higher frequencies.

2) *Random vs. Isotropic* (R-I): The *random* source denotes a static speaker with head movements, whose position is randomly chosen anywhere in the room for each new utterance. This prevents the Eigennet from learning the position of the speaker, but rather to distinguish the characteristics of the isotropic background noise and the speaker.

3) *Dynamic1 vs. Isotropic* ( $D_1$ -I): This scenario tests the tracking capabilities of the Eigennet architecture. A speaker moving in region  $D_1$  has to be tracked in the presence of ambient background noise.

4) *Static1 vs. Static2 + Isotropic* ( $S_1$ - $S_2$ I): This scenario tests the separation capabilities of two simultaneous speakers embedded in background noise. The second speaker is randomly chosen from the same WSJ0 subset.

5) *Dynamic1 vs. Dynamic2 + Isotropic* ( $D_1$ - $D_2$ I): This scenario tests the separation capabilities of two moving speakers embedded in background noise. The second speaker is randomly chosen from the same WSJ0 subset.

## B. Data Generation

To generate multi-channel recordings from monaural sources, we simulate the ATFs in Eq. (3) using the *Image Source Method* (ISM) [37], [47]. The living room is modeled as shoebox with a reflection coefficient of  $\beta = 0.85$  for each wall, and a reflection order of 10. This results in a reverberation time of approximately 500 ms. As we have moving sources, we generate a new set of ATFs every 32 ms. The isotropic background noise is generated using

$$\mathbf{X}_n(k, t) = \mathbf{U}(k, t)\mathbf{X}_n(k, t), \quad (31)$$

with  $\mathbf{U}(k, t) = \mathbf{E}(k)\mathbf{\Lambda}(k)^{0.5} e^{j\varphi(k,t)}$ . The matrices  $\mathbf{\Lambda}(k)$  and  $\mathbf{E}(k)$  are the eigenvalues and eigenvectors of the spatial coherence matrix  $\mathbf{\Gamma}(k)$  for a spherical sound field [40]. The  $M \times 1$  vector  $\varphi(k, t)$  denotes a uniformly distributed phase between  $-\pi, \dots, \pi$ . It can easily be seen that the PSD matrix of  $\mathbf{X}_n(k, t)$  has the properties of a spherical sound field, i.e.  $\mathbb{E}\{\mathbf{X}_n(k)\mathbf{X}_n^H(k)\} = \mathbf{\Gamma}(k)\Phi_{X_n X_n}(k)$ , where  $\Phi_{X_n X_n}(k)$  is the power spectrum of the monaural recording  $X_n(k, t)$ .

## C. Training and Testing

For training, we use 12776 utterances from the *si\_tr\_s* set of the WSJ0 [36] corpus for the speech sources in Eq. (3), and 20 hours of 20 different sound categories from YouTube [48] as isotropic background noise.<sup>1</sup> All recordings are sampled at 16 kHz, and converted to the frequency domain with  $K = 513$  bins and 75% overlapping blocks. The sources in Eq. (1) are mixed with equal volume. For testing, we use 2907 utterances from the *si\_et* set of the WSJ0 corpus mixed with another 20 hours of Youtube noise of 20 different categories.

Each of the five scenarios in Table II is tested against the four features introduced in Table I. The resulting 20 experiments demonstrate the performance of each feature in different environments. For each experiment, a separate Eigennet is trained. Model optimization is done using stochastic gradient descent with ADAM [49]. We use the weighted cross-entropy between the optimal binary mask  $p_{opt}$  and the estimated mask  $p_{est}$  of the respective model as loss function [45], i.e.

$$\mathcal{L} = \frac{\sum_{k=1}^K \sum_{t=1}^T \|\mathbf{Z}(k, t)\|_2^2 \mathcal{L}_{cce}(k, t)}{\sum_{k=1}^K \sum_{t=1}^T \|\mathbf{Z}(k, t)\|_2^2}, \quad (32)$$

and

$$\mathcal{L}_{cce}(k, t) = - \sum_{i=1}^3 p_{opt}(k, l, i) \log(p_{est}(k, l, i)). \quad (33)$$

The LSTM units are trained using *back-propagation through time* [50]. To regularize the NN, *batch normalization* is performed along the time axis of the features for each frequency bin [51]. To avoid overfitting, we use early stopping by observing the error on the validation set every 20 epochs.

<sup>1</sup>The noise categories are: highway, TV, office, restaurant, etc.

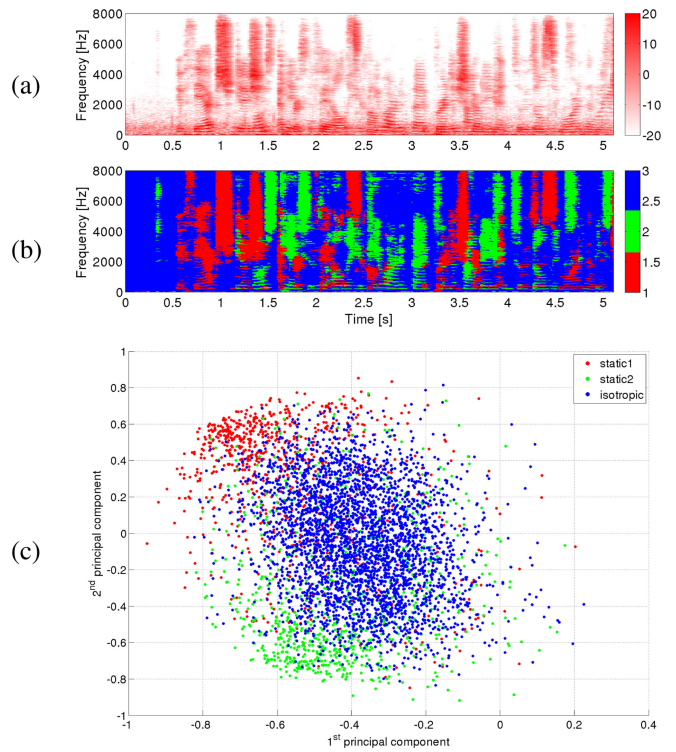


Fig. 5. Visualization of the Evs feature. (a) spectrogram of the noisy mixture. (b) optimal binary masks. (c) PCA plot of  $\mathbf{v}_{Z,i}(k, t)$ .

## D. Performance Evaluation

For each of the resulting gain masks, three different beamformers are determined: the MVDR, GEV-BAN and GEV-PAN (see Section III). The required PSD matrices are obtained using Eq. (22), where  $L = 32$  blocks. As a baseline, we use the *BeamformIt* toolkit [35]. It performs DOA estimation [8] followed by a MVDR beamformer. Hence, BeamformIt will fail to separate multiple speakers, as there is no prior information about the target speaker's location. To evaluate the performance of the enhanced signals  $Y(k, t)$ , we use the Google Speech-to-Text API [38] to perform *Automatic Speech Recognition* (ASR). Further, we measure the improvement in SNR with:

$$\Delta SNR = 10 \log_{10} \frac{\sum_{K,L} |Y(k, t)p_{opt}(k, l, 1)|^2}{\sum_{K,L} |Y(k, t)p_{opt}(k, l, 2)|^2} - 10 \log_{10} \frac{\sum_{K,L} \|\mathbf{Z}(k, t)p_{opt}(k, l, 1)\|_2^2}{\sum_{K,L} \|\mathbf{Z}(k, t)p_{opt}(k, l, 2)\|_2^2}, \quad (34)$$

where we use the optimal binary mask  $p_{opt}$  to measure the energy of the desired and interfering components in the beamformer output  $Y(k, t)$ , and the noisy inputs  $\mathbf{Z}(k, t)$ , respectively. This allows to calculate the  $\Delta SNR$  without having access to the beamforming weights  $\mathbf{W}(k, t)$ , as is the case with the BeamformIt toolkit.

## E. Visualization of the Eigenvectors

Fig. 5(a) shows the magnitude spectrogram for a single utterance of the  $S_1$ - $S_2$ I experiment. Panel (b) shows the optimal

TABLE III  
WSJ0, WER IN [%]

Experiment	Beamformer	Scenario				
		S <sub>1</sub> -I	R-I	D <sub>1</sub> -I	S <sub>1</sub> -S <sub>2</sub> I	D <sub>1</sub> -D <sub>2</sub> I
Mag	MVDR	12.899	19.197	27.560	95.634	97.718
	GEV-BAN	9.187	13.397	19.530	93.172	96.929
	GEV-PAN	10.175	11.458	17.576	82.940	88.806
Evd	MVDR	21.791	20.800	39.596	89.839	98.996
	GEV-BAN	14.771	14.957	30.038	84.108	98.011
	GEV-PAN	13.639	12.743	23.486	73.146	88.664
Evs	MVDR	10.767	11.161	18.653	18.410	30.594
	GEV-BAN	7.987	8.662	13.291	14.497	22.161
	GEV-PAN	9.304	8.958	14.298	17.256	24.279
EvsMag	MVDR	10.780	11.501	19.264	17.845	26.439
	GEV-BAN	7.946	9.001	13.996	14.544	19.612
	GEV-PAN	9.482	9.351	14.957	16.047	19.670
BeamformIt	MVDR	22.765	21.388	22.950	84.687	80.903

TABLE IV  
WSJ0,  $\Delta$ SNR IN [dB]

Experiment	Beamformer	Scenario				
		S <sub>1</sub> -I	R-I	D <sub>1</sub> -I	S <sub>1</sub> -S <sub>2</sub> I	D <sub>1</sub> -D <sub>2</sub> I
Mag	MVDR	7.174	5.644	5.333	1.289	0.408
	GEV-BAN	7.971	6.477	6.304	1.758	0.588
	GEV-PAN	8.303	7.138	7.094	1.431	0.725
Evd	MVDR	5.843	5.823	5.469	1.029	0.390
	GEV-BAN	6.887	6.746	6.629	1.649	-0.072
	GEV-PAN	7.421	7.251	7.288	1.430	0.225
Evs	MVDR	7.467	7.105	7.056	9.716	8.954
	GEV-BAN	8.203	7.900	8.044	11.387	12.231
	GEV-PAN	8.473	8.252	8.467	11.749	12.841
EvsMag	MVDR	7.930	7.299	7.619	10.388	10.756
	GEV-BAN	8.501	8.035	8.414	11.601	12.776
	GEV-PAN	8.683	8.322	8.700	11.872	13.207
BeamformIt	MVDR	-0.460	-0.567	-0.185	-0.202	0.354

binary masks  $p_{opt}(k, t, i)$  for the three sources, colored with the respective class label  $i = \{1, 2, 3\}$ . Panel (c) shows the first two principal components of the Evs feature  $v_{Z,i}(k, t)$ , colored accordingly. It can be seen that the three classes form clusters based on the location of their respective sources. Clearly, those features are well suited to separate multiple multiple speakers from a noisy mixture. Note that this plot shows data points for a single frequency of  $\approx 550$  Hz. To obtain 5000 data points, several utterances have been concatenated along the time axis.

## F. Results

1) *ASR Performance*: Table III shows the WER obtained by the Google-ASR system using WSJ0 data. For experiments with more than one dominant source (i.e. S<sub>1</sub>-S<sub>2</sub>I and D<sub>1</sub>-D<sub>2</sub>I) the Mag and Evd features and the BeamformIt method fail, while the Evs and EvsMag methods provide reasonable results. This is to be expected, as spatial features are required to separate multiple sources. Further, it can be seen that combining magnitude and spatial features (EvsMag) does not increase the performance significantly. Consequently, the spatial information contributes a major part to the results. Regarding the beamformers, the GEV-BAN beamformer gives the best results in terms of WER. As expected, the MVDR performs poor due to numerical problems. This has also been reported in [24].

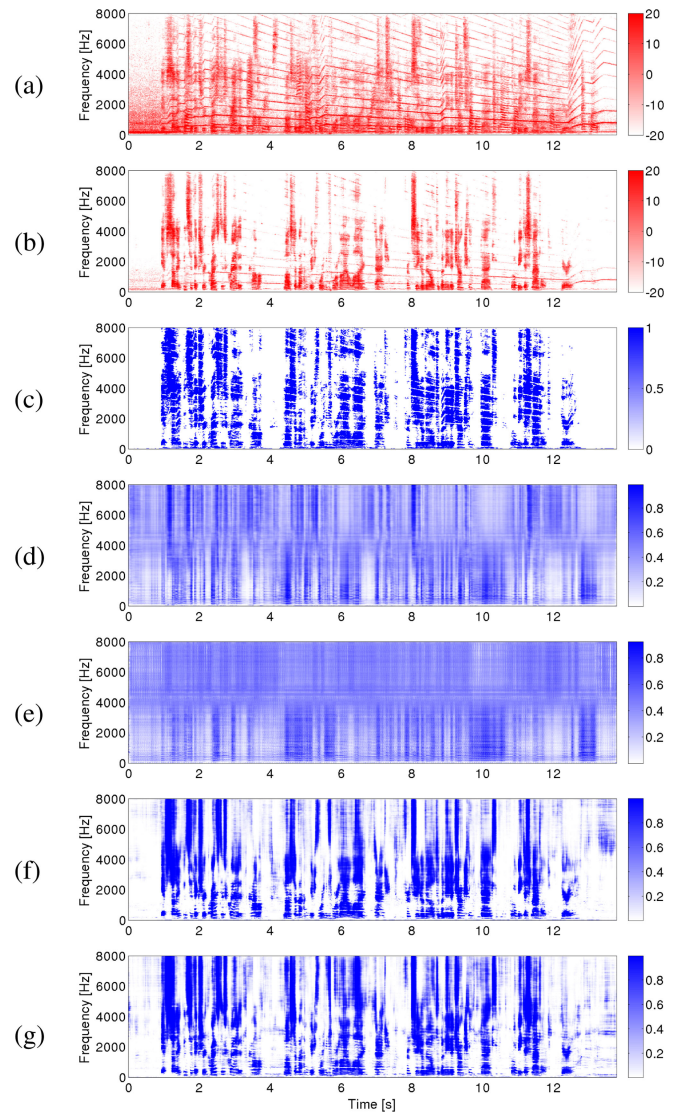


Fig. 6. Estimated masks for the WSJ0 utterance 440e090x. Panel (a) shows the noisy mixture, (b) the enhanced signal using the EvsMag features and the GEV-BAN beamformer, (c) optimal binary mask, (d) estimated mask using the Evd, (e) Mag, (f) Evs, (g) EvsMag.

2) *SNR Performance*: Table IV shows the SNR improvement after beamforming. Similar to the WER scores, improvement in SNR is poor for scenarios with more than one dominant source and the Mag or Evd features, or the BeamformIt method. The GEV-BAN beamformer yields the best scores in this category.

Fig. 6(a) shows the noisy mixture of the utterance 440e090x from the WSJ0 test set, as seen by the first microphone, i.e.  $Z_1(k, t)$ . The scenario is S<sub>1</sub>-S<sub>2</sub>I, where a second speaker and ambient noise are the interfering signals. Panel (b) shows the enhanced output  $Y(k, t)$  using the EvsMag features and the GEV-BAN beamformer. It can be seen that the interfering noise energy (the siren of a fire truck, and another speaker) is significantly reduced. Panel (c) shows the optimal binary mask for the desired speech source, i.e.  $p_{opt}(k, t, 1)$ . Panel (d) shows the estimated mask  $p_{est}(k, t, 1)$  using the Evd features, and Panel (e) shows the estimated mask using the Mag features. It can be



seen that neither the `Evd` nor the `Mag` feature is able to separate multiple speakers. Panel (f) shows the estimated mask using `Evs` features, and Panel (g) shows the estimated mask using `EvsMag` features. Both masks closely match the ideal one, i.e. those features are able to separate multiple speakers.

## VI. EXPERIMENTS USING CHiME4

### A. Experimental Setup

To test the performance of the Eigennet on real-world recordings, we used the CHiME4 corpus [21]. It provides 6-channel recordings of multiple speakers corrupted by four different types of ambient noise: *pedestrian*, *street*, *cafe* and *bus*. The recordings are sampled at 16 kHz. The ground truth (i.e. the separated speech and noise signals) is available for all recordings via a close-talking microphone (channel0). It is used to augment the database by *simulated* data, which is obtained by mixing different noises with clean speech, convolved by delay-only ATFs (see Eq. (11)). The training set comprises 1600 real and 7138 simulated utterances. The development set consists of 1640 real and 1640 simulated utterances. And the test set is composed of 1320 real and 1320 simulated utterances. There is also a 2-channel track, where 2 out of the 6 microphones have been chosen randomly, so that the array geometry is unknown for each utterance. For more details on the CHiME4 data the interested reader is referred to [21]. We use a STFT size of 1024 samples (32 ms) and an overlap of 75% to process the audio data.

### B. Training and Testing

For training, we use the 8738 utterances from the `train` set. The groundtruth of the `real` utterances contains a small amount of background interference, due to the use of a close-talking microphone. However, this influence is negligible as we use binary masks as labels. We test the 2 and 6 channel tracks against the `Evd` and `EvsMag` features introduced in Table I. Optimization of the Eigennet is identical to the WSJ0 experiments.

### C. Performance Evaluation

For each of the resulting gain masks, three different beamformers are computed: the `MVDR`, `GEV-BAN` and `GEV-PAN`. We use the `BeamformIt` toolkit as baseline [35]. Further, we compare our models against two *state-of-the-art* reference systems:

1) *Cgmm-em*: The first reference system uses a CGMM to estimate the speech and noise PSD matrices. The noise PSD matrix is initialized from a patch of noise-only data that precedes each CHiME4 utterance. The model parameters are estimated with an EM algorithm, and the posterior probabilities are used as gain masks for the speech and noise components [29].

2) *nn-gev*: The second reference system uses a NN with four fully-connected layers [24]. The first layer is a BLSTM, while the remaining layers are MLPs. For each of the six microphone channels, independent masks for speech and noise are estimated. The resulting masks are combined into a single mask for speech and noise using the median operator. The NN is trained on magnitude features similar to  $x_{\text{Mag}}(k, t)$ .

TABLE V  
CHiME4, 6-CHANNEL, KALDI-WER IN [%]

Experiment	Beamformer	development		test	
		simu	real	simu	real
Evd	MVDR	3.81	3.89	6.72	9.11
	GEV-BAN	3.20	3.79	4.90	6.47
	GEV-PAN	3.68	3.58	4.98	6.38
EvsMag	MVDR	3.43	3.90	5.50	8.22
	GEV-BAN	2.84	3.82	4.09	6.18
	GEV-PAN	3.29	3.57	4.69	6.24
BeamformIt	MVDR	4.60	3.80	7.56	6.71
nn-gev	GEV-BAN	3.39	3.68	4.27	6.03
cgmm-em	MVDR	4.04	3.59	4.88	6.17

TABLE VI  
CHiME4, 6-CHANNEL, GOOGLE-WER IN [%]

Experiment	Beamformer	development		test	
		simu	real	simu	real
Evd	MVDR	9.016	6.770	14.218	18.830
	GEV-BAN	7.850	6.680	11.541	15.640
	GEV-PAN	8.837	6.509	13.231	17.338
EvsMag	MVDR	8.263	6.806	12.794	17.964
	GEV-BAN	7.390	6.692	10.176	13.901
	GEV-PAN	8.483	6.813	12.089	17.547
BeamformIt	MVDR	11.252	7.984	18.961	20.514
nn-gev	GEV-BAN	7.805	6.158	10.162	16.033
cgmm-em	MVDR	10.070	7.743	13.901	20.793

TABLE VII  
CHiME4, 6-CHANNEL,  $\Delta$ SNR IN [dB]

Experiment	Beamformer	development		test	
		simu	real	simu	real
Evd	MVDR	5.642	2.754	4.403	2.925
	GEV-BAN	7.725	6.249	7.612	6.408
	GEV-PAN	7.750	6.290	7.700	6.483
EvsMag	MVDR	6.719	3.052	5.167	3.192
	GEV-BAN	7.845	6.217	7.721	6.047
	GEV-PAN	7.854	6.246	7.775	6.091
BeamformIt	MVDR	2.101	-1.118	3.752	0.673
nn-gev	GEV-BAN	7.657	5.467	7.624	5.452
cgmm-em	MVDR	4.258	2.231	3.679	2.402

To evaluate the performance of the enhanced signals  $Y(k, t)$ , an *Automatic Speech Recognition* (ASR) baseline system is provided as part of the CHiME4 challenge [52]. It uses the Kaldi speech recognition toolkit [39]. Additionally, we report the WER obtained by Google Speech-to-Text API [38], and the improvement in SNR using Eq. (34).

### D. Results

1) *ASR Performance*: Tables V and VI show the WER for the Kaldi and Google ASR systems using CHiME4 data. It can be seen that the `Evd` and `EvsMag` features, and the `nn-gev` system show the best performance in combination with the `GEV-BAN` beamformer. As there is only one dominant source in the CHiME4 data, the `Evd` and `EvsMag` features show comparable results.

2) *SNR Performance*: Table VII shows the SNR improvement after beamforming. Again, the beamformers using the `Evd` and `EvsMag` features, as well as the `nn-gev` system exhibit a

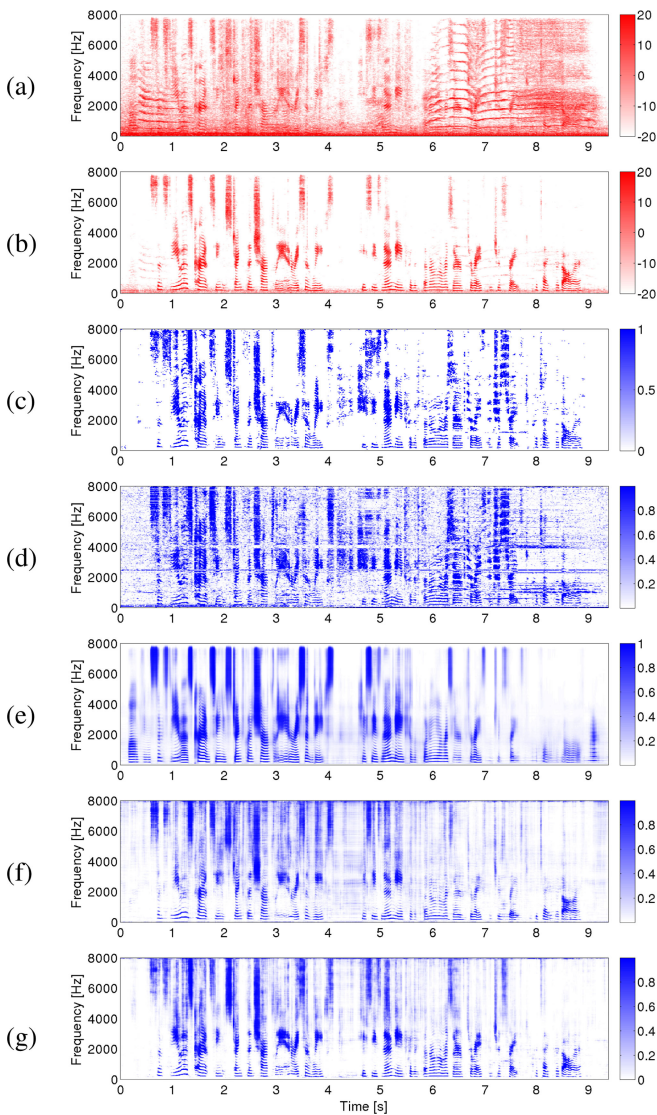


Fig. 7. Estimated masks for the CHiME4 utterance F01-22GC010X-BUS. Panel (a) shows the noisy mixture, (b) the enhanced signal using the EvsMag features and the GEV-BAN beamformer, (c) optimal binary mask, (d) estimated mask for the cgmm-em system, which is quite noisy, (e) estimated mask for the nn-gev system. It looks very clean, but is wrong at both ends of the utterance. Panel (f) shows the estimated mask using Evd features, and Panel (g) shows the estimated mask using EvsMag features. Both masks look similar.

similar performance. Here, the GEV-PAN beamformer provides the best results.

Fig. 7(a) shows the noisy mixture of the utterance F01-22GC010X-BUS from the CHiME4 test set, as seen by the first microphone, i.e.  $Z_1(k, t)$ . Panel (b) shows the enhanced output  $Y(k, t)$  using the EvsMag features and the GEV-BAN beamformer. It can be seen that the interfering noise energy (bus noise and a screeching baby) is significantly reduced. Panel (c) shows the optimal binary mask for the desired speech source, i.e.  $p_{opt}(k, t, 1)$ . Panel (d) shows the estimated mask  $p_{est}(k, t, 1)$  for the cgmm-em system, which is quite noisy. Panel (e) shows the estimated mask for the nn-gev system. It looks very clean, but is wrong at both ends of the utterance. Panel (f) shows the estimated mask using Evd features, and Panel (g) shows the estimated mask using EvsMag features. Both masks look similar.

TABLE VIII  
CHiME4 2-CHANNEL, GOOGLE-WER IN [%]

Experiment	Beamformer	development		test	
		simu	real	simu	real
Evd	MVDR	12.971	8.726	17.658	20.417
	GEV-BAN	12.247	8.319	16.969	20.025
	GEV-PAN	12.332	8.479	16.628	19.803
EvsMag	MVDR	12.569	8.646	16.688	19.845
	GEV-BAN	12.460	8.515	16.381	19.447
	GEV-PAN	12.314	8.599	16.038	19.296
BeamformIt	MVDR	14.613	10.505	22.784	25.809
	nn-gev	12.400	8.858	16.497	20.751
	cgmm-em	13.651	9.888	17.973	23.810

TABLE IX  
CHiME4 2-CHANNEL, KALDI-WER IN [%]

Experiment	Beamformer	development		test	
		simu	real	simu	real
Evd	MVDR	5.76	5.07	8.84	10.16
	GEV-BAN	5.51	5.65	7.47	9.82
	GEV-PAN	5.85	4.97	7.45	9.56
EvsMag	MVDR	5.63	4.88	8.57	9.83
	GEV-BAN	5.54	5.00	7.33	9.07
	GEV-PAN	5.61	5.13	7.56	9.12
BeamformIt	MVDR	6.71	5.97	11.87	11.96
nn-gev	GEV-BAN	5.65	5.03	7.22	9.34
cgmm-em	MVDR	6.27	8.44	10.31	16.52

TABLE X  
CHiME4 2-CHANNEL,  $\Delta$ SNR IN [dB]

Experiment	Beamformer	development		test	
		simu	real	simu	real
Evd	MVDR	2.397	0.929	2.185	1.024
	GEV-BAN	3.418	2.550	3.623	2.413
	GEV-PAN	3.436	2.573	3.675	2.446
EvsMag	MVDR	2.747	0.973	2.496	1.100
	GEV-BAN	3.390	2.239	3.846	2.346
	GEV-PAN	3.400	2.251	3.888	2.371
BeamformIt	MVDR	0.309	-1.427	1.226	-0.073
nn-gev	GEV-BAN	3.140	1.823	3.393	1.797
cgmm-em	MVDR	1.683	0.650	2.016	0.658

3) *2 Channel Track*: Tables VIII to X show the WER and  $\Delta$ SNR results for the 2-channel CHiME4 track. For both scores, the EvsMag features yield the best performance.

## VII. CONCLUSION

In this paper, we have introduced our Eigennet architecture for estimating gain masks from noisy, multi-channel microphone observations. While many mask estimators use magnitude features, the proposed eigenvector features also exploit the spatial information embedded in the phase of the data. The obtained gain masks are used to construct the MVDR or GEV beamformer. We derived the PAN postfilter, which corrects both magnitude and phase distortions caused by the GEV. We tested our approach on the WSJ0 and CHiME4 datasets, where we demonstrated the benefits of using spatial features over magnitude information alone. We further reported the improvement in SNR, as well as the WER obtained by the Google-ASR and the Kaldi-ASR systems. The Eigennet architecture yields competitive results compared to state-of-the-art mask estimators.

## ACKNOWLEDGMENT

The authors acknowledge NVIDIA for providing GPU computing resources.

## REFERENCES

- [1] B. D. V. Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Mag.*, vol. 5, no. 2, pp. 4–24, Apr. 1988.
- [2] O. Hoshuyama, A. Sugiyama, and A. Hirano, "A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2677–2684, Oct. 1999.
- [3] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1614–1626, Aug. 2001.
- [4] W. Herbordt and W. Kellermann, "Analysis of blocking matrices for generalized sidelobe cancellers for non-stationary broadband signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, May 2002, pp. I-4187–IV-4187.
- [5] S. Gannot and I. Cohen, "Speech enhancement based on the general transfer function GSC and postfiltering," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 6, pp. 561–571, Nov. 2004.
- [6] M. G. Shmulik, S. Gannot, and I. Cohen, "Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 6, pp. 1071–1086, Aug. 2009.
- [7] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 5, pp. 1529–1539, Jul. 2007.
- [8] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Berlin, Germany: Springer, 2008.
- [9] L. Pfeifenberger and F. Pernkopf, "Blind source extraction based on a direction-dependent a-priori SNR," in *Proc. INTERSPEECH*, May 2014, pp. 2700–2704.
- [10] R. Talmon, I. Cohen, and S. Gannot, "Relative transfer function identification using convolutive transfer function approximation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 4, pp. 546–555, May 2009.
- [11] I. Cohen, "Relative transfer function identification using speech signals," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 5, pp. 451–459, Sep. 2004.
- [12] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Berlin, Germany: Springer, 2008.
- [13] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and Andrew, "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, Dec. 2015, pp. 30–36.
- [14] T. N. Sainath *et al.*, "Multichannel signal processing with deep neural networks for automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 5, pp. 965–979, May 2017.
- [15] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "BLSTM supported GEV beamformer front-end for the 3rd CHiME challenge," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, Dec. 2015, pp. 444–451.
- [16] L. Deng, M. Seltzer, D. Yu, A. Acero, A. Mohamed, and G. Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *Proc. INTERSPEECH*, 2010, pp. 1692–1695.
- [17] T. Schrank, L. Pfeifenberger, M. Zöhrer, J. Stahl, P. Mowlae, and F. Pernkopf, "Deep beamforming and data augmentation for robust speech recognition: Results of the 4th CHiME challenge," in *Proc. 4th Intl. Workshop Speech Process. Everyday Environ.*, 2016, pp. 18–20.
- [18] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, "DNN-based speech mask estimation for eigenvector beamforming," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2017, pp. 66–70.
- [19] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, "Eigenvector-based speech mask estimation using logistic regression," in *Proc. 18th Annu. Conf. Int. Speech Commun. Assoc.*, Aug. 2017, pp. 2660–2664.
- [20] M. Zöhrer, L. Pfeifenberger, G. Schindler, H. Fröning, and F. Pernkopf, "Resource efficient deep eigenvector beamforming," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2018, pp. 3354–3358.
- [21] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2015, pp. 504–511.
- [22] Y. Wang and D. Wang, "Cocktail party processing via structured prediction," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 224–232.
- [23] F. Weninger, J. L. Roux, J. R. Hershey, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *Proc. Global Conf. Signal Inf. Process.*, Dec. 2014, pp. 577–581.
- [24] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2016, pp. 196–200.
- [25] H. Erdogan, J. Hershey, S. Watanabe, M. Mandel, and J. L. Roux, "Improved MVDR beamforming using single-channel mask prediction networks," in *Proc. INTERSPEECH*, Sep. 2016, pp. 504–511.
- [26] Z. Meng, S. Watanabe, J. R. Hershey, and H. Erdogan, "Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2017, pp. 271–275.
- [27] H. Erdogan *et al.*, "Multi-channel speech recognition: LSTMs all the way through," in *Proc. CHiME 4 Workshop*, 2016, pp. 45–48.
- [28] M. Souden, J. Chen, J. Benesty, and S. Affes, "An integrated solution for online multichannel noise tracking and reduction," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2159–2169, Sep. 2011.
- [29] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, Mar. 2016, pp. 5210–5214.
- [30] T. Yoshioka, H. Erdogan, Z. Chen, X. Xiao, and F. Alleva, "Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks," in *Proc. INTERSPEECH*, Sep. 2018, pp. 3038–3042.
- [31] H. Sawada, S. Araki, and S. Makino, "Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 3, pp. 516–527, Mar. 2011.
- [32] Z. Wang and D. Wang, "Combining spectral and spatial features for deep learning based blind speaker separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 2, pp. 457–468, Feb. 2019.
- [33] Z. Wang, X. Zhang, and D. Wang, "Robust speaker localization guided by deep learning-based time-frequency masking," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 1, pp. 178–188, Jan. 2019.
- [34] P. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2014, pp. 1562–1566.
- [35] X. Anguera, C. Wooters, and J. Hernandez, "Acoustic beamforming for speaker diarization of meetings," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 7, pp. 2011–2021, Sep. 2007.
- [36] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Workshop Speech Natural Lang.*, Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 357–362.
- [37] E. A. P. Habets and S. Gannot, "Generating sensor signals in isotropic noise fields," *J. Acoustical Soc. Amer.*, vol. 122, no. 6, pp. 3464–3470, 2007.
- [38] "SpeechRecognition – a library for performing speech recognition, with support for several engines and apis, online and offline," 2018. [Online]. Available: <https://pypi.org/project/SpeechRecognition/>
- [39] D. Povey *et al.*, "The kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, Big Island, HI, USA, 2011.
- [40] H. Kuttruff, *Room Acoustics*, 5th ed. London, U.K.: Spon Press, 2009.
- [41] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. Berlin, Germany: Springer, 2006.
- [42] M. Brandstein and D. Ward, *Microphone Arrays*. Berlin, Germany: Springer, 2001.
- [43] E. Warsitz, A. Krueger, and R. Haeb-Umbach, "Speech enhancement with a new generalized eigenvector blocking matrix for application in a generalized sidelobe canceller," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2008, pp. 73–76.
- [44] C. Böddeker, H. Erdogan, T. Yoshioka, and R. Haeb-Umbach, "Exploring practical aspects of neural mask-based beamforming for far-field speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 6697–6701.
- [45] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.
- [46] M. Zöhrer, R. Peharz, and F. Pernkopf, "Representation learning for single-channel source separation and bandwidth extension," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 12, pp. 2398–2409, Dec. 2015.



- [47] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: A python package for audio room simulations and array processing algorithms," 2017, *arXiv:1710.04196*.
- [48] "PyTube – a lightweight, pythonic, dependency-free, library for downloading youtube videos," 2018. [Online]. Available: <https://python-pytube.readthedocs.io/en/latest/>
- [49] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, Jul. 2015.
- [50] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," Oct. 2015, *arXiv:1506.00019*.
- [51] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, 2015, vol. 37, pp. 448–456.
- [52] S. Chen, A. S. Subramanian, H. Xu, and S. Watanabe, "Building state-of-the-art distant speech recognition using the CHiME-4 challenge with a setup of speech enhancement baseline," 2018, *arXiv:1803.10109*.



**Lukas Pfeifenberger** received the M.Sc. (Dipl. Ing. FH) degree in computer science from the University of Applied Sciences, Salzburg, Austria, in 2004. In 2013, he received the M.Sc. (Dipl. Ing.) degree in Telematik, Graz University of Technology, Graz, Austria. Since 2005 he has been working in the electronics industry on projects pertaining to FPGA design, DSP programming, and communication acoustics. Since 2015 he has been a Research Associate with the Laboratory of Signal Processing and Speech Communication, Graz University of Technol-

ogy, Graz, Austria. His research interests include machine learning, computer vision, and speech enhancement. He currently pursues research projects in blind source separation and acoustic echo control.



**Matthias Zöhrer** received the M.Sc. (Dipl. Ing.) degree in Telematik from the Graz University of Technology, Graz, Austria, in summer 2013. Since 2013 he has been a Research Associate with the Laboratory of Signal Processing and Speech Communication, Graz University of Technology, Graz, Austria. His research interests include machine learning, representation learning, deep learning architectures, GPU optimized processing, and image- and speech-processing applications.



**Franz Pernkopf** (SM'14) received the M.Sc. (Dipl. Ing.) degree in electrical engineering from the Graz University of Technology, Graz, Austria, in summer 1999. He earned the Ph.D. degree from the University of Leoben, Leoben, Austria, in 2002. In 2002, he was awarded the Erwin Schrödinger Fellowship. He was a Research Associate with the Department of Electrical Engineering, University of Washington, Seattle, WA, USA, from 2004 to 2006. From 2010 to 2019, he was an Associate Professor with the Laboratory of Signal Processing and Speech Communication, Graz

University of Technology, Graz, Austria. Since 2019, he has been a Professor for Intelligent Systems with the Signal Processing and Speech Communication Laboratory, Graz University of Technology, Graz, Austria. His research is focused on pattern recognition, machine learning, and computational data analytics with applications in various fields ranging from signal and speech processing to medical data analysis and other data-modeling problems from industrial applications. He is particularly interested in probabilistic graphical models for reasoning under uncertainty, discriminative and hybrid learning paradigms, deep neural networks, and sequence modeling.

## **A.8 IEEE/ICASSP 2019**

*"Deep complex-valued neural beamformers"*, Lukas Pfeifenberger, Matthias Zöhrer and Franz Pernkopf, The 44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, 2019



# DEEP COMPLEX-VALUED NEURAL BEAMFORMERS

Lukas Pfeifenberger<sup>1</sup>, Matthias Zöhrer<sup>1</sup>, Franz Pernkopf

Signal Processing and Speech Communication Lab  
Graz University of Technology

## ABSTRACT

We propose a complex-valued deep neural network (cDNN) for speech enhancement and source separation. While existing *end-to-end* systems use complex-valued gradients to pass the training error to a real-valued DNN used for gain mask estimation, we use the full potential of complex-valued LSTMs, MLPs and activation functions to estimate complex-valued beamforming weights directly from complex-valued microphone array data. By doing so, our cDNN is able to locate and track different moving sources by exploiting the phase information in the data. In our experiments, we use a typical living room environment, mixtures of the WallStreet Journal corpus, and YouTube noise. We compare our cDNN against the BeamformIt toolkit as a baseline, and a mask-based beamformer as a state-of-the-art reference system. We observed a significant improvement in terms of PESQ, STOI and WER.

**Index Terms**— beamforming, complex-valued deep neural networks, Wirtinger Calculus

## 1. INTRODUCTION

Recent contributions to data-driven beamforming propose a DNN to estimate a spectral gain mask from noisy, multi-microphone speech signals. This mask is used to obtain the power spectral density (PSD) matrices of the desired and interfering sound sources. With those PSD estimates, statistical beamformers such as the *Minimum Variance Distortionless Response* (MVDR) beamformer [1] or the *Generalized Eigenvalue* (GEV) beamformer [2] are used to estimate the desired signal. DNN-based gain mask estimators have been proposed in [3, 4, 5]. As those approaches use magnitude spectrograms as features, they do not exploit the spatial information contained in the phase of the data. In [6, 7], we circumvent this limitation by using the eigenvectors of the short-time PSD matrix of the noisy speech as features. This allows for a significantly smaller DNN to estimate the gain mask, with comparable performance in both ASR results and perceptual speech quality [7]. However, mask-based beamforming requires an entire block of audio data at a time. During this

period, the signal statistics are assumed to be constant. This limits the capability to track moving sound sources. An attempt towards online processing has been proposed in [8], where the PSD matrices are recursively estimated.

With recent trends towards *end-to-end* ASR systems, the DNN-based mask estimator, the beamformer and the acoustic front-end of the ASR system are combined into a fully interconnected model. This allows to back-propagate the training error from the acoustic modelling cost function through the beamformer and the DNN-based mask estimator [9, 10, 11, 12]. As beamforming involves non-holomorphic functions (i.e. conjugation or absolute value), their gradients do not exist. A widely adopted solution for this problem is to split complex-valued functions into their *real* and *imaginary* parts, and treat them like real-valued functions. However, this results in losing important properties like complex rotation or symmetry. Using *Wirtinger Calculus*, it is possible to derive complex-valued gradients from non-holomorphic functions with respect to a real-valued variable [13, 14, 15].

While end-to-end systems make use of the complex-valued gradient of statistical beamformers, they still use a real-valued DNN to estimate the gain mask. We aim to explore the full potential of complex-valued gradients and propose a fully complex DNN (cDNN) beamformer, with complex LSTM and MLP layers, as well as complex-valued activation functions. By doing so, we do not need to rely on a gain mask, as the cDNN is able to predict complex-valued beamforming weights directly from complex-valued microphone signals. Unlike a statistical beamformer, such a model estimates a set of optimal beamforming weights for each time-frequency bin. This leverages the source tracking and separation performance. To demonstrate the capabilities of our cDNN, we perform simulations involving moving and static sound sources in a typical living room setup, using mixtures of the WallStreet Journal corpus (WSJ) [16] and YouTube noise [17]. We compare the performance of the cDNN against a baseline using *BeamformIt* [18] and a reference system using a mask-based beamformer [6] with online tracking [8]. We further report performance metrics, i.e.  $\Delta$ SNR, PESQ [19], STOI [20], as well as WER scores. Compared to the mask-based beamformer, the proposed system reaches an average relative improvement of 58.47% WER.

<sup>1</sup> Both authors contributed equally.

This work was supported by the Austrian Science Fund (FWF) under the project number I2706-N31 and NVIDIA for providing GPUs.

## 2. COMPLEX-VALUED MULTI LAYER PERCEPTRONS

A complex-valued MLP (cMLP) is defined analogously to its real-valued counterpart. i.e.

$$\mathbf{h}^{(t)} = g(\mathbf{W}_h \mathbf{z}^{(t)} + \mathbf{b}_h), \quad (1)$$

where  $\mathbf{z}^{(t)}$  denotes the input, and  $\mathbf{W}_h$  and  $\mathbf{b}_h$  are the internal weights and biases, respectively. All variables are defined over  $\mathbb{C}$ . Based on recent contributions on complex-valued neural networks [21, 22, 23], we propose the non-linear complex-valued activation function  $g(\cdot)$  as natural extension of a real-valued  $\tanh$  unit, i.e.

$$g(\mathbf{z}) = \tanh(|\mathbf{z}|) \odot \frac{\mathbf{z}}{|\mathbf{z}|}, \quad (2)$$

where  $\odot$  denotes element-wise multiplication. The function  $g(\mathbf{z})$  is symmetric, with a magnitude bounded by 1.0. The phase of  $\mathbf{z}$  is not modified. For comparison, we demonstrate the behavior of a  $\tanh$  activation function with non-complex gradients (i.e. the real and imaginary parts are stacked and treated as individual values). It is given as  $g_2(\mathbf{z}) = \tanh(\text{Re}\{\mathbf{z}\}) + i \tanh(\text{Im}\{\mathbf{z}\})$ . Figure 1 shows the magnitude and phase response of  $g(\mathbf{z})$  in panel (a) and (b), and the magnitude and phase response of  $g_2(\mathbf{z})$  in panel (c) and (d), respectively. It can be seen that  $g_2(\mathbf{z})$  is not bounded to 1.0. It also modifies the phase to a constant value per quadrant of the complex plane.

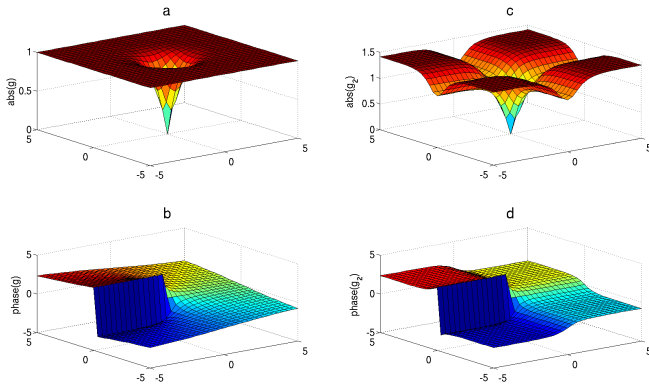


Fig. 1: Magnitude and phase of  $g(\mathbf{z})$  and  $g_2(\mathbf{z})$ .

## 3. COMPLEX-VALUED LONG SHORT TERM MEMORY NETWORKS

In complex-valued LSTMs (cLSTM) the input  $\mathbf{i}^{(t)}$ , forget  $\mathbf{f}^{(t)}$  and output  $\mathbf{o}^{(t)}$  gate are calculated as follows:

$$\mathbf{i}^{(t)} = \sigma\left(\text{Re}\left\{\mathbf{W}_{zi}\mathbf{z}^{(t)} + \mathbf{W}_{hi}\mathbf{h}^{(t-1)} + \mathbf{b}_i\right\}\right), \quad (3a)$$

$$\mathbf{f}^{(t)} = \sigma\left(\text{Re}\left\{\mathbf{W}_{zf}\mathbf{z}^{(t)} + \mathbf{W}_{hf}\mathbf{h}^{(t-1)} + \mathbf{b}_f\right\}\right), \quad (3b)$$

$$\mathbf{o}^{(t)} = \sigma\left(\text{Re}\left\{\mathbf{W}_{zo}\mathbf{z}^{(t)} + \mathbf{W}_{ho}\mathbf{h}^{(t-1)} + \mathbf{b}_o\right\}\right). \quad (3c)$$

Similar to real-valued LSTMs [24], the memory cell  $\mathbf{c}^{(t)}$  is updated according to

$$\tilde{\mathbf{c}}^{(t)} = g(\mathbf{W}_{zc}\mathbf{z}^{(t)} + \mathbf{W}_{hc}\mathbf{h}^{(t-1)} + \mathbf{b}_c), \quad \text{and} \quad (4a)$$

$$\mathbf{c}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \tilde{\mathbf{c}}^{(t)}. \quad (4b)$$

The hidden state is determined as

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot g(\mathbf{c}^{(t)}). \quad (5)$$

Figure 2 shows the network graph of the resulting cLSTM. Again, all variables are defined over  $\mathbb{C}$ . Note that  $g_2(\mathbf{z})$  cannot be used in Eq. (5), as its magnitude is greater than one. This would cause the gradient of  $\mathbf{h}^{(t)}$  to grow exponentially when using back-propagation through time. The activations  $\sigma(\cdot)$  for the gating variables in Eq. (3a) - (3c) are real-valued sigmoid functions, to ensure that the gating mechanism is not altering the phase information of the input signal.

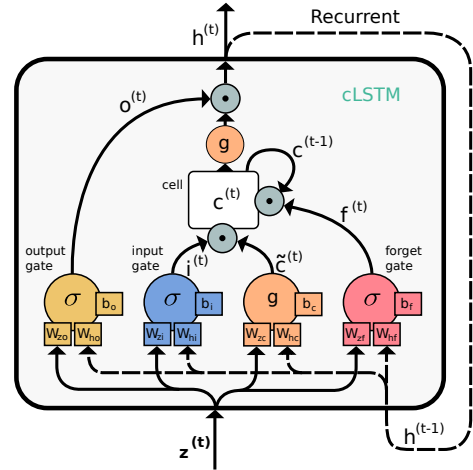


Fig. 2: Complex LSTM unit with internal connections.

We use *Wirtinger Calculus* [13, 25, 15] to obtain complex-valued gradients for each component. It allows us to iteratively apply the chain-rule to complex derivatives, i.e. complex-valued back-propagation. It can also be applied to stochastic gradient descent optimization algorithms like ADAM [26]. For further details on complex-valued back-propagation we refer the interested reader to [11].

#### 4. DEEP COMPLEX-VALUED NEURAL BEAMFORMING

The cDNN uses the complex-valued microphone samples  $Z(k, t, m)$  as features, with  $k = 1, \dots, K$  frequency bins and  $m = 1, \dots, M$  microphones. To speed up the learning process, the features are decorrelated using *Principal Component Analysis* (PCA) whitening. For each time frame  $t$ , the cDNN processes a matrix of  $M \times K$  features  $\mathbf{Z}(t)$ , and predicts a  $M \times K$  matrix of complex-valued beamforming weights  $\mathbf{W}(t)$ . The cDNN composed of three cLSTM layers and three cMLP layers with  $2MK$  neurons between each hidden layer. The beamforming step is a *filter-and-sum* operation, i.e.  $Y(k, t) = \mathbf{W}(k, t)^H \mathbf{Z}(k, t)$ . Figure 3 provides a system overview.

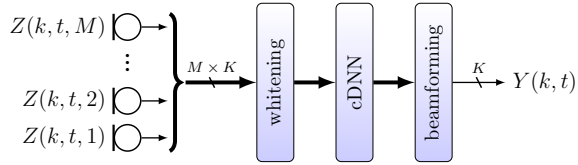


Fig. 3: System overview.

The signal arriving at the microphones is composed of an additive mixture of  $N$  sound sources, i.e.

$$\mathbf{Z}(k, t) = \sum_{n=1}^N \mathbf{S}_n(k, t), \quad (6)$$

where  $\mathbf{S}_n(k, t)$  represents the  $n^{\text{th}}$  sound source at frequency bin  $k$  and time frame  $t$ . Each sound source is composed of a monaural recording  $X_n(k, t)$  convolved with an *Acoustic Transfer Function* (ATF) denoted by  $\mathbf{A}_n(k, t)$ , i.e.

$$\mathbf{S}_n(k, t) = \mathbf{A}_n(k, t)X_n(k, t). \quad (7)$$

The ATFs model the acoustic path from a sound source to the microphones, including all reverberations and reflections caused by the room acoustics [27]. To simulate the ATFs for point sources, we use the *Image Source Method* (ISM) [28]. The living room is modeled as shoebox with a reflection coefficient of  $\beta = 0.85$  for each wall, and a reflection order of 10. This results in a reverberation time of approximately  $250ms$ . For static sources, software libraries such as [29] are readily available. For dynamic sources, we generate a new set of ATFs every  $32ms$ . We also generate an isotropic background noise using

$$\mathbf{S}_n(k, t) = \mathbf{U}(k, t)X_n(k, t), \quad (8)$$

with  $\mathbf{U}(k, t) = \mathbf{E}(k)\mathbf{\Lambda}(k)^{0.5} e^{i\varphi(k, t)}$ . The matrices  $\mathbf{\Lambda}(k)$  and  $\mathbf{E}(k)$  are the eigenvalues and eigenvectors of the spatial coherence matrix  $\mathbf{\Gamma}(k)$  for a spherical sound field [30]. The  $M \times 1$  vector  $\varphi(k, t)$  denotes a uniformly distributed phase

between  $-\pi, \dots, \pi$ . It can easily be seen that the PSD matrix of  $\mathbf{S}_n(k, t)$  has the properties of a spherical sound field, i.e.  $\mathbb{E}\{\mathbf{S}_n(k)\mathbf{S}_n^H(k)\} = \mathbf{\Gamma}(k)\Phi_{X_n X_n}(k)$ , where  $\Phi_{X_n X_n}(k)$  is the power spectrum of the monaural recording  $X_n(k, t)$ .

#### 5. EXPERIMENTAL SETUP

To test the performance of our cDNN, we simulate a typical living room scenario with two static speakers  $\mathbf{S}_1$  and  $\mathbf{S}_2$ , a TV set  $\mathbf{S}_3$ , and two moving speakers  $\mathbf{D}_1$  and  $\mathbf{D}_2$ . The dynamic paths  $\mathbf{D}_1$  and  $\mathbf{D}_2$  change randomly within a region of 2m on each side, as indicated in Figure 4. To simulate head movements of the static sources  $\mathbf{S}_1$  and  $\mathbf{S}_2$ , random position changes occur within a cube of  $20cm$  in size. We use a circular microphone array with  $M = 6$  microphones and a diameter of  $86mm$ , located next to the TV set. Within this environment, we define the five experiments given in Table 1.

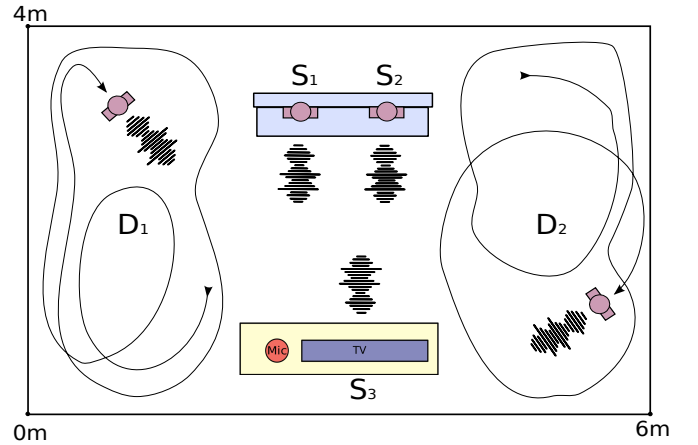


Fig. 4: Shoebox model of a living room showing stationary sound sources  $\mathbf{S}_1$  to  $\mathbf{S}_3$ , and dynamic sound sources  $\mathbf{D}_1$  and  $\mathbf{D}_2$ . The microphone array is located next to the TV set.

Experiment #	Desired source	Interfering source(s)
1	$\mathbf{D}_1$	$\mathbf{D}_2$
2	$\mathbf{D}_1$	isotropic
3	$\mathbf{S}_1$	isotropic
4	$\mathbf{S}_1$	$\mathbf{S}_3$
5	$\mathbf{S}_2$	$\mathbf{D}_1, \mathbf{S}_3$

Table 1: Experimental setups.

For each experiment, the cDNN predicts beamforming weights  $\mathbf{W}(k, t)$  which preserve the desired source  $\mathbf{S}_1(k, t)$ , and cancel out the interfering sources  $\mathbf{S}_{2 \dots N}(k, t) = \sum_{n=2}^N \mathbf{S}_n(k, t)$ . The cost function  $\mathcal{L}(k, t)$  of the cDNN is designed to maximize the  $\Delta$ SNR after applying the beamforming weights, i.e.

$$\mathcal{L}(k, t) = 10 \log_{10} \frac{|\mathbf{W}^H \mathbf{S}_1|^2}{|\mathbf{W}^H \mathbf{S}_{2 \dots N}|^2} - 10 \log_{10} \frac{\|\mathbf{S}_1\|_2^2}{\|\mathbf{S}_{2 \dots N}\|_2^2} \quad (9)$$

for each time-frequency bin<sup>1</sup>. The mean over all time steps  $T$  and frequency bins  $K$  is then used for back-propagation. Note that the weights  $\mathbf{W}(k, t)$  do not represent a statistical beamformer like the MVDR or GEV, but rather an optimal filter-and-sum beamformer for each time-frequency bin in a max-SNR fashion. To avoid unbounded weights, we normalize each predicted beamforming vector to unit length, i.e.  $|\mathbf{W}(k, t)| \stackrel{\dagger}{=} 1$ . As a consequence, speech distortions will occur. However, it is possible to control those distortions using *Blind Statistical Normalization* (BSN) [2].

### 5.1. Training and Testing

For training, we use 12776 utterances from the *si\_tr\_s* set of the WSJ0 [16] corpus for the speech sources in Eq. (7), and 27 hours of 32 different sound categories from YouTube [17] as isotropic background noise in Eq. (8). All recordings are sampled at 16kHz, and converted to frequency domain with  $K = 513$  bins and 50% overlapping blocks. The sources in Eq. (6) are mixed with equal volume. For testing, we use 651 utterances from the *si\_et\_05* set of the WSJ0 corpus mixed with another 5 hours of Youtube noise of the same 32 categories. For each of the five experiments in Table 1, a separate cDNN and mask-based beamformer is trained.

### 5.2. Results

We use the *BeamformIt* toolkit as baseline, and the mask-based beamformer in [6] with online tracking from [8] as reference system. For each method and each experiment, we report the  $\Delta$ SNR from Eq. (9), the *Perceptual Evaluation of Speech Quality* score (PESQ) [19], the *Short-Time Objective Intelligibility* measure (STOI), and the WER obtained by the Google Speech-to-Text API [31]. In particular, the WER was computed using the clean WSJ0 test set as reference, for which the Google Speech-to-Text API reports a WER of 6.1%. From Table 2 it can be seen that *BeamformIt* performs poorly for experiments with more than one source, i.e. experiments 1, 4 and 5. This is to be expected, as *BeamformIt* relies on blind DOA estimation to localize a single source. The mask-based beamformer with online tracking shows better performance for those experiments, which has also been observed in [8]. However, our cDNN outperforms this approach significantly, as we are able to estimate the optimal beamformer weights for each time-frequency bin in a max-SNR sense. Figure 5 shows an utterance from the test set using the 1<sup>st</sup> experiment, where two dynamic sound sources  $D_1$  and  $D_2$  are constantly moving around the living room. Panel (a) shows the mixture  $Z(k, t, 1)$  for the first microphone, and panel (b) shows the estimate  $Y(k, t)$ . It can be seen that the cDNN predicts beamforming weights according to the occurrence of the sound sources, i.e. source signal  $D_1$  is preserved, and  $D_2$  is canceled.

<sup>1</sup>For enhanced readability, the indices  $k, t$  have been omitted in Eq. (9).

Method	Experiment #	$\Delta$ SNR	PESQ	STOI	WER
BeamformIt	1	-	1.325	0.699	76.7%
	2	-	1.222	0.774	17.7%
	3	-	1.222	0.764	17.9%
	4	-	1.179	0.632	43.2%
	5	-	1.186	0.588	88.3%
mask-based BF + online tracking	1	4.445	1.514	0.834	46.1%
	2	4.286	1.576	0.837	32.8%
	3	4.516	1.751	0.866	18.5%
	4	8.690	1.439	0.811	45.6%
	5	7.011	1.402	0.792	58.3%
cDNN	1	6.156	1.688	0.825	21.5%
	2	8.736	2.263	0.882	9.0%
	3	9.558	2.551	0.902	6.1%
	4	10.306	1.652	0.792	13.4%
	5	9.212	1.441	0.758	33.7%

Table 2: Results

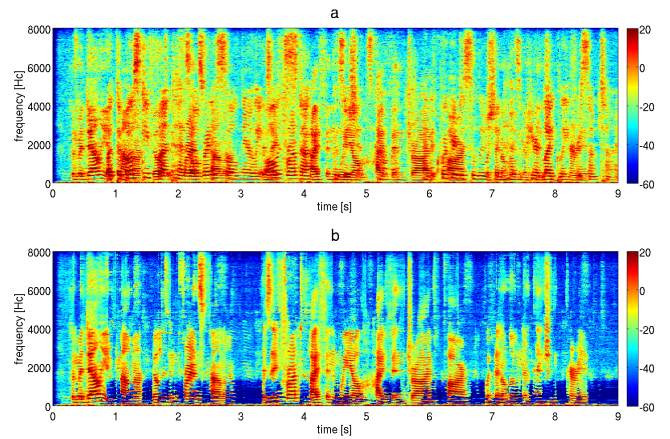


Fig. 5: (a) mixture of two dynamic sound sources  $D_1$  and  $D_2$ . (b) separated source  $D_1$  predicted by the cDNN.

## 6. CONCLUSIONS AND FUTURE WORK

We presented a complex-valued deep neural network (cDNN) to estimate complex-valued beamforming weights directly from complex-valued microphone data. Unlike existing approaches, our cDNN uses fully complex-valued LSTM and MLP layers, as well as complex-valued activation functions. Comparisons against *BeamformIt* and a state-of-the-art mask-based beamforming system showed a significant improvement in terms of  $\Delta$ SNR, PESQ, STOI and WER. Future work includes experiments on real multi-channel recordings, and the inclusion of our model in an end-to-end system.

## 7. REFERENCES

- [1] B. D. V. Veen and K. M. Buckley, “Beamforming: a versatile approach to spatial filtering,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, no. 5, pp. 4–24, Apr. 1988.
- [2] E. Warsitz and R. Haeb-Umbach, “Blind acoustic beamform-

- ing based on generalized eigenvalue decomposition,” *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, Jul. 2007.
- [3] J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming,” in *2016 IEEE ICASSP*, 2016.
- [4] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, “Blstm supported gev beamformer front-end for the 3rd chime challenge,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 444–451.
- [5] H. Erdogan, J. Hershey, S. Watanabe, M. Mandel, and J. L. Roux, “Improved mvdr beamforming using single-channel mask prediction networks,” in *Interspeech*, Sep. 2016.
- [6] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, “Dnn-based speech mask estimation for eigenvector beamforming,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, 2017, pp. 66–70.
- [7] —, “Eigenvector-based speech mask estimation using logistic regression,” in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2017, pp. 2660–2664.
- [8] C. Bøddeker, H. Erdogan, T. Yoshioka, and R. Haeb-Umbach, “Exploring practical aspects of neural mask-based beamforming for far-field speech recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 6697–6701.
- [9] Z. Meng, S. Watanabe, J. R. Hershey, and H. Erdogan, “Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 271–275.
- [10] T. Menne, R. Schlüter, and H. Ney, “Speaker Adapted Beamforming for Multi-Channel Automatic Speech Recognition,” *ArXiv e-prints*, 2018.
- [11] C. Bøddeker, P. Hanebrink, L. Drude, J. Heymann, and R. Haeb-Umbach, “On the computation of complex-valued gradients with application to statistically optimum beamforming,” *CoRR*, vol. abs/1701.00392, 2017.
- [12] J. Heymann, L. Drude, C. Bøddeker, P. Hanebrink, and R. Haeb-Umbach, “Beamnet: End-to-end training of a beamformer-supported multi-channel asr system,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 5325–5329.
- [13] M. F. Amin, M. I. Amin, A. Y. H. Al-Nuaimi, and K. Murase, “Wirtinger calculus based gradient descent and levenberg-marquardt learning algorithms in complex-valued neural networks,” in *Neural Information Processing*. Springer Berlin Heidelberg, 2011, pp. 550–559.
- [14] P. Bouboulis, “Wirtinger’s calculus in general hilbert spaces,” *CoRR*, vol. abs/1005.5170, 2010. [Online]. Available: <http://arxiv.org/abs/1005.5170>
- [15] R. F. H. Fischer, “Appendix A: Wirtinger calculus,” in *Pre-coding and Signal Shaping for Digital Transmission*. Wiley-Blackwell, 2005, pp. 405–413.
- [16] D. B. Paul and J. M. Baker, “The design for the wall street journal-based csr corpus,” in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT ’91. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 357–362.
- [17] “PyTube – a lightweight, pythonic, dependency-free, library for downloading youtube videos.” 2018. [Online]. Available: <https://python-pytube.readthedocs.io/en/latest/>
- [18] X. Anguera, C. Wooters, and J. Hernando, “Acoustic beamforming for speaker diarization of meetings,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2021, September 2007.
- [19] “ITU-T recommendation P.862. Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs,” 2000.
- [20] H.-Y. Dong and C.-M. Lee, “Speech intelligibility improvement in noisy reverberant environments based on speech enhancement and inverse filtering,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2018, no. 1, p. 3, May 2018.
- [21] C. Trabelsi, O. Bilaniuk, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, “Deep complex networks,” *CoRR*, vol. abs/1705.09792, 2017.
- [22] C.-A. Popa, “Complex-valued stacked denoising autoencoders,” in *Advances in Neural Networks – ISNN 2018*, T. Huang, J. Lv, C. Sun, and A. V. Tuzikov, Eds. Cham: Springer International Publishing, 2018, pp. 64–71.
- [23] M. Tytgert, J. Bruna, S. Chintala, Y. LeCun, S. Piantino, and A. Szlam, “A mathematical motivation for complex-valued convolutional networks,” *Neural Computation*, vol. 28, no. 5, pp. 815–825, 2016.
- [24] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [25] P. Bouboulis and S. Theodoridis, “Extension of wirtinger’s calculus to reproducing kernel hilbert spaces and the complex kernel lms,” *Trans. Sig. Proc.*, vol. 59, no. 3, pp. 964–978, Mar. 2011.
- [26] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.
- [27] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Berlin–Heidelberg–New York: Springer, 2008.
- [28] E. A. P. Habets and S. Gannot, “Generating sensor signals in isotropic noise fields,” *The Journal of the Acoustical Society of America*, vol. 122, no. 6, pp. 3464–3470, 2007.
- [29] R. Scheibler, E. Bezzam, and I. Dokmanic, “Pyroomacoustics: A python package for audio room simulations and array processing algorithms,” *CoRR*, vol. abs/1710.04196, 2017.
- [30] H. Kuttruff, *Room Acoustics*, 5th ed. London–New York: Spon Press, 2009.
- [31] “SpeechRecognition – a library for performing speech recognition, with support for several engines and apis, online and offline.” 2018. [Online]. Available: <https://pypi.org/project/SpeechRecognition/>

## A.9 InterSpeech 2020

*"Nonlinear residual echo suppression using a recurrent neural network"*, Lukas Pfeifenberger and Franz Pernkopf, International Conference on Spoken Language Processing (InterSpeech), Shanghai, 2020





# Nonlinear Residual Echo Suppression using a Recurrent Neural Network

Lukas Pfeifenberger, Franz Pernkopf

Signal Processing and Speech Communication Laboratory  
Graz University of Technology, Graz, Austria

lukas.pfeifenberger@alumni.tugraz.at  
pernkopf@tugraz.at

## Abstract

The acoustic front-end of hands-free communication devices introduces a variety of distortions to the linear echo path between the loudspeaker and the microphone. While the amplifiers may introduce a memory-less non-linearity, mechanical vibrations transmitted from the loudspeaker to the microphone via the housing of the device introduce non-linearities with memory, which are much harder to compensate. These distortions significantly limit the performance of linear Acoustic Echo Cancellation (AEC) algorithms. While there already exists a wide range of Residual Echo Suppressor (RES) techniques for individual use cases, our contribution specifically aims at a low-resource implementation that is also real-time capable. The proposed approach is based on a small Recurrent Neural Network (RNN) which adds memory to the residual echo suppressor, enabling it to compensate both types of non-linear distortions. We evaluate the performance of our system in terms of Echo Return Loss Enhancement (ERLE), Signal to Distortion Ratio (SDR) and Word Error Rate (WER), obtained during realistic double-talk situations. Further, we compare the postfilter against a state-of-the-art implementation. Finally, we analyze the numerical complexity of the overall system.

**Index Terms:** Acoustic echo cancellation, residual echo suppression, non-linear echo, recurrent neural networks

## 1. Introduction

In hands-free speech communication devices, an *Acoustic Echo Canceller* (AEC) is an essential building block which models the acoustic path between loudspeaker output and microphone input with a linear *Finite Impulse Response* (FIR) filter. The AEC subtracts the echo replica from the microphone signal, enabling echo-free voice communication [1]. Unfortunately, the task of echo cancellation is complicated by additional non-linear distortions in the loudspeaker and the amplifier, and also by mechanical vibrations transmitted from the loudspeaker via the case of the device to the microphone [2]. These distortions cannot be modeled by linear echo cancelers. Consequently, the practically achievable Echo Return Loss Enhancement (ERLE) is limited, which results in a degraded speech quality and intelligibility. This problem is even more relevant today as speakerphones or smart speakers are portable devices with small enclosure dimensions and tiny loudspeakers, which are prone to non-linear distortions. Despite their size, they produce high sound pressure levels by using amplifiers which pre-distort the loudspeaker signal [3]. This introduces even more distortions to the echo path. Non-linear distortions can be categorized into two groups:

1) Non-linearities without memory, i.e. harmonic distortions caused by non-linear loudspeaker drivers, or clipping of the microphone signal [4]. Non-linear systems without mem-

ory can be approximated by polynomials in the form  $f_{NL}(x) = \sum_{i=0}^{\infty} \alpha_i \cdot x^i$ . The parameters  $\alpha_i$  may be determined using non-linear system identification, i.e. by using a chirp signal [5]. This is also a standard procedure for measuring the individual harmonics and the overall *Total Harmonic Distortions* (THD) of loudspeakers and amplifiers. Harmonic distortions may be compensated by incorporating *power-filters* into the AEC algorithm [6–8], or by using a residual echo suppressor [9–12].

2) Non-linearities with memory, i.e. partial vibrations of the loudspeaker membrane, or structure-borne sounds and mechanical vibrations [4]. Non-linear systems with memory can be approximated by Volterra series [13]. As the size of a Volterra kernels grows exponentially with its order, this concept is of limited use in real-world applications. Further, tuning the kernels for a given non-linearity is a non-trivial task, as system identification requires *Higher Order Statistics* (HOS) and *spectral analysis*. However, several echo suppressors with Volterra series have been proposed, e.g.: sparse Volterra kernels [14, 15], or Hammerstein models [16, 17].

Both Volterra series and *Multilayer Perceptrons* (MLP) are universal approximators for non-linearities with memory. Consequently, neural networks have been proposed for non-linear residual echo cancellation [18–23]. However, we found that many contributions in this field are limited by one or more of the following aspects: (i) Only memoryless non-linearities are considered, even though both types always occur in a real-world scenario [4]. (ii) The neural network features a lot of weights, making the postfilter computationally more expensive than the actual AEC itself. (iii) The system is not real-time capable due to the data flow of the neural network.

In this paper, we consider *Recurrent Neural Networks* (RNNs) as postfilter to address these shortcomings. (i) Due to the recurrent structure of our neural network, non-linearities with memory can be learned directly from real-world audio examples. The use of internal memory in form of an LSTM layer allows for a smaller network compared to an MLP [18, 23]. (ii) Our approach is real-time capable, due to the LSTM layer operating only in forward direction of the data stream. It introduces no additional delay to the overall system, as it operates on one block of data at a time. (iii) With only two Dense layers and one LSTM layer with 25 units in its smallest variant, our neural network is considerably smaller than comparable approaches with 1024 or more units. Further, our system can be trained with as little as 1.75h of echo recordings, which allows for a fast training process even without a GPU.

## 2. System Model

We assume a classical, monaural speakerphone with a loudspeaker and a microphone for hands-free telephony applications, i.e. *Voice over IP* (VoIP). The system model is shown

in Fig. (1). In this setup, the far-end speaker signal is received via the network (RX), and the near-end speaker signal is transmitted (TX) back over the network. Due to acoustic echoes, the microphone picks up both the near-end speaker and the acoustic echo from the loudspeaker. Hence, an AEC is required. In Fig. (1), all signals are denoted in the *Short Time Fourier Transform* (STFT) domain with a frequency index  $k$  and a time index  $t$ . The loudspeaker and microphone signals are represented by  $X(k, t)$  and  $D(k, t)$ , respectively. The echo model, which is obtained from the AEC filter, is given as  $Y(k, t)$ . By subtracting the echo model from the microphone signal, we obtain the residual signal  $E(k, t)$ . The proposed postfilter operates on the residual and the microphone signal, and outputs the enhanced signal  $Z(k, t)$ .

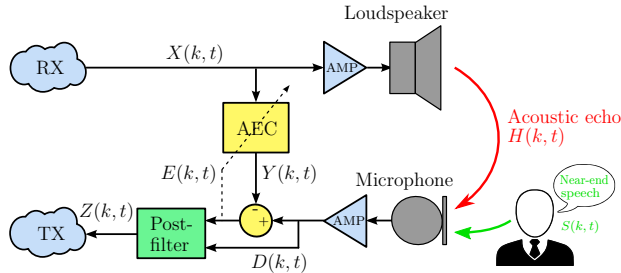


Figure 1: System Model with signals in the STFT domain.

The *Echo Impulse Response* (EIR)  $H(k, t)$  is modeled as FIR filter. Usually, it is much longer than the STFT block length, therefore it is partitioned into  $L$  blocks. Using this notation, the microphone signal can be written as

$$D(k, t) = S(k, t) + f_{NL}(X(k, t)) + \sum_{l=t-L}^t X(k, l)H(k, l), \quad (1)$$

where  $S(k, t)$  denotes near-end speech signal, and  $f_{NL}(\cdot)$  denotes an unknown non-linear relationship with memory. The AEC in Fig. (1) estimates the EIR  $\hat{H}(k, t)$ , such that the echo model is given as

$$Y(k, t) = \sum_{l=t-L}^t X(k, l)\hat{H}(k, l) \quad (2)$$

After the subtraction stage, the residual is given as

$$E = D - Y = \sum X\tilde{H} + f_{NL}(X) + S, \quad (3)$$

where  $\tilde{H} = H - \hat{H}$ . The frequency and time indices have been omitted for readability. Ideally, the filter mismatch  $\tilde{H}$  and the non-linearity  $f_{NL}(\cdot)$  are small, so that the residual signal contains only the near-end speech signal  $S(k, t)$ .

### 2.1. AEC Framework

We use a frequency-domain, block-based Acoustic Echo Canceled (AEC), which partitions the echo filter into multiple blocks using a STFT. This reduces the overall system delay of the algorithm to a single STFT block length, allowing for real-time operation. We chose the state-space block-partitioned AEC implementation from [24], which we found to be both robust and well-performing in real-world scenarios. We use a block length

of 1024 samples, 50% overlap, and  $L = 16$  blocks in total at  $f_s = 16\text{kHz}$  to model a tail length of up to 512ms.

In a practical application there is always a mismatch between the filter estimated by the AEC, and the actual EIR. The linear echo path may change over time as the near-end speaker moves in front of the device. The device itself may be carried around, causing a constantly changing EIR. These changes must be tracked by the AEC algorithm.

### 3. RNN postfilter

In a real-world scenario with actual loudspeakers and amplifiers, both non-linearities with and without memory are always present. These distortions cannot be compensated by the AEC. Residual echo suppressors have been proposed for both non-linearities without memory [8–12], and for non-linearities with memory [14–17]. With the advent of machine learning, the performance of residual echo suppressors has dramatically increased [18–23].

However, our contribution differs in the following key aspects: (i) Due to the recurrent structure of our neural network, non-linearities with memory can be learned directly from real-world audio examples, while most contributions only use memoryless non-linearities. (ii) Our approach is real-time capable, due to the LSTM layer operating only in forward direction of the data stream. It introduces no additional delay to the overall system. (iii) With only three layers and 25 LSTM cells in its smallest variant, our neural network is considerably smaller than comparable approaches [18, 23].

Fig. (2) outlines the architecture of our RNN postfilter. It consists of three layers, and operates on log-differences of the power of the microphone signal  $D(k, t)$  and the echo model  $Y(k, t)$ . The first layer is a simple dense layer, which performs data compression from  $K$  frequency bands to  $M$  bands. This is useful to facilitate a small LSTM layer, which is the second layer of the system and the computationally most complex one.  $M$  can be as low as 25 units, whereas  $K = 513$  in our implementation. The third layer expands the data back to  $K$  bands. It predicts a *gain mask*  $p(k, t)$ , which is multiplied element-wise to the residual signal  $E(k, t)$  to produce the enhanced output, i.e.:

$$Z(k, t) = E(k, t)p(k, t), \quad (4)$$

with  $p(k, t) \in [0, 1]$ .

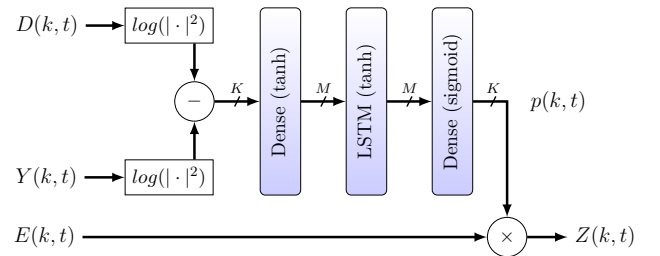


Figure 2: RNN architecture with  $K$  frequency bins and  $M$  LSTM units.

#### 3.1. Hybrid loss function

To train the RNN postfilter, we consider two use cases during a conversation:



- *Single-talk*: Only the far-end speaker  $X(k, t)$  is talking, the near-end is silent, i.e.  $S(k, t) = 0$ .
- *Double-talk*: Both near- and far-end speakers talk simultaneously.

During single-talk, we want to maximize the *Echo Return Loss Enhancement* (ERLE), i.e.: the output  $Z(k, t)$  is ideally zero. The ERLE is defined as follows:

$$\mathcal{L}_{\text{ERLE}} = 10 \log_{10} \frac{\sum_{K,T} |D(k, t)|^2}{\sum_{K,T} |Z(k, t)|^2} \quad (5)$$

During double-talk, we want to maximize the *Signal to Distortion Ratio* (SDR), i.e.: the output  $Z(k, t)$  is identical to the near-end signal  $S(k, t)$ . The SDR is defined as:

$$\mathcal{L}_{\text{SDR}} = 10 \log_{10} \frac{\sum_{K,T} |S(k, t)|^2}{\sum_{K,T} |S(k, t) - Z(k, t)|^2} \quad (6)$$

To fulfill both constraints, we use a hybrid objective to train the RNN postfilter. The overall loss function to be minimized by the RNN is given as:

$$\mathcal{L} = -\mathcal{L}_{\text{ERLE}} - \lambda \mathcal{L}_{\text{SDR}}, \quad (7)$$

where the parameter  $\lambda$  allows to adjust the importance of either the ERLE or SDR constraint during training.

## 4. Experiments

### 4.1. Recording Setup

In order to obtain realistic distortions which contain both types of non-linearities, it is essential to use a real-world setup, i.e. a speakerphone or smart speaker with a loudspeaker and a microphone in the same case. Otherwise it would be difficult to accurately simulate realistic non-linearities with memory, as well as changing EIR paths over time. Therefore, we used a small speakerphone (EasyAcc-MC) with a 3W loudspeaker and an electret microphone. We disconnected the internal electronics and used an external amplifier to drive the loudspeaker. The amplifier and the microphone were plugged into the line-out and mic-in jack of a sound card, respectively. We measured the *Total Harmonic Distortion* (THD) of the speakerphone at 3W, which is about 12%. Therefore, a reasonable amount of non-linear distortions is present in our setup [2]. To drive the speakerphone from a Linux-based PC with ALSA [25], we use the *PlayRec* Python module [26], which simultaneously plays and records audio from a sound card. We further implemented the block-based AEC from [24] in Python, to obtain the relevant signals for training the RNN postfilter.

The speakerphone was placed in 7 different office rooms in 10 different positions each. The rooms had a  $RT_{60}$  between 250ms and 500ms. For each position, we generated 3 training examples. Each training example consists of the excitation signal  $X(k, t)$ , and the recorded echo response  $D(k, t)$ . We use 30s of randomly concatenated utterances from the TIMIT speech corpus [27] as excitation signal  $X(k, t)$ . The simultaneously recorded microphone signal  $D(k, t)$  contains 30s echo response. In total, 1.75 hours of reverberated samples have been obtained. All samples were recorded at  $f_s = 16\text{kHz}$ . Fig. (3) illustrates the recording setup, using the speakerphone. Green arrows represent the linear echo path (EIR), and red parts depict potential sources of non-linear distortions.

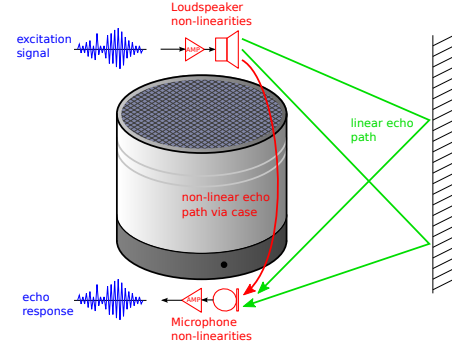


Figure 3: Recording setup using a small speakerphone.

### 4.2. Training

We used the recordings from the first 6 rooms for training, and the rest for evaluating the RNN postfilter. Note that the neural network does not learn speech or speaker characteristics, but rather the non-linearities embedded in the microphone signal  $D(k, t)$ . Hence, a small training set is sufficient. We train the RNN as follows:

First, we process each of the 30s long data samples with the AEC algorithm. The AEC provides the residual  $E(k, t)$  and the echo model  $Y(k, t)$ , which are required as inputs for the RNN (see Fig. 2). To train on time-varying EIRs, we reset the AEC weights at the beginning of each 30s long training example.

To optimize the RNN for both ERLE and SDR, we use each training example twice: In the first pass, the ERLE from Eq. (5) is calculated for the single-talk case, i.e. the near-end speaker  $S(k, t) = 0$ . In the second pass, the SDR from Eq. (6) is calculated for the double-talk case. We used randomly selected utterances from the *si-tr-s* set of the WSJ0 [28] corpus to simulate the near-end speaker  $S(k, t)$ , which we mixed into the microphone signal with a *Signal to Echo Ratio* (SER) of  $-12\text{dB}$ . This corresponds to the SER encountered when driving the loudspeaker at 3W and speaking into the device from approximately 0.5m distance. The trade-off parameter  $\lambda$  in Eq. (7) was set to 1. We trained 7 different versions of the RNN postfilter, where we parametrized the size of the LSTM layer from 25 to 250 units, see also Table (1).

### 4.3. Testing

Testing the RNN postfilter was done with the unused recordings from the 7<sup>th</sup> room. ERLE and SDR are evaluated as during training. We also measured the *Word Error Rate* (WER) for the enhanced signal  $Z(k, t)$  during double talk. The WER is obtained by the Google Speech-to-Text API [29]. In particular, it was measured using clean WSJ0 data set as reference, for which the Google Speech-to-Text API reports a WER of 6.1%.

### 4.4. Results

Table 1 reports the ERLE, SDR and WER scores for experiments using a varying LSTM layer size from  $M = 25$  to 250 units. As a baseline, we also evaluated the AEC without the postfilter. Further, we compare our postfilter to a state-of-the-art reference AEC implementation (Speex-DSP) [30]. Speex also uses a frequency-domain, block-based echo canceler [31], and a residual echo-suppressor. We configured the same echo-tail length of 512ms. It can be seen that Speex slightly outperforms the baseline in all scores. However, our RNN postfilter yields a significant improvement in all scores.

Table 1: ERLE, SDR and WER scores for the RNN postfilter, the reference system (Speex-DSP) and the AEC without a postfilter as a baseline.

LSTM cells $M$	ERLE	SDR	WER
25	44.868	11.079	17.08%
50	51.802	12.084	16.41%
75	55.303	12.656	14.87%
100	60.447	12.902	12.56%
150	61.650	13.294	11.72%
200	60.637	13.404	11.33%
250	63.019	13.434	10.64%
Speex-DSP	21.726	6.716	25.16%
no postfilter	19.206	5.454	44.73%

#### 4.5. Performance

Fig. (4) illustrates a 30s example from the test set with  $M = 100$  LSTM cells. Panel (a) shows the far-end and near-end signals, respectively. Panel (b) shows the residual signal  $E(k, t)$ . It can be seen that the AEC needs approximately 10s to adjust to the EIR. During that time, the error in the residual is quite large. About 16s into the sample, the near-end speaker  $S(k, t)$  starts talking. Panel (c) shows the enhanced output  $Z(k, t)$  of the RNN postfilter. It can be seen that the enhanced signal only contains the desired speech signal. Panel (d) shows the ERLE, measured over time and split into the contribution of both the AEC and the postfilter, respectively.

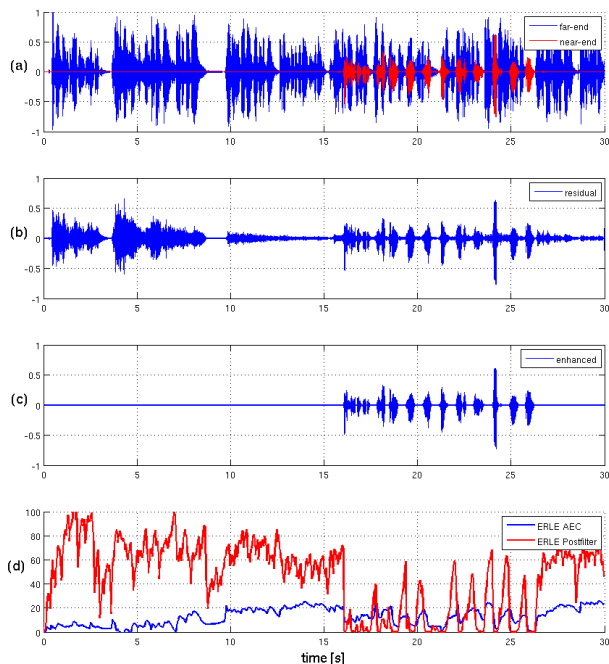


Figure 4: Performance of the RNN postfilter for a 30s test example: (a) far-end and near-end signals  $X(k, t)$  and  $S(k, t)$ , respectively. (b) AEC residual  $E(k, t)$ . (c) enhanced output of the postfilter  $Z(k, t)$ . (d) ERLE of the AEC and the postfilter.

#### 4.6. Numerical complexity

In this section we will discuss the numerical complexity of the overall system. We count the total number of *Multiply and Accumulate* (MAC) operations, which can be performed on either a dedicated DSP or CPU with a vector floating point unit (i.e.: ARM NEON). The RNN postfilter consists of 3 layers. The first layer is a dense layer with  $K$  inputs and  $M$  outputs. Its forward path is defined as  $y = \mathbf{W}x + b$ , where  $\mathbf{W}$  is a  $K \times M$  weight matrix and  $b$  is a bias vector of size  $M$ , and the input  $x \in \mathbb{R}^K$ . Hence, the layer requires  $(K \cdot M + M)$  MAC operations<sup>1</sup>. In the same manner, the LSTM layer requires  $(8M^2 + 7M)$  MACs, and the third layer requires  $(M \cdot K + K)$  MACs.

The complexity of the state-space block-partitioned AEC can be assessed using Eq. (26-32) in [24]. For  $L = 16$  blocks and  $K = 513$  frequency bins, we obtain 143k MACs including complex operations. Additionally, there are  $L + 3$  complex FFTs and  $L + 1$  complex IFFTs required for zero-padding and processing the time-domain inputs, adding another 737k MACs to the algorithm. Table (2) summarizes the numerical complexity for each RNN postfilter and the AEC. It can be seen that the postfilter adds only a fraction to the overall complexity, especially for small LSTM layers. At  $f_s = 16$ kHz and a block length of 1024 samples with 50% overlap, we process 31.25 blocks per second. In total, the smallest postfilter+AEC requires 25M MACs/s, while the largest postfilter+AEC requires 51M MACs/s, which is well within the reach of modern embedded systems.

Table 2: Numerical complexity per block.

LSTM cells $M$	MAC operations
25	31k
50	72k
75	123k
100	184k
150	336k
200	527k
250	758k
AEC	880k

## 5. Conclusion

In this paper, we proposed a residual echo suppressor which uses a recurrent neural network to model distortions such as non-linearities with memory, which are often found in small speakerphones housing a loudspeaker and a microphone in the same case. We showed that our approach uses very little resources, while still being real-time capable as it introduces no additional delay to the echo canceler. We also showed that the performance in terms of ERLE, SDR and WER is greatly improved compared to a state-of-the-art echo canceler and residual echo suppressor. In particular, the RNN postfilter lowers the WER by up to 14.52%

## 6. References

- [1] S. Haykin, *Adaptive Filter Theory*, 4th ed. New Jersey: Prentice Hall, 2002.

<sup>1</sup>We excluded the numerical complexity of the *tanh* or *sigmoid* activation functions, as they are negligible compared to the matrix multiplications, and also implemented in highly optimized libraries.

- [2] H. Kuttruff, *Room Acoustics*, 5th ed. London–New York: Spon Press, 2009.
- [3] Y. A. Huang and J. Benesty, *Audio Signal Processing For Next-Generation Multimedia Communication Systems*. Boston: Kluwer Academic Publishers, 2004.
- [4] T. D. Rossing, *Springer Handbook of Acoustics*. Berlin–Heidelberg–New York: Springer, 2007.
- [5] A. Novak, L. Simon, F. Kadlec, and P. Lotton, “Nonlinear system identification using exponential swept-sine signal,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2220–2229, 2010.
- [6] F. Kuech, A. Mitnacht, and W. Kellermann, “Nonlinear acoustic echo cancellation using adaptive orthogonalized power filters,” in *Proceedings. (ICASSP ’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 3, 2005, pp. iii/105–iii/108 Vol. 3.
- [7] F. Kuech and W. Kellermann, “Orthogonalized power filters for nonlinear acoustic echo cancellation,” *Signal Processing*, vol. 86, no. 6, pp. 1168 – 1181, 2006, applied Speech and Audio Processing.
- [8] F. Kuech and W. Kellermann, “Nonlinear residual echo suppression using a power filter model of the acoustic echo path,” in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07*, vol. 1, 2007, pp. 1–73–1–76.
- [9] D. A. Bendersky, J. W. Stokes, and H. S. Malvar, “Nonlinear residual acoustic echo suppression for high levels of harmonic distortion,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing, 2008*, pp. 261–264.
- [10] Kun Shi, Xiaoli Ma, and G. Tong Zhou, “A residual echo suppression technique for systems with nonlinear acoustic echo paths,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing, 2008*, pp. 257–260.
- [11] S. Malik and G. Enzner, “State-space frequency-domain adaptive filtering for nonlinear acoustic echo cancellation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 2065–2079, 2012.
- [12] S. Malik and G. Enzner, “A variational bayesian learning approach for nonlinear acoustic echo control,” *Signal Processing, IEEE Transactions on*, vol. 61, pp. 5853–5867, 12 2013.
- [13] H. Enzinger, K. Freiburger, G. Kubin, and C. Vogel, “Fast time-domain volterra filtering,” in *2016 50th Asilomar Conference on Signals, Systems and Computers, 2016*, pp. 225–228.
- [14] A. Guerin, G. Faucon, and R. Le Bouquin-Jeannes, “Nonlinear acoustic echo cancellation based on volterra filters,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 672–683, 2003.
- [15] F. Kuech and W. Kellermann, “A novel multidelay adaptive algorithm for volterra filters in diagonal coordinate representation [nonlinear acoustic echo cancellation example],” in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2004, pp. ii–869.
- [16] S. Malik and G. Enzner, “Fourier expansion of hammerstein models for nonlinear acoustic system identification,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011*, pp. 85–88.
- [17] A. Schwarz, C. Hofmann, and W. Kellermann, “Combined nonlinear echo cancellation and residual echo suppression,” in *Speech Communication; 11. ITG Symposium, 2014*, pp. 1–4.
- [18] C. M. Lee, J. W. Shin, and N. S. Kim, “Dnn-based residual echo suppression,” in *INTERSPEECH, 2015*.
- [19] T. V. Huynh, “A new method for a nonlinear acoustic echo cancellation system,” 2017.
- [20] H. Zhang and D. Wang, “Deep learning for acoustic echo cancellation in noisy and double-talk scenarios,” 09 2018, pp. 3239–3243.
- [21] G. Carbajal, R. Serizel, E. Vincent, and E. Humbert, “Multiple-input neural network-based residual echo suppression,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018*, pp. 231–235.
- [22] G. Carbajal, R. Serizel, E. Vincent, and E. Humbert, “Joint dnn-based multichannel reduction of acoustic echo, reverberation and noise,” 2019.
- [23] Q. Lei, H. Chen, J. Hou, L. Chen, and L. Dai, “Deep neural network based regression approach for acoustic echo cancellation,” in *Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing*, ser. ICMSSP 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 94–98.
- [24] F. Kuech, E. Mabande, and G. Enzner, “State-space architecture of the partitioned-block-based acoustic echo controller,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014*, pp. 1295–1299.
- [25] “Alsa-project,” Website, visited on February 19th 2020. [Online]. Available: [https://alsa-project.org/wiki/Main\\_Page](https://alsa-project.org/wiki/Main_Page)
- [26] “python-sounddevice,” Website, visited on February 19th 2020. [Online]. Available: <https://python-sounddevice.readthedocs.io/en/0.3.15/>
- [27] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, “Darpa timit acoustic phonetic continuous speech corpus cdrom,” 1993.
- [28] D. B. Paul and J. M. Baker, “The design for the wall street journal-based csr corpus,” in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT ’91. Stroudsburg, PA, USA: Association for Computational Linguistics, 1992, pp. 357–362.
- [29] “SpeechRecognition – a library for performing speech recognition, with support for several engines and apis, online and offline.” Website, 2018, visited on March 25th 2020. [Online]. Available: <https://pypi.org/project/SpeechRecognition/>
- [30] “Speex-dsp,” Website, visited on February 19th 2020. [Online]. Available: <https://github.com/xiongyihui/speexdsp-python>
- [31] J. . Soo and K. K. Pang, “Multidelay block frequency domain adaptive filter,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 2, pp. 373–376, 1990.

## **A.10 IEEE/ACM 2020**

*"Blind speech separation and dereverberation using neural beamforming"*, Lukas Pfeifenberger and Franz Pernkopf, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020

# Blind Speech Separation and Dereverberation using Neural Beamforming

Lukas Pfeifenberger and Franz Pernkopf, *Senior Member, IEEE*

**Abstract—**

In this paper, we present the *Blind Speech Separation and Dereverberation (BSSD) network*, which performs simultaneous speaker separation, dereverberation and speaker identification in a single neural network. Speaker separation is guided by a set of predefined spatial cues. Dereverberation is performed by using neural beamforming, and speaker identification is aided by embedding vectors and triplet mining. We introduce a *frequency-domain* model which uses complex-valued neural networks, and a *time-domain* variant which performs beamforming in latent space. Further, we propose a *block-online* mode to process longer audio recordings, as they occur in meeting scenarios. We evaluate our system in terms of *Scale Independent Signal to Distortion Ratio (SI-SDR)*, *Word Error Rate (WER)* and *Equal Error Rate (EER)*.

**Index Terms—**Multi-channel speech separation, beamforming, dereverberation, speaker identification, triplet mining

## I. INTRODUCTION

**S**PEAKER separation and speech enhancement is of paramount significance in many voice applications, such as hands-free teleconferencing or meeting scenarios. Especially in human-machine interfaces, where high-performance Automatic Speech Recognition (ASR) systems are essential, both speech intelligibility and quality play an important role. Fueled by the success of deep learning, both speaker separation and speech enhancement have made major advances over the last years [1].

When multiple microphones are available, spatial information can be exploited as speaker sources are directional. Mask-based beamforming has been shown to be advantageous for this task [2], [3]. In particular, a neural network is leveraged to estimate a time-frequency mask of the desired signal [4], [5], [6]. This mask is then used to compute the spatial covariance matrices required to construct a frequency-domain beamformer [7]. This approach has been further extended into the domain of complex numbers, where complex-valued neural networks [8] are used to directly estimate complex beamforming weights from noisy observations [9], [10].

Simultaneously to the success of neural beamforming, single-channel speaker separation techniques have also progressed dramatically. Frequency-domain algorithms such as *Deep Clustering (DC)* [11], *Permutation Invariant Training (PIT)* [12] and *Deep Attractor Network (DAN)* [13] rely

solely on spectral features. Time-domain algorithms such as *Wave-U-Net* [14], *TasNet* [15] and *Conv-TasNet* [16] delivered promising results.

Recently, some of these single-channel algorithms have been combined with a mask-based beamformer. In particular, a neural network estimates a gain mask of the desired signal, which is then used to construct a frequency-domain beamformer, i.e.: *Beam-TasNet* [17], *SpeakerBeam* [18], [19], *Neural Speech Separation* [20], *Multi-Channel Deep Clustering* [21], [22], and *Convolutional Beamforming* [23]. More recently, end-to-end multi-channel speech separation has been done entirely in time domain. By leveraging spatial cues among the multi-channel signals such as the Inter-channel Time Difference (ITD) or Inter-channel Phase Difference (IPD), the desired signal is estimated directly in time domain [24]. We further extend this approach by addressing the following three issues:

1) *Open number of sources*: Many source separation algorithms are limited to a pre-defined number of sources which they can separate [12], [13], [15], [16], [14], [17]. Exceptions are k-means clustering [11] and *Recurrent Selective Attention Network (RSAN)* [25]. We propose an iterative approach, by leveraging the spatial information encoded within the data.

2) *Distant speaker separation*: While close-talk speech separation models yield impressive performance, far-field speech separation is still a challenging task [26], [27]. Especially in real-world scenarios, reverberation and echoes cannot be ignored, as they severely degrade speech intelligibility and ASR performance [28]. Various deep learning based methods have been proposed for dereverberation [29], [30], [31], most of which are based on the Weighted Prediction Error (WPE) algorithm [32]. As this algorithm is restricted to the frequency domain, we chose a more general approach which learns the echo directly from the reverberated data.

3) *Speaker Identification*: To be useful in real-world applications, a speaker separation algorithm has to be at least *block-online* capable, i.e.; a short block of audio is processed at a time. This implies a permutation problem at block level, requiring *speaker diarization* [33], [34]. Therefore, identifying the separated speakers in each block of audio is necessary. A speaker identification algorithm is agnostic to the spoken text, and only relies on the speaker characteristics embedded in the waveform. Embedding vectors are used to map utterances into a feature space where distances correspond to speaker similarity [35]. Typically, i-Vectors [36] or x-Vectors [37] are used for this task. Algorithms such as *Deep Speaker* [38] rely on contrastive loss or triplet loss to learn embeddings on a very large set of speakers [39], [40], [41], [42]. We chose the triplet loss as its performance on small batch sizes is advantageous

Lukas Pfeifenberger and Franz Pernkopf are with the Intelligent Systems Group at the Signal Processing and Speech Communication Laboratory, Graz University of Technology, Graz, Austria.

This work was supported by the Austrian Science Fund (FWF) under the project number P27803-N15. Furthermore, we acknowledge NVIDIA for providing GPU computing resources.

[43].

In this paper, we introduce our *Blind Speech Separation and Dereverberation* (BSSD) network, which performs *separation*, *dereverberation* and *speaker identification* in a single neural network. We propose a frequency-domain variant (BSSD-FD), and time-domain variant (BSSD-TD). Our contributions are: Unsupervised speaker localization; separation of each speaker using adaptive beamforming; dereverberation of each source; and speaker diarization using embedding vectors. We evaluate our system in both *offline* and *block-online* mode. Further, we report the performance in terms of SI-SDR, WER and EER against similar state-of-the-art algorithms for speaker separation.

## II. SYSTEM MODEL

We assume a standard meeting scenario, where multiple speakers may talk simultaneously in an arbitrary room, i.e. an office. The position and number  $C$  of the speakers is unknown. We place a circular microphone array with  $M$  microphones in the center of the room, i.e. on a table. Figure 1 provides an example with three speakers. We assume that each speaker has a direct line of sight to the microphone array, i.e. the speaker is not obscured by a corner, or standing in the next room. Each speaker is assumed to be stationary, except for minor movements. Further, the room may have a significant amount of reverberation.

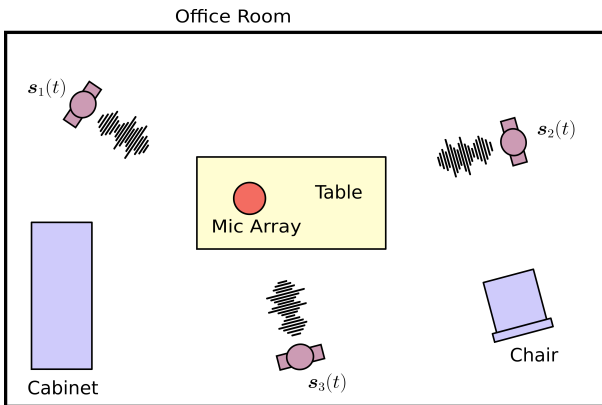


Fig. 1: Meeting room scenario with three independent speakers.

The signal arriving at the microphone array is composed of an additive mixture of  $C$  independent sound sources  $s_c(t)$ . In time domain, the samples of all  $M$  microphones at sampling time  $t$  can be stacked into a single  $M \times 1$  vector, i.e.

$$\mathbf{z}(t) = \sum_{c=1}^C s_c(t), \quad (1)$$

where

$$\mathbf{z}(t) = [z_1(t), \dots, z_M(t)]^T. \quad (2)$$

We use bold symbols for vectors, i.e.  $\mathbf{z}(t)$ , and plain symbols for scalars, i.e.  $z_m(t)$ . The vector  $s_c(t)$  represents the  $c^{\text{th}}$  sound source at sample time  $t$ . Each sound source is composed

of a monaural recording  $s_c(t)$  convolved with a *Room Impulse Response* (RIR) denoted by  $\mathbf{h}_c(t)$ , i.e.

$$s_c(t) = \mathbf{h}_c(t) \otimes s_c(t), \quad (3)$$

where  $\otimes$  denotes the convolution operator. The RIRs model the acoustic path from a sound source to the microphones as FIR filter, which includes all reverberations and reflections caused by the room acoustics [44]. Modern office rooms are made of laminate flooring and concrete walls, which have a low acoustic absorption coefficient. Consequently, the reverberation time  $RT_{60}$  may be very large, which significantly affects the performance of speech separation and speech recognition algorithms [26], [27], [28].

To cope with this environment, we propose the BSSD network, which iteratively extracts an unknown number of speakers from a multi-channel input mixture  $\mathbf{z}(t)$ . During each iteration, the Direction Of Arrival (DOA) of the loudest speech source is estimated by a *localization* module, which correlates the input mixture against a pre-defined set of DOA bases. The DOA is subtracted from a spatial speech presence probability map, so that the second-loudest source is extracted during the subsequent iteration. The DOA information is then fed into a Neural Network (NN), which extracts and dereverberates the corresponding speech source. The network also predicts a speaker embedding vector for each extracted speech source, which is used to assign the utterance to a speaker for block-online processing. This iterative process is repeated until no new speaker embedding is found. Figure 2 illustrates two iterations of the BSSD network, which consists of the following modules: *DOA bases*, *localization*, *beamforming and dereverberation*, and *speaker identification*. In the following chapters, we will introduce each module in detail.

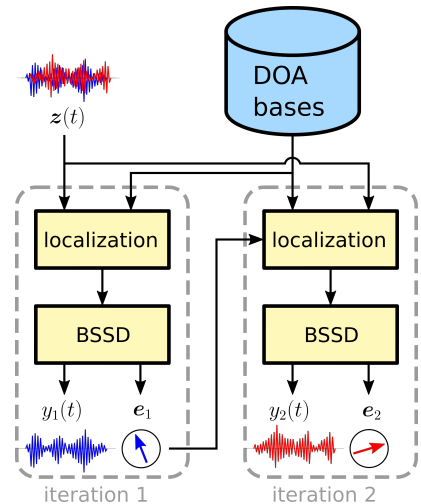


Fig. 2: Overview of the BSSD system, showing two iterations. During each iteration, the *localization* module estimates the DOA of a source from a set of pre-defined DOA bases. The DOA is then used to extract and dereverberate the corresponding speech source from the multi-channel input mixture  $\mathbf{z}(t)$ . The neural network also assigns a speaker embedding vector  $e_i$  to each enhanced source  $i$ .



### III. DOA BASES

As each source in Figure 1 has a direct line of sight towards the microphone array, it is possible to assign a unique DOA to each individual source in the mixture. Even if there is a significant amount of reverberation, there will always be an anechoic component in the RIR (i.e. the earliest peak) that corresponds to the DOA [44]. We therefore define a set of  $D$  unique DOA vectors on a unit sphere around the microphone array, where each impinging sound wave is modeled as plane wave, i.e.

$$V(d, k, m) = e^{-i2\pi f_k \tau_{d,m}}, \quad (4)$$

where  $f_k$  is the frequency for index  $k$  and  $\tau_{d,m}$  is the time delay from a point on the sphere to the  $m^{\text{th}}$  microphone, i.e.

$$\tau_{d,m} = \frac{\sqrt{(x_m - x_d)^2 + (y_m - y_d)^2 + (z_m - z_d)^2}}{c}, \quad (5)$$

where  $c$  is the speed of sound. The cartesian coordinates of the  $m^{\text{th}}$  microphone are denoted by  $x_m, y_m, z_m$ , and  $x_d, y_d, z_d$  are the coordinates of the  $d^{\text{th}}$  point on the sphere. We define these points to be equally distributed on the surface of the sphere using a fibonacci spiral [45], i.e.

$$\begin{aligned} \phi_d &= g \cdot d, \\ \theta_d &= \arcsin \frac{d}{D-1}, \\ x_d &= \cos \theta_d \cos \phi_d, \\ y_d &= \cos \theta_d \sin \phi_d, \\ z_d &= \sin \theta_d, \end{aligned} \quad (6)$$

where  $g = \pi(3 - \sqrt{5})$  is known as the golden angle [45], and  $d = 1 \dots D$  is the DOA index. We use a circular microphone array with  $M$  channels. Hence, the array is flat and we cannot distinguish between positive and negative  $z$  coordinates. It is therefore sufficient to only use half a sphere for the DOA bases. To assign a DOA index to a given RIR  $i$ , we utilize GCC-PHAT [46], i.e.

$$d_i = \operatorname{argmax}_d \sum_{k=1}^K \frac{|\mathbf{H}_i^H(k) \cdot \mathbf{V}(d, k)|^2}{|\mathbf{H}_i(k)|_2^2}, \quad (7)$$

where  $\mathbf{H}_i$  represents the FFT of the RIR  $\mathbf{h}_i(t)$ . Note that the amplitude of the DOA vector  $\mathbf{V}(d, k)$  is defined as 1 in Eq. (4). Figure 3 illustrates the DOA locations as black dots, where  $D = 100$  and  $M = 6$ . The color gradient is obtained from Eq. (7), with a randomly chosen RIR.

### IV. SOURCE LOCALIZATION

To estimate the direction of a speech source relative to the microphone array, we use GCC-PHAT to obtain a spatial speech presence probability map for the input mixture  $\mathbf{z}(t)$  and the DOAs  $\mathbf{V}(d, k)$ . First, we transform the input mixture to the frequency domain using the Short-Time Fourier Transform (STFT), i.e.

$$\mathbf{z}(t) \rightarrow \mathbf{Z}(l, k), \quad (8)$$

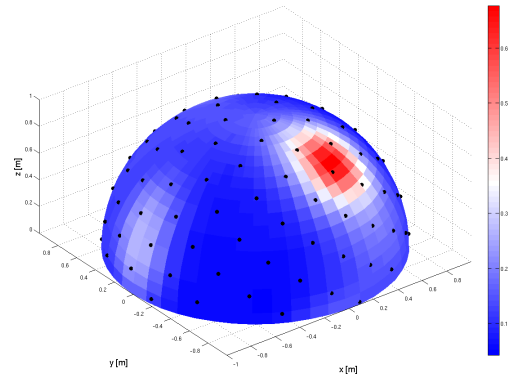


Fig. 3: Unit sphere with  $D = 100$  equi-distant DOA points and a circular microphone array with  $M = 6$  channels. The color gradient indicates the location of a single speaker, using Eq. (7).

where  $\mathbf{Z}(l, k)$  contains  $M$  samples of frequency bin  $k$  and STFT frame index  $l$ . Next, we compute the spatial speech presence probability map  $\gamma \in [0 \dots 1]$  as:

$$\gamma(l, k, d) = \frac{|\mathbf{Z}^H(l, k) \cdot \mathbf{V}(d, k)|^2}{|\mathbf{Z}(l, k)|_2^2}. \quad (9)$$

#### A. Spatial Whitening

To separate speakers based on their location, Eq. (9) utilizes the IPDs, which are encoded in the phase of the complex-valued input mixture  $\mathbf{Z}(l, k)$ . However, it is well known that microphone array recordings are strongly correlated towards low frequencies [46], [7], [47], [48]. This is due to the fact that the wavelength of low frequencies is large compared to the aperture of the microphone array. As a consequence, the IPDs will be small, and the overall separation performance is degraded. To mitigate this effect, we decorrelate the noisy inputs  $\mathbf{Z}(l, k)$  using *Zero-phase Component Analysis* (ZCA) whitening [49] from our previous works [9], [6]. In particular, we use the whitening matrix

$$\mathbf{U}(k) = \mathbf{E}_\Gamma(k) \mathbf{D}_\Gamma^{-\frac{1}{2}}(k) \mathbf{E}_\Gamma^H(k), \quad (10)$$

where  $\mathbf{E}_\Gamma$  and  $\mathbf{D}_\Gamma$  are  $M \times M$  sized eigenvector and eigenvalue matrices of the real-valued spatial coherence matrix of the ideal isotropic sound field  $\Gamma(k)$  [44]. Its elements are given as  $\Gamma_{i,j}(k) = \frac{\sin(2\pi f_k d_{i,j}/c)}{2\pi f_k d_{i,j}/c}$ , and  $d_{i,j}$  is the distance between the  $i^{\text{th}}$  and the  $j^{\text{th}}$  microphone. To avoid a division by zero, the diagonal elements of  $\mathbf{D}_\Gamma$  are loaded with a small constant  $\epsilon = 10^{-3}$ . We prefer ZCA whitening over PCA whitening, as the ZCA preserves the orientation of the distribution of the data [49]. Using the whitening matrix  $\mathbf{U}(k)$ , we rewrite Eq. (9) as

$$\gamma_U(l, k, d) = \frac{|\mathbf{Z}^H(l, k) \mathbf{U}^H(k) \cdot \mathbf{U}(k) \mathbf{V}(d, k)|^2}{|\mathbf{U}(k) \mathbf{Z}(l, k)|_2^2 \cdot |\mathbf{U}(k) \mathbf{V}(d, k)|_2^2}, \quad (11)$$

where  $\mathbf{U}(k) \mathbf{Z}(l, k)$  can be recognized as the whitened input mixture, and  $\mathbf{U}(k) \mathbf{V}(d, k)$  as whitened DOA vector. Figure 4 demonstrates the effect of spatial whitening. Panel (a) shows

shows  $\gamma(l, k)$  for a single speaker and a matching DOA vector. Panel (b) shows  $\gamma_U(l, k)$  with whitening. It can be seen that the separation performance is greatly increased for low frequencies.

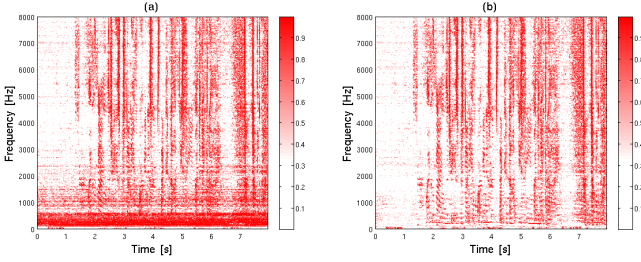


Fig. 4: Effectiveness of spatial whitening at low frequencies. (a)  $\gamma(l, k)$  from Eq. (9) for a single speaker. (b)  $\gamma_U(l, k)$  from Eq. (11) with whitening.

### B. Speaker Separation and Diarization

To iteratively estimate the DOA index  $d$  of all speech sources in the mixture  $\mathbf{Z}(l, k)$ , we use the pseudo code in Algorithm 1. First, we create a weighted spatial speech presence probability map  $\gamma_W(l, k, d)$ , using the energy  $P_Z(l, k)$  in each time-frequency bin of the input mixture  $\mathbf{Z}(l, k)$ . Next, we copy that map into  $\gamma'_W(l, k, d)$ . Then, we initialize an empty list of speaker embeddings  $\mathcal{E}$ . During each iteration, we average over the frame and frequency axes of  $\gamma'_W(l, k, d)$  to determine its global maximum over the  $D$  possible DOAs. The index of the maximum is denoted as  $\hat{d}$ , which is used as input for the BSSD network, which outputs an estimate of the desired signal  $y(t)$  at the direction of the DOA index  $\hat{d}$ , and a speaker embedding vector  $e$  for that output. Then, we compare this newly found embedding against the previously stored ones in the list  $\mathcal{E}$ , using the distance function  $\text{distance}(\mathcal{E}, e)$ . If the distance is greater than a threshold  $\delta$ , we append the embedding to the list, and subtract  $\gamma_W(l, k, \hat{d})$  from all DOA indices of the weighted spatial speech presence probability map  $\gamma'_W(l, k, :)$ . This ensures that each speech source is only extracted once<sup>1</sup>. If the threshold  $\delta$  is not met, the same embedding is already a member of the list  $\mathcal{E}$ . This may happen due to reflections or sidelobes [46] of the beamformer in the BSSD module. In this case, we stop the iterations and consider all speech sources within the mixture  $\mathbf{z}(t)$  to be extracted. We will discuss the BSSD architecture and the distance function in the following chapters.

## V. BSSD NETWORK - FREQUENCY DOMAIN

Well-established beamformers such as the *Minimum Variance Distortionless Response* (MVDR) beamformer [50] or the *Generalized Eigenvalue* (GEV) beamformer [51] use the signal statistics (i.e., the power spectral density matrices) to derive a set of beamforming weights  $\mathbf{W}(k) \in \mathbb{C}$  in frequency domain. As those weights are static over time, the signal separation

<sup>1</sup>Note that this algorithm is different to just sorting the DOA indices by energy, as multiple DOA indices may share the energy from the same speaker, due to the limited spatial resolution of the beamformer array.

### Algorithm 1 Source localization

---

```

1:  $P_Z(l, k) \leftarrow \frac{1}{M} \sum_{m=1}^M |\mathbf{Z}(l, k, m)|^2$ 
2:  $\gamma_W(l, k, d) \leftarrow \gamma_U(l, k, d) \cdot P_Z(l, k)$ 
3:  $\gamma'_W(l, k, d) \leftarrow \gamma_W(l, k, d)$ 
4:  $\mathcal{E} \leftarrow \square$ 
5:  $\mathcal{Y} \leftarrow \square$ 
6: while true do
7:    $\hat{d} \leftarrow \underset{d}{\operatorname{argmax}} \left( \sum_{l=1}^L \sum_{k=1}^K \gamma'_W(l, k, d) \right)$ 
8:    $y(t), e \leftarrow \text{BSSD}(\mathbf{z}(t), \hat{d})$ 
9:    $\mathcal{Y}.append(y(t))$ 
10:  if  $\text{distance}(\mathcal{E}, e) > \hat{\delta}$  then
11:     $\mathcal{E}.append(e)$ 
12:     $\gamma'_W(l, k, :) \leftarrow \max \left( \gamma'_W(l, k, :) - \gamma_W(l, k, \hat{d}), 0 \right)$ 
13:  else
14:    break
15:  end if
16: end while

```

---

performance is limited especially in reverberant conditions [46]. Therefore, a beamformer is often used in conjunction with a *post-filter* [7]. The post-filter acts as a single-channel gain mask on the output of the beamformer.

In [9], we proposed the *Complex-valued Neural Beamformer* (CNBF), which combines the properties of a beamformer and a post-filter using a neural network. Unlike a statistical beamformer, the CNBF estimates a set of individual beamforming weights  $\mathbf{W}(l, k) \in \mathbb{C}$  for each time-frequency bin. Those weights act as a spatio-temporal, complex-valued gain mask, which allows for a higher flexibility in the design of the beamformer, i.e. higher suppression rates or dereverberation. The CNBF uses complex-valued, non-holomorphic activation functions like vector normalization, phase normalization or conjugation. To back-propagate the complex-valued gradient, Wirtinger calculus is used [52], [53], [54]. A Tensorflow implementation of the CNBF network can be found at<sup>2</sup>.

We extend the CNBF to include dereverberation and a speaker embedding vector. Figure 5 shows the architecture of the BSSD-FD network. The left branch performs beamforming and dereverberation, and the right branch outputs an embedding vector per utterance.

### A. Speaker Separation

The STFT layer transforms the multi-channel input mixture to the frequency domain using Eq. (8). The STFT produces  $L$  time frames and  $K$  frequency bins per frame. Source separation is based on the DOA index  $\hat{d}$  from Algorithm 1, which is a scalar. The *Adaption* layer uses this input to modify the IPDs of the multi-channel input mixture in frequency domain, i.e.

$$\tilde{\mathbf{Z}}(l, k) = (\mathbf{U}(k)\mathbf{V}(\hat{d}, k))^* \odot (\mathbf{U}(k)\mathbf{Z}(l, k)), \quad (12)$$

<sup>2</sup><https://github.com/rrbluke/CNBF>



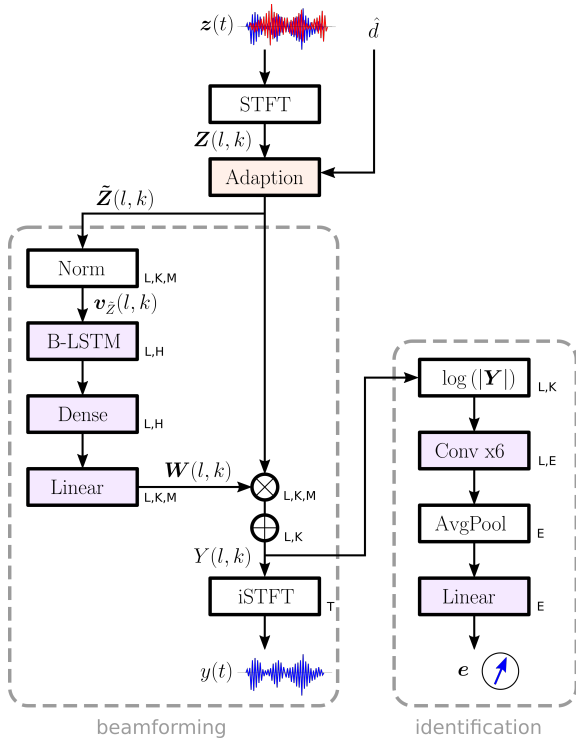


Fig. 5: Layers of the frequency-domain BSSD-FD network. The left branch performs beamforming and dereverberation, the right branch assigns an embedding vector to the enhanced output signal  $y(t)$ . The symbols next to each layer denote the dimensionality of the respective output tensor.

where  $*$  denotes complex conjugation, and  $\odot$  element-wise multiplication. The adaption layer performs two tasks: (i) It whitens the input signal  $\mathbf{Z}(l, k)$  as shown in Eq. (11). (ii) It subtracts the phase of the whitened DOA vector  $\mathbf{V}(\hat{d}, k)$  from the phase of the whitened input signal. This operation acts like a steering vector in a conventional beamformer, as it modifies the IPDs of the input signal to be approximately zero for signals originating from the direction of  $\mathbf{V}(\hat{d})$ , i.e. the desired signal. The unwanted signals (i.e. interfering speakers) are moved further away from the zero-IPD by the whitening process. Hence, the NN sees the desired signal always at the same spatial location, enabling it to distinguish between the desired and unwanted signal components. Consequently, the NN extracts the speaker towards the direction of  $\mathbf{V}(\hat{d})$ . We refer to Eq. (12) as the *Analytic Adaption* (AA). Hence, this system is abbreviated as BSSD-FD-AA.

Instead of modifying the phase of the input with the DOA vector, it is also possible to modify the input directly with a set of trainable weights, i.e.

$$\tilde{\mathbf{Z}}(l, k) = \mathbf{A}(\hat{d}, k)\mathbf{Z}(l, k), \quad (13)$$

where  $\mathbf{A}(\hat{d}, k)$  is a complex-valued matrix of shape  $M \times M$ . It allows to scale, shift and mix the  $M$  channels of the complex-valued inputs  $\mathbf{Z}(l, k)$  freely. Note that the DOA index  $\hat{d}$  selects the location from which we want to extract the desired speech signal. Hence, during training, all possible  $D$  DOA locations must be presented to the NN to train all complex-valued

weights in the tensor  $\mathbf{A}$ . We refer to Eq. (13) as *Statistic Adaption* (SA). Hence, this system is abbreviated as BSSD-FD-SA.

### B. Beamforming and Dereverberation

The structure of the left branch of the NN in Figure 5 resembles a traditional *filter-and-sum* beamformer, which can be written as:

$$Y(l, k) = \mathbf{W}^T(l, k)\tilde{\mathbf{Z}}(l, k), \quad (14)$$

where  $Y(l, k)$  denotes the beamformed output in frequency domain, and  $\mathbf{W}(l, k)$  are the beamforming filters. The inner vector product of Eq. (14) is computed before the inverse STFT layer in Figure 5. The NN estimates the weights  $\mathbf{W}(l, k)$  solely from the spatial information in  $\tilde{\mathbf{Z}}(l, k)$ , which is obtained by the *Norm* layer. In particular, this layer normalizes the magnitude of the  $M$  dimensional input vector  $\tilde{\mathbf{Z}}(l, k)$  to 1, and aligns its phase to the first microphone, i.e.

$$v_{\tilde{\mathbf{Z}}}(l, k) = \frac{\tilde{\mathbf{Z}}(l, k) \cdot \tilde{\mathbf{Z}}^*(l, k, m=1)}{|\tilde{\mathbf{Z}}(l, k) \cdot \tilde{\mathbf{Z}}^*(l, k, m=1)|}. \quad (15)$$

Then, a bidirectional LSTM layer creates a latent space of  $H$  neurons, followed by a dense with a complex-valued tanh activation function [9]. A linear layer outputs a set of unconstrained filter weights  $\mathbf{W}(l, k) \in \mathbb{C}$  to calculate the enhanced output  $Y(l, k)$  as shown in Eq. (14). After the inverse STFT layer, we obtain the enhanced time-domain signal  $y(t)$ .

By using a neural beamformer, the design goal is not limited to MVDR constraints or similar concepts [9]. In fact, we can also include a dereverberation objective by using an appropriate loss function for the NN. In particular, we use the negative SI-SDR [55] between the output  $y(t)$ , and a clean anechoic reference utterance  $r(t)$ , i.e.

$$\mathcal{L}_{\text{SI-SDR}} = -10 \log_{10} \left( \frac{|\alpha r(t)|_2^2}{|\alpha r(t) - y(t)|_2^2} \right), \quad (16)$$

where  $\alpha = \frac{y(t)^T r(t)}{r(t)^T r(t)}$ . We use  $r(t) = s_c(t)$  from Eq. (3) as anechoic reference signal.

### C. Speaker Identification

The right branch of the NN in Figure 5 extracts an embedding vector  $e$  to identify the speaker in the enhanced output signal  $Y(l, k)$ . The embedding vector maps the utterance into a feature space where distances correspond to speaker similarity [35]. Therefore, the NN must be agnostic to the spoken text, and only rely on the speaker characteristics embedded in the waveform. We use the log-power spectral density  $\log(|Y(l, k)|^2)$  as input features. Then, a series of 6 convolutional layers with a filter length of 10, and increasing dilation factors of (1,2,4,8,16,32) frames create a latent space of  $L \times E$  dimensional embeddings. We use a softplus<sup>3</sup> activation function and layer normalization after each convolutional layer. A skip connection is added between every two convolutional layers. The  $L$  time frames are averaged

<sup>3</sup>The softplus activation function is defined as  $f(z) = \log(1 + e^z)$ .

to obtain a single,  $E$  dimensional embedding for the whole utterance (AvgPool layer). The linear layer at the end of the stack outputs the unconstrained embedding vector  $e$ .

We want to identify an open set of speakers, i.e. we need to be able to compare two utterances and determine whether they belong to the same speaker. Therefore we employ the triplet loss [39], which has been successfully used for speaker identification and diarization tasks [38], [40], [41], [42]. The goal of the triplet loss is to ensure that two utterances from the same speaker have their embeddings close together in the embedding space, and two examples from different speakers have their embeddings farther away by some margin  $\beta$ . In other words, we want the embeddings of the same speaker to form clusters, and these clusters must be separated by the margin, i.e.

$$\mathcal{L}_{\text{TL}} = \sum_{B^3} \left[ |e_a - e_p|_2 - |e_a - e_n|_2 + \beta \right]_+, \quad (17)$$

where the embedding  $e_a$  denotes an *anchor*,  $e_p$  is an embedding from the same speaker as the anchor (positive example), and  $e_n$  is an embedding from a different speaker (negative example). In a batch of  $B$  utterances, there can be as much as  $B^3$  triplets. It is therefore crucial to only select a subset of valid triplets, where the positive example is from the same speaker as the anchor, and the negative example belongs to a different speaker. Further, we only need to consider triplets where the loss  $\mathcal{L}_{\text{TL}}$  is actually greater than zero. To select relevant triplets, we utilize *Hard Triplet Mining* [56], where we select the hardest positive and negative example per anchor. In particular, we randomly select  $P$  utterances from  $B$  speakers, where we determine the largest distance  $|e_a - e_p|_2$  between an anchor and a positive example within the  $P$  utterances per speaker, and the smallest distance  $|e_a - e_n|_2$  between an anchor and a negative example from the  $P(B-1)$  remaining utterances. More formally, this procedure can be written as:

$$\mathcal{L}_{\text{TL-HTM}} = \frac{1}{B \cdot P} \sum_{i=1}^B \sum_{a=1}^P \left[ \beta + \max_{p=1 \dots P} (|e_a^i - e_p^i|_2) - \min_{\substack{j=1 \dots B \\ n=1 \dots P \\ i \neq j}} (|e_a^i - e_n^j|_2) \right]_+. \quad (18)$$

When the batch size  $P \cdot B$  is small, the embeddings may collapse into a single point during training [57]. To avoid this, we propose to minimize the cross-entropy between embeddings of different speakers as follows:

$$\mathcal{L}_{\text{TL-CE}} = \frac{-1}{(B^2 - B)P^2} \sum_{a=1}^B \sum_{\substack{n=1 \\ n \neq a}}^B \sum_{i=1}^P \sum_{j=1}^P \log(|(\tilde{e}_a^i)^T \tilde{e}_n^j|^2), \quad (19)$$

where  $\tilde{e} = \frac{e}{|e|_2}$  is the magnitude-normalized embedding vector  $e$ . This regularization ensures that the embeddings  $e_a$  and  $e_n$  will be different. The overall cost function for the entire BSSD-FD architecture is then defined as:

$$\mathcal{L}_{\text{BSSD-FD}} = \mathcal{L}_{\text{SI-SDR}} + \lambda_1 \mathcal{L}_{\text{TL-HTM}} + \lambda_2 \mathcal{L}_{\text{TL-CE}}, \quad (20)$$

where  $\lambda_1$  and  $\lambda_2$  are weights for the individual terms.

#### D. Distance Measure

In order to determine whether two embeddings  $e_1$  and  $e_2$  belong to the same speaker, we use the euclidian distance from Eq. (17), i.e.  $|e_1 - e_2|_2$ . If the distance falls below a certain threshold  $\delta$ , we consider the two embeddings to belong to the same speaker. If it exceeds the threshold, the speakers are considered different. Hence, two types of errors exist: (i) A false positive is triggered when two embeddings from two different speakers are incorrectly classified as belonging to the same speaker, which we measure using the False Acceptance Rate (FAR), i.e.

$$\text{FAR}(\delta) = \frac{1}{(B^2 - B)P^2} \sum_{a=1}^B \sum_{\substack{n=1 \\ n \neq a}}^B \sum_{i=1}^P \sum_{j=1}^P \mathbb{1}(|e_a^i - e_n^j|_2 < \delta), \quad (21)$$

where  $\mathbb{1}(x)$  denotes an indicator function, i.e.

$$\mathbb{1}(x) = \begin{cases} 1, & \text{if condition } x \text{ is true.} \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

(ii) A false negative is triggered when two embeddings from the same speaker are classified as belonging to different speakers, which we measure using the False Rejection Rate (FRR), i.e.

$$\text{FRR}(\delta) = \frac{1}{B(P^2 - P)} \sum_{\substack{a=1 \\ p=a}}^B \sum_{i=1}^P \sum_{\substack{j=1 \\ j \neq i}}^P \mathbb{1}(|e_a^i - e_p^j|_2 > \delta). \quad (23)$$

The FAR is positively correlated to the decision threshold  $\delta$ , and the FRR is correlated negatively. The value at which the FAR and FRR are equal, is known as the Equal Error Rate (EER). It is determined by:

$$\hat{\delta} = \underset{\delta}{\operatorname{argmin}} |\text{FAR}(\delta) - \text{FRR}(\delta)| \quad (24)$$

$$\text{EER} = \text{FAR}(\hat{\delta}) = \text{FRR}(\hat{\delta}),$$

where  $\hat{\delta}$  is considered the optimal threshold belonging to the EER.

## VI. BSSD NETWORK - TIME DOMAIN

With the recent success of time-domain speech separation algorithms [14], [15], [16], [24], we also formulate a time-domain variant of our BSSD network. Figure 6 shows the architecture of the BSSD-TD network. Similar to the frequency-domain variant, the left branch performs beamforming and dereverberation, and the right branch outputs an embedding vector per utterance.

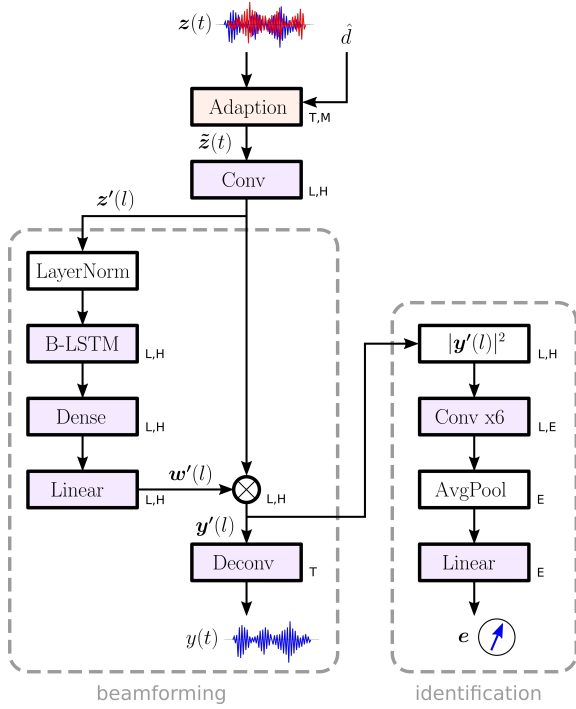


Fig. 6: Layers of the time-domain BSSD-TD network. The left branch performs beamforming and dereverberation, the right branch assigns an embedding vector to the enhanced output signal  $y(t)$ .

### A. Speaker Separation

Analogous to the frequency-domain network, source separation is based on the *Adaption* layer, which uses the DOA index  $\hat{d}$  from Algorithm 1 to modify the ITD of the input signal  $z(t)$ . By rearranging Eq. (12), we can formulate an identical operation in time-domain, i.e.

$$\begin{aligned} \tilde{z}(l, k, m) &= (\mathbf{U}^T(k, m)\mathbf{V}(\hat{d}, k))^* \cdot (\mathbf{U}^T(k, m)\mathbf{Z}(l, k)), \\ &= \sum_{i=1}^M \mathbf{U}^H(k, m)\mathbf{V}^*(\hat{d}, k)U(k, m, i) \cdot Z(l, k, i), \\ &= \sum_{i=1}^M V'(\hat{d}, k, m, i) \cdot Z(l, k, i), \end{aligned} \quad (25)$$

where we can identify the convolutional kernel  $V'(\hat{d}, k, m, i)$  in frequency domain. We can see from Eq. (4), that the DOA  $\mathbf{V}(\hat{d}, k)$  resembles  $M$  sinc pulses with a positive time-delay  $\tau_{d,m}$ , and Eq. (10) shows that the elements of the whitening matrix  $U(k, m, i)$  are real-valued. Therefore,  $V'(\hat{d}, k, m, i)$  will be a causal IIR filter in time domain [58], which we truncate to  $T_A$  samples to obtain the FIR filter  $v'(\hat{d}, t_A, m, i)$  by using the inverse FFT. This allows to formulate the time-domain adaption layer as:

$$\tilde{z}(t, m) = \sum_{i=1}^M z(t, i) \otimes v'(\hat{d}, t_A, m, i), \quad (26)$$

which can be implemented using a single convolution layer. Similar to the frequency-domain adaption layer, Eq. (26)

synchronizes the ITD to be zero for signals originating from the direction of  $\mathbf{V}(\hat{d})$ , i.e. the desired signal. The subsequent NN sees the desired signal always at the same spatial location, which makes it easier to distinguish between the desired and unwanted signal components. Consequently, the NN extracts the speaker towards the direction of  $\mathbf{V}(\hat{d})$ . We refer to Eq. (26) as *Analytic Adaption (AA)*. Hence, this system is abbreviated as BSSD-TD-AA.

Instead of modifying the ITDs of the input signal with the DOA vector, it is also possible to replace the fixed convolutional kernels  $\tilde{v}(\hat{d}, t_A, m, i)$  with a set of trainable weights, i.e.

$$\tilde{z}(t, m) = \sum_{i=1}^M z(t, i) \otimes a(\hat{d}, t_A, m, i), \quad (27)$$

where  $\mathbf{a}$  is a tensor of shape  $(D, T_A, M, M)$ , and  $T_A$  is the filter length of the learnable convolutional kernels. This allows to scale, shift and mix the  $M$  channels of the input signal  $z(t)$  freely. Note that the DOA index  $\hat{d}$  provides the location from which we want to extract the desired speech signal. Hence, during training, all  $D$  possible DOA locations must be presented to the NN to train the weights  $\mathbf{a}$ . We refer to Eq. (27) as *Statistic Adaption (SA)*. Hence, this system is abbreviated as BSSD-TD-SA.

### B. Beamforming and Dereverberation

The structure of the left branch of the NN in Figure 6 resembles a time-domain beamformer [46], where the first convolution layer right after the adaption layer transforms the time-domain input  $\tilde{z}(t)$  into a latent space  $z'(l, h)$  with  $L$  frames and  $H$  filters. The stride of this convolution layer is set to  $\frac{H}{4}$ , and the activation function is linear.

Similar to the frequency-domain beamformer, filtering is performed in latent space. The beamforming weights  $w'(l, h)$  are estimated from the spatial information embedded in  $z'(l, h)$ , using layer normalization, followed by a bidirectional LSTM layer, a dense layer with tanh activation, and a linear layer. The linear layer allows the NN to freely choose the amplitude and phase of the beamforming weights. The enhanced output  $y'(l, h)$  is obtained by

$$y'(l, h) = w'(l, h) \odot z'(l, h), \quad (28)$$

where all variables are of shape  $L \times H$ . Finally, a deconvolution layer with a linear activation function produces the enhanced time-domain signal  $y(t)$ . Analogous to the BSSD-FD architecture, we use the negative SI-SDR from Eq. (16) between the output  $y(t)$ , and a clean anechoic reference utterance  $r(t)$ .

### C. Speaker Identification

The right branch of the NN in Figure 6 extracts an embedding vector  $e$  to identify the speaker in the enhanced output signal  $y(t)$ . The NN is identical to the BSSD-FD architecture, except for the input layer which uses the enhanced signal  $y'(l, h)$  as input features. Identically to Eq. (20), the overall cost function for the entire BSSD-TD architecture is defined as:

$$\mathcal{L}_{\text{BSSD-TD}} = \mathcal{L}_{\text{SI-SDR}} + \lambda_1 \mathcal{L}_{\text{TL-HTM}} + \lambda_2 \mathcal{L}_{\text{TL-CE}}. \quad (29)$$

## VII. BLOCK ONLINE PROCESSING

For realtime applications, it is possible to use the BSSD system in block-online mode. We split the input mixture  $z(t)$  into blocks of equal length. Each block  $b$  is iteratively processed using Algorithm 1. It returns the DOA index  $\hat{d}$ , a list  $\mathcal{Y}_b$  of extracted speakers  $y(t)$ , and a list  $\mathcal{E}_b$  of speaker embeddings  $e$ . Figure 7 illustrates the block-online processing scheme of the BSSD system for  $C = 2$  speakers and 4 blocks. Note that the speakers may change their position from block to block, as their respective DOA indices are re-estimated for each block. The order and the number of speakers which are extracted may vary from block to block. To solve this permutation problem, we employ *diarization* [33], [34], by using Algorithm 2.

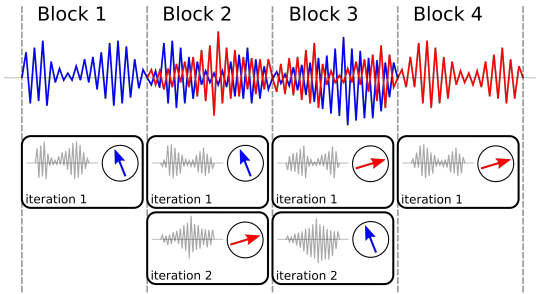


Fig. 7: Block-online processing mode of the BSSD system, showing a mixture of  $C = 2$  speakers being split into 4 blocks. Each block is processed separately using Algorithm 1.

---

### Algorithm 2 Diarization in block-online mode.

---

```

1:  $\mathcal{Y} \leftarrow \square$ 
2:  $\mathcal{E} \leftarrow \square$ 
3: for all blocks  $b$  do
4:   for  $c = 1 : \text{length}(\mathcal{E}_b)$  do
5:     if  $\min(|\mathcal{E} - \mathcal{E}_b(c)|_2) > \hat{\delta}$  then
6:        $\mathcal{E}.\text{append}(\mathcal{E}_b(c))$ 
7:     else
8:        $i \leftarrow \text{argmin}(|\mathcal{E} - \mathcal{E}_b(c)|_2)$ 
9:        $\mathcal{Y}(i).\text{append}(\mathcal{Y}_b(c))$ 
10:    end if
11:  end for
12: end for

```

---

First, we initialize empty lists for all speakers  $\mathcal{Y}$  and all embeddings  $\mathcal{E}$ . Then, we iterate over all blocks  $b$ , where Algorithm 1 is executed for each block. It returns a list  $\mathcal{Y}_b$  of extracted speakers and a list  $\mathcal{E}_b$  of speaker embeddings for that block. Next, we iterate over each extracted source  $c$  within that block, and we compare the distance of the embedding  $\mathcal{E}_b(c)$  against all embeddings  $\mathcal{E}$ . If the threshold  $\hat{\delta}$  (see Eq. 24) is exceeded, we have found a new speaker. In that case, this speaker is added to the list of known embeddings  $\mathcal{E}$ . Otherwise, we have found an utterance belonging to a known

embedding. In that case, we determine the index  $i$  of that embedding, and append the source  $\mathcal{Y}_b(c)$  to the speaker at position  $\mathcal{Y}(i)$ . To preserve the time alignment of each extracted speaker, we append a block of silence to each source in  $\mathcal{Y}$  that did not receive an update. This may happen if a speaker is silent during block  $b$ .

A trade-off has to be made when choosing the block length  $T_B$ . If it is too large, short utterances followed by periods of silence might not get detected. If it is too small, the predicted speaker embeddings may be inaccurate, causing Algorithm 2 to assign the sources  $\mathcal{Y}_b(c)$  to the wrong speaker. We examine this behavior in our experiments.

## VIII. RIR RECORDINGS

We use both recorded and simulated RIRs to generate spatialized recordings with Eq. (3). Real RIRs are obtained through multi-channel room impulse response measurements, and simulated RIRs are obtained by the Image Source Method (ISM) [22], [59].

### A. Real RIRs

To obtain realistic room impulse responses (RIRs), we use a circular microphone array with  $M = 6$  channels and a diameter of  $92.6\text{mm}$  [60], and a 5W measurement loudspeaker. To drive the loudspeaker from a Linux-based PC with ALSA [61], we use the PlayRec Python module [62], which simultaneously plays and records audio from a sound card. We use an exponential chirp with a duration of 5s sweeping from  $24\text{kHz}$  down to  $20\text{Hz}$  as excitation signal [44]. However, we only use a bandwidth of  $8\text{kHz}$  for our experiments. We recorded 120 6-channel RIRs in 24 different, fully furnished office rooms with a reverberation time  $RT_{60} \in [200 \dots 900]\text{ms}$ . The distance from the loudspeaker to the microphone array was varied from  $1\text{m} \dots 3\text{m}$ , and the direction was chosen randomly. Figure 8 shows the recording setup. We augmented the number of RIR recordings to 720 by virtually rotating the array by  $6 \times 60^\circ$ , i.e. shifting the microphone channels.



Fig. 8: RIR recording setup using a 5W measurement loudspeaker and a 6-channel microphone array [60].

## B. Simulated RIRs

To obtain simulated RIRs, we further generated 720 artificial RIRs for the same array geometry with 6 channels, but with a shorter reverberation time  $RT_{60} \in [200 \dots 400]ms$ , which is randomly chosen. The room is modeled as a simple rectangular shoebox with random dimensions ranging from  $3m \dots 6m$ , where the microphone array and the sound sources are placed randomly. RIR generation was done using the Image Source Method [22], [59] using *Pyroomacoustics* [63].

## IX. EXPERIMENTS

### A. Experimental Setup

1) *Speech mixtures*: We use the WSJ0 speech database which contains 12776 utterances from 101 different speakers for training, and 5895 utterances from 18 different speakers for testing. To generate mixtures, we use the wsj0-2mix from [11], which we extended to 3 and 4 speakers. To generate reverberated, multi-channel mixtures from Eq. (3), we convolve the monaural signals with both the *real* and *simulated* RIRs, as described in Section VIII. All recordings use a sample rate of  $f_s = 16kHz$ .

2) *DOA bases*: We use  $D = 100$  DOA bases, which are distributed on a sphere, as shown in Figure 3. This provides sufficient spatial resolution for the scenario described in Section II. We use a different DOA index  $d_c \in [1 \dots D]$  for each source  $s_c$  in the mixture  $z(t)$ . To achieve this, we randomly select a RIR  $h_c(t)$  belonging to a the DOA index  $d_c$  using Eq. (7). From the 720 *real* and *simulated* RIRs available, 640 are used for training, and 80 for testing.

3) *BSSD-FD system*: For the BSSD-FD network in Figure 5, we use a FFT length of 1024 samples, and an overlap of 75%. This results in  $K = 513$  frequency bins. Further, we have  $M = 6$  microphones as determined by the RIR recordings. The *beamforming* branch uses  $H = 500$  neurons to create the beamforming weights  $\mathbf{W}(l, k)$ , and to predict the enhanced signal  $Y(l, k)$ . The *identification* branch uses an embedding dimension of  $E = 100$  to predict the speaker embeddings  $e$ .

4) *BSSD-TD system*: For the BSSD-TD network in Figure 6, we use a filter length of  $T_A = 100$  samples for the filter kernels in the adaption layer in Eq. (26) and (27). The first convolutional layer uses a filter length of 200 samples and a stride of 50 samples to create a latent space of  $H = 500$  neurons. The *beamforming* branch predicts the beamforming weights  $w'(l)$  and the enhanced signal  $y'(l)$  in latent space. This signal is transformed back to time domain using the deconvolution layer, which uses a filter length of 200 samples, a stride of 50 samples, and overlap-add to produce the enhanced signal  $y(t)$ . The *identification* branch uses an embedding dimension of  $E = 100$  to predict the speaker embeddings  $e$ .

### B. Related Systems

To compare our BSSD system against other state-of-the art speech separation algorithms, we evaluate Conv-TasNet [16] and PIT with spatial features [22].

1) *Conv-TasNet*: Conv-TasNet separates 2 speakers in time domain. It operates on chunks of 4s of audio, where it separates the two speakers in a latent space by using a speech mask  $\in [0 \dots 1]$ . The mask is obtained from a series of convolutions. The system operates on single-channel inputs. However, we trained it to perform dereverberation, by providing anechoic utterances as target signal. We use the implementation of [16].

2) *Spatial PIT*: Spatial PIT separates 2 speakers in frequency domain. It uses log-spectrograms and the sine and cosine of the IPDs of frequency-domain, multi-channel mixtures to predict a speech mask for each speaker [22]. This mask is used to construct a frequency-domain beamformer [6]. Note that there is no explicit dereverberation constraint, but the target speech mask is obtained from the anechoic reference signal  $r(t)$ . Hence, the beamformer will remove late echoes.

### C. Training

All four variants of the BSSD network are trained on mixtures of  $C = 2$  sources, where each mixture  $z(t)$  is truncated to 5s length. The location (i.e. the DOA index  $d$ ) of each source is chosen randomly for each example. We use a batch size of 60 mixtures from the 101 speakers of the WSJ0 training set. To enable efficient triplet mining with Eq. (18), we use  $P = 3$  different utterances from  $B = 20$  speakers for the first source  $s_{c=1}(t)$  of each mixture. The second source  $s_{c=2}(t)$  is chosen randomly from the remaining 100 speakers from the WSJ0 training set. We use the clean first source as a reference utterance, i.e.  $r(t) = s_{c=1}(t)$ . The ground truth DOA index  $\hat{d}$  is used to train the network. We use  $\lambda_1 = 10^{-2}$  and  $\lambda_2 = 10^{-4}$  for the cost function in Eq. (20). This ensures that the *beamforming* path is trained faster than the *identification* path, as the latter depends on the former. As the combination of the different RIRs and WSJ0 utterances allows for billions of combinations, we randomly create new batches for training and validation for each epoch. Adam is used as optimizer [64], with a learning rate of  $10^{-3}$ . A Tensorflow implementation of the BSSD network can be found at<sup>4</sup>.

### D. Testing

We compare the frequency-domain (FD) and time-domain (TD) variants of our BSSD system, as well as the analytic adaption (AA) and statistic adaption (SA) layers introduced in Section V and VI. Further, we use the Conv-TasNet and spatial PIT as baseline systems. We test the BSSD network both in *offline* mode in *block-online* mode.

1) *Offline Mode*: In offline mode, we use 5s long mixtures of  $C = \{1, 2, 3, 4\}$  speakers from the test set. To test the performance of the *speaker separation* and *speaker identification* modules separately, we use the ground truth DOA index  $\hat{d}$  as input to the BSSD network. We report the separation and dereverberation performance in terms of SI-SDR using Eq. (16). Further, we report the WER using the Google Speech-to-Text API [65] to perform ASR. Speaker identification performance is reported in terms of EER on the enhanced output, using Eq. (24).

<sup>4</sup><https://github.com/rrbluke/BSSD>

2) *Block-online Mode*: In block-online mode, we use 20s long mixtures from the test set, which we divide into  $N_b$  blocks of  $T_B = \{1, 2.5, 5\}s$  length. Each block is processed by Algorithm 1, which outputs the DOA index  $\hat{d}$ , a list of extracted signals  $\mathcal{Y}_b$ , and a list of speaker embeddings  $\mathcal{E}_b$  for each block  $b$ . Then, Algorithm 2 is used to assign the extracted utterances of each block to the same speaker. This solves the speaker permutation problem. We report the SI-SDR, WER and the Block Error Rate (BER) for the extracted speakers. The BER indicates the percentage of falsely assigned blocks due to erroneous embeddings. It is determined by comparing the speaker embedding of the reference utterance  $r_c(t)$  against the extracted chunks  $y_{b,c}(t)$  for each speaker  $c$ , i.e.

$$\text{BER} = \frac{1}{C \cdot N_b} \sum_{c=1}^C \sum_{b=1}^{N_b} \mathbb{1}(|e_{r,c} - e_{b,c}|_2 > \hat{\delta}) \quad (30)$$

## X. RESULTS

### A. Offline mode

Table I reports SI-SDR, WER and EER for the *real* RIRs. For  $C = 1$  speaker, the BSSD models only perform dereverberation. Hence, the SI-SDR is the highest for this case. The low WER of 9.65% indicates that Google-ASR recognizes reverberant audio quite well. However, all BSSD models could lower the WER even further. Also, the EER is lowest for one speaker. This is to be expected, as no interfering components of other speakers reduce the quality of the speaker embeddings  $e$ . For  $C = 2$  speakers, it can be seen that all BSSD models outperform Conv-TasNet and spatial PIT, even though both methods have been trained to perform dereverberation as well. However, Conv-TasNet only operates on a single channel, and spatial PIT uses a static beamformer, which performs poorly in reverberant environments [6]. In these conditions, Conv-TasNet could only achieve a WER of 80.27%. For  $C = 3$  and 4 speakers, performance drops rapidly. I.e.: the SI-SDR gets lower, and both the WER and EER rise. The statistic adaption (SA) clearly outperforms the analytic adaption (AA) variants for all number of speakers. This is also expected, as the SA variant allows the network to find an optimal transformation to separate the speakers both in time- and frequency domain, while the AA enforces a fixed scheme for spatial whitening and source localization. When comparing the time-domain (TD) to the frequency-domain (FD) variants, it can be seen that the FD models perform slightly better in terms of EER. This indicates that the speaker embeddings are easier to estimate in frequency domain, as they are calculated from the enhanced spectrograms (see Figure 5).

Table II reports SI-SDR, WER and EER for the significantly shorter *simulated* RIRs. Consequently, all systems perform better in all scores. In these almost ideal conditions, Google-ASR achieved a WER of 3.04% for a single speaker without any enhancement. However, all BSSD variants could lower the WER even further. Also, Conv-TasNet and spatial PIT perform better compared to the *real* RIRs. However, Conv-TasNet still could not separate the speakers perfectly. The static beamformer of spatial PIT performs quite well for the

TABLE I: Speech separation, dereverberation and speaker identification performance for the *real* RIRs in *offline* mode.

model	$C$	SI-SDR	WER	EER
no enhancement	1	-	9.65 %	-
Conv-TasNet	2	3.99 dB	80.27 %	-
spatial PIT	2	2.26 dB	42.27 %	-
BSSD-FD-AA	1	16.91 dB	5.09 %	2.87 %
	2	8.65 dB	28.74 %	5.92 %
	3	6.75 dB	51.90 %	8.94 %
	4	5.61 dB	66.20 %	11.32 %
BSSD-FD-SA	1	14.92 dB	6.10 %	3.02 %
	2	10.23 dB	23.70 %	4.22 %
	3	8.34 dB	42.49 %	6.20 %
	4	7.17 dB	56.43 %	7.22 %
BSSD-TD-AA	1	10.07 dB	9.81 %	4.18 %
	2	6.74 dB	45.79 %	9.94 %
	3	5.22 dB	71.63 %	15.68 %
	4	4.31 dB	84.39 %	21.65 %
BSSD-TD-SA	1	14.40 dB	5.72 %	2.89 %
	2	9.33 dB	26.19 %	5.75 %
	3	7.92 dB	42.32 %	7.28 %
	4	6.84 dB	56.57 %	9.39 %

short *simulated* RIRs, achieving a WER of 17.1%. Still, all BSSD variants achieved a lower WER for  $C = 2$  speakers. Again, the FD variants perform slightly better than the TD models, and the SA layer outperforms the AA layer.

TABLE II: Speech separation, dereverberation and speaker identification performance for the *simulated* RIRs in *offline* mode.

model	$C$	SI-SDR	WER	EER
no enhancement	1	-	3.04 %	-
Conv-TasNet	2	4.74 dB	55.15 %	-
spatial PIT	2	3.06 dB	17.10 %	-
BSSD-FD-AA	1	22.72 dB	1.72 %	2.89 %
	2	10.93 dB	14.71 %	6.11 %
	3	8.62 dB	27.82 %	8.27 %
	4	7.25 dB	37.58 %	9.65 %
BSSD-FD-SA	1	22.02 dB	2.72 %	3.07 %
	2	12.06 dB	12.80 %	5.25 %
	3	9.06 dB	25.39 %	7.37 %
	4	7.40 dB	40.36 %	8.99 %
BSSD-TD-AA	1	16.87 dB	2.71 %	3.44 %
	2	10.62 dB	15.55 %	7.23 %
	3	8.31 dB	30.86 %	10.05 %
	4	6.75 dB	46.48 %	14.52 %
BSSD-TD-SA	1	22.75 dB	2.02 %	2.84 %
	2	12.82 dB	9.84 %	5.56 %
	3	10.23 dB	20.92 %	7.77 %
	4	8.57 dB	34.45 %	9.23 %

### B. Block-online mode

Table III reports SI-SDR, WER and BER for the *real* RIRs, for block lengths of  $T_B = 1s, 2.5s$  and  $5s$ . We only performed these experiments on the SA variants of the BSSD network, as the SA layer consistently outperforms the AA layer. The SI-SDR and WER are worse compared to *offline* mode, as many sources of errors build up throughout the processing chain. I.e.: Algorithm 1 may produce wrong DOA indices for short blocks and many speakers. Consequently, speaker separation is poor, resulting in erroneous speaker embeddings. Further, both the speaker separation and speaker identification modules



introduce errors on their own. For the shortest block length of  $T_B = 1s$ , there are 20 blocks for 20s of audio. In order to achieve a perfect BER score, the embeddings for the same speaker in all 20 blocks must be identical (see Eq. (30)). If the speaker is silent in one or more blocks, a perfect BER score cannot be achieved. Clearly, performance is better for larger block lengths and fewer speakers. For  $C = 2$  speakers and a block length of  $T_B = 5s$ , the WER is 34.79% for the FD variant, and 28.51% for the TD variant. The BER is 10% for the FD variant, and 4.5% for the TD variant. In contrast to the experiments in *offline* mode, all scores are slightly better for the TD models.

TABLE III: Speech separation and dereverberation performance for the *real* RIRs in *block-online* mode.

model	$C$	$T_B$	SI-SDR	WER	BER
BSSD-FD-SA	2	1.0s	3.80 dB	66.94 %	37.40 %
		2.5s	7.05 dB	44.22 %	15.88 %
		5.0s	8.64 dB	34.79 %	10.00 %
	3	1.0s	3.00 dB	75.68 %	48.90 %
		2.5s	5.19 dB	59.94 %	27.13 %
		5.0s	6.73 dB	52.98 %	14.75 %
	4	1.0s	2.49 dB	78.21 %	64.40 %
		2.5s	3.71 dB	71.96 %	43.75 %
		5.0s	4.76 dB	68.07 %	35.00 %
BSSD-TD-SA	2	1.0s	4.49 dB	60.67 %	25.30 %
		2.5s	7.04 dB	36.24 %	10.75 %
		5.0s	8.47 dB	28.51 %	4.50 %
	3	1.0s	3.30 dB	74.11 %	39.70 %
		2.5s	4.81 dB	63.82 %	23.88 %
		5.0s	6.31 dB	48.68 %	22.50 %
	4	1.0s	2.70 dB	76.32 %	47.85 %
		2.5s	3.93 dB	70.83 %	32.25 %
		5.0s	4.86 dB	65.14 %	32.50 %

Table IV reports SI-SDR, WER and BER for the *simulated* RIRs, for block lengths of  $T_B = 1s, 2.5s$  and  $5s$ . All scores are better compared to the significantly longer *real* RIRs. For  $C = 2$  speakers and a block length of  $T_B = 5s$ , the WER is 19.80% for the FD variant, and 16.75% for the TD variant. The BER is 3.25% for the FD variant, and 1.25% for the TD variant. Again, In contrast to the experiments in *offline* mode, all scores are slightly better for the TD models.

TABLE IV: Speech separation and dereverberation performance for the *simulated* RIRs in *block-online* mode.

model	$C$	$T_B$	SI-SDR	WER	BER
BSSD-FD-SA	2	1.0s	4.24 dB	58.08 %	27.55 %
		2.5s	8.13 dB	27.70 %	7.88 %
		5.0s	10.67 dB	19.80 %	3.25 %
	3	1.0s	3.49 dB	75.03 %	40.00 %
		2.5s	6.46 dB	46.98 %	17.50 %
		5.0s	7.91 dB	35.83 %	11.75 %
	4	1.0s	2.89 dB	81.74 %	49.40 %
		2.5s	5.25 dB	52.42 %	24.63 %
		5.0s	6.05 dB	49.52 %	23.75 %
BSSD-TD-SA	2	1.0s	5.82 dB	51.17 %	18.55 %
		2.5s	10.94 dB	18.21 %	3.63 %
		5.0s	11.91 dB	16.75 %	1.25 %
	3	1.0s	4.40 dB	73.55 %	30.50 %
		2.5s	7.75 dB	47.37 %	15.25 %
		5.0s	9.66 dB	39.12 %	9.75 %
	4	1.0s	3.10 dB	81.24 %	37.65 %
		2.5s	5.11 dB	60.12 %	26.00 %
		5.0s	7.80 dB	52.99 %	10.75 %

### C. Performance

Figure 9 shows the performance of the BSSD-TD-SA model with  $C = 3$  speakers and *real* RIRs. From panel (a) it can be seen that there is a significant amount of reverberation in the input mixture  $z(t)$ . Panel (b) and (d) show the extracted and dereverberated signals of male speakers. Panel (c) shows the extracted and dereverberated signal of a female speaker.

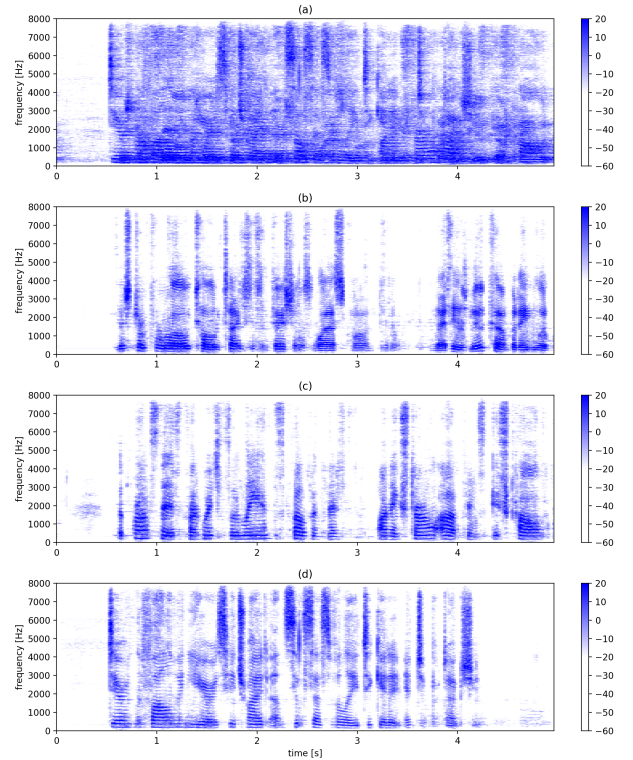


Fig. 9: Performance plot of the BSSD-TD-SA model with  $C = 3$  speakers and *real* RIRs. (a) STFT plot of the first microphone of the input mixture  $z(t)$ . (b-d) STFT plots of the extracted and dereverberated speakers  $y_c(t)$ .

### D. Model complexity

Table V reports the number of trainable parameters per variant of the BSSD network. The frequency domain (FD) variants use mostly complex-valued weights, which are counted as 2 real-valued weights. Hence, these models are significantly larger than the time domain (TD) variants. The size of the statistic adaption (SA) layer in the time domain network is comparatively small with 720,000 parameters. However, the analytic adaption (AA) layer requires additional convolutions from Eq. (26). Similar to Conv-TasNet, the time domain variant also has the advantage of a small step size of 50 samples. The number of parameters for speaker identification is almost the same for all variants.

## XI. CONCLUSION

In this paper, we introduced the *Blind Speech Separation and Dereverberation* (BSSD) network, which performs simultaneous *speaker separation, dereverberation and speaker*

TABLE V: Number of parameters for the *beamforming* and *identification* branches of the BSSD network.

model	parameters beamformer	parameters identification
BSSD-FD-AA	11,064,384	2,664,100
BSSD-FD-SA	14,757,984	2,664,100
BSSD-TD-AA	5,456,700	2,526,500
BSSD-TD-SA	6,176,700	2,526,500

*identification* in a single neural network. We proposed four variants of our system, which operate in frequency-domain and time-domain, and use analytic adaption and statistic adaption layers to perform blind speaker separation. We have shown that 100 DOA bases provide enough spatial resolution to separate up to four speakers. Further, we proposed the *block-online* mode to process longer audio recordings, as they occur in meeting scenarios. In our experiments, we could show that the BSSD network outperforms similar state-of-the art algorithms for speaker separation in terms of SI-SDR and WER.

#### REFERENCES

- [1] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.
- [2] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *IEEE 2015 Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.
- [3] E. Vincent, S. Watanabe, J. Barker, and R. Marxer, "The fourth CHiME speech separation and recognition challenge: Dataset, task and baselines," in *Proc. of the 4th Intl. Workshop on Speech Processing in Everyday Environments (CHiME 2016)*, 2016.
- [4] H. Erdogan, J. Hershey, S. Watanabe, M. Mandel, and J. L. Roux, "Improved MVDR beamforming using single-channel mask prediction networks," in *Interspeech*, Sep. 2016.
- [5] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, "Eigenvector-based speech mask estimation using logistic regression," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, 2017*, Aug. 2017, pp. 2660–2664.
- [6] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, "Eigenvector-based speech mask estimation for multi-channel speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2162–2172, 2019.
- [7] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Berlin–Heidelberg–New York: Springer, 2008.
- [8] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. Pal, "Deep complex networks," in *International Conference for Learning Representations (ICLR)*, 05 2017.
- [9] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, "Deep complex-valued neural beamformers," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2902–2906.
- [10] Y. Koyama and B. Raj, "W-net BF: dnn-based beamformer using joint training approach," *CoRR*, vol. abs/1910.14262, 2019. [Online]. Available: <http://arxiv.org/abs/1910.14262>
- [11] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 31–35.
- [12] D. Yu, M. Kolbæk, Z. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 241–245.
- [13] Z. Chen, Y. Luo, and N. Mesgarani, "Deep attractor network for single-microphone speaker separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 246–250.
- [14] D. Stoller, S. Ewert, and S. Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *CoRR*, vol. abs/1806.03185, 2018. [Online]. Available: <http://arxiv.org/abs/1806.03185>
- [15] Y. Luo and N. Mesgarani, "Tasnet: Time-domain audio separation network for real-time, single-channel speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 696–700.
- [16] —, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [17] T. Ochiai, M. Delcroix, R. Ikeshita, K. Kinoshita, T. Nakatani, and S. Araki, "Beam-tasnet: Time-domain audio separation network meets frequency-domain beamformer," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6384–6388.
- [18] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Higuchi, A. Ogawa, and T. Nakatani, "Speaker-aware neural network based beamformer for speaker extraction in speech mixtures," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, 2017*, 08 2017, pp. 2655–2659.
- [19] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černocký, "Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.
- [20] T. Yoshioka, H. Erdogan, Z. Chen, and F. Alleva, "Multi-microphone neural speech separation for far-field multi-talker speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5739–5743.
- [21] Z.-Q. Wang and D. Wang, "Integrating spectral and spatial features for multi-channel speaker separation," in *Interspeech 2018 - 19th Annual Conference of the International Speech Communication Association, Sep 2018, Hyderabad, India*, 09 2018, pp. 2718–2722.
- [22] Z. Wang, J. Le Roux, and J. R. Hershey, "Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1–5.
- [23] T. Nakatani, R. Takahashi, T. Ochiai, K. Kinoshita, R. Ikeshita, M. Delcroix, and S. Araki, "Dnn-supported mask-based convolutional beamforming for simultaneous denoising, dereverberation, and source separation," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6399–6403.
- [24] X. Chang, W. Zhang, Y. Qian, J. L. Roux, and S. Watanabe, "Mimo-speech: End-to-end multi-channel multi-speaker speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 237–244.
- [25] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umbach, "All-neural online source separation, counting, and diarization for meeting analysis," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 02 2019, pp. 91–95.
- [26] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'chime' speech separation and recognition challenge: Dataset, task and baselines," in *Interspeech 2018 - 19th Annual Conference of the International Speech Communication Association, Sep 2018, Hyderabad, India*, 03 2018.
- [27] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, D. Snyder, A. S. Subramanian, J. Trmal, B. B. Yair, C. Boeddeker, Z. Ni, Y. Fujita, S. Horiguchi, N. Kanda, T. Yoshioka, and N. Ryant, "Chime-6 challenge: tackling multispeaker speech recognition for unsegmented recordings," 2020.
- [28] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann, "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 114–126, 2012.
- [29] K. Kinoshita, M. Delcroix, H. Kwon, T. Mori, and T. Nakatani, "Neural network-based spectrum estimation for online wpe dereverberation," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, 2017*, 08 2017, pp. 384–388.
- [30] D. Williamson and D. Wang, "Time-frequency masking in the complex domain for speech dereverberation and denoising," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. PP, pp. 1–1, 04 2017.



- [31] T. Nakatani and K. Kinoshita, "A unified convolutional beamformer for simultaneous denoising and dereverberation," *IEEE Signal Processing Letters*, vol. 26, no. 6, pp. 903–907, 2019.
- [32] T. Yoshioka, T. Nakatani, K. Kinoshita, and M. Miyoshi, *Cohen I., Benesty J., Gannot S. (eds) Speech Processing in Modern Communication. Springer Topics in Signal Processing*. Springer, Berlin, Heidelberg, 12 2009, vol. 3, ch. Speech Dereverberation and Denoising Based on Time Varying Speech Model and Autoregressive Reverberation Model, pp. 151–182.
- [33] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge," in *Interspeech 2018 - 19th Annual Conference of the International Speech Communication Association, Sep 2018, Hyderabad, India*, 09 2018, pp. 2808–2812.
- [34] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.
- [35] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 165–170.
- [36] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 788 – 798, 06 2011.
- [37] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 04 2018, pp. 5329–5333.
- [38] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *CoRR*, vol. abs/1705.02304, 05 2017. [Online]. Available: <http://arxiv.org/abs/1705.02304>
- [39] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.
- [40] C. Wang, X. Lan, and X. Zhang, "How to train triplet networks with 100k identities?" in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 1907–1915.
- [41] H. Song, M. Willi, J. J. Thiagarajan, V. Berisha, and A. Spanias, "Triplet network with attention for speaker diarization," in *Interspeech 2018 - 19th Annual Conference of the International Speech Communication Association, Sep 2018, Hyderabad, India*, 08 2018.
- [42] C. Zhang, K. Koishida, and J. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 1–1, 04 2018.
- [43] C. Wu, R. Manmatha, A. J. Smola, and P. Krähénbühl, "Sampling matters in deep embedding learning," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2859–2867.
- [44] H. Kuttruff, *Room Acoustics*, 5th ed. London–New York: Spoon Press, 2009.
- [45] B. Duvenhage, K. Bouatouch, and D. Kourie, "Numerical verification of bidirectional reflectance distribution functions for physical plausibility," in *SAICSIT '13: Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, 10 2013, pp. 200–208.
- [46] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Berlin–Heidelberg–New York: Springer, 2008.
- [47] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. Berlin–Heidelberg–New York: Springer, 2006.
- [48] M. Brandstein and D. Ward, *Microphone Arrays*. Berlin–Heidelberg–New York: Springer, 2001.
- [49] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters." *VISION RESEARCH*, vol. 37, pp. 3327–3338, 1997.
- [50] B. D. V. Veen and K. M. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, no. 5, pp. 4–24, Apr. 1988.
- [51] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, Jul. 2007.
- [52] M. F. Amin, M. I. Amin, A. Y. H. Al-Nuaimi, and K. Murase, "Wirtinger calculus based gradient descent and levenberg-marquardt learning algorithms in complex-valued neural networks," in *Neural Information Processing*. Springer Berlin Heidelberg, 2011, pp. 550–559.
- [53] P. Bouboulis, "Wirtinger's calculus in general hilbert spaces," *CoRR*, vol. abs/1005.5170, 2010. [Online]. Available: <http://arxiv.org/abs/1005.5170>
- [54] R. F. H. Fischer, "Appendix A: Wirtinger calculus," in *Precoding and Signal Shaping for Digital Transmission*. Wiley-Blackwell, 2005, pp. 405–413.
- [55] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr – half-baked or well done?" in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 626–630.
- [56] E. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *CoRR*, vol. abs/1703.07737, 03 2017. [Online]. Available: <http://arxiv.org/abs/1703.07737>
- [57] X. Zhang, F. X. Yu, S. Kumar, and S. Chang, "Learning spread-out local feature descriptors," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4605–4613.
- [58] S. Haykin, *Adaptive Filter Theory*, 4th ed. New Jersey: Prentice Hall, 2002.
- [59] E. A. P. Habets and S. Gannot, "Generating sensor signals in isotropic noise fields," *The Journal of the Acoustical Society of America*, vol. 122, no. 6, pp. 3464–3470, 2007.
- [60] "Respeaker core v2.0," Website, available online at <https://www.seedstudio.com/ReSpeaker-Core-v2-0.html>; visited on September 24th 2020.
- [61] "Alsa-project," Website, visited on February 19th 2020. [Online]. Available: [https://alsa-project.org/wiki/Main\\_Page](https://alsa-project.org/wiki/Main_Page)
- [62] "python-sounddevice," Website, visited on February 19th 2020. [Online]. Available: <https://python-sounddevice.readthedocs.io/en/0.3.15/>
- [63] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: A python package for audio room simulations and array processing algorithms," *CoRR*, vol. abs/1710.04196, 2017.
- [64] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations, San Diego, 2015, Jul. 2015*.
- [65] "SpeechRecognition – a library for performing speech recognition, with support for several engines and apis, online and offline." Website, 2018, visited on March 25th 2020. [Online]. Available: <https://pypi.org/project/SpeechRecognition/>



**Lukas Pfeifenberger** received the M.Sc. (Dipl. Ing. FH) degree in computer science from the University of Applied Sciences, Salzburg, Austria, in 2004. Since 2005 he has been working in the electronics industry on projects pertaining to FPGA design, DSP programming and communication acoustics. In 2013, he received the M.Sc. (Dipl. Ing.) degree in Telematics at Graz University of Technology, Austria. Since 2015 he has been a Research Associate at the Laboratory of Signal Processing and Speech Communication, Graz University of Technology,

Austria. His research interests include signal processing, machine learning, pattern recognition, computer vision and speech processing. He currently focuses on research projects in speech enhancement, acoustic echo control, speaker separation, and data analysis for industrial applications.



**Franz Pernkopf** received his MSc (Dipl. Ing.) degree in Electrical Engineering at Graz University of Technology, Austria, in summer 1999. He earned a PhD degree from the University of Leoben, Austria, in 2002. In 2002 he was awarded the Erwin Schrödinger Fellowship. He was a Research Associate in the Department of Electrical Engineering at the University of Washington, Seattle, from 2004 to 2006. From 2010-2019 he was Associate Professor at the Laboratory of Signal Processing and Speech Communication, Graz University of Technology,

Austria. Since 2019, he is Professor for Intelligent Systems at the Signal Processing and Speech Communication Laboratory at Graz University of Technology, Austria. His research is focused on pattern recognition, machine learning, and computational data analytics with applications in signal and speech processing.





# Bibliography

- [1] P. Vary and R. Martin, *Digital Speech Transmission*. West Sussex: Wiley, 2006.
- [2] S. Haykin, *Adaptive Filter Theory*, 4th. New Jersey: Prentice Hall, 2002.
- [3] R. Lyon, “A computational model of binaural localization and separation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 8, 1983, pp. 1148–1151.
- [4] Y. Ephraim and D. Malah, “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 2, Apr. 1985.
- [5] P. C. Loizou, *Speech Enhancement: Theory and Practice*. Boca Raton: CRC Press, 2007.
- [6] J. Benesty, S. Makino, and J. Chen, *Speech Enhancement*. Berlin–Heidelberg–New York: Springer, 2005.
- [7] Y. A. Huang and J. Benesty, *Audio Signal Processing For Next-Generation Multimedia Communication Systems*. Boston: Kluwer Academic Publishers, 2004.
- [8] H. Fastl and E. Zwicker, *Psychoacoustics: Facts and Models*. Berlin–Heidelberg–New York: Springer, 1999.
- [9] R. Martin, “Noise power spectral density estimation based on optimal smoothing and minimum statistics,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, Jul. 2001.
- [10] I. Cohen, “Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, Sep. 2003.
- [11] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Berlin–Heidelberg–New York: Springer, 2008.
- [12] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Berlin–Heidelberg–New York: Springer, 2008.
- [13] S. Schmitt, M. Sandrock, and J. Cronemeyer, “Single channel noise reduction for hands free operation in automotive environments,” *AES 112-th Convention, Munich, Germany*, May 2002.
- [14] P. Mowlae, J. Kulmer, J. Stahl, and F. Mayer, *Single Channel Phase-Aware Signal Processing in Speech Communication: Theory and Practice*. West Sussex: John Wiley & Sons, 2016.
- [15] P. Mowlae and R. Martin, “On phase importance in parameter estimation for single-channel source separation,” *International Workshop on Acoustic Signal Enhancement in Aachen*, Sep. 2012.
- [16] T. Gerkmann, M. Krawczyk, and R. Rehr, “Phase estimation in speech enhancement - unimportant important or impossible?” *IEEE 27-th Convention of Electrical and Electronics Engineers in Israel*, Nov. 2012.
- [17] M. Zöhrer, R. Peharz, and F. Pernkopf, “Representation learning for single-channel source separation and bandwidth extension,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2398–2409, 2015, ISSN: 2329-9290. DOI: 10.1109/TASLP.2015.2470560.
- [18] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, “Deep learning for monaural speech separation,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 1562–1566.

- [19] Y. Wang and D. Wang, “Cocktail party processing via structured prediction,” in *Neural Information Processing Systems (NIPS)*, 2012, pp. 224–232.
- [20] F. Weninger, J. L. Roux, J. R. Hershey, and B. Schuller, “Discriminatively trained recurrent neural networks for single-channel speech separation,” in *Proc. GlobalSIP Symposium on Machine Learning Applications in Speech Processing*, Dec. 2014.
- [21] L. Deng, M. Seltzer, D. Yu, A. Acero, A. Mohamed, and G. Hinton, “Binary coding of speech spectrograms using a deep auto-encoder.,” in *Interspeech*, 2010, pp. 1692–1695.
- [22] M. Zöhrer and F. Pernkopf, “Single channel source separation with general stochastic networks,” in *Interspeech*, 2014.
- [23] Y. Wang, A. Narayanan, and D. Wang, “On training targets for supervised speech separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1849–1858, 2014.
- [24] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, “A regression approach to speech enhancement based on deep neural networks,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015, ISSN: 2329-9290.
- [25] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: <http://arxiv.org/abs/1609.03499>.
- [26] D. Stoller, S. Ewert, and S. Dixon, “Wave-u-net: A multi-scale neural network for end-to-end audio source separation,” *CoRR*, vol. abs/1806.03185, 2018. arXiv: 1806.03185. [Online]. Available: <http://arxiv.org/abs/1806.03185>.
- [27] Y. Luo and N. Mesgarani, “Tasnet: Time-domain audio separation network for real-time, single-channel speech separation,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 696–700.
- [28] —, “Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [29] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: John Wiley, 2001.
- [30] B. Sallberg, N. Grbic, and I. Claesson, “Online maximization of subband kurtosis for blind adaptive beamforming in realtime speech extraction,” in *15th International Conference on Digital Signal Processing*, Jul. 2007, pp. 603–606.
- [31] S. Makino, S. Araki, R. Mukai, and H. Sawada, “Audio source separation based on independent component analysis,” in *in Proc. ISCAS*, 2004, pp. 668–671.
- [32] P. Smaragdis, “Convolutional speech bases and their application to supervised speech separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 1–12, 2007.
- [33] E. M. Grais and H. Erdogan, “Single channel speech music separation using nonnegative matrix factorization and spectral masks,” in *International Conference on Digital Signal Processing (DSP)*, Jul. 2011, pp. 1–6.
- [34] C. Ding, X. He, H. D. Simon, and R. Jin, “On the equivalence of nonnegative matrix factorization and k-means- spectral clustering,” *International Conference on Data Mining (SDM)*, no. 4, pp. 606–610, 2005.
- [35] Z. Wang, J. Le Roux, and J. R. Hershey, “Multi-channel deep clustering: Discriminative spectral and spatial embeddings for speaker-independent speech separation,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1–5.

- [36] D. Yu, M. Kolbæk, Z. Tan, and J. Jensen, “Permutation invariant training of deep models for speaker-independent multi-talker speech separation,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 241–245.
- [37] B. D. V. Veen and K. M. Buckley, “Beamforming: A versatile approach to spatial filtering,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, no. 5, pp. 4–24, Apr. 1988.
- [38] O. Hoshuyama, A. Sugiyama, and A. Hirano, “A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters,” *IEEE Transactions on Signal Processing*, vol. 47, no. 10, Oct. 1999.
- [39] S. Gannot, D. Burshtein, and E. Weinstein, “Signal enhancement using beamforming and nonstationarity with applications to speech,” *IEEE Transactions on Signal Processing*, vol. 49, no. 8, Aug. 2001.
- [40] W. Herbordt and W. Kellermann, “Analysis of blocking matrices for generalized side-lobe cancellers for non-stationary broadband signals,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May 2002.
- [41] S. Gannot and I. Cohen, “Speech enhancement based on the general transfer function GSC and postfiltering,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 6, Nov. 2004.
- [42] M. G. Shmulik, S. Gannot, and I. Cohen, “Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, Aug. 2009.
- [43] E. Warsitz and R. Haeb-Umbach, “Blind acoustic beamforming based on generalized eigenvalue decomposition,” *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, Jul. 2007.
- [44] L. Pfeifenberger and F. Pernkopf, “Blind source extraction based on a direction-dependent a-priori SNR,” in *Interspeech 2014 - 15th Annual Conference of the International Speech Communication Association, Sep 2014, Singapore*, May 2014.
- [45] H. Kuttruff, *Room Acoustics*, 5th. London–New York: Spoon Press, 2009.
- [46] R. Talmon, I. Cohen, and S. Gannot, “Relative transfer function identification using convolutive transfer function approximation,” *IEEE Transactions on audio, speech, and language processing*, vol. 17, no. 4, May 2009.
- [47] I. Cohen, “Relative transfer function identification using speech signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, Sep. 2004.
- [48] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, “Dnn-based speech mask estimation for eigenvector beamforming,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017*, Mar. 2017, pp. 66–70.
- [49] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, “BLSTM supported GEV beamformer front-end for the 3rd CHiME challenge,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2015, pp. 444–451.
- [50] T. Schrank, L. Pfeifenberger, M. Zöhrer, J. Stahl, P. Mowlaee, and F. Pernkopf, “Deep beamforming and data augmentation for robust speech recognition: Results of the 4th CHiME challenge,” in *Proc. of the 4th Intl. Workshop on Speech Processing in Everyday Environments (CHiME 2016)*, 2016.
- [51] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, “Eigenvector-based speech mask estimation using logistic regression,” in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, 2017*, Aug. 2017, pp. 2660–2664.

- [52] M. Zöhrer, L. Pfeifenberger, G. Schindler, H. Fröning, and F. Pernkopf, “Resource efficient deep eigenvector beamforming,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*, Apr. 2018.
- [53] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, “Eigenvector-based speech mask estimation for multi-channel speech enhancement,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2162–2172, 2019.
- [54] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and Andrew, “Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2015, pp. 30–36.
- [55] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variiani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, “Multichannel signal processing with deep neural networks for automatic speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 965–979, May 2017.
- [56] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umbach, “All-neural online source separation, counting, and diarization for meeting analysis,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Feb. 2019, pp. 91–95.
- [57] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, “Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge,” in *Interspeech 2018 - 19th Annual Conference of the International Speech Communication Association, Sep 2018, Hyderabad, India*, Sep. 2018, pp. 2808–2812.
- [58] L. Pfeifenberger and F. Pernkopf, “Blind speech separation and dereverberation using neural beamforming,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. -, no. -, pp. -, 2020, submitted.
- [59] M. Delcroix, T. Ochiai, K. Zmolikova, K. Kinoshita, N. Tawara, T. Nakatani, and S. Araki, “Improving speaker discrimination of target speech extraction with time-domain speakerbeam,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 691–695.
- [60] M. Brandstein and D. Ward, *Microphone Arrays*. Berlin–Heidelberg–New York: Springer, 2001.
- [61] T. Wolff and M. Buck, “A generalized view on microphone array postfilters,” *International Workshop on Acoustic Signal Enhancement*, Sep. 2010.
- [62] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, *Robust Automatic Speech Recognition - A Bridge to Practical Applications (1st Edition)*. Elsevier, Oct. 2015.
- [63] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin–Heidelberg–New York: Springer, 2006.
- [64] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, “The third CHiME speech separation and recognition challenge: Dataset, task and baselines,” in *IEEE 2015 Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.
- [65] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, “Robust MVDR beamforming using time-frequency masks for online/offline asr in noise,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 5210–5214, Mar. 2016.
- [66] R. F. H. Fischer, “Appendix A: Wirtinger calculus,” in *Precoding and Signal Shaping for Digital Transmission*, Wiley-Blackwell, 2005, pp. 405–413.

- [67] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, “Deep complex-valued neural beamformers,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2902–2906.
- [68] T. Ochiai, M. Delcroix, R. Ikeshita, K. Kinoshita, T. Nakatani, and S. Araki, “Beam-tasnet: Time-domain audio separation network meets frequency-domain beamformer,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6384–6388.
- [69] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, *Deep speech 2: End-to-end speech recognition in english and mandarin*, 2015. arXiv: 1512.02595 [cs.CL].
- [70] E. Vincent, S. Watanabe, J. Barker, and R. Marxer, “The fourth CHiME speech separation and recognition challenge: Dataset, task and baselines,” in *Proc. of the 4th Intl. Workshop on Speech Processing in Everyday Environments (CHiME 2016)*, 2016.
- [71] C. Kim, D. Gowda, D. Lee, J. Kim, A. Kumar, S. Kim, A. Garg, and C. Han, *A review of on-device fully neural end-to-end automatic speech recognition algorithms*, 2020. arXiv: 2012.07974.
- [72] L. Pfeifenberger, T. Schrank, M. Zöhrer, M. Hagmüller, and F. Pernkopf, “Multi-channel speech processing architectures for noise robust speech recognition: 3rd CHiME challenge results,” in *Proc. IEEE ASRU*, 2015.
- [73] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The kaldii speech recognition toolkit,” in *ASRU*, 2011.
- [74] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. Berlin–Heidelberg–New York: Springer, 2006.
- [75] M. Souden, J. Benesty, and S. Affes, “On optimal frequency-domain multichannel linear filtering for noise reduction,” *IEEE Trans. Audio, Speech, Lang. Process*, pp. 260–275, Feb. 2010.
- [76] M. G. Shmulik, S. Gannot, and I. Cohen, “A sparse blocking matrix for multiple constraints GSC beamformer,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2012.
- [77] E. Warsitz, A. Krueger, and R. Haeb-Umbach, “Speech enhancement with a new generalized eigenvector blocking matrix for application in a generalized sidelobe canceller,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 73–76, May 2008.
- [78] J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 196–200.
- [79] *Respeaker core v2.0*, Website, Available online at <https://www.seeedstudio.com/ReSpeaker-Core-v2-0.html>; visited on September 24th 2020.
- [80] M. Taseska and E. A. Habets, “MMSE-based blind source extraction in diffuse noise fields using a complex coherence-based a priori SAP estimator,” *International Workshop on Acoustic Signal Enhancement*, Sep. 2012.
- [81] I. Cohen, “Analysis of two-channel generalized sidelobe canceller (gsc) with post-filtering,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, Nov. 2003.



- [82] I. Cohen, "Multichannel post-filtering in nonstationary noise environments," *IEEE Transactions on Signal Processing*, vol. 52, no. 5, May 2004.
- [83] L. Pfeifenberger and F. Pernkopf, "A multi-channel postfilter based on the diffuse noise sound field," in *European Association for Signal Processing Conference 2014*, Jun. 2014.
- [84] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters.," *VISION RESEARCH*, vol. 37, pp. 3327–3338, 1997.
- [85] N. Schinkel-Bielefeld, "Audio quality evaluation in mushra tests—influences between loop setting and a listeners' ratings," in *Audio Engineering Society Convention 142*, May 2017. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=18655>.
- [86] I. Holube and B. Kollmeier, "Speech intelligibility prediction in hearing-impaired listeners based on a psychoacoustically motivated perception model," *The Journal of the Acoustical Society of America*, vol. 100, pp. 1703–16, 1996.
- [87] J. B. Allen, *How do Humans Process and Recognize Speech?* R. P. Ramachandran and R. J. Mammone, Eds. Springer US, 1995, pp. 251–275, ISBN: 978-1-4615-2281-2.
- [88] M. Wältermann, *Dimension-based Quality Modeling of Transmitted Speech*. Springer Publishing Company, Incorporated, 2013, ISBN: 3642350186, 9783642350184.
- [89] J. Ma and P. C. Loizou, "Snr loss: A new objective measure for predicting the intelligibility of noise-suppressed speech," *Speech Communication*, vol. 53, no. 3, pp. 340–354, 2011.
- [90] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.
- [91] T. Houtgast and H. Steeneken, "Evaluation of speech transmission channels by using artificial signals," *Acta Acustica united with Acustica*, vol. 25, pp. 355–367, 1971.
- [92] D. U. Ebem, J. G. Beerends, J. Van Vugt, C. Schmidmer, R. E. Kooij, and J. O. Uguru, "The impact of tone language and non-native language listening on measuring speech quality," *Journal of the Audio Engineering Society (JAES)*, vol. 59, no. 9, pp. 647–655, 2011.
- [93] Huber and Kollmeier, "Pemo-q a new method for objective audio quality assessment using a model of auditory perception," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 1902–1911, Nov. 2006, ISSN: 1558-7916.
- [94] *ITU-T recommendation P.862. Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, 2000.
- [95] E. Vincent, "Improved perceptual metrics for the evaluation of audio source separation," *10th Int. Conf. on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pp. 430–437, 2012.
- [96] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr – half-baked or well done?" In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 626–630.
- [97] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics Doklady*, vol. 10, p. 707, 1966.
- [98] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "A short-time objective intelligibility measure for time-frequency weighted noisy speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 4214–4217.
- [99] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, Sep. 2011.

- [100] M. Souden, J. Chen, J. Benesty, and S. Affes, “An integrated solution for online multichannel noise tracking and reduction,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 7, Sep. 2011.
- [101] —, “Gaussian model-based multichannel speech presence probability,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 5, Jul. 2010.
- [102] H. Sawada, S. Araki, and S. Makino, “Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 516–527, Mar. 2011.
- [103] H. Erdogan, J. Hershey, S. Watanabe, M. Mandel, and J. L. Roux, “Improved MVDR beamforming using single-channel mask prediction networks,” in *Interspeech*, Sep. 2016.
- [104] Z. Meng, S. Watanabe, J. R. Hershey, and H. Erdogan, “Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 271–275.
- [105] H. Erdogan, T. Hayashi, J. R. Hershey, T. Hori, C. Hori, W.-N. Hsu, S. Kim, J. L. Roux, Z. Meng, and S. Watanabe, “Multi-channel speech recognition : Lstms all the way through,” 2016.
- [106] C. Bøddeker, H. Erdogan, T. Yoshioka, and R. Haeb-Umbach, “Exploring practical aspects of neural mask-based beamforming for far-field speech recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 6697–6701.
- [107] J. Heymann, L. Drude, C. Boeddeker, P. Hanebrink, and R. Haeb-Umbach, “Beamnet: End-to-end training of a beamformer-supported multi-channel asr system,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5325–5329.
- [108] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [109] D. Rumelhart, G. Hintont, and R. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [110] X. Yao, “Evolving artificial neural networks,” *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.
- [111] R. Neal, “Bayesian training of backpropagation networks by the hybrid monte carlo method,” Dept. of Computer Science, University of Toronto, Tech. Rep., 1993.
- [112] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *AISTATS*, 2011.
- [113] J. D. V. Dorsel, “Serial order: A parallel distributed processing approach,” in *Neural-Network Models of Cognition*, vol. 121, 1997.
- [114] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [115] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.
- [116] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference for Learning Representations, San Diego, 2015*, Jul. 2015.
- [117] Y. Koyama and B. Raj, “W-net BF: dnn-based beamformer using joint training approach,” *CoRR*, vol. abs/1910.14262, 2019. arXiv: 1910.14262. [Online]. Available: <http://arxiv.org/abs/1910.14262>.

- [118] D. Williamson and D. Wang, “Time-frequency masking in the complex domain for speech dereverberation and denoising,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. PP, pp. 1–1, Apr. 2017. DOI: 10.1109/TASLP.2017.2696307.
- [119] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. Pal, “Deep complex networks,” in *International Conference for Learning Representations (ICLR)*, May 2017.
- [120] M. F. Amin, M. I. Amin, A. Y. H. Al-Nuaimi, and K. Murase, “Wirtinger calculus based gradient descent and levenberg-marquardt learning algorithms in complex-valued neural networks,” in *Neural Information Processing*, Springer Berlin Heidelberg, 2011, pp. 550–559.
- [121] P. Bouboulis and S. Theodoridis, “Extension of wirtinger’s calculus to reproducing kernel hilbert spaces and the complex kernel lms,” *Trans. Sig. Proc.*, vol. 59, no. 3, pp. 964–978, Mar. 2011, ISSN: 1053-587X.
- [122] D. H. Brandwood, “A complex gradient operator and its application in adaptive array theory,” *IEE Proceedings F - Communications, Radar and Signal Processing*, vol. 130, no. 1, pp. 11–16, 1983.
- [123] K. Kreutz-Delgado, “The complex gradient operator and the cr-calculus,” *CoRR*, vol. abs/0906.4835, 2009.
- [124] E. Kreyszig, *Advanced Engineering Mathematics*. John Wiley & Sons, 2010, ch. 13.
- [125] C. Bøddeker, P. Hanebrink, L. Drude, J. Heymann, and R. Haeb-Umbach, “On the computation of complex-valued gradients with application to statistically optimum beamforming,” *CoRR*, vol. abs/1701.00392, 2017.
- [126] A. Hirose and S. Yoshida, “Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 4, pp. 541–551, 2012.
- [127] E. A. P. Habets and S. Gannot, “Generating sensor signals in isotropic noise fields,” *The Journal of the Acoustical Society of America*, vol. 122, no. 6, pp. 3464–3470, 2007.
- [128] D. B. Paul and J. M. Baker, “The design for the wall street journal-based csr corpus,” in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT ’91, Harriman, New York: Association for Computational Linguistics, 1992, pp. 357–362, ISBN: 1-55860-272-0.
- [129] *PyTube – a lightweight, pythonic, dependency-free, library for downloading youtube videos*. 2018. [Online]. Available: <https://python-pytube.readthedocs.io/en/latest/>.
- [130] *SpeechRecognition – a library for performing speech recognition, with support for several engines and apis, online and offline*. Website, Visited on March 25th 2020., 2018. [Online]. Available: <https://pypi.org/project/SpeechRecognition/>.
- [131] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, “Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 708–712.
- [132] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Higuchi, A. Ogawa, and T. Nakatani, “Speaker-aware neural network based beamformer for speaker extraction in speech mixtures,” in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, 2017*, Aug. 2017, pp. 2655–2659.
- [133] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černocký, “Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

- [134] T. Yoshioka, H. Erdogan, Z. Chen, and F. Alleva, “Multi-microphone neural speech separation for far-field multi-talker speech recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5739–5743.
- [135] Z.-Q. Wang and D. Wang, “Integrating spectral and spatial features for multi-channel speaker separation,” in *Interspeech 2018 - 19th Annual Conference of the International Speech Communication Association, Sep 2018, Hyderabad, India*, Sep. 2018, pp. 2718–2722.
- [136] T. Nakatani, R. Takahashi, T. Ochiai, K. Kinoshita, R. Ikeshita, M. Delcroix, and S. Araki, “Dnn-supported mask-based convolutional beamforming for simultaneous denoising, dereverberation, and source separation,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6399–6403.
- [137] X. Chang, W. Zhang, Y. Qian, J. L. Roux, and S. Watanabe, “Mimo-speech: End-to-end multi-channel multi-speaker speech recognition,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 237–244.
- [138] R. Gu, J. Wu, S. Zhang, L. Chen, Y. Xu, M. Yu, D. Su, Y. Zou, and D. Yu, “End-to-end multi-channel speech separation,” *CoRR*, vol. abs/1905.06286, 2019. [Online]. Available: <http://arxiv.org/abs/1905.06286>.
- [139] M. D. Zeiler, G. W. Taylor, and R. Fergus, “Adaptive deconvolutional networks for mid and high level feature learning,” in *2011 International Conference on Computer Vision*, 2011, pp. 2018–2025.
- [140] N. Wiener, “The linear filter for a single time series,” in *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. 1964, pp. 81–103.
- [141] T. Yoshioka, T. Nakatani, K. Kinoshita, and M. Miyoshi, “Cohen i., benesty j., gannot s. (eds) speech processing in modern communication. springer topics in signal processing,” in Springer, Berlin, Heidelberg, Dec. 2009, vol. 3, ch. Speech Dereverberation and Denoising Based on Time Varying Speech Model and Autoregressive Reverberation Model, pp. 151–182. DOI: 10.1007/978-3-642-11130-3\_6.
- [142] R. Scheibler, E. Bezzam, and I. Dokmanic, “Pyroomacoustics: A python package for audio room simulations and array processing algorithms,” *CoRR*, vol. abs/1710.04196, 2017.
- [143] *Alsa-project*, Website, Visited on February 19th 2020. [Online]. Available: [https://alsa-project.org/wiki/Main\\_Page](https://alsa-project.org/wiki/Main_Page).
- [144] *Python-sounddevice*, Website, Visited on February 19th 2020. [Online]. Available: <https://python-sounddevice.readthedocs.io/en/0.3.15/>.
- [145] A. Farina, “Advancements in impulse response measurements by sine sweeps,” in *Audio Engineering Society Convention 122*, May 2007. [Online]. Available: <http://www.aes.org/e-lib/browse.cfm?elib=14106>.
- [146] D. B. Paul and J. M. Baker, “The design for the wall street journal-based csr corpus,” in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT ’91, Harriman, New York: Association for Computational Linguistics, 1992, pp. 357–362, ISBN: 1-55860-272-0.
- [147] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 31–35.
- [148] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 788–798, Jun. 2011. DOI: 10.1109/TASL.2010.2064307.

- [149] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5329–5333.
- [150] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, “Deep speaker: An end-to-end neural speaker embedding system,” *CoRR*, vol. abs/1705.02304, May 2017. arXiv: 1705.02304. [Online]. Available: <http://arxiv.org/abs/1705.02304>.
- [151] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, “The fifth ‘chime’ speech separation and recognition challenge: Dataset, task and baselines,” in *Interspeech 2018 - 19th Annual Conference of the International Speech Communication Association, Sep 2018, Hyderabad, India*, Mar. 2018.
- [152] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, D. Snyder, A. S. Subramanian, J. Trmal, B. B. Yair, C. Boeddeker, Z. Ni, Y. Fujita, S. Horiguchi, N. Kanda, T. Yoshioka, and N. Ryant, *Chime-6 challenge: tackling multispeaker speech recognition for unsegmented recordings*, 2020. arXiv: 2004.09249 [cs.SD].
- [153] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann, “Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 114–126, 2012.
- [154] K. Kinoshita, M. Delcroix, H. Kwon, T. Mori, and T. Nakatani, “Neural network-based spectrum estimation for online wpe dereverberation,” in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, 2017*, Aug. 2017, pp. 384–388.
- [155] T. Nakatani and K. Kinoshita, “A unified convolutional beamformer for simultaneous denoising and dereverberation,” *IEEE Signal Processing Letters*, vol. 26, no. 6, pp. 903–907, 2019.
- [156] F. Kuech and W. Kellermann, “Orthogonalized power filters for nonlinear acoustic echo cancellation,” *Signal Processing*, vol. 86, no. 6, pp. 1168–1181, 2006, Applied Speech and Audio Processing, ISSN: 0165-1684.
- [157] F. Kuech and W. Kellermann, “Nonlinear residual echo suppression using a power filter model of the acoustic echo path,” in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 1, 2007, pp. I-73–I-76.
- [158] A. Guerin, G. Faucon, and R. Le Bouquin-Jeannes, “Nonlinear acoustic echo cancellation based on volterra filters,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 672–683, 2003.
- [159] F. Kuech and W. Kellermann, “A novel multidelay adaptive algorithm for volterra filters in diagonal coordinate representation [nonlinear acoustic echo cancellation example],” in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2004, pp. ii–869.
- [160] D. A. Bendersky, J. W. Stokes, and H. S. Malvar, “Nonlinear residual acoustic echo suppression for high levels of harmonic distortion,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 261–264.
- [161] Kun Shi, Xiaoli Ma, and G. Tong Zhou, “A residual echo suppression technique for systems with nonlinear acoustic echo paths,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 257–260.
- [162] S. Malik and G. Enzner, “State-space frequency-domain adaptive filtering for nonlinear acoustic echo cancellation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 2065–2079, 2012.

- [163] S. Malik and G. Enzner, “A variational bayesian learning approach for nonlinear acoustic echo control,” *Signal Processing, IEEE Transactions on*, vol. 61, pp. 5853–5867, Dec. 2013. DOI: 10.1109/TSP.2013.2281021.
- [164] C. M. Lee, J. W. Shin, and N. S. Kim, “Dnn-based residual echo suppression,” in *INTERSPEECH*, 2015.
- [165] T. V. Huynh, “A new method for a nonlinear acoustic echo cancellation system,” 2017.
- [166] H. Zhang and D. Wang, “Deep learning for acoustic echo cancellation in noisy and double-talk scenarios,” Sep. 2018, pp. 3239–3243. DOI: 10.21437/Interspeech.2018-1484.
- [167] G. Carbajal, R. Serizel, E. Vincent, and E. Humbert, “Multiple-input neural network-based residual echo suppression,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 231–235.
- [168] G. Carbajal, R. Serizel, E. Vincent, and E. Humbert, *Joint dnn-based multichannel reduction of acoustic echo, reverberation and noise*, 2019. arXiv: 1911.08934 [cs.SD].
- [169] Q. Lei, H. Chen, J. Hou, L. Chen, and L. Dai, “Deep neural network based regression approach for acoustic echo cancellation,” in *Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing*, ser. ICMSSP 2019, New York, NY, USA: Association for Computing Machinery, 2019, pp. 94–98, ISBN: 9781450371711.
- [170] L. Pfeifenberger and F. Pernkopf, “Nonlinear Residual Echo Suppression Using a Recurrent Neural Network,” in *Proc. Interspeech 2020*, 2020, pp. 3950–3954. DOI: 10.21437/Interspeech.2020-1473.
- [171] F. Kuech, E. Mabande, and G. Enzner, “State-space architecture of the partitioned-block-based acoustic echo controller,” *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014.
- [172] —, “State-space architecture of the partitioned-block-based acoustic echo controller,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1295–1299.
- [173] *Speex-dsp*, Website, Visited on February 19th 2020. [Online]. Available: <https://github.com/xiongyihui/speexdsp-python>.
- [174] J. .-. Soo and K. K. Pang, “Multidelay block frequency domain adaptive filter,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 2, pp. 373–376, 1990.
- [175] J. Heitkaemper, D. Jakobeit, C. Böddeker, L. Drude, and R. Haeb-Umbach, “Demystifying tasnet: A dissecting approach,” *CoRR*, vol. abs/1911.08895, 2019. [Online]. Available: <http://arxiv.org/abs/1911.08895>.
- [176] Z. Chen, Y. Luo, and N. Mesgarani, “Deep attractor network for single-microphone speaker separation,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 246–250.
- [177] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, “Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.
- [178] B. Duvenhage, K. Bouatouch, and D. Kourie, “Numerical verification of bidirectional reflectance distribution functions for physical plausibility,” in *SAICSIT ’13: Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, Oct. 2013, pp. 200–208.

- [179] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, “Deep neural network-based speaker embeddings for end-to-end speaker verification,” in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 165–170.
- [180] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.
- [181] C. Wang, X. Lan, and X. Zhang, “How to train triplet networks with 100k identities?” In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 1907–1915.
- [182] H. Song, M. Willi, J. J. Thiagarajan, V. Berisha, and A. Spanias, “Triplet network with attention for speaker diarization,” in *Interspeech 2018 - 19th Annual Conference of the International Speech Communication Association, Sep 2018, Hyderabad, India*, Aug. 2018.
- [183] C. Wu, R. Manmatha, A. J. Smola, and P. Krähenbühl, “Sampling matters in deep embedding learning,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2859–2867.
- [184] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” *CoRR*, vol. abs/1703.07737, Mar. 2017. arXiv: 1703.07737. [Online]. Available: <http://arxiv.org/abs/1703.07737>.
- [185] X. Zhang, F. X. Yu, S. Kumar, and S. Chang, “Learning spread-out local feature descriptors,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4605–4613.

# List of Figures

1.1	General architecture of spectral subtraction algorithms [13]. . . . .	16
2.1	Microphone array with four microphones in an arbitrary geometry, and two point sound sources. . . . .	24
2.2	Filter-and-sum beamformer. The microphone signals and the beamformer output are denoted as $Z_m(l, k)$ and $Y(l, k)$ , respectively. . . . .	25
2.3	Block diagram of the GSC beamformer, with the steering vector $\mathbf{v}_S(k)$ , the blocking matrix $\mathbf{B}(k)$ and the AIC $\mathbf{H}_{AIC}(k)$ . . . . .	27
2.4	Squared spatial coherence for the near-field and the far-field between two microphones with a distance of $d_{12} = 46.3mm$ . . . . .	32
2.5	Directivity pattern for the delay-and-sum beamformer. (a) $\Psi(k, \theta, \phi)$ from Eq. 2.56 for a single speaker at $\theta_Z = 50^\circ$ and $\phi_Z = 0^\circ$ . (b) $\Psi_U(k, \theta, \phi)$ from Eq. 2.68 with whitening. . . . .	38
2.6	Effectiveness of spatial whitening at low frequencies. (a) $G(l, k)$ from Eq. 2.56 for a single speaker. (b) $G_U(l, k)$ from Eq. 2.69 with whitening. . . . .	39
2.7	Block diagram of the PESQ algorithms [5]. . . . .	42
2.8	Output mapping between narrowband PESQ (dashed line), and wideband PESQ (solid line) to the subjective MOS score [5]. . . . .	43
2.9	Block diagram of the NN used by the PEASS algorithm for the computation of the OPS, TPS, IPS and APS scores. [95]. . . . .	44
3.1	Speech masks. (a) First channel of the input mixture $\mathbf{Z}(l, k)$ . (b) IBM. (c) IRM. (d) CSM. . . . .	47
3.2	Block diagram of the Eigennet architecture [53]. . . . .	52
3.3	(a) Cosine distance obtained by Eq. 3.17. (b) IRM obtained by Eq. 3.1. (c) Estimated speech mask using the Eigennet from [48]. (d) First channel of the noisy CHiME4 utterance F01_22GC010X_BUS. (e) Enhanced utterance obtained from the Eigennet and the GEV beamformer from Section 2.2.4. . . . .	54
3.4	Floorplan of a rectangular room with a circular microphone array and a region of acceptance (green) and a region of rejection (red). . . . .	55
4.1	Visualization of the complex-valued chain rule: The forward path includes the complex-valued functions $s = g(z)$ and $J = f(s)$ . Each function has its intermediate derivatives, which contribute to the backward gradient. . . . .	58
4.2	(a) Magnitude and (b) phase response of Eq. 4.26a. . . . .	63
4.3	(a) Magnitude and (b) phase response of Eq. 4.26b. . . . .	63
4.4	(a) Magnitude and (b) phase response of Eq. 4.26c. . . . .	64
4.5	(a) Magnitude and (b) phase response of Eq. 4.29. . . . .	65
4.6	(a) Magnitude and (b) phase response of Eq. 4.30. . . . .	65
4.7	Block diagram of the CNBF architecture [67]. . . . .	69
4.8	Shoebox model of a living room showing stationary sound sources $\mathbf{S}_1$ to $\mathbf{S}_3$ , and dynamic sound sources $\mathbf{D}_1$ and $\mathbf{D}_2$ . The microphone array is located next to the TV set [67]. . . . .	69
4.9	(a) First channel of the input mixture. (b) Enhanced output of the CNBF. (c) Enhanced output of the Eigennet. (d) First channel of the clean target speaker. . . . .	71
5.1	Cross-domain learning: Signal flow of the CNBF and TDNBF architectures. . . . .	75
5.2	(a) RIR recording setup for moving speakers in a $6m \times 7m$ office room. (b) Floorplan with 448 grid points. (c) Microphone array and measurement loudspeaker. . . . .	77



5.3	TDNBF architecture. . . . .	79
5.4	Separation of a reverberant RIR into its directional component $\mathbf{h}_{\text{DIR}}(t)$ and its diffuse component $\mathbf{h}_{\text{DIFF}}(t)$ . . . . .	81
5.5	Dereverberation performance using the TDNBF and WPE approaches. (a) STFT of the first microphone of the input $z(t, m = 1)$ . (b) Dereverberated speaker $y(t)$ using the TDNBF. (c) Dereverberated speaker $y_{\text{WPE}}(t)$ using the WPE algorithm [141]. (d) Anechoic reference $r(t)$ . . . . .	83
5.6	System model of an AEC with a NRES post-filter, with signals in the STFT domain. . . . .	84
5.7	NRES architecture in time-domain. . . . .	86
5.8	(a) Microphone signal $d(d)$ . (b) Enhanced output of the frequency-domain NRES. (c) Enhanced output of the time-domain NRES. (d) Ground truth of the near-end speaker $s(t)$ . . . . .	87
6.1	Histogram showing the number of RIRs from Section 5.3 assigned to a DOA index $d$ using 6.4. . . . .	91
6.2	Speaker localization using Algorithm 2. (a) $\bar{\gamma}_W(d)$ at iteration 1. (b) $\bar{\gamma}_W(d)$ at iteration 2. (c) $\bar{\gamma}_W(d)$ at iteration 3. (d) Spectrogram of the first channel of the input mixture $Z(l, k, m = 1)$ . (e) Speech presence probability $\gamma_U(l, k, \hat{d}_1)$ . (f) $\gamma_U(l, k, \hat{d}_2)$ . (g) $\gamma_U(l, k, \hat{d}_3)$ . . . . .	94
6.3	Architecture of the BSSD network. The left branch performs beamforming and dereverberation, and the right branch performs speaker identification. The symbols next to each layer denote the dimensionality of the respective output tensor. . . . .	99
6.4	Performance of the BSSD network during training. (a) Loss $\mathcal{L}_{\text{SI-SDR}}$ from Eq. 5.9 versus training epochs. (b) EER from Eq. 6.19 versus training epochs. (c) FAR and FRR from Eq. 6.17 and 6.18 versus the threshold $\delta$ , after $10^5$ training epochs. . . . .	102
6.5	Isolated and dereverberated speakers of a mixture with $C = 3$ speakers, using the BSSD network. (a) Spectrogram of the first microphone of the input $z(t, m = 1)$ . (d) Reference signal $r_1(t)$ . (c) Isolated source $y_1(t)$ . (d) $y_2(t)$ . (e) $y_3(t)$ . . . . .	103

---

## List of Tables

4.1	Performance comparison of the Eigennet and the CNBF methods. . . . .	70
5.1	Performance comparison of the CNBF and TDNBF methods with the SI-SDR objective. . . . .	80
5.2	ERLE, SI-SDR and WER scores for the time- and frequency-domain NRES post-filter, the reference system (Speex-DSP) and the AEC without a postfilter as a baseline. . . . .	86
6.1	Parameters of the BSSD network shown in Figure 6.3. . . . .	100
6.2	Performance of the time-domain BSSD network on $C = \{1, 2, 3, 4\}$ speakers. . . .	101



# Acronyms

- AEC** Acoustic Echo Canceler. 18, 26, 38, 82–85, 106
- AIC** Adaptive Interference Canceler. 25, 26
- APS** Artifact Perceptual Score. 42
- ASR** Automatic Speech Recognition. 15–17, 38, 39, 44, 67, 78, 79, 89, 101
- ATF** Acoustic Transfer Function. 15, 22–28, 30–32, 51
- BAN** Blind Analytical Normalization. 17
- BSS** Blind Source Separation. 15, 18, 43, 72, 89
- BSSD** Blind Source Separation and Dereverberation. 89, 93, 96, 98–104
- cGMM** complex Gaussian Mixture Model. 43, 46
- CNBF** Complex-Valued Neural Beamformer. 16, 65–71, 73, 78, 79, 81, 95
- CNN** Convolutional Neural Network. 48, 64
- CSM** Cosine Similarity Mask. 44, 45
- DD-SNR** Direction-Dependent SNR. 15, 16, 19, 31, 32
- DNN** Deep Neural Network. 14, 15, 39, 43, 44
- DOA** Direction Of Arrival. 15, 16, 30, 50, 51, 90–93, 95, 96, 98–101, 104, 106
- EER** Equal Error Rate. 97, 98, 101, 102
- EIR** Echo Impulse Response. 83
- EM** Expectation Maximization. 43
- ERLE** Echo Return Loss Enhancement. 84–86
- EVD** Eigenvalue Decomposition. 13, 35, 46, 49, 51, 53, 55
- FAR** False Acceptance Rate. 97, 102
- FFT** Fast Fourier Transform. 65, 71, 73, 78, 84, 87, 106
- FIR** Finite Impulse Response. 22, 28, 30, 51, 83, 95, 106
- FRR** False Reject Rate. 97, 102
- GAN** Generative Adversarial Network. 106
- GCC-PHAT** Generalized Cross Coherence Phase Transform. 19, 33, 36, 90–93, 106
- GEV** Generalized Eigenvalue. 15–18, 26, 27, 34, 46, 50–52, 54, 55, 65, 69, 72
- GMM** Gaussian Mixture Model. 16, 43, 46

- GSC** Generalized Sidelobe Canceler. 15, 16, 18, 25, 26, 31, 33, 43
- HMM** Hidden Markov Model. 16
- HRTF** Head-Related Transfer Function. 37
- IBM** Ideal Binary Mask. 44, 45
- ICA** Independent Component Analysis. 13, 15
- iFFT** inverse Fast Fourier Transform. 71, 73, 78, 84, 87, 106
- IMCRA** Improved Minima-Controlled Recursive Averaging. 14, 33
- IPD** Inter-channel Phase Difference. 28, 30, 34, 43, 45, 46, 71, 72, 95
- IPS** Interference Perceptual Score. 42
- IRM** Ideal Ratio Mask. 44, 45, 50–53
- ISM** Image Source Method. 67
- ITD** Inter-channel Time Difference. 34, 43, 71, 72, 95, 98
- LCMV** Linearly Constrained Minimum Variance. 24
- LSTM** Long Short-Term Memory. 17, 47, 48, 50, 63–66, 77, 84, 98
- MAP** Maximum A-Posteriori. 13, 43
- MCSE** Multi-Channel Speech Enhancement. 13–15, 18, 38, 39, 71, 72
- MIMO** Multiple Input - Multiple Output. 18, 21
- ML** Maximum Likelihood. 13
- MLP** Multi-Layer Perceptron. 46, 47
- MMSE** Minimum Mean-Squared Error. 13
- MOS** Mean Opinion Score. 37, 38, 40, 41
- MS** Minimum Statistics. 14
- MSE** Mean Square Error. 24, 26, 78
- MUSIC** MUltiple SIgnal Classification. 15, 16
- MVDR** Minimum Variance Distortionless Response. 15–18, 24–28, 31, 33, 34, 38, 43, 46, 50, 51, 54, 55, 65, 69, 72, 90
- NLMS** Normalized Least Mean Squares. 16, 26
- NMF** Non-negative Matrix Factorization. 15
- NN** Neural Network. 14–18, 41, 42, 46–51, 53–55, 63, 65, 66, 70–73, 76, 77, 79, 81–85, 87, 89, 90, 95–98, 100
- NRES** Nonlinear Residual Echo Suppression. 82–86

- 
- OPS** Overall Perceptual Score. 42
- PAN** Phase Aware Normalization. 17, 27, 51
- PCA** Principal Component Analysis. 35
- PEASS** Perceptual Evaluation methods for Audio Source Separation. 19, 38, 41, 42, 106
- PEMO-Q** Perceptual Model-Quality Assessment. 38, 41, 42
- PESQ** Perceptual Evaluation of Speech Quality. 19, 38, 40, 41, 67, 106
- PHAT** PHase Acoustic Transform. 15
- PMWF** Parametric Multichannel Wiener Filter. 46
- POLQA** Perceptual Objective Listening Quality Analysis. 38
- PSD** Power-Spectral Density. 15, 17, 19, 23–28, 30–32, 34, 43, 46, 49–51, 54, 105
- PSQM** Perceptual Speech Quality Measure. 40
- ReLU** Rectified Linear Unit. 47, 48, 62, 64, 77
- RIR** Room Impulse Response. 67, 73, 74, 76, 78–81, 89, 91–93, 95, 99–101
- RNN** Recurrent Neural Network. 47
- SCSE** Single-Channel Speech Enhancement. 13, 14, 18, 33, 38, 39, 71, 82
- SDR** Signal to Distortion Ratio. 19, 37–39, 78, 84, 87
- SI-SDR** Scale Independent Signal to Distortion Ratio. 38, 39, 71, 78–81, 84, 85, 98, 101, 102, 106
- SII** Speech Intelligibility Index. 37
- SNR** Signal to Noise Ratio. 16, 17, 19, 25, 27, 31, 32, 34, 37–39, 43, 44, 65–68, 71, 78, 87
- SPP** Speech Presence Probability. 17, 44
- SRP-PHAT** Steered Response Power Phase Transform. 19, 30–32
- STFT** Short-time Fourier Transform. 22, 23, 39–41, 49, 53, 54, 68, 71–73, 77, 82, 83, 85, 92, 105
- STI** Speech Transmission Index. 37, 39
- STOI** Short-Time Objective Intelligibility measure. 19, 38–40, 67
- SVD** Singular Value Decomposition. 13
- TDNBF** Time-Domain Neural Beamformer. 16, 73, 76, 78–82, 84, 87, 95, 96, 98, 104
- TPS** Target Perceptual Score. 42
- TTS** Text-To-Speech. 39
- VAD** Voice Activity Detector. 26

**VoIP** Voice over IP. 40

**WER** Word Error Rate. 19, 37–39, 67, 78, 79, 85, 89, 101

**WPE** Weighted Prediction Error. 73, 79, 81, 82

**ZCA** Zero-phase Component Analysis. 35, 49, 66, 91