Robust Speech Enhancement Using Statistical Signal Processing and Machine-Learning

Dissertation zur Erlangung des Doktorgrades
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik
der Universität Hamburg

vorgelegt von
Robert Rehr

Hamburg, 2018

Robert Rehr: *Robust Speech Enhancement Using Statistical Signal Processing and Machine-Learning*

Erstgutachter:
Prof. Dr.-Ing. Timo Gerkmann

Weitere Gutachter:
Prof. Dr. Simone Frintrop
Prof. Dr.-Ing. Dorothea Kolossa

Vorsitz der Prüfungskommision:
Prof. Dr. Chris Biemann

Tag der Einreichung:
15. Januar 2018

Tag der Disputation:
27. April 2018

ABSTRACT

With the availability of powerful mobile electronic devices, speech communication plays an important role in many applications such as telecommunications, hearing aids and voice-controlled devices. Due to their mobility, such devices are often used in noisy acoustic environments. In such situations, the microphones do not only capture the desired speech signal but also undesired background noises. This degrades the perceived quality and the intelligibility of the speech signal. Further, the performance of subsequent speech processing algorithms may be impaired by background noises. To restore the quality and possibly also the intelligibility of noise corrupted speech, speech enhancement algorithms are employed.

In this thesis, single-channel speech enhancement algorithms that either process the signal captured by a single microphone or the output of a spatial filtering algorithm are considered. The aim of this thesis is to increase the robustness of machine-learning (ML)-based and non-ML-based single-channel speech enhancement algorithms by exploiting synergies between both approaches. In conventional non-ML-based speech enhancement such as Wiener filtering based approaches, spectral gain functions are applied to the complex coefficients of the short-time Fourier spectra to enhance the noisy input signal. These gain functions are derived in a statistical framework where the clean speech and the noise Fourier coefficients are modeled using parametric probability density functions (PDFs). The parameters of the PDFs are estimated blindly from the noisy observation. Contrarily, ML-based algorithms use representative examples to learn the statistics of speech and noise which are then used for the enhancement. Often, ML-based approaches are motivated by the fact that conventional approaches are unable to follow highly non-stationary background noise types. However, it is still unclear how well ML-based approaches generalize unseen acoustic conditions.

The first part of this thesis deals with non-ML-based noise power spectral density (PSD) estimators that rely on first-order recursive smoothing filter structures. In contrast to usual linear smoothing filters, the considered noise PSD estimators adaptively change the smoothing factor based on the previously estimated noise PSD and the noisy input. We show that such noise PSD estimators are generally biased and present approaches to analytically quantify and compensate for the bias.

Second, we address a specific group of speech enhancement approaches where the speech PSD estimates are obtained using ML techniques. As the considered techniques only represent coarse spectral envelopes of speech, we refer to them as machine-learning spectral envelope (MLSE)-based approaches. The coarse speech PSD estimates of an MLSE approach result in an overestimation of the speech PSD between speech spectral harmonics. As a consequence, noise between these harmonics is not suppressed, if Gaussian speech enhancement filters, e.g., the Wiener filter, are employed. As a result, the enhanced signal exhibits noise bursts in speech active segments which reduce the perceived quality. Our analysis shows that super-Gaussian estimators are able to suppress the background noise even if the speech PSD is overestimated. Correspondingly, we propose to use these estimators to improve the quality of MLSE speech enhancement approaches. Further, an alternative approach to suppress the noise between speech spectral harmonics is proposed. Instead of using super-Gaussian models, an ML and a non-ML-based approach are combined.

In the last part of the thesis, the generalization of unseen noise conditions of deep neural network (DNN)-based enhancement schemes is considered. To make the ML approach more robust to unseen noise conditions, it is proposed to use normalized features based on speech and noise PSD estimates obtained from conventional non-ML-based enhancement algorithms. More specifically, we propose to use the *a priori* signal-to-noise ratio (SNR), i.e., the ratio between the speech PSD and the noise PSD, and the *a posteriori* SNR, i.e., the ratio between the noisy periodogram and the noise PSD, as input features. In comparison to the already existing noise aware training approaches, where an estimate of the noise PSD is appended to the features extracted from the noisy observation, the proposed approach has two major advantages: First, the proposed features are scale-invariant, i.e., their value is not influenced by the overall level of the input signal. As a result, also the performance of the DNN-based speech enhancement scheme becomes independent of the overall signal level. Second, the results show that the proposed features generally outperform noise aware training features in terms of enhancement quality in unseen noise conditions.

ZUSAMMENFASSUNG

Durch die Verfügbarkeit von leistungsfähigen, elektronischen Mobilgeräten spielt Sprachkommunikation eine immer wichtigere Rolle, inbesondere in Anwendungen wie Telekommunikation, Hörhilfen und sprachgesteuerten Geräte. Aufgrund ihrer Mobilität werden solche Geräte oft in akustischen Umgebungen eingesetzt, in denen Hintergrundgeräusche auftreten. In solchen Situationen nehmen die Mikrofone nicht nur das gewünschte Sprachsignal sondern auch die ungewünschten Geräusche auf. Dies verschlechtert die wahrgenommene Qualität und Verständlichkeit des Sprachsignals. Außerdem kann die Leistungsfähigkeit von nachfolgenden Sprachverarbeitungsalgorithmen durch die Störgeräusche verschlechtert werden. Um die Qualität und, wenn möglich, auch die Verständlichkeit der gestörten Sprache wiederherzustellen, werden Sprachverbesserungsalgorithmen eingesetzt.

In dieser Arbeit werden einkanalige Sprachverbesserungsalgorithmen betrachtet, die entweder das Signal eines einzelnen Mikrofons oder den Ausgang eines räumlichen Filters verarbeiten. Das Ziel dieser Arbeit ist es, die Robustheit einkanaliger, maschinenlernbasierter (ML-basiert) Verfahren und nicht-maschinenlernbasierte (nicht-ML-basiert) Sprachverbesserungsalgorithmen durch das Ausnutzen von Synergien zu erhöhen. In konventioneller nicht-ML-basierter Sprachverbesserung, z. B. Ansätze, die auf Wiener-Filterung basieren, werden spektrale Gewichtungsfunktionen auf die komplexen Koeffizienten der Kurzzeit-Fourier-Transformation angewendet, um das verrauschte Eingangssignal zu verbessern. Diese Gewichtungsfunktionen werden in einem statistischen Rahmenwerk hergeleitet, in dem die Koeffizienten der unverrauschten Sprache und des Rauschens durch parametrische Wahrscheinlichkeitsdichten modelliert werden. Die Parameter der Verteilungen werden blind aus den verrauschten Beobachtungen geschätzt. Im Gegensatz dazu nutzen ML-basierte Algorithmen repräsentative Beispiele, um die statistischen Eigenschaften der Sprache und des Rauschens zu lernen, die anschließend für die Verbesserung verwendet werden. Häufig sind ML-basierte Ansätze dadurch motiviert, dass konventionelle Ansätze nicht in der Lage sind, hochinstationären Geräuschtypen zu folgen. Allerdings ist weiterhin unklar, wie gut ML-basierte Ansätze ungesehene akustische Konditionen generalisieren können.

Im ersten Teil dieser Arbeit geht es um nicht-ML-basierte Geräuschleistungsdichteschätzer, die auf Glättungsfilter erster Ordnung basieren. Im Gegensatz zu herkömmlichen linearen Glättungsfiltern verändern die betrachteten Geräuschleistungsdichteschätzer den Glättungsparameter adaptiv basierend auf der zuvor geschätzten Geräuschleistungsdichte und dem verrauschten Eingang. Wir zeigen, dass die Schätzung solcher Geräuschleistungsdichteschätzer im Allgemeinen fehlerbehaftet ist, und stellen Ansätze zur analytischen Bestimmung und zur Kompensation des Fehlers vor.

Als zweites wird eine spezifische Gruppe von Sprachverbesserungsansätzen adressiert, bei denen die Sprachleistungsdichtespektren durch ML-basierte Verfahren bestimmt werden.

Da die betrachteten Methoden nur grobe spektrale Einhüllende der Sprache abbilden, bezeichnen wir diese als ML-basierte Spracheinhüllendenverfahren. Die groben Sprachleistungsdichteschätzungen der ML-basierten Spracheinhüllendenverfahren führen zu einer Überschätzung der Sprachleistungsdichte zwischen den spektralen Harmonischen der Sprache. Dadurch wird das Geräusch zwischen diesen Harmonischen nicht unterdrückt, wenn gaußsche Sprachverbesserungsfilter, z. B. das Wiener Filter, eingesetzt werden. Infolgedessen ist die Geräuschreduktion in sprachaktiven Segmenten stark begrenzt, wodurch die wahrgenommene Qualität reduziert wird. Unsere Analyse zeigt, dass supergaußsche Schätzer in der Lage sind, das Geräusch zu reduzieren, auch wenn die Sprachleistungsdichte überschätzt wird. Dementsprechend schlagen wir vor, diese Art von Schätzer zur Verbesserung der Signalqualität bei ML-basierten Verbesserungsalgorithmen einzusetzen, die nur die Spracheinhüllende abbilden. Zusätzlich, schlagen wir einen alternativen Ansatz vor, um das Geräusch zwischen den spektralen Harmonischen der Sprache zu unterdrücken. Bei diesem Ansatz werden ML- und nicht-ML-basierte Ansätze miteinander kombiniert, anstatt supergaußsche Sprachmodelle zu verwenden.

Im letzten Teil dieser Arbeit wird die Generalisierbarkeit eines ML-basierten Verbesserungsverfahrens, das auf tiefen neuronalen Netzwerken (DNNs) basiert, in ungesehenen Geräuschtypen betrachtet. Um den ML-basierten Ansatz robuster gegen ungesehene Geräuschkonditionen zu machen, werden normalisierte Merkmale basierend auf der Sprach- und Geräuschleistungsdichte vorgeschlagen, die durch konventionelle, nicht-ML-basierte Verbesserungsalgorithmen bestimmt werden. Im Speziellen schlagen wir vor, das *a priori* Signal-zu-Rauschverhältnis (SNR), also das Verhältnis zwischen Sprach- und Rauschleistungsdichte, und das *a posteriori* SNR, also das Verhältnis zwischen dem verrauschten Eingangsperiodogram und der Geräuschleistungsdichte, als Eingangsmerkmale einzusetzen. Im Vergleich zu den zuvor vorgeschlagenen Ansätzen zum geräuschbewusstem Training, bei denen eine Schätzung der Geräuschleistungsdichte an die Merkmale, die aus der verrauschten Beobachtung extrahiert wurden, angehängt werden, hat der vorgeschlagene Ansatz zwei wesentliche Vorteile: Erstens sind die vorgeschlagenen Merkmale skalierungsinvariant, d. h., dass ihr Wert nicht durch den Gesamtpegel des Eingangssignals beeinflusst wird. Aufgrund dessen ist die Verbesserungsleistung des DNN-basierten Sprachverbesserungsverfahrens entsprechend unabhängig vom Gesamtpegel. Zweitens zeigen die Ergebnisse, dass die vorgeschlagenen Merkmale das geräuschbewusste Training im Hinblick auf die Verbesserungsqualität in ungesehenen Geräuschkonditionen schlagen.

EIDESTATTLICHE ERKLÄRUNG

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

R. Rh

Hamburg, den 13.01.2018

Robert Rehr

ACKNOWLEDGMENTS

This thesis has been written as a PhD student at the Speech Signal Processing Group of the Carl-von-Ossietzky Universität Oldenburg and the Signal Processing Group of the Universität Hamburg. In this part of the thesis, I would like to express my gratitude to the many people who made this work possible.

First, I want to thank my supervisor Prof. Dr.-Ing. Timo Gerkmann for all his support, the helpful advises, discussions, and ideas. I would also like to thank Prof. Dr. ir. Simon Doclo for his insightful comments, support and collaborations with his research group.

Furthermore, I want to thank Prof. Dr.-Ing. Dorothea Kolossa and Prof. Dr. Simone Frintrop for their participation in my thesis committee and their interest in my work. In this regard, I would also like to thank Prof. Dr. Chris Biemann for chairing the thesis committee.

I am especially grateful to all my current and former colleagues in Oldenburg and in Hamburg. Working with you has been a great pleasure and the very enjoyable evenings at home and at conferences, the barbecues and PlayStation battles will remain in my memory. A special thanks goes to Martin Krawczyk-Becker for all the guidance and support that you provided since the beginning of my master thesis and throughout the PhD time.

I am also very thankful to my family and my friends who supported and encouraged me during this time. Especially, I would like to thank my dad for putting up with me on many phone calls. Probably, I couldn't have done this without you.

CONTENTS

I. Introduction 1

1. Introduction 3
   - 1.1. Motivation . . . 3
   - 1.2. Non-Machine-Learning Based Single-Channel Speech Enhancement . . . 4
     - 1.2.1. Filtering Based Speech Enhancement in the Short-Time Fourier Transform Domain . . . 5
     - 1.2.2. Noise and Speech PSD estimators . . . 7
   - 1.3. Machine-Learning Based Single-Channel Speech Enhancement . . . 9
     - 1.3.1. Generative Models and Codebook Based Enhancement . . . 9
     - 1.3.2. Non-Negative Matrix Factorization . . . 13
     - 1.3.3. Neural Networks . . . 17
   - 1.4. Outline of the Thesis and Main Contributions . . . 22

2. Statistical Framework For Speech Enhancement and Instrumental Measures 27
   - 2.1. Statistical Models for Spectral Enhancement . . . 27
     - 2.1.1. Spectral Domain Speech Enhancement . . . 27
     - 2.1.2. Statistical Models for Spectral Speech Enhancement . . . 31
     - 2.1.3. Log-Spectral Models and Their Relation to Spectral Models . . . 33
     - 2.1.4. Non-ML Estimation of the Spectral Speech and Noise PSD . . . 34
   - 2.2. Instrumental Measures . . . 37
     - 2.2.1. Estimation Accuracy of Noise PSD Estimators . . . 37
     - 2.2.2. Instrumental Measures of the Perceived Signal Quality . . . 38

II. Bias of Adaptive Smoothing Based Noise PSD Estimators 41

3. On the Bias of Adaptive Smoothing Based Noise PSD Estimation 43
   - 3.1. Basic Properties and Bias Compensation . . . 45
     - 3.1.1. Stationarity and Ergodicity . . . 45
     - 3.1.2. Scale-Invariance . . . 45
     - 3.1.3. Bias Compensation . . . 46
   - 3.2. Iterative Bias Compensation . . . 46
   - 3.3. Estimating the Bias Using Transition Densities . . . 48
   - 3.4. Noise PSD Estimators in the Context of Adaptive Smoothing . . . 50
     - 3.4.1. Signal Model . . . 50
     - 3.4.2. Two Different Smoothing Factors Based on Thresholding . . . 50
     - 3.4.3. Speech Presence Probability Based Noise PSD Estimation . . . 51
   - 3.5. Bias Compensation for Noise PSD Estimation . . . 52

3.6. Evaluation 53
3.6.1. Verification of the Bias Estimation Methods 54
3.6.2. Applications to Speech-Enhancement 58
3.7. Summary 62

4. Non-Iterative Bias Compensation for Adaptive Smoothing Based Noise PSD Estimation 63
4.1. Non-Iterative Bias Compensation 64
4.2. Bias Compensation for Noise PSD Estimation 66
4.3. Evaluation 68
4.3.1. Verification of the Estimation Methods 68
4.3.2. Applications to Speech Enhancement 71
4.4. Summary 73

III. Improvements for Envelope Based Enhancement Schemes 75

5. Effects of Super-Gaussian Priors on Envelope Based Speech Enhancement 77
5.1. Speech Estimators 78
5.2. DNN-Based Speech Enhancement Scheme 80
5.3. NMF-Based Speech Enhancement Scheme 82
5.4. Importance of Super-Gaussianity for MLSE-Based Speech Enhancement . 84
5.4.1. Analysis of the Gain Functions 84
5.4.2. Effects of Super-Gaussian Estimators on the Enhancement 85
5.5. Instrumental Evaluation 89
5.5.1. Performance Impact of MOSIE's Parameters 89
5.5.2. Comparison with Common Enhancement Schemes 90
5.6. Subjective Evaluation 92
5.7. Summary 94

6. Super-Gaussian MLSE-Based Speech Enhancement Under the MixMax Model 95
6.1. MixMax Based Speech Estimator 96
6.2. Analysis of the MixMax Gain Functions 97
6.3. Evaluation 98
6.3.1. Evaluation Setup 99
6.3.2. Results 100
6.4. Summary 101

7. Combination of Non-ML and ML-Based Algorithms for Envelope Based Speech Enhancement 103
7.1. Signal Model and Statistical Models 104
7.2. MLSE-Based Speech Enhancement 105
7.3. Non-MLSE-Based Speech Enhancement 108
7.3.1. Linear Log-Spectral Filter 108
7.3.2. Log-Spectral Amplitude Estimator 109

7.4. Proposed Combination 110
7.5. Evaluation 112
7.6. Summary 114

IV. Generalization of ML-Based Enhancement 117

8. Normalized Features for Improving the Generalization of DNN-Based Speech Enhancement 119
8.1. Non-ML-Based Enhancement Algorithms 120
8.2. ML-Based Enhancement Algorithms 121
8.2.1. ML-Based Enhancement Framework 121
8.2.2. Non-Normalized Features 121
8.2.3. Proposed Normalized Features 122
8.3. Instrumental Evaluation 123
8.3.1. Audio Material, Parameters and Instrumental Measures 123
8.3.2. Analysis 125
8.3.3. Comparisons 127
8.4. Subjective Evaluation 130
8.4.1. Audio Material, Parameters and Setup 131
8.4.2. Results 132
8.5. Summary 133

V. Conclusions and Further Research 135

9. Conclusions and Further Research 137
9.1. Conclusions 137
9.2. Suggestions to Further Research 140

A. Analytic Solutions for Bias Estimation 143
A.1. Analytical Results for the Iterative Bias Estimation 143
A.2. Analytic Solutions for the Self-Similarity Optimization 143

Bibliography 147

List of Publications 167

Further Publications (Not Related to This Theisis) 169

LIST OF FIGURES

1.1. Block scheme of a general DFT-based speech enhancement scheme. Here, y_t denotes the noise corrupted sampled time-domain signal and $\hat{s}_t$ the estimated clean speech signal. 5

1.2. Gaussian, Laplace and gamma densities fitted to a histogram of the real part of clean speech coefficients. 6

1.3. Example of an HMM that models the observed weather using the underlying hidden states which are given by high and low atmospheric pressure. Here, q_ℓ denotes the random variable of the hidden state at time ℓ. Further, o_ℓ is the random variable which describes whether the observation is that it rains or it is dry. 10

1.4. General approach to enhance noisy speech using NMF. 14

1.5. Block diagram of an example feed-forward neural network. 17

2.1. Block scheme of a non-ML-based STFT filter based single-channel based enhancement scheme. 28

2.2. Block scheme showing the steps / components of a speech enhancement scheme operating in the spectral domain. The block scheme shows the blocks of the enhancement step shown in Fig. 2.1. 29

2.3. Block scheme of the master-slave filtering approach used to compute the SegSSNR and SegNR. 40

3.1. Block diagram of a first-order recursive filter structure. 44

3.2. Value of the adaptive smoothing factors $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$, and $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$, as functions of the *a posteriori* SNR $y_\ell/\overline{y}_{\ell-1} = |Y_{k,\ell}|^2/\hat{\Lambda}^{n}_{k,\ell-1} = \gamma_{k,\ell}$. . . . 51

3.3. Bias correction factor $\mathcal{C}_i = \mathbb{E}\{y_\ell\}/\overline{y}_i^{(\mathrm{fix})}$ computed for each iteration step in Algorithm 1 for various adaptive smoothing factors and the true bias correction term $\mathcal{C}_{\mathrm{MC}}$ obtained from Monte-Carlo simulations with 10^6 realizations. 54

3.4. Shape of the fitted model distribution $\tilde{f}(\overline{y}_\ell|\hat{\boldsymbol{\theta}})$, the marginalized distribution $\tilde{g}(\overline{y}_\ell|\hat{\boldsymbol{\theta}})$ obtained by using the optimized model in (3.11), and the true PDF of the filter output obtained from Monte-Carlo simulations with 10^6 samples for various adaptive smoothing factors. 56

3.5. Correction factor $\mathcal{C}$ determined using Monte-Carlo simulations on white noise for different overlaps Ω in the STFT domain and various adaptive smoothing functions. Additionally shown: the correction factors reported in Table 3.1. 57

3.6. Log-error distortion of the adaptive smoothing function $\alpha_{\text{Thr}}(y_\ell, \overline{y}_{\ell-1})$ described in (3.17) with and without the proposed correction method for speech in noise at different SNRs. The lower part (gray) of the bars represents the overestimation $\text{LogErr}_\uparrow$, whereas the upper part (black / white) is the underestimation $\text{LogErr}_\downarrow$. 60

3.7. Same as Fig. 3.6, but for the adaptive smoothing function $\alpha_{\text{SPP}}(y_\ell, \overline{y}_{\ell-1})$ described in (3.18). 61

3.8. PESQ improvement scores for a simple speech enhancement framework where the adaptive smoothing functions $\alpha_{\text{Thr}}(y_\ell, \overline{y}_{\ell-1})$ and $\alpha_{\text{SPP}}(y_\ell, \overline{y}_{\ell-1})$ are used as noise PSD estimators with and without the correction proposed in Algorithm 3. The fixed correction factor $\mathcal{C}$ was estimated using Algorithm 2. 62

4.1. Iteration steps of (4.5) normalized to the expected value of the filter input, i.e., $\overset{\circ}{\overline{y}}{}_i^{(\text{fix})}/\mathbb{E}\{y_\ell\}$, for the noise PSD estimators in (3.17) and (3.18) described in Section 3.4. 69

4.2. Estimation of the SNR depended correction $\mathcal{G}_{k,\ell}^{(\text{a})}$ obtained via Monte-Carlo simulations and Algorithm 4 for the noise PSD estimators (3.17) and (3.18) described in Section 3.4. 70

4.3. Comparison of the proposed correction methods in terms of the log-error distortion with respect to the noise PSD estimators $\alpha_{\text{Thr}}(y_\ell, \overline{y}_{\ell-1})$ given in (3.17) for noisy speech. 72

4.4. Same as in Fig. 4.3 but for the noise PSD estimator $\alpha_{\text{SPP}}(y_\ell, \overline{y}_{\ell-1})$ proposed in (3.18). 73

4.5. PESQ improvement scores depending on the adaptive smoothing functions embedded in the speech enhancement framework. Here, the adaptive smoothing functions $\alpha_{\text{Thr}}(y_\ell, \overline{y}_{\ell-1})$ and $\alpha_{\text{SPP}}(y_\ell, \overline{y}_{\ell-1})$ are used as noise PSD estimators with and without the correction proposed in Algorithm 5. The correction factor $\mathcal{G}_{k,\ell}^{(\text{a})}$ is estimated using Algorithm 4. 74

5.1. Architecture of the employed DNN. 80

5.2. Gain function $G_{k,\ell}$ of MOSIE over the *a posteriori* SNR $\gamma_{k,\ell}$ for different values of shape ν and compression β. The upper row shows the results for an *a priori* SNR of -5 dB and the lower for an *a priori* SNR of 10 dB. See Table 5.1 for related estimators for the values of ν and β. 85

5.3. Same as Fig. 5.2 but over the *a priori* SNR $\xi_{k,\ell}$ and for two fixed *a posteriori* SNRs $\gamma_{k,\ell} = 0$ dB and $\gamma_{k,\ell} = 10$ dB. 86

5.4. Spectrogram of the example speech signal in stationary pink noise at at 5 dB SNR. Here, f denotes frequency and t time. 86

5.5. *A priori* SNR $\hat{\xi}_{k,\ell}$ estimated using different enhancement schemes for the excerpt shown in Fig. 5.4. Here, f denotes frequency and t time. 87

5.6. Gain applied to the noisy input coefficients $Y_{k,\ell}$ by MOSIE for different MLSE-based enhancement schemes for the excerpt shown in Fig. 5.4. In the upper rows, $\nu = 1$ and $\beta = 0.001$ which approximates the Gaussian LSA as shown in Table 5.1. In the lower rows, $\nu = 0.2$ and $\beta = 0.001$ is used which corresponds to a super-Gaussian LSA. Here, f denotes frequency and t time. 88
5.7. PESQ improvement scores of MOSIE for all considered enhancement schemes in dependence of the shape ν and compression β. For relations to other clean speech estimators, see Table 5.1. 90
5.8. PESQ improvement scores and segmental SNR measures for different clean speech estimators employed in the non-MLSE-based, the DNN-based, and the NMF-based enhancement scheme. While LSA and STSA employ Gaussian speech priors, MOSIE ($\nu = 0.2, \beta = 0.001$) and MOSIE ($\nu = 0.2, \beta = 1$) represent modern super-Gaussian speech estimators (see Table 5.1). . . . 91
5.9. Box plot of the subjective ratings for different enhancement schemes. . . . 93

6.1. Gain functions of the super-Gaussian LSA derived under an additive model in the spectral domain and the MixMax based clean speech estimator. . . 98
6.2. PESQ improvement score (left panel) and log-kurtosis ratio (right panel) of the super-Gaussian LSA and the MixMax based estimator averaged over all noise types. 100

7.1. PESQ improvements of the combined and non-combined enhancement approaches for four different noise types over different SNRs. 113

8.1. PESQ and STOI improvements for the considered enhancement algorithms in dependence of the peak level of the clean speech signal averaged over all noise types at an input SNR of 0 dB. 125
8.2. Computational complexity of the considered algorithms in terms of the real-time factor. This factor describes how many seconds of the audio material can be processed within a second in the real world. The number on top of the bar shows the actual value of the real-time factor. 126
8.3. PESQ improvements that results for the considered enhancement algorithms in dependence of the background noise type and the SNR. The ML-based algorithms are always trained on CSNE and the noise types not given in the respective plot, i.e., the background noise is unseen in all cases. 128
8.4. Same as Fig. 8.3 but for STOI. 129
8.5. Box plots for the subjective rating of different enhancement schemes. The left column shows the results for factory 1 noise and the right column for traffic noise as test signals. The rows show different training strategies. The linking lines show pairings that are *not* identified as statistically significant by the post-hoc tests. 132

LIST OF TABLES

1.1. List of target functions used in DNN-based speech enhancement. The symbol $S_{k,\ell}$ denotes the complex speech time-frequency points while $N_{k,\ell}$ denotes noise time-frequency points. Further, $\Phi^{s}_{k,\ell}$ and $\Phi^{y}_{k,\ell}$ are the respective phases of the complex coefficients $S_{k,\ell}$ and $N_{k,\ell}$. The index k represents frequency while ℓ represents time. Here, $I(\cdot)$ denotes the indicator function which is 1 if the condition in the argument is true and 0 otherwise. 19

3.1. Correction factor $\mathcal{C} = \mathbb{E}\{y_\ell\}/\mathbb{E}\{\overline{y}_\ell\}$ for the adaptive smoothing functions in (3.17) and (3.18) without replacement of $\overline{y}_{\ell-1}$. 56

4.1. Estimation of the correction factor $\mathcal{C}^{(\mathrm{a})}$ by Monte-Carlo simulations and Algorithm 4 for the adaptive smoothing functions in (3.17) and (3.18) of Section 3.4. The correction factors are shown as linear quantities. 70

5.1. List of clean speech estimators that MOSIE generalizes. 79

LIST OF ALGORITHMS

1. Iterative estimation of the fixed correction factor $\mathcal{C}$ for adaptive functions depending on $\overline{y}_{\ell-1}$ proposed in Section 3.2. 48

2. Estimation of the fixed correction factor $\mathcal{C}$ by maximizing the self-similarity with respect to the transition density $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ as proposed in Section 3.3. Here, we refer to solutions of specific noise PSD estimators where appropriate. 49

3. Proposed algorithm for the bias compensation which is aware of the speech level. 53

4. Estimation of the time-varying correction factor $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$ for the alternative correction method considered in Chapter 4. 67

5. Bias compensation for adpative smoothing filters where the bias is corrected by scaling only the y_ℓ not occurring in $\alpha(y_\ell, \overline{y}_{\ell-1})$ by $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$. 68

6. DNN-based enhancement scheme. 81

7. NMF-based enhancement scheme. 83

8. Algorithm for combining MLSE based and non-MLSE based estimators using Bayesian inference. In the algorithm, it is assumed that the MLSE approach is either combined with the log-spectral Wiener Filter (Section 7.3.1) or the LSA (Section 7.3.2). 115

GLOSSARY

ACRONYMS AND ABBREVIATIONS

AMS	amplitude modulation spectrum
ANOVA	analysis of variance
AR	auto-regressive
ASR	automatic speech recognition
CDF	cumulative distribution function
cIRM	complex ideal ratio mask
CMVN	cepstral mean and variance normalization
CSNE	concatenated short noise excerpts
DFT	discrete Fourier transform
DNN	deep neural network
EM	expectation maximization
GAN	generative adversarial network
GMM	Gaussian mixture model
HMM	hidden Markov model
IBM	ideal binary mask
IDFT	inverse discrete Fourier transform
IRM	ideal ratio mask
IS	Itakura-Saito
LSA	log-spectral amplitude estimator
LSTM	long short-term memory
MAP	maximum *a posteriori*
MFCC	Mel-frequency cepstral coefficient
ML	machine-learning
MLSE	machine-learning spectral envelope

MOSIE	(M)MSE estimation with (o)ptimizable (s)peech (m)odel and (i)nhomogeneous (e)rror criterion
MSE	mean-squared error
MUSHRA	multi-stimulus test with hidden reference and anchor
NMF	non-negative matrix factorization
OMLSA	optimally modified log-spectral amplitude estimator
PDF	probability density function
PESQ	Perceptual Evaluation of Speech Quality
PLP	Perceptive Linear Prediction
PSD	power spectral density
RASTA	relative spectral
ReLU	rectified linear unit
SegNR	segmental noise reduction
SegSNR	segmental SNR
SegSSNR	segmental speech SNR
SNR	signal-to-noise ratio
SPP	speech presence probability
STFT	short-time Fourier transform
STOI	short-time objective intelligibility
STSA	short-term spectral amplitude estimator
TCS	temporal cepstrum smoothing
VAD	voice activity detector
VTS	vector Taylor series

MATHEMATICAL NOTATION

a	scalar a
$\mathbf{a}$	vector $\mathbf{a}$
$\mathbf{A}$	matrix $\mathbf{A}$
$\hat{a}$	estimate of a
$\breve{a}$	bias corrected a
$\mathbf{a}^T$	transpose of vector $\mathbf{a}$
$\mathbf{A}^T$	transpose of matrix $\mathbf{A}$
$(\mathbf{A})_{i,j}$	element at the ith row and jth column of matrix $\mathbf{A}$
$\mathbb{E}\{\cdot\}$	expectation operator

x_t	time-domain signal at sample t
$X_{k,\ell}$	STFT at frequency k and segment ℓ
$x_{k,\ell}^{(\log)}$	log-spectral STFT at frequency k and segment ℓ
$\lvert\cdot\rvert$	magnitude
$\lvert\cdot\rvert_1$	L_1 norm

FIXED SYMBOLS

k	frequency bin index
ℓ	segment index
o	cepstral index
r	scaling factor
t	sample index
H_x	number of units in xth hidden layer
$I^{(x)}$	number of NMF bases vectors for signal component x
J	cost function
K	number of DFT coefficients
L	number of segments
M	segment length in samples
Q	number of phonemes in Chapter 5 and 6 or number of mixtures in the speech GMM in Chapter 7
R	segment shift in samples
V	dimensionality of the feature vector
$h_{i,j}$	output of jth unit in the ith hidden layer
n_t	noise signal at sample t
s_t	speech signal at sample t
$v_{i,\ell}$	element of the feature vector $\mathbf{v}_\ell$
y_t	noisy signal at sample t
y_ℓ	input of adaptive recursive smoothing filter at segment ℓ
$\overline{y}_\ell$	output of adaptive recursive smoothing filter at segment ℓ
$A_{k,\ell}$	magnitude of complex speech coefficients
$b_{k,\ell}$	$b_{k,\ell} = f_s(y_{k,\ell}^{(\log)})F_n(y_{k,\ell}^{(\log)})/f_y(y_{k,\ell}^{(\log)})$ used in the MixMax clean speech estimator
$G_{k,\ell}$	spectral gain
$\tilde{G}_{k,\ell}$	limited spectral gain

$z_{k,\ell}$ state indicator of the enhancement algorithm used for the combination in Chapter 7

$\gamma_{k,\ell}$ *a posteriori* SNR $\gamma_{k,\ell} = |Y_{k,\ell}|^2/\Lambda^n_{k,\ell}$

$\hat{\Lambda}^{s,\mathrm{ml}}_{k,\ell}$ maximum likelihood estimate of the speech PSD

$\lambda^x_{k,\ell}$ variance of the signal component x in the log-spectral domain

$\Lambda^x_{k,\ell}$ spectral PSD of signal component x

$\lambda^{x|y}_{k,\ell}$ variance of signal component x given the value of y in the log-spectral domain

$\lambda^{xy}_{k,\ell}$ cross-covariance between signal component x and y in the log-spectral domain

$\mu^x_{k,\ell}$ mean of the signal component x in the log-spectral domain

$\mu^{x|y}_{k,\ell}$ mean of the signal component x given the value of y in the log-spectral domain

$\Phi^x_{k,\ell}$ phase of the complex STFT coefficients of signal component x

$\xi_{k,\ell}$ *a priori* SNR $\xi_{k,\ell} = \Lambda^s_{k,\ell}/\Lambda^n_{k,\ell}$

$\zeta_{k,\ell}$ equals Wiener gain times *a posteriori* SNR which often occurs in clean speech estimators

$G_{\min}$ minimum value for spectral gain

q phoneme in Chapter 5 and 6 or mixture of the speech GMM in Chapter 7

$\overline{y}^{(\mathrm{fix})}_i$ fixed value used to replace $\overline{y}_{\ell-1}$ in $\alpha(y_\ell, \overline{y}_{\ell-1})$ which is updated iteratively to estimate the bias of adaptive recursive smoothing filter (see Section 3.2)

$\overset{\circ}{\overline{y}}{}^{(\mathrm{fix})}_i$ fixed value used to replace $\overline{y}_{\ell-1}$ in $\alpha(y_\ell, \overline{y}_{\ell-1})$ for the alternative correction method in Section 4.1

z^{LSA} indicator of the LSA in Chapter 7

z^{MLSE} indicator of the MLSE approach in Chapter 7

z^{WF} indicator of the linear log-spectral filter in Chapter 7

α smoothing constant of first-order recursive smoothing filters

α_{DD} fixed smoothing constant of the decision-directed approach

α_{LogErr} fixed smoothing constant to obtain reference noise PSD for the log-error distortion measure

$\alpha^{(\mathrm{fix})}_{\mathrm{SPP}}$ fixed smoothing constant used in SPP-based noise PSD estimator

$\alpha^{\uparrow}$	smoothing factor of the threshold based noise PSD estimator if the threshold is exceeded
$\alpha^{\downarrow}$	smoothing factor of the threshold based noise PSD estimator if the threshold is not exceeded
β	parameter of exponential compression function $c(S_{k,\ell}) = \lvert S_{k,\ell}\rvert^{\beta}$
δ	controls sparsity of NMF activations
ϵ	bias in DNN cost function
κ_x	Kurtosis of the signal x
$\Delta\kappa^{(\log)}$	log-kurtosis ratio
$\varkappa$	bias correction term used in TCS
$\mu_{\log(\mathcal{N})}$	mean of the log-normal distribution
ν	shape parameter of the χ-distribution
Ω	overlap of the STFT segments
ρ	correlation coefficient
$\lambda_{\log(\mathcal{N})}$	variance of the log-normal distribution
$\boldsymbol{\theta}$	vector of parameters
$\xi_{\mathcal{H}_1}$	fixed *a priori* SNR used in SPP-based noise PSD estimator
$\xi_{\min}^{\mathrm{ml}}$	lower limit used in the maximum likelihood estimator of the *a posteriori* SNR
$\mathcal{C}$	correction factor to compensate the bias of adaptive recursive smoothing filters via scaling (see Chapter 3)
$\mathcal{C}_{\mathrm{MC}}$	correction factor obtained from Monte-Carlo simulations
$\mathcal{C}^{(\mathrm{a})}$	correction factor for adaptive recursive smoothing filter used for the alternative correction method (see Chapter 4)
$\mathcal{C}_{\mathrm{MC}}^{(\mathrm{a})}$	correction factor of the alternative correction method determined using Monte-Carlo simulations
$\mathcal{G}_{k,\ell}$	time-varying correction factor to compensate the bias of recursive adaptive smoothing factors using scaling (see Chapter 3)
$\mathcal{G}_{k,\ell}^{(\mathrm{a})}$	time-varying correction factor to compensate the bias of adaptive recursive smoothing factors using the alternative correction method (see Chapter 4)
$\mathcal{H}_0$	speech absence hypothesis used in SPP-based noise PSD estimator
$\mathcal{H}_1$	speech presence hypothesis used in SPP-based noise PSD estimator
$\mathcal{P}_0$	linearization point for VTS approach

$\mathcal{P}_0^{(x)}$	linearization point for VTS approach of the signal component x
$\mathcal{V}$	simplified mixing function used in VTS-based MLSE approach
$\mathcal{V}_x$	simplified mixing function used in VTS-based MLSE approach derived by x
$\mathcal{V}_x^{\mathcal{P}_0}$	simplified mixing function used in VTS-based MLSE approach derived by component x and evaluated at linearization point $\mathcal{P}_0$
$\mathcal{V}^{\mathcal{P}_0}$	simplified mixing function used in VTS-based MLSE approach evaluated at linearization point $\mathcal{P}_0$

$\alpha(\cdot,\cdot)$	adaptive smoothing factor
$\alpha_{\text{SPP}}(\cdot,\cdot)$	adaptive smoothing factor of the SPP-based noise PSD estimator
$\alpha_{\text{Thr}}(\cdot,\cdot)$	adaptive smoothing factor of the threshold based noise PSD estimator
$\mathcal{B}(\cdot,\cdot)$	Bhattacharyya distance of two PDFs
$c(\cdot)$	compression function used in MSE optimal spectral clean speech estimators
$\eta(\cdot,\cdot)$	Bhattacharyya coefficient of two PDFs
$f(\cdot)$	probability density function
$\tilde{f}(\cdot\vert\cdot)$	model distribution used in Section 3.3 for estimating the bias of adaptive recursive smoothing filters
${}_pF_q(\cdot;\cdot)$	generalized hypergeometric function
$F(\cdot)$	cumulative density function
$\mathcal{F}(\cdot)$	arbitrary function of a random variable
$\Gamma(\cdot)$	gamma function
$\tilde{g}(\cdot\vert\cdot)$	marginalized distribution used in Section 3.3 for estimating the bias of adaptive recursive smoothing filters
$m(\cdot)$	function that extracts mean of a PDF from a set of parameters
$\mathcal{M}(\cdot,\cdot;\cdot)$	confluent hypergeometric function
$\mathcal{N}(x\vert\cdot,\cdot)$	normal distribution of x with mean (first parameter) and variance (second parameter)
$\mathcal{N}_{\mathbb{C}}(x\vert\cdot,\cdot)$	complex circular-symmetric normal distribution of x with mean (first parameter) and variance (second parameter)
$\phi(\cdot)$	arbitrary function of the present and past inputs of a filter
$\psi(\cdot)$	digamma function
$\psi_1(\cdot)$	trigamma function
$\mathcal{W}(\cdot)$	Lambert's W-function

$\omega_{\mathrm{a}}(\cdot)$	spectral analysis window in the time domain
$\omega_{\mathrm{s}}(\cdot)$	synthesis window in the time domain
$\mathcal{Z}(x)$	bias of the recursive smoothing factor given a fixed value x for the previous filter output
$\mathbf{v}_\ell$	feature vector of segment ℓ
$\mathbf{w}_\ell$	ℓth column of the activation matrix $\mathbf{W}$
$\mathbf{w}_\ell^{(x)}$	ℓth column of the activation matrix $\mathbf{W}^(x)$
$\mathbf{z}_\ell$	vector of enhancement algorithm states $z_{k,\ell}$ at segment ℓ
$\mathbf{B}$	non-negative matrix of the basis vectors
$\mathbf{B}^{(x)}$	non-negative basis matrix of signal component x, e.g., speech or noise
$\mathbf{W}$	non-negative matrix of the activations
$\mathbf{W}^{(x)}$	non-negative activation matrix of signal component x
$\mathbf{Y}$	non-negative matrix of the noisy periodogram
$\mathbb{L}_k^{(n)}$	set of segments in a frequency band k where noise is dominant
$\mathbb{L}^{(q)}$	set of segments belonging to a specific phoneme or mixture q
$\mathbb{L}^{(s)}$	set of speech active segments
$\mathrm{IRM}(\cdot)$	function that computes the ideal ratio mask from the noisy observation
LogErr	total log-error distortion
$\mathrm{LogErr}_\uparrow$	overestimation of the total log-error distortion
$\mathrm{LogErr}_\downarrow$	underestimation of the total log-error distortion

Part I.

Introduction

CHAPTER 1

INTRODUCTION

1.1. MOTIVATION

Speech is the most natural forms of communication for human beings and poses an effective tool to exchange ideas or to express needs and emotions. Due to technical advances, speech communication is no longer restricted to face-to-face conversations but is also performed over long distances, e.g., in the form of telecommunication, or is even used as a natural way for human-machine interaction. As computationally powerful computer hardware has become available to many users, the number of speech processing devices such as smart phones, tablets and notebooks, has increased. As a consequence, speech plays an important role in many applications, e.g., hands-free telephony, digital hearing aids, speech-based computer interfaces, or home entertainment systems.

With the increasing use of mobile devices, also the demand for processing algorithms to ensure high quality speech application is constantly increasing. In many speech processing applications, one or more microphones are used to capture the voice of the targeted speaker. As the microphones are often placed at a considerable distance from the target speaker, e.g., in hearing aids or hands-free telephony, the received signal does not only contain the sound of the target speaker, but possibly also sounds of other speakers or background noises. Understanding speech becomes increasingly difficult if additional sounds interfere with the desired speech sound, especially with increasing level of the interferers [1]–[5]. Also moderate amounts of background noise that may not effect the intelligibility can reduce the perceived quality of speech [6], [7]. Speech signals may additionally be degraded by reverberation, which is caused by reflections of the speech sound on walls and other surfaces in closed rooms. While moderate levels of reverberation may improve the speech intelligibility, high amounts aggravate speech understanding such that conversations become harder to follow [5], [8].

As noisy and reverberant environments are often encountered in our daily life, approaches to reduce noise and reverberation, e.g., to restore the quality or to improve the intelligibility of corrupted speech sounds, are of particular interest. Algorithms specifically tailored to this task are commonly referred to as *speech enhancement algorithms* and have been a research topic for many decades [6], [7], [9]. Initially, noise suppression has been only considered for signals captured by a single microphone, e.g., [10]–[12], where mainly spectral and temporal features have been exploited. Meanwhile, also a significant body of work has been dedicated to speech enhancement algorithms that employ multiple microphones, e.g., [13]–[16]. Such algorithms allow spatial features to be exploited and, as a result, generally have a higher performance than single-channel approaches. Recently, also dereverberation has been considered in the context of speech enhancement [17], [18] which can be explained by its challenging nature and its higher computational complexity.

This thesis focuses on single-channel speech enhancement algorithms where mainly the noise suppression aspect is considered in this thesis. Hence, the term speech enhancement is used synonymously to noise reduction. Despite the higher performance of multi-channel noise reduction algorithms, single-channel speech enhancement is still a topic of active research. One of the reasons is that there are still applications where no extra microphones can be fitted to a device due to physical size limitations or economical reasons. Further, advances in this field also benefit multi-channel algorithms because some approaches are mathematically equivalent to a spatial filter followed by a spectral filter. Hence, by improving single-channel spectral filters, the overall performance of such multi-channel algorithms can be improved.

Another important factor for the continuing interest in single-speech enhancement is the constantly increasing use of machine-learning (ML) algorithms [19]–[24]. ML-based approaches have been considered to improve the performance of single-channel approaches in acoustic conditions where non-ML-based approaches fail, e.g., in highly non-stationary noise types. Correspondingly, research on ML algorithms constantly accompanies the research on speech enhancement algorithms. ML-based enhancement schemes generally follow a two-step approach to enhance a noisy speech signal. First, the parameters of a model are tuned using an ML algorithm on training examples. After that, the obtained models are used to separate speech from the background noise. However, an often raised concern with ML-based speech enhancement is their generalization towards unseen acoustic conditions [21], [24]–[27]. On the contrary, non-ML-based approaches do not learn any models from training data prior to processing. Instead the parameters required for the enhancement are estimated on-line and blindly from the noisy observation independent of the speaker and the noise type. Generally, non-ML-based algorithms are more robust to unseen noise conditions but are often not able to track very non-stationary noises. Further, non-ML approaches are usually based on strong assumptions about the independence and the distribution of the speech and noise spectral coefficients, where powerful ML-based models may find more appropriate descriptions. This thesis focuses on the robustness of ML-based and non-ML-based enhancement schemes, e.g., the generalization of ML-based algorithms in unseen acoustic environments and the tracking capabilities of non-ML noise estimators. For this, synergies between both approaches are exploited.

In the following sections of this chapter, an overview of existing ML-based and non-ML-based single-channel enhancement schemes is given. Further, the contributions and the structure of the thesis are described.

1.2. NON-MACHINE-LEARNING BASED SINGLE-CHANNEL SPEECH ENHANCEMENT

Single-channel speech enhancement algorithms are employed to obtain an estimate of the clean speech signal from the noisy input signal. For this, many different non-ML approaches exist in the literature. This thesis mainly focuses on a specific class of approaches where

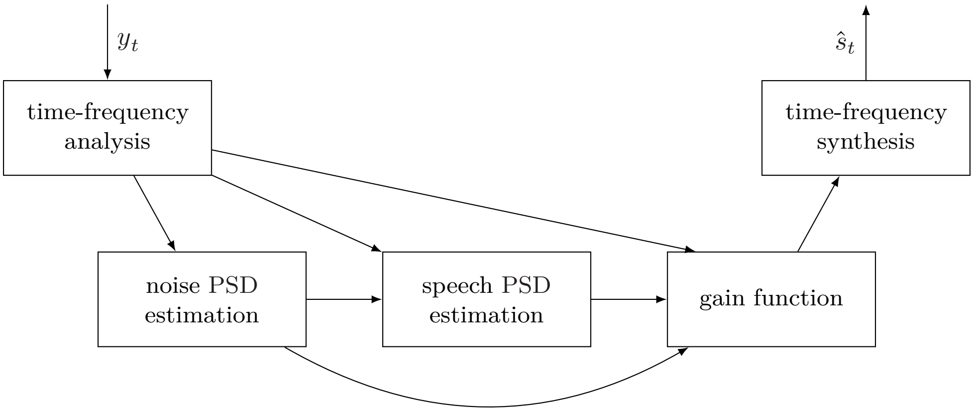

Fig. 1.1.: Block scheme of a general discrete Fourier transform based speech enhancement scheme. Here, y_t denotes the noise corrupted sampled time-domain signal and $\hat{s}_t$ the estimated clean speech signal.

the noisy input signal is filtered in the short-time Fourier transform (STFT) domain using a time-varying filter, e.g., [12], [28]–[33]. Therefore, the overview in this section focuses on this type of single-channel enhancement schemes. However, it is worth noting that in the literature also many other possible ways to enhance a noisy signal have been described, e.g., subspace based methods [34]–[38] and Kalman-filter based methods [39]–[42]. Further note that some of the approaches described in the following sections are explained in more detail in Chapter 2 as they form the basis of the work presented in this thesis.

1.2.1. Filtering Based Speech Enhancement in the Short-Time Fourier Transform Domain

Fig. 1.1 shows a general framework which is applicable to a wide range of STFT-based filtering methods in speech enhancement. Most STFT-based filtering schemes assume that the speech signal is corrupted by additive noise which yields the noisy signal. This assumption is often used because it is well motivated by the physical properties of sound [43]. The task of the enhancement system is to estimate the clean speech signal from the corrupted version of the speech signal. For this, the signal is transformed to a time-frequency representation using the STFT which is used to obtain an estimate of the clean speech spectrum. Often, the estimation the clean speech coefficients can be represented as the component-wise multiplication of a real-valued gain function and the noisy complex-valued spectrum. In many cases, the computation of the gain function depends on the noise power spectral density (PSD) and the speech PSD as shown in Fig. 1.1. The noise PSD, as well as, the speech PSD are estimated blindly from the noisy observation. The estimation of the speech PSD is often based on the estimate of the noise PSD, which has to be obtained in advance. The enhanced spectra are transformed back to

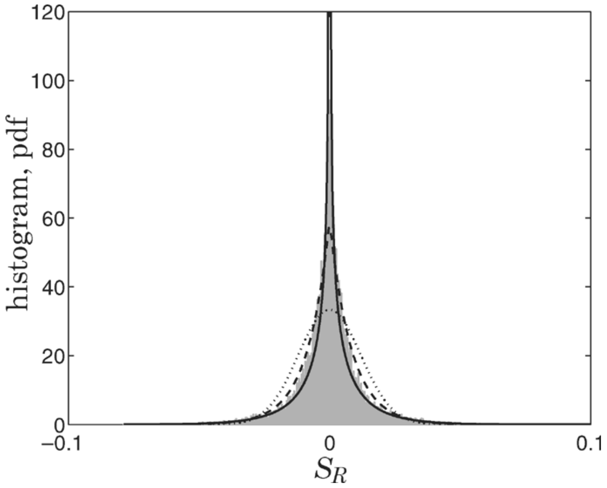

Fig. 1.2.: Gaussian (dotted), Laplace (dashed) and gamma (solid) densities fitted to a histogram (shaded) of the real part of clean speech coefficients. (Taken from [46], © IEEE 2005).

the time-domain and the estimated clean speech signal is obtained from an overlap-add procedure [44].

One of the first methods that has been proposed to reconstruct the clean speech coefficients from the noisy observation are the spectral subtraction methods presented in [11], [45]. Here, an averaged noise spectrum is subtracted from the noisy observation and is used as the speech estimate. In contrast to that, modern clean speech estimator are commonly derived in a statistical framework, where the clean speech and the noise coefficients are modeled by parametric probability density functions (PDFs) [9], [12], [28]. This allows the derivation of statistically optimal estimators that minimize specific cost functions, e.g., the mean-squared error (MSE), or maximize the posterior distribution.

In such statistical frameworks, the clean speech coefficients and the noise coefficients are often assumed to follow a complex Gaussian distribution. This is justified by the central limit theorem [47, Chapter 4]. Here, the time-domain coefficients are interpreted as random variables and due to the discrete Fourier transform (DFT), which is a linear combination of the time-domain samples, the distribution of the spectral coefficients converges towards a Gaussian distribution with increasing number of time-domain coefficients [48], [47, Chapter 4]. Deriving the MSE optimal estimator of the complex speech coefficients under a Gaussian model for speech and noise results in the well-known Wiener filter [44]. The use of this statistical framework further allows the derivation of estimators that minimize the MSE with respect to functions of complex speech coefficients, e.g., the speech spectral magnitude [12], the logarithmized magnitude [28], or exponential compressions of the speech magnitude [49].

However, experiments that have been conducted to measure the PDF of clean speech spectral coefficients showed that the Gaussian assumption may not be appropriate [46], [50], [51]. Instead, it has been found that the PDF of the complex speech coefficients is rather super-Gaussian which is explained by the strong correlations over time [46], [52]. In comparison to Gaussian distributions, a super-Gaussian distribution has a more spiky peak and more heavy tails. Fig. 1.2 shows the results of an experiment conducted in [46] where the estimated PDF of the clean speech coefficients is compared to various parametric distributions. The super-Gaussian distributions, i.e., the Laplace and the gamma densities, show a better fit than the Gaussian distribution. This observation has motivated research on so-called super-Gaussian clean speech estimators, e.g., [30]–[32], [46], [50], [53], where the clean speech spectral coefficients are modeled by super-Gaussian densities. Also the noise coefficients have been modeled by super-Gaussian distributions, e.g., [54], [55] where a Laplace distribution has been considered. Still, most publications use the Gaussian model to describe the noise coefficients.

Most clean speech estimators only focus on enhancing the amplitude of the noisy DFT coefficients and combine the estimated clean speech amplitudes with the noisy phase. For a long time, it has been believed that enhancing the phase is unimportant [56]. However, more recent experiments indicate that enhancing the phase may be beneficial [57], [58]. The findings inspired algorithms for phase estimation [59] and phase-aware clean speech estimators [33], [60], [61]. An overview of this topic is given in [62].

1.2.2. Noise and Speech PSD estimators

As shown in Fig. 1.1, statistically motivated estimators require estimates of the speech and the noise PSDs. Both quantities are blindly estimated from the noisy observation. The following paragraphs give an overview over methods that have been proposed in the literature.

For many non-ML-based enhancement schemes, updating the noise PSD is the first step when a new noisy observation is processed. A simple method for estimating the noise PSD is to smooth the noisy input periodogram over time and to suspend the update for segments where speech is present. Such methods belong to the group of voice activity detector (VAD)-based approaches where VADs form a subject of its own research, e.g., [63]–[65]. This noise estimation technique is, however, very limited as it allows only stationary background noises to be tracked.

The restrictions of VAD-based noise PSD estimators motivated more sophisticated approaches such as minimum statistics based methods [29], [66]. Here, the noise PSD is tracked by following the minima of the temporarily smoothed noisy periodograms. For this, the smoothed periodograms of the last 1.5 s are stored in a buffer and the noise PSD is obtained by finding the minimum value in the buffer for each frequency band. Due to the temporal sparsity of speech, i.e., the short pauses between speech bursts, this method allows the noise PSD to be updated also in speech active segments. However, the minimum search is a biased estimator and methods for estimating and correcting the bias have been analyzed in [66]. Despite the improvements over the VAD-based approach, this method still suffers from insufficient tracking capabilities in non-stationary noises, especially in cases where noise level increases. This is due to the employed search buffer because low energy spectra that remain in the buffer may delay the increase in noise PDF estimate by up to 1.5 s in the worst case. Improvements of this approach have been considered in [67] where the temporal smoothing for the noisy periodograms, which are later used for the minimum search, has been enhanced.

Furthermore, methods have been proposed which compute an MSE optimal estimate of the noise periodogram, which is then averaged over time to estimate the noise PSD. Such methods have been shown to be able to track moderately non-stationary noises [68]–[71]. An improved approach related to [69] has been proposed in [70], [71] where the MSE optimal estimator of the noise periodogram is derived under a speech presence and speech absence model. This results in a noise PSD estimator where the noisy periodogram is recusively smoothed with an adaptively changing and frequency dependent smoothing constant. The adaption is based on the speech presence probability (SPP) which is estimated for each time-frequency point in the STFT. If speech is likely to be present in a time-frequency bin the noise tracking in the respective frequency band is slowed down to avoid speech energy from leaking into the noise PSD estimate. This approach is related to the previously proposed method in [72], where a frequency dependent VAD is employed. As in the approach in [70], [71], the noise tracking in [72] is stopped if a time-frequency point is marked as speech active. In [73] this general idea has been used to derive an ML-based noise PSD estimator. Here, the noise presence probability, i.e., the opposite of the SPP, is estimated using a deep neural network (DNN) which is used to control the smoothing in the frequency bands of the STFT. These methods are further related to minimum-controlled recursive averaging based approaches [74], [75]. In contrast to the method considered in [70], [71], a minimum statistics based noise PSD estimate is required to determine the SPP in [74], [75] which is then used to control the amount of smoothing of first-order recursive smoothing filters. Various variations of the minimum-controlled recursive averaging methods have been proposed, e.g., [76], [77]. Other methods that have been proposed for noise PSD estimation employ subspace techniques [78], high-resolution DFTs [79] or baseline tracking [80]. Further, an overview over heuristically motivated but computationally low complex methods has been presented in [81, Chapter 5].

As shown in Fig. 1.1, the speech PSD is estimated based on the noise PSD and the time-domain representation of the noisy observation. In [12], the maximum likelihood optimal speech PSD estimator and the widely used decision-directed approach have been presented. The decision-directed approach can be considered an extension of the maximum likelihood speech PSD estimator, which combines the estimated clean speech coefficients with the maximum likelihood speech PSD estimate. Temporal cepstrum smoothing (TCS) [82], [83] has considerable advantages over the decision-directed approach and the maximum likelihood estimator. In this approach, the maximum likelihood speech PSD estimate is transformed to the cepstral domain where only the coefficients are smoothed that are irrelevant for speech. In comparison to the maximum likelihood estimator and the decision-directed approach, this method causes less musical tone artifacts.

1.3. MACHINE-LEARNING BASED SINGLE-CHANNEL SPEECH ENHANCEMENT

In contrast to non-ML-based enhancement schemes where the required statistical parameters are estimated blindly and on-line from the noisy observations, ML-based speech enhancement algorithms learn these statistics from training data. After training, the learned statistics are employed to enhance the noisy speech signal. Several different motivations have been given to use ML algorithms in the context of speech enhancement. One is that noise and speaker specific properties can be learned from training data, i.e., more prior knowledge is available [21], which cannot be easily included in non-ML-based schemes. Another often raised argument against non-ML-based enhancement schemes is the limited tracking capability for highly non-stationary background noises [20], [21], [25].

Many different ML-based speech enhancement methods have been proposed in the literature. Here, the algorithms are categorized based on the employed ML algorithm where the following types are distinguished

1. Gaussian mixture models (GMMs), hidden Markov models (HMMs) and codebook methods
2. Non-negative matrix factorization (NMF)
3. Deep neural networks (DNNs)

As some of the methods overlap and also combinations of various ML algorithms are possible for speech enhancement, this categorization is not necessarily exclusive. This overview tries to give a broad overview of ML-based speech enhancement methods, but is not meant to be comprehensive and, correspondingly, many methods are covered only with little algorithmic detail. The following subsections give an overview over these approaches.

1.3.1. Generative Models and Codebook Based Enhancement

HMMs have been among the first ML algorithms that have been considered for speech enhancement [19], [84], [85]. An HMM is a statistical model which has been widely employed to capture the temporal correlations of sequential data [86]. A schematic of an HMM is depicted in Fig. 1.3. It is assumed that a sequence can be described by a set of states where each state is linked to a PDF that models the observable data belonging to the respective state. The temporal evolution is captured by modeling the underlying states as a Markov chain, i.e., the probability that a specific state occurs depends only on the previous state. The PDF that describes the data given the underlying state is referred to as emission PDF while the probability of the state occurrence given the previous state is often called transition probability [86]. Often, only a sequence of observations is given and the underlying states are unknown. A typical task is then to infer the unknown states

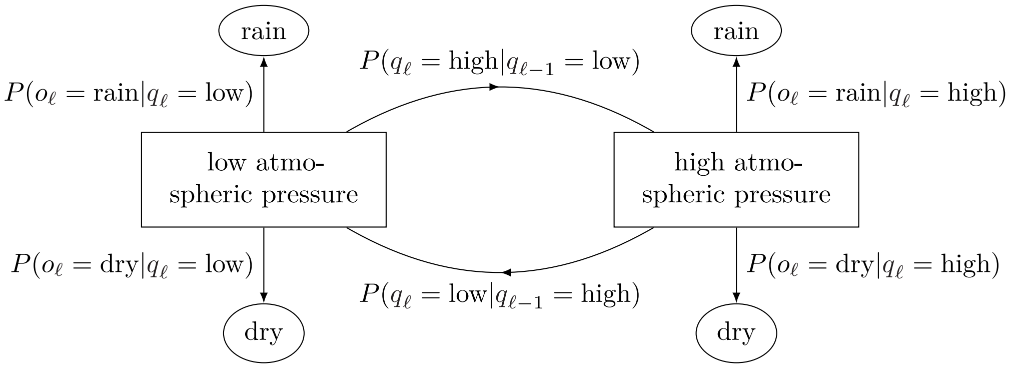
rain
rain
$P(o_\ell = \text{rain}|q_\ell = \text{low})$
$P(q_\ell = \text{high}|q_{\ell-1} = \text{low})$
$P(o_\ell = \text{rain}|q_\ell = \text{high})$
low atmospheric pressure
high atmospheric pressure
$P(o_\ell = \text{dry}|q_\ell = \text{low})$
$P(q_\ell = \text{low}|q_{\ell-1} = \text{high})$
$P(o_\ell = \text{dry}|q_\ell = \text{high})$
dry
dry

Fig. 1.3.: Example of an HMM that models the observed weather using the underlying hidden states which are given by high and low atmospheric pressure. Here, q_ℓ denotes the random variable of the hidden state at time ℓ. Further, o_ℓ is the random variable which describes whether the observation is that it rains or it is dry.

from the observation sequence which is why the states are often referred to as hidden or latent. Commonly, the parameters of an HMM are optimized on training data using the expectation maximization (EM) approach [86]–[88]. This iterative approach maximizes the likelihood on the training data and it can be shown that this algorithm always converges to a locally optimal solution [88].

The first HMM-based speech enhancement schemes have been proposed in [19], [84], [85]. Initially, only the speech component has been modeled by an HMM [84], [85] whereas the background noise has been described by a single, fixed distribution. The emission probabilities of the HMM are GMMs where each component describes the time-domain representation of speech. For this, the GMM components were assumed to have zero mean while the covariance of the time-domain signal was modeled based on auto-regressive (AR) coefficients. This type of HMM has also been referred to as AR-HMM and has also found applications in speech recognition [89]. In [84], [85], the clean speech coefficients were estimated using a maximum *a posteriori* (MAP) optimal estimator where the clean speech coefficients are iteratively updated using the EM algorithm. In each iteration, the clean speech coefficients are estimated using a weighted sum of Wiener filter based estimations. The weights are obtained by inferring the probability of the state in the HMM and GMM model to which the respective observation belongs. This probability is referred to as state posterior probability and is determined by the forward-backward algorithm [86].

In [19], the fixed distribution used to model the background noise has been replaced by a noise HMM. To infer the clean speech from the speech and the noise HMM, both HMMs are combined to form an HMM of noisy speech. For this, each state of the speech HMM is combined with each state of the noise HMM. Correspondingly, the number of states in the combined HMM corresponds to the number of states in the speech HMM times the number of states in the noise HMM. The states of the speech component and the noise component are allowed to evolve freely. Such a type of HMM is referred to as factorial HMM [90]. Further, an MSE optimal estimator of the clean speech coefficients has been derived for HMM-based speech enhancement. Similar to the MAP approach, also the MSE optimal estimator is given by a weighted average of the state-dependent estimator, i.e., for each possible combination of the speech and the noise states, the MSE optimal clean speech estimate is computed. The weights are, again, given by the state posterior probabilities.

The MSE approach has also been pursued in other publications [22], [25], [91]. In [25], an important issue with ML-based approaches has been addressed by considering acoustic environments where the background noise is not known *a priori*. Such situations are expected, e.g., in hearing-aid based applications, where the user moves freely and, with that, the acoustic environment is constantly changing. To solve this issue, various noise HMMs are trained in [25], each specializing on a single or a small group of noise types. During enhancement, the type of background noise is identified using segments containing only noise which allows the selection of the appropriate noise HMM. Another approach has been pursued in [22], where a sparsity constraint has been added to the state transition probabilities and the emission probabilities of speech. The former forces the system to prefer changes from a state that lead to only few other states, while the latter emphasizes that the overlap of the emission probabilities should be small. This allows the number of states of the speech and noise HMMs to be increased such that an appropriate background noise model can be identified by the forward-backward-algorithm without an additional selection method. Another motivation for using sparsity given in [22] is the issue that the MSE estimator combines the MSE optimal speech estimate over all possible speech and noise states. This potentially allows a combination of phonemes which could not be produced by humans or for combinations where the speech signal is explained by the noise states. By employing sparsity, the amount of combinations that exhibit a large weight in the overall MSE estimate is reduced.

In [85], another common issue with HMM-based enhancement schemes has been identified. HMMs can be easily employed to learn the spectral shapes of speech and also noise signals, but modelling the gain, i.e., the overall level of speech and noise, requires extra consideration. To resolve this issue, a MAP approach similar to [84] has been proposed in [85] to estimate the gain of each observation in an iterative EM scheme for a speech recognition application. In [19], this approach has been extended to speech enhancement. In [91], [92], this problem has been addressed by including a prior distribution of the signal level into the AR-HMM to model changes of speech and noise levels. For this, a log-normal distribution has been used where some of the parameters are considered time-invariant and some are considered time-variant. The time-invariant parameters model the overall shape of the gain distribution while the time-variant parameter, which is given by a shift of the mean in the log-spectral domain, is used to model the changes of the level during processing. The time-invariant parameters are learned off-line while the time-variant parameter is updated using an on-line EM algorithm [93]–[95].

By now, only HMM-based enhancement schemes have been considered that employ features based on AR coefficients. But also many other representation of the speech and the noise segments have been considered for modeling in HMMs. Further, some of the proposed methods in the literature neglect the temporal correlations explicitly modeled in an HMM and assume that the observations are independent, i.e., the speech and noise models are replaced by GMMs. In [96]–[99], spectral features are employed where a special focus is laid on the super-Gaussian distribution of speech as discussed in [46], [50]. Inspired by automatic speech recognition (ASR) also log-spectral and cepstral representations, e.g., Mel-frequency cepstral coefficients (MFCCs), of the speech and noise have been considered for speech enhancement and source separation [100]–[107]. However, the necessity of taking the logarithm and the absolute value to go from the spectral domain to the log-spectral and cepstral domain turns the additive signal model into a complicated, non-linear relationship between speech and noise. Hence, the expression for the noisy factorial HMM (or GMM), which is required for computing the MSE optimal clean speech estimator, often cannot be easily derived. This is why approximations of the relationship between speech and noise are employed if a non-linear feature space is used. Among the most common approaches are vector Taylor series (VTS) [103]–[105] which have originally been applied to increase the robustness of ASR towards noise [108]–[111]. Here, the non-linear function is approximated by a first-order Taylor series such that, again, an additive relationship can be exploited. But the selection of an appropriate linearization point for the Taylor series is a common issue. Alternatively, the MixMax approximation, also known as log-max approximation, is often used, which also finds its roots in noise robust ASR [112]. Here, it is assumed that the noisy log-spectrum can be approximated by the maximum of the speech and the noise log-spectral coefficients.

Another challenge that is faced with HMM and GMM-based enhancement schemes is that the emission probabilities often only represent the spectral envelope but not the spectral fine structure. In other words, only the vocal tract shape but not the excitation are modeled. This fact stems from either the low amount of states that is employed for the speech signal or by the employed features, e.g., AR coefficient based features. This issue has been addressed in [22] by employing an estimator that enforces higher suppression for low energies and in [104], [113] using a harmonic model to suppress the residual noise between harmonics. In [26], [27], an estimate of the SPP is used to achieve the same goal. For this, the estimated clean speech spectrum is multiplied by the SPP to reduce the residual noise between spectral harmonics.

Codebook based approaches are closely related to HMM and GMM-based enhancement schemes. However, instead of employing a generative model and deriving a training method for the speech and the noise model, e.g., based on EM, these methods build a using general vector quantization algorithms, e.g., [114]. Similar, to HMM-based enhancement schemes, AR coefficients or related quantities are commonly used for the quantization [20], [27], [115]. The learned codebook entries are then used as parameters in a parametric PDF which describe the statistical models of the speech, the noise and the noisy observations. These are used to obtain maximum likelihood [20] or MSE optimal estimators [115]. A potential advantage of codebook based speech enhancement is that larger codebooks can be more easily trained compared to HMM and GMM-based enhancement methods. For codebook based approaches [20], [115], 1024 codebook entries are employed for speech while only hundreds of states are employed for the speech model in HMM and GMM-based enhancement schemes [91], [97]. The challenges of codebook based approaches are similar to HMM-based and GMM-based approaches. As AR features are often employed, also here, only the spectral envelope of speech can be learned and, consequently, for reducing the noise between spectral harmonics post-processing techniques need to be employed. Further, gain-adaption techniques are required, e.g., [20], [27], which, however, are generally simpler than for HMM-based methods.

1.3.2. Non-Negative Matrix Factorization

Another widely used ML technique is NMF which has been considered in many different publications for source separation and speech enhancement, e.g., [21], [116]–[118]. Let $\mathbb{R}_+$ denote the set of real positive numbers including zero. Then, NMF is an algorithm that allows a non-negative matrix $\mathbf{Y} \in \mathbb{R}_+^{K \times L}$ to be split into two non-negative matrices $\mathbf{B} \in \mathbb{R}_+^{K \times I}$ and $\mathbf{W} \in \mathbb{R}_+^{I \times L}$ such that $\hat{\mathbf{Y}} = \mathbf{B}\mathbf{W}$ is an approximation of the matrix $\mathbf{Y}$ [119]. The notation $\hat{\cdot}$ is used to indicate an estimate of a quantity. Here, k and ℓ are the row and column index which correspond to frequency and time, respectively. The matrix $\mathbf{B}$ is often referred to as dictionary or basis matrix while the matrix $\mathbf{W}$ is the NMF coefficient matrix or activation matrix [120]. For speech enhancement application, often $K > I$. Thus, NMF is also referred to as a low-rank approximation of the matrix $\mathbf{Y}$. In signal processing applications, the spectral magnitude or the periodograms of the noisy input signal are often considered for NMF.

The non-negative decomposition of a given matrix $\mathbf{Y}$ is obtained by a two-step approach. First, a distortion $D(\mathbf{Y}\|\mathbf{B}\mathbf{W})$ between the matrix $\mathbf{Y}$ and its decomposition $\mathbf{B}\mathbf{W}$ is defined. Afterwards, this distortion is minimized with respect to the basis matrix $\mathbf{B}$ and the activation matrix $\mathbf{W}$ under the constraint that the elements of $\mathbf{B}$ and $\mathbf{W}$ have to be non-negative. Common choices for the distortion are the Euclidean distance, the generalized Kullback-Leibler divergence and the Itakura-Saito divergence, which all can be generalized by the β-divergence [121]–[123]. Generally, the optimization problem is non-convex and is solved using iterative algorithms. For the β-divergence based cost functions, such algorithms are given by a set of update rules where the elements of the basis matrix $\mathbf{B}$ and the activation matrix $\mathbf{W}$ are updated by an element-wise multiplication with an update matrix.

Probabilistic models have been a common approach in non-ML-based enhancement schemes and form also the basis for HMM-based and GMM-based ones. Interestingly, it is possible for many NMF algorithms to find formulations in a probabilistic framework. All these relationships are established by comparing the log-likelihood function that results from the

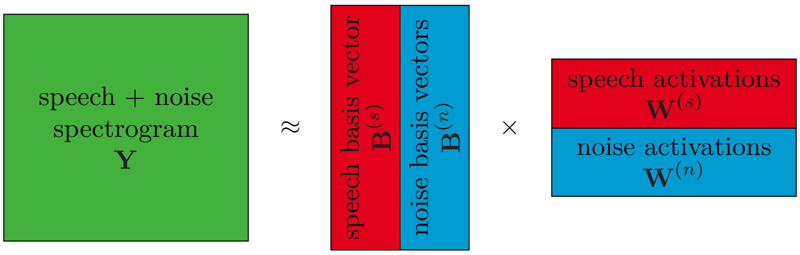

Fig. 1.4.: General approach to enhance noisy speech using NMF.

statistical model to the corresponding cost function. The statistical models generally assume that the time-frequency points are independent. As noted in [122], [124], the Euclidean distance can be associated with the assumption that the elements of $\mathbf{Y}$ are normally distributed with mean $(\hat{\mathbf{Y}})_{k,\ell} = \sum_i (\mathbf{B})_{k,i}(\mathbf{W})_{i,\ell}$ and a constant variance. Here, $(\cdot)_{i,j}$ denotes the element at the ith row and the jth column of a matrix. In [125], it was shown that NMF-based on minimizing the Kullback-Leibler divergence is similar to modeling the elements $(\mathbf{Y})_{k,\ell}$ using a Poisson distribution. Under the Poisson model, an EM algorithm was derived in [125] which resulted in the same update rules as in [121]. However, as the Poisson distribution describes a discrete random variable, this interpretation requires that the input data is scaled to integer values, which may have theoretical implications as outlined in [126]. In [127], [128], Kullback-Leibler divergence based NMF was found to be also related to probabilistic latent semantic indexing and analysis [129], [130] — a technique for document indexing. Using the Itakura-Saito divergence for NMF results in the same cost function as if the complex DFT coefficients $Y_{k,\ell}$ are modelled by a zero-mean complex Gaussian distribution with variance given by $(\hat{\mathbf{Y}})_{k,\ell} = \sum_i (\mathbf{B})_{k,i}(\mathbf{W})_{i,\ell}$ [122]. Similarly, considering the noisy periodogram and modeling it using a gamma distribution where the mean equals $(\hat{\mathbf{Y}})_{k,\ell}$, as above, leads to the same cost function and log-likelihood function, respectively [122]. Both cases correspond to an Itakura-Saito divergence based NMF when the elements of $\mathbf{Y}$ are given by the periodogram.

NMF has been employed for source separation [117], [120], [122], [131], [132] and single-channel speech enhancement [21], [98], [116], [118], [133]–[136]. For the enhancement, the basis matrix is split into a speech part $\mathbf{B}^{(s)}$ and a noise part $\mathbf{B}^{(n)}$, where $\mathbf{B} = [\mathbf{B}^{(s)}, \mathbf{B}^{(n)}]$ as shown in Fig. 1.4. Similarly, also the activation matrix is split into a speech activation part $\mathbf{W}^{(s)}$ and a noise activation part $\mathbf{W}^{(n)}$. Commonly, the basis matrix parts, i.e., $\mathbf{B}^{(s)}$ and $\mathbf{B}^{(n)}$, are learned from training data prior to processing. During processing, the factorization of an unknown vector $\mathbf{y}_\ell$ is obtained by applying the NMF algorithm where only the activation matrix $\mathbf{W}$ is updated and $\mathbf{B}$ remains fixed. Splitting the basis matrix into a speech and a noise dependent part is used to approximate a noisy observation as $\mathbf{y}_\ell \approx \mathbf{B}\mathbf{w}_\ell = \mathbf{B}^{(s)}\mathbf{w}_\ell^{(s)} + \mathbf{B}^{(n)}\mathbf{w}_\ell^{(n)}$. Here, the vectors $\mathbf{w}_\ell^{(s)}$ and $\mathbf{w}_\ell^{(n)}$ denote the activations of the basis functions for a single segment, i.e., they are columns of the matrices $\mathbf{W}^{(s)}$ and $\mathbf{W}^{(n)}$, respectively. Further, $\mathbf{B}^{(s)}\mathbf{w}_\ell^{(s)}$ and $\mathbf{B}^{(n)}\mathbf{w}_\ell^{(n)}$ are interpreted as speech and noise spectrum, respectively. Given the approximation of $\mathbf{y}_\ell$, it can be concluded that most NMF approaches make the assumptions that either the magnitude spectra or the periodograms of speech and noise are additive. Generally, this assumption is incorrect as the phase relation between speech and noise is neglected. This simplification is, however, often justified by experiments which indicate that satisfying results with respect to sound quality can be obtained [117], [137]. Based on the separated speech spectrum and the corresponding noise spectrum, a gain function is computed to estimate the clean speech spectrum similar to the Wiener filter. Depending on the used non-negative transform, i.e., if magnitudes or periodograms have been employed, the speech estimate needs to be converted to a clean speech magnitude before it is combined with the noisy phase. The clean speech signal is reconstructed by using an overlap-add scheme.

An advantage of NMF-based algorithms over HMM or GMM-based approaches is that no explicit gain modeling is required. Instead, changes in the overall level of speech and noise are implicitly captured by the activations. However, the performance of NMF-based source separation or speech enhancement depends highly on the difference in the subspaces spanned by speech and noise basis matrices. In cases where the speech basis vectors are able to explain the background and, conversely, the noise basis vectors are capable to explain speech, the enhancement quality may suffer [138]. Hence, further regularizations terms are often imposed on the NMF cost function to increase the separability. Enforcing sparsity or employing temporal continuity constraints on the rows of $\mathbf{B}$ or columns of $\mathbf{W}$ are commonly used in speech processing applications [136], [139]–[141]. Additionally, the generalization towards unseen noise types is an issue that is shared with other ML-based enhancement approaches. Hence, NMF algorithms that are capable to estimate the background noise in an unsupervised fashion are also a topic of research, e.g., [133], [134], [142], [143]. In the following paragraphs, the current state of the research, especially with respect to these challenges, is presented.

Several NMF enhancement algorithms have been described which are able to learn the noise basis matrix blindly from a noisy observation under the constraint that the complete noisy speech utterance is given. In [133], a VAD is used to identify segments in the noisy spectrogram that contain only noise. The non-speech segments of the signal are used to train a noise basis matrix, which is used for the enhancement afterwards. For this, an algorithm similar to the general approach that has been sketched above is used. In [143], a combination of a non-ML-based noise PSD estimator, e.g., [29], [70], and an NMF-based noise estimation has been proposed. Here, the noise PSD estimate is considered a fixed component in the NMF and an NMF-based noise estimate is allowed to be added to it. Previous to processing, clean speech basis vectors are trained and during enhancement only the activation matrix and the basis vectors of the background noise matrix are updated. As a noise basis matrix with several basis vectors is learned to support the non-ML estimate, the updates are performed on a complete utterance. Also in [142], [144], a clean speech model is trained off-line on training data. Here, however, a non-negative HMM is used which uses statistical models similar to probabilistic latent component analysis for the emission PDFs. Similar to HMM-based approaches, a factorial HMM is constructed by expanding the non-negative speech HMM by a noise component. Using EM and the observations of a complete utterance, the noise specific parameters of the model are updated which allow the computation of a gain function. A completely unsupervised NMF enhancement scheme, i.e., neither speech basis vectors nor noise basis vectors are trained off-line, has been proposed in [134].

Methods that are able to estimate the noise basis matrix on segment-by-segment basis have also been considered. Two of them have been proposed in [21]. In the first method, a statistical NMF approach is combined with an environment classifier. For the environment classification, an HMM is used. The states of the HMM are related to the noise environment that should be detected. Similar to the HMM and GMM-based enhancement schemes discussed in Section 1.3.1, the final clean speech estimate is obtained by a weighted average over the state specific clean speech estimates where the weights are again given by the state posterior probabilities. The second method proposed in [21], is related to the minimum statistics approach in [29]. Also here, a buffer is used to track a set of segments that exhibit low energy. This set is used to update a collection of spectra that contain only noise which are used to update the noise basis vectors in regular intervals. A related approach has been proposed in [145], where the on-line estimation is also performed using a sliding window. In contrast to [21], the pre-selection of the noise only segments is omitted. Instead, all matrices except the speech basis matrix are updated when new data are available. To prevent speech information from leaking into the noise basis vectors, the NMF iterations are interrupted prematurely, i.e., only a fixed and low number of iterations are performed.

Several studies show that the inclusion of the temporal dynamics of the considered sources make it possible to reduce the overlap of the subspaces spanned by the speech and noise basis vectors. Hence, a part of NMF research is dedicated to the inclusion of temporal dependencies in NMF-based enhancement approaches [146]. Often, the dynamics are incorporated by including additional regularization terms into the NMF cost function. One of the first approaches to include this information has been proposed in [147], where the mean and the covariance of the NMF activations and similar statistics about the temporal evolution of these coefficients are learned prior to the enhancing the signal. Based on these statistics, a heuristic regularization term is included in the NMF cost function to take the temporal dependencies into account. Similarly, other approaches impose additional constraints on the activations by restricting them to be close to the activations of the previous time step [117], [122], [148]. The approach has been advanced by more flexible state-space models, e.g., [140], [141], [149]. Here, the temporal progression of the activation functions is modeled using a vector-AR process which makes it possible to include more complex temporal dependencies. Further, also combinations of HMMs and NMF have been considered in [98], [142], [144], [150]. Similar to HMM-based enhancement schemes, factorial HMMs are used to model the temporal evolution of multiple sources [142], [144], [150].

Another approach that has been considered to reduce the overlap between speech and noise subspaces are exemplar based approaches, e.g., [135], [138], [151], [152]. In contrast

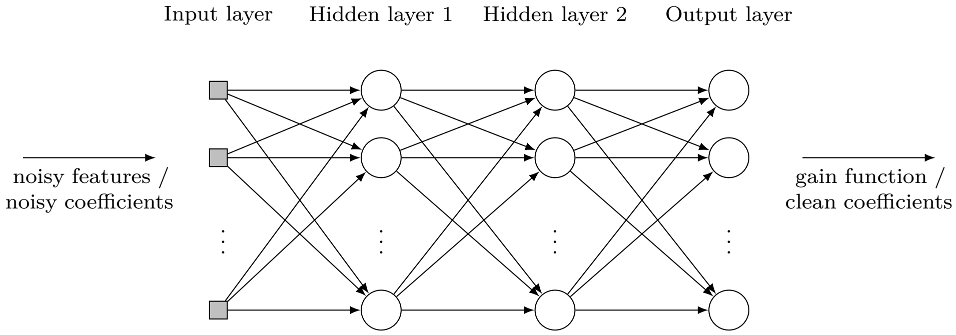

Fig. 1.5.: Block diagram of an example feed-forward neural network.

to the NMF approaches discussed until now, exemplar based approaches do not learn a low-rank approximation for the speech and noise basis matrices from training data. Instead the speech and noise basis matrices are obtained by selecting samples from the respective training data. The motivation for this is that by using a low-rank approximation in the basis vectors, the model may become too general such that other sounds not lying in the signal space may be explained too easily [138]. By forcing the basis vectors to be samples of the training data, the basis vectors are more likely to span the manifold of the speech signal which reduces the overlap with the noise basis functions. Exemplar based dictionaries are usually overcomplete, i.e., the number basis vectors exceeds the dimensionality. Hence, regularizations in form of sparsity constraints or temporal constraints as discussed above become mandatory. Other disadvantages of this approach are the increased demands in computational complexity and memory requirements. In [135], the dictionaries for speech and noise include tens of thousands of vectors which is considerably higher than in previous studies [21].

Another approach to reduce the overlap of the subspaces has been introduced in [153]–[155]. In contrast to exemplar based approaches, which may be demanding on computational and memory resources, an additional regularization term is included here which penalizes similarities between speech and noise subspaces. This is referred to as discriminative training. In [153], the cross-coherence between the basis vectors of two different sources is employed to quantify the similarity between the basis vectors. In [154], [155], the reconstruction error of a specific source extracted from a mixture is added as a regularization term that is minimized additionally during learning.

1.3.3. Neural Networks

The origin of the name “neural network“ can be found in the first attempts to describe neural information processing mathematically [88], [156], [157]. In general, neural networks expose a structure which is similar to the block diagram shown in Fig. 1.5 where a feed-forward network is depicted. The nodes in the first hidden layer of a neural network compute multiple affine linear transforms of the input values and the results are non-linearly warped. The following layers perform the same operations on the outputs of the previous layers until the output layer is reached. Interestingly, it has been shown that neural networks are universal approximators, which make it possible to learn arbitrary functions on a compact range, i.e., a limited subset, of the input space [88], [158]. Correspondingly, such networks can be trained to yield values of a pre-defined target function given the input data, e.g., the clean speech coefficients given the noisy ones. Despite the powerful universal approximation property, the training of such networks, i.e., finding suitable values for the parameters, is a challenging task. The resulting cost function is generally non-convex and the solutions are not unique [88]. The advances in [159]–[161] allowed more complex networks to be trained, e.g., networks with many hidden layers which became known as DNNs. Such models allow very complex relationships to be captured and have lead to considerable progress in various fields such as speech and image recognition.

Similar to HMM and non-ML-based approaches, which have been investigated for several centuries, also neural networks have been considered for speech enhancement nearly 30 years ago [162]. In [162], a feed-forward network with two hidden layers was used to map the noisy time-domain signal to a clean version which are processed in 60 sample long segments that do not overlap. Each hidden layer comprised 60 units, where sigmoid functions are employed as non-linear activation functions. Similarly, also the output layer consists of 60 units where linear activations functions are used. However, due to the limited computational resources, the performance of this approach had been analyzed using only few processed examples in [162]. It took several years until neural networks were reconsidered for speech enhancement [163]–[165].

Nowadays, neural networks have become a common tool to approach single-channel speech enhancement, e.g., [23], [26], [166], [167]. DNN-based single-channel speech enhancement algorithms often leverage spectral representations. For this, many approaches extract features from the spectral representation of the noisy input signal which are mapped directly to the clean speech coefficients or to a multiplicative gain function [23], [168]–[170] as shown in Fig. 1.5. As for the other speech enhancement approaches that have been considered in this overview, the time-domain signal is reconstructed using overlap-add procedures. Research topics in the field of DNN-based speech enhancement are the selection of input features, the DNN architecture, the target functions and/or cost functions.

In [168], [171], [172] various features are compared with respect to their performance for speech enhancement and also dereverberation. Many of these features are inspired by ASR and include MFCCs [173], amplitude modulation spectra (AMS) [174], Perceptive Linear Prediction (PLP) in combination with relative spectral (RASTA) processing features [175], as well as, Gabor filterbank features [176]. Even though it has been proposed in [168], [171], [172] to combine multiple features, many DNN enhancement schemes restrict themselves to a single set of features. Commonly, magnitude spectra or periodograms are used where often the logarithmic representation is computed, e.g., [23]. Similarly, Mel filterbank features and the corresponding logarithmized variants are also often encountered [170], [177].

name	formula
ideal binary mask (IBM) [169]	$I\left(\lvert S_{k,\ell}\rvert^2/\lvert N_{k,\ell}\rvert^2 > \tau\right)$
ideal ratio mask (IRM) [169]	$\left(\lvert S_{k,\ell}\rvert^2/(\lvert S_{k,\ell}\rvert^2 + \lvert N_{k,\ell}\rvert^2)\right)^{\beta}$
DFT magnitude [169]	$\lvert S_{k,\ell}\rvert$
DFT mask [169], [170]	$\lvert S_{k,\ell}\rvert/\lvert Y_{k,\ell}\rvert$
phase-sensitive filter [170]	$\lvert S_{k,\ell}\rvert/\lvert Y_{k,\ell}\rvert\cos(\Phi^s_{k,\ell} - \Phi^y_{k,\ell})$
complex ideal ratio mask (cIRM) [170], [178]	$S_{k,\ell}/Y_{k,\ell}$

Table 1.1.: List of target functions used in DNN-based speech enhancement. The symbol $S_{k,\ell}$ denotes the complex speech time-frequency points while $N_{k,\ell}$ denotes noise time-frequency points. Further, $\Phi^s_{k,\ell}$ and $\Phi^y_{k,\ell}$ are the respective phases of the complex coefficients $S_{k,\ell}$ and $N_{k,\ell}$. The index k represents frequency while ℓ represents time. Here, $I(\cdot)$ denotes the indicator function which is 1 if the condition in the argument is true and 0 otherwise.

Similarly, also various target functions of the DNN have been investigated for speech enhancement, e.g., [169], [170]. Most of the target functions are so-called mask functions. Similar to the Wiener filter gain function, the noisy spectral coefficients are multiplied by the mask obtained from a trained DNN to estimate the clean speech coefficients. During training, oracle knowledge about the speech and the noise signal is used to compute the ideal target values, while the desired mask has to be reproduced blindly by the DNN during processing. Table 1.1 gives an overview of target functions that have been proposed in the literature [169], [170], [178]. In comparisons [169], [170], the ideal binary mask (IBM) shows the largest improvements in signal-to-noise ratio (SNR) as the coefficients that only contain background noise are completely rejected. Even though studies show that binary masks can improve the intelligibility of speech in noisy conditions [179], other studies indicate that these masks do not have advantages over soft masks [170], [180], [181]. Correspondingly, [170], [181] conclude that the phase-sensitive masks such as the phase-sensitive filter are best suited if a real gain function needs to be learned. As shown in [182], it may further be beneficial for speech enhancement applications to use multiple targets during training. For this, the DNN is used in [182] to directly predict the clean speech coefficients, as well as, an ideal ratio mask (IRM). The clean speech coefficients are estimated by averaging the direct estimate and the IRM-based estimate.

The target functions considered above are real, i.e., they do not enhance the phase of the clean speech signal even though the phase-sensitive filter makes it possible to react to some phase dependent variations in the input data. Due to this limitation, target functions have been proposed that makes it possible to enhance the complex speech coefficients. In [178], the complex ideal ratio mask (cIRM) (see Table 1.1) is trained by separating the complex function into its real and its imaginary part. Consequently, the DNN's task is to predict both parts separately and, correspondingly, both parts form a separate term in the cost function. Even though improvements are reported in [178], most input features used in [178] remove the phase information. Splitting the target function into the real and imaginary parts avoids the problem of training complex-valued networks. However, learning complex-valued mappings using a DNN is doable using the derivations in [183], [184]. Similarly, also [185] derived analytic expressions of the gradients required for back-propagation and applied a complex-valued DNN to a beamforming task. From the obtained results, it is however concluded that “complex-valued neural network[s] do not perform dramatically better than real-valued” [185]. In [186], complex-valued DNNs have been used for singing-voice extraction where improvements over non-complex-valued DNNs are reported.

Additionally to the input features and the employed target function, the performance of a DNN-based enhancement scheme depends also on the employed cost function. In [155], two approaches namely the *mask approximation* and the *signal approximation* are compared for training the target functions. For mask approximation the error criterion is defined directly on the target functions, i.e., the mask which should be estimated. Correspondingly, the error between the target mask and the predicted mask is optimized, e.g., by minimizing the MSE. In [177] it is shown that defining the error function directly on the target speech signal instead of the mask function can improve the performance of DNN-based enhancement schemes. Correspondingly, the error function is defined between the clean speech signal and the masked, i.e., the enhanced, noisy signal. However, to be able to obtain the advantage from the signal approximation loss function, the network has been pre-trained in [155] using a mask based target function before learning the signal approximation. In [187]–[189], it has been proposed to include also the reconstruction of the interfering source, e.g., noise, in the cost function. Additionally, [187], [188] proposed to further add a discriminative constraint to the cost functions that reduces the similarity between two sources, e.g., speech and noise.

Another way of finding an appropriate cost function is to use generative adversarial networks (GANs) [190]. During training, the goal is to find the optimal parameters of a network which maps an observable space to a target space. For speech enhancement, this may correspond to the mapping of the noisy observation to clean speech. Further, a discriminator network is trained whose task is to distinguish between the true observations in the target space and the generated ones. For speech enhancement, this corresponds to the classification of true clean speech samples and the estimated ones. After training the discriminator, it can be used to update the parameters of the generator network. For this, the generator is optimized such that it becomes harder for the discriminator to distinguish between the generated or estimated samples and the true samples. To train the generator network in this specific way, the error is propagated through the discriminator network. This procedure can be repeated until it becomes impossible for the discriminator to decide whether an observation is generated or not. In [191], this type of network has been employed to train a feed-forward network that enhances noisy speech signals in the time-domain via end-to-end learning.

Another factor of a DNN’s performance in speech enhancement applications is the structure of the units in the layers. Some initial approaches such as [23], [169], [192], [193] employed feed-forward networks as shown in Fig. 1.5. In [194], an evaluation of various feed-forward architectures is conducted where various dimensionalities such as the non-linearities, number of hidden units and context size are compared. However, as speech is highly correlated in time and frequency, deep recurrent neural networks have been quickly adopted for speech enhancement and speech separation, e.g., [163], [195], [196]. As recurrent networks are often hard to train [197], [198], long short-term memory (LSTM) cells [199] are a natural choice [170], [196]. Further, also convolutional neural networks have been employed for speech enhancement [200], [201]. Some recent approaches use novel network designs such as WaveNet [202] which models the evolution of the time-domain signal using a conditional PDF. The PDF is conditioned on the previous speech samples in the time domain, i.e., the PDF has the form $f(s_t|s_{t-1},\ldots,s_{t-\tau})$. The initial application of WaveNet has been text-to-speech synthesis but also found its way into speech enhancement [203], [204] which is also performed in the time-domain due to the model definition.

As for the other ML-based enhancement schemes considered in the previous subsection, also for neural network based enhancement scheme the generalization of the approaches is discussed. Using a DNN-based speech enhancement scheme in acoustic conditions that have not been seen during training is often referred to as mismatching conditions. In [24], the concern is raised that for several studies the overlap of the acoustic conditions seen during training and used for testing is rather high, e.g., [188], [194], [205], [206]. Hence, the generalization capability of the DNN-based speech enhancement algorithm has been investigated in [24]. For this, the effect of changes in the training set diversity on the performance in seen and unseen acoustic conditions are analyzed. The training set diversity is changed in terms of the number of speakers, SNR range and noise types. The authors of [24] conclude that mismatches in the speaker and the SNR are less critical, while "matching the noise type is critical in acquiring good performance for DNN based SE [speech enhancement] algorithms" [24]. Even though [24] mainly considers the noise type as the most critical one with respect to generalization, [207] highlight the speaker. In [207], it is hypothesized that the speaker plays an important role for DNN-based speech enhancement. Further, LSTM networks are proposed as a remedy based on the assumption that non-recurrent DNNs do not have the modeling capabilities to generalize to unseen speakers. This statement is supported by empirical evaluations where it is shown that the LSTM states are correlated with speaker identities.

To increase the generalization to unseen noise conditions, various studies employ a large amount of training data where as many noise types as possible are covered [23], [167], [207]. However, there is no consent on which amount of training data can be considered sufficient. Hence, some authors argue, e.g., [26], that a limited data set is never sufficient because infinitely many noise types are encountered under real-world conditions. As a consequence, a noise PSD estimation algorithm similar to non-ML-based approaches is employed in [26] which, however, is supported by a DNN-based phoneme classifier. Another approach to improve the generalization is noise-aware training [208] which has been employed in [23], [209], [210]. In noise-aware training, an estimate of the noise PSD is appended to the feature vector used for enhancement to support the learning algorithm with additional information. In [23], [208], a fixed noise PSD estimate is used, which is obtained on the first segments of the input signal. In [209], [210], this estimate has been replaced by a dynamic noise estimate which is obtained from a non-ML noise PSD estimator or an IBM-based noise estimating DNN. Both studies report improvements of the noise-aware DNNs over the corresponding non-noise-aware counterparts. The approach in [211] continues the work in [209] and employs two separate DNNs for the enhancement. The first DNN is used to estimate the background noise as in [209], but is also used to jointly predict an IRM. Both quantities are used in a second DNN to predict the clean speech spectra. In [212], the influence of the DNN structure on the generalization is analyzed. The experiments indicate that predicting a gain function instead of the clean speech spectra results in more robust enhancement algorithms. Further, using a structure of multiple DNNs that follows the structure of a conventional single-channel speech enhancement scheme similar to Fig. 1.1 may further improve the robustness.

In [213], a speech enhancement approach is proposed which relies only on a speech specific model given by an deep auto-encoder. This model is trained only on clean speech before processing noisy signals. Similarly, also the background noise is modeled by an autoencoder, but its parameters are not trained off-line on training noises. Instead, the parameters are updated for each noisy input segment based on a noise estimate which is obtained by subtracting the estimated clean speech from the noisy observation. As this problem is underdetermined, various constraints and regularizations are applied, e.g., that the estimated speech and noise sum up to the noisy spectrum, that the speech spectrum needs to lie in the subspace spanned by pre-trained speech NMF basis vectors, and that speech and noise should be dissimilar. A somewhat related approach has been proposed in [214] where a denoising autoencoder is trained on various noise types while a speech autoencoder is trained separately on clean speech. In the enhancement stage, the speech autoencoder is stacked on top of the denoising autoencoder and is used as a controlling instance. More specifically, the weights of the denoising autoencoder are adapted by minimizing the error between the output of denoising autoencoder and the output of the speech autoencoder. This allows the quality of the enhanced speech signal obtained from the denoising autoencoder to be fine-tuned.

1.4. OUTLINE OF THE THESIS AND MAIN CONTRIBUTIONS

The main topic of this thesis deals with improving single-channel speech enhancement. Non-ML-based single-channel speech enhancement schemes, ML-based enhancement schemes and the combination of both approaches are considered. Non-ML approaches are generally more robust to unseen and moderately changing noise conditions, but are unable to suppress highly non-stationary noise. In such noise conditions, non-ML approaches fail to track very fast changes. Contrarily, ML-based approaches have the ability to follow such changes and, hence, are an intriguing approach to improve the quality of the enhanced signal in adverse acoustic conditions. However, as the previous sections on ML-based enhancement indicate, the generalization of ML enhancement schemes to unseen acoustic conditions is a discussed topic. In this work, various shortcomings of ML and non-ML-based enhancement schemes are considered and potential solutions are presented. Among these solutions, approaches to combine both methods are proposed which aim to exploit the advantages of ML and non-ML speech enhancement at the same time.

The main contributions are three-fold. First, improvements to non-ML-based noise PSD estimators are presented. We show that non-ML-based noise PSD estimation algorithms that can be described as first-order recursive smoothing filters with an adaptively changing smoothing factor, e.g., [70], [71], [81], are biased. Methods for quantifying the bias are presented and approaches are derived that compensate for the bias. Second, we show that super-Gaussian models have considerable perceptual advantages for ML-based speech enhancement algorithms that model only the spectral envelope of speech but not its fine structure. If spectral envelope models are used, it is not easily possible to reduce noise between the harmonics of the speech fundamental frequency and its harmonics if the speech coefficients are modeled by a Gaussian PDF. However, we show that super-Gaussian estimators are able to suppress the residual noise if only spectral envelope models are employed. Third, we propose non-ML-based features to improve the generalization of a DNN-based speech enhancement scheme. For this, we propose to use normalized, i.e., SNR-based features, which are obtained from non-ML-based estimates of the speech PSD and the noise PSD. In contrast to the existing noise aware training [23], [208]–[210], the proposed features result in better performance in unseen noise conditions.

The following paragraphs give a section by section overview of the thesis, which summarizes the contributions of the thesis in more detail. Here, also the related publications that have been worked on during the thesis are referenced.

In Chapter 2, general aspects of single-channel speech enhancement algorithms are considered. The STFT and commonly used statistical models for the speech and noise spectral coefficients are explained. As various approaches in this thesis operate in the log-spectral domain, also log-spectral models are introduced. Additionally, an overview of single-channel speech and noise PSD estimation methods is given. In the last part of this chapter, the instrumental measures used to assess the performance of the proposed algorithms are presented.

In Chapter 3, single-channel noise PSD estimators are considered which can be described as first-order recursive smoothing filter with an adaptively changing smoothing factor, e.g., [70], [71] and [81, Section 14.1.3]. To avoid speech energy from leaking into the noise PSD estimate, the value of the smoothing factor is changed to larger values, i.e., the tracking speed is reduced, if speech is present. In this chapter, we show that such approaches are generally biased estimators of the noise PSD. A method for bias compensation is introduced where the bias is corrected by scaling the input or the output of the adaptive smoothing filter. We present two iterative algorithms that allow an approximate estimation of the required scaling factor. The correction factors are first derived under the assumption that the input signal is stationary and contains only noise. To make the correction method aware of the speech signal, further extensions of the correction method are presented. The results indicate that the proposed bias estimation algorithms are able to estimate the bias with sufficient precision and make it possible to reduce the error in the estimation of the noise PSD. For cases, where the noise PSD estimation is considerably biased, the proposed correction method allows the perceived quality of the enhanced signal to be improved. This chapter is mainly based on [215], [216].

In Chapter 4, another correction method is proposed to compensate for the bias of adaptive recursive smoothing filters. Here, a correction factor is applied to one of the quantities in the adaptive smoothing filters, but at a different point in the smoothing function. As a result, this correction method can no longer be interpreted as a scaling of the filter input or the output. The value of the correction factor required to obtain an unbiased noise PSD estimate is generally different from the ones used in Chapter 3. To determine the correction factor, one of the methods used in Chapter 3 is modified to meet the requirements of the alternative correction method considered in this chapter. It is shown that the required modifications turn the iterative estimation method of the correction factor into a non-iterative procedure. Similar to Chapter 3, also here, the correction factor is first derived under the assumption of a stationary signal containing only noise and is extended afterwards. The results of this method are compared to the bias compensation from Chapter 3 where it is shown that the alternative method yields similar results. The publication related to this chapter is [217].

In Chapter 5, an issue specific to ML-based speech enhancement algorithms which use speech models that only describe the speech spectral envelope is considered. Due to the missing fine structure, clean speech estimators that are based on Gaussian priors, e.g., the Wiener filter, are unable to reduce the noise between the spectral harmonics. Existing methods approach this issue by applying post-processing to the estimated clean speech spectrum. For this, the estimated speech spectrum is multiplied by an estimate of the SPP, [26], [27] or a harmonic model is employed [104]. In this chapter, it is shown that by using super-Gaussian PDFs for the speech priors the residual noise can be reduced without additional post-processing steps. An analysis of the estimators that result from the super-Gaussian modeling reveals that they make it possible to reduce the noise in cases where the speech PSD is overestimated. For ML-based enhancement algorithms that model speech only by its envelope, super-Gaussian clean speech estimators improve the quality of the enhanced signals considerably compared to Gaussian estimators. The chapter is based on [218].

In Chapter 6, we show that the advantages of super-Gaussian enhancement filters do not only apply if the noisy speech spectrum is modeled as an additive mixture of speech and noise in the spectral domain as in Chapter 5. Instead, the advantages of super-Gaussian modelling can also be observed if the MixMax or log-max approximation [26], [101], [112] is employed. Using a Gaussian distribution [26], [100], [101], [112] for modeling the log-spectral speech and noise coefficients, an MSE optimal clean speech estimator of the log-spectral speech coefficients can be derived [26], [101], [112]. In this chapter, the relation between the parameters of a spectral super-Gaussian model and the log-spectral mean and variance is exploited (see Section 2.1.3 [83]). The use of log-spectral means and variances that correspond to a spectral super-Gaussian model also allows clean speech estimators derived under the MixMax model to suppress the residual noise if the speech PSD is overestimated. Comparing the results with estimators derived in the spectral domain, the estimators based on the MixMax model yields similar results in terms of quality but cause less musical tone artifacts. This chapter is based on [219].

In Chapter 7, we propose another solution for reducing the residual background noise if speech is modeled by spectral envelopes in an ML-based enhancement scheme. In contrast to using super-Gaussian models as in the previous chapters, we combine a non-ML-based speech enhancement scheme with the ML-based enhancement scheme. The combination is embedded in a statistical framework where each enhancement method in the combination has its own likelihood model for the noisy observation. Using Bayesian statistics, a posterior probability is computed which is used to identify the enhancement scheme which describes the noisy observation best. The final clean speech estimate is obtained by mixing the estimated clean speech spectra of the combined enhancement schemes based on the previously computed posterior probability. The results show that the proposed enhancement scheme outperforms a purely non-ML-based speech enhancement as shown by instrumental measures. The related publication is given by [220].

In Chapter 8, the generalization of a DNN-based enhancement scheme towards unseen noise conditions is considered. It is proposed to improve the generalization by using input features that are based on classic non-ML methods. A similar approach, referred to as noise aware training, has been previously used in [23], [208]–[210]. Here, an estimate of the noise PSD is appended to the features extracted from the noisy observation. In the proposed method, the estimated noise PSD is not appended but used as normalization term. More specifically, we propose to use the *a priori* SNR and the *a posteriori* SNR as input features which are commonly used in non-ML-based clean speech estimators, e.g., [9], [12], [28], [31]. We show that the normalized SNR-based features are advantageous over non-normalized features such as noise aware training. In contrast to the non-normalized features, the proposed features are scale-invariant which has the effect that also the performance of the DNN-based enhancement scheme becomes independent of the overall level of the input signal. Further, the proposed features yield a considerably higher performance in unseen noise conditions. The related publication is [221].

Chapter 9 summarizes the main contributions of this thesis and presents suggestions for further research.

CHAPTER 2

STATISTICAL FRAMEWORK FOR SPEECH ENHANCEMENT AND INSTRUMENTAL MEASURES

In this chapter, the general mathematical notations and the statistical signal models are introduced. These form the basis of the framework which is used for derivations of the enhancement algorithms proposed in this thesis. Furthermore, the instrumental measures are introduced which are used to assess the performance of the proposed algorithms in terms of signal quality and intelligibility.

In Section 2.1, spectral enhancement methods are considered. In Section 2.1.1, the general procedure and mathematical notations used in this context are introduced. Section 2.1.2 deals with statistical signal models that describe the interaction between speech and noise based on an additive interaction model in the frequency domain. Section 2.1.3 presents the relationship between spectral and log-spectral models which is used to combine ML-based approaches that employ log-spectral representations and conventional non-ML approaches. Finally, the non-ML-based speech and noise PSD estimation algorithms that are used throughout this thesis are described mathematically in Section 2.1.4. Section 2.2 presents the instrumental measures that are used to assess the quality of the enhanced speech signals. In addition, measures for quantifying the estimation error of noise PSD estimation algorithms, which are required for the evaluation in Chapter 3 and Chapter 4, are introduced.

2.1. STATISTICAL MODELS FOR SPECTRAL ENHANCEMENT

2.1.1. Spectral Domain Speech Enhancement

In this section, a general speech enhancement procedure is presented, which is shared among many single-channel speech enhancement algorithms. If speech is recorded in a noisy acoustic environment, e.g., as shown in Fig. 2.1, the employed microphone does not only capture the clean speech time-domain signal s_t but also the background noise signal n_t. Here, t is the sample index. Under some mild constraints, the interaction of sound waves is physically well described by their superposition [43]. Hence, an adequate expression of the noisy microphone signal y_t is given by

$$y_t = s_t + n_t. \tag{2.1}$$

As speech is known to be non-stationary, i.e., the speech sounds change considerably over time, the noisy input signal is split into short overlapping time segments. Each segment of the input signal is transformed to the frequency domain using the DFT which results in

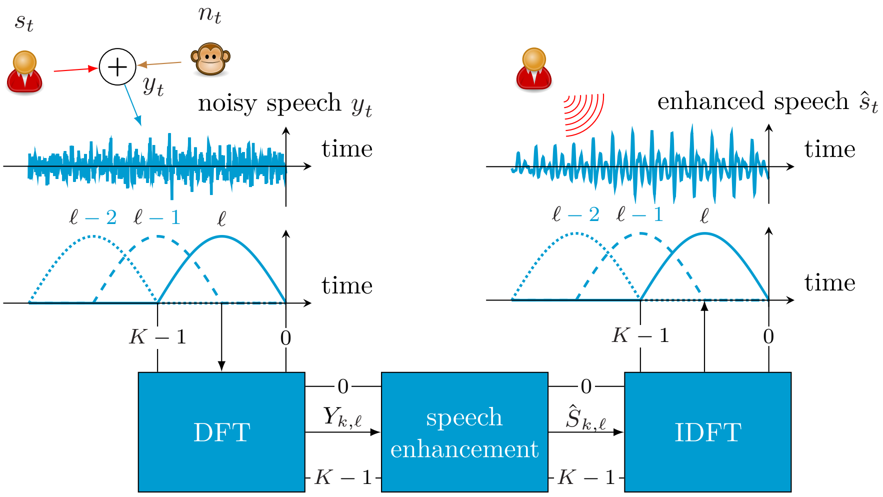

Fig. 2.1.: Block scheme of a non-ML-based STFT filter based single-channel based enhancement scheme.

the STFT. Mathematically, this procedure is given by

$$Y_{k,\ell} = \sum_{t=0}^{K-1} \omega_{\mathrm{a}}(t) y_{\ell R+t} e^{-j2\pi kt/K}. \tag{2.2}$$

Here, k is the frequency index, ℓ is the segment index and $j = \sqrt{-1}$. Each segment has K samples which is equivalent to the number of frequency bins. Zero-padding, where the number of DFT coefficients is set to a higher value than the segment length, is not considered in this thesis. Further, R is the segment shift and $\omega_{\mathrm{a}}(\cdot)$ denotes the spectral analysis window. The multiplication with the window function $\omega_{\mathrm{a}}(\cdot)$ in the time-domain corresponds to a convolution in the frequency domain, i.e., the frequency-domain representation of the window is convolved with the spectrum of the noisy input signal. A rectangular window has often undesirable properties, e.g., low sidelobe attenuation, and therefore smooth window functions such as the Hann window, the Hamming window or the square-root Hann window are often used. Due to the linearity of the DFT, the additive mixing model in the time domain also applies to the STFT, i.e.,

$$Y_{k,\ell} = S_{k,\ell} + N_{k,\ell}. \tag{2.3}$$

Here, $S_{k,\ell}$ and $N_{k,\ell}$ denote the STFT of the clean speech signal s_t and the background noise n_t, respectively.

The aim of speech enhancement algorithms is to find an estimate of the clean speech coefficients $\hat{S}_{k,\ell}$ from the noisy observations $Y_{k,\ell}$. Here, $\hat{\cdot}$ indicates that $\hat{S}_{k,\ell}$ is an estimated

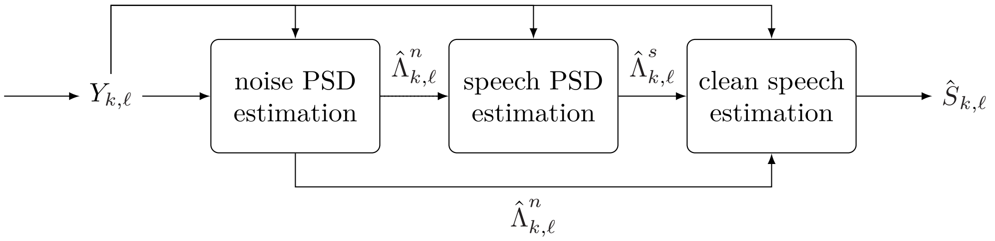

Fig. 2.2.: Block scheme showing the steps / components of a speech enhancement scheme operating in the spectral domain. The block scheme shows the blocks of the enhancement step shown in Fig. 2.1.

quantity. Most spectral clean speech estimators can be written in the form

$$\hat{S}_{k,\ell} = G_{k,\ell} Y_{k,\ell}, \tag{2.4}$$

where $G_{k,\ell}$ is the so-called gain function. Often, the gain function is real-valued as many statistical models of the clean speech coefficients do not provide prior knowledge about the phase. Further, many algorithms estimate only the magnitude $A_{k,\ell} = |S_{k,\ell}|$ of the clean speech signal. In these cases, the enhanced speech magnitude $\hat{A}_{k,\ell}$ is combined with the phase of the noisy observation $\Phi^y_{k,\ell}$ as

$$\hat{S}_{k,\ell} = \hat{A}_{k,\ell} \exp(j\Phi^y_{k,\ell}). \tag{2.5}$$

Often, the gain function can be expressed in terms of the *a priori* SNR $\xi_{k,\ell}$ and the *a posteriori* SNR $\gamma_{k,\ell}$, e.g., the Wiener filter as in Section 2.1.2 or the gain function used in Section 5. The quantities have been defined in [12] as

$$\xi_{k,\ell} = \frac{\Lambda^s_{k,\ell}}{\Lambda^n_{k,\ell}}, \tag{2.6}$$

$$\gamma_{k,\ell} = \frac{|Y_{k,\ell}|^2}{\Lambda^n_{k,\ell}}, \tag{2.7}$$

respectively. The complex spectral coefficients are assumed to be zero-mean such that $\Lambda^s_{k,\ell} = \mathbb{E}\{|S_{k,\ell}|^2\}$ and $\Lambda^n_{k,\ell} = \mathbb{E}\{|N_{k,\ell}|^2\}$ denote the speech and the noise variance, respectively. Here, $\mathbb{E}\{\cdot\}$ is the expectation operator. In non-ML-based enhancement schemes, the speech PSD and the noise PSD are estimated blindly from the noisy observation $Y_{k,\ell}$. The steps taken to enhance the noisy observation $Y_{k,\ell}$ for a new time segment ℓ are depicted in Fig. 2.2. They are given by:

1. Estimating the noise PSD $\Lambda^n_{k,\ell}$ from the noisy observation $Y_{k,\ell}$.

2. Estimating the speech PSD $\Lambda^s_{k,\ell}$ using the estimated noise PSD $\hat{\Lambda}^n_{k,\ell}$ and the noisy observation $Y_{k,\ell}$.

3. Estimating the clean speech coefficients $\hat{S}_{k,\ell}$ using both PSD estimates, i.e., $\hat{\Lambda}^{n}_{k,\ell}$ and $\hat{\Lambda}^{s}_{k,\ell}$ and the noisy observation $Y_{k,\ell}$.

More details on the estimation of clean speech spectral coefficients and the resulting gain functions $G_{k,\ell}$ is given in Section 2.1.2 and Section 2.1.3. The estimation of the speech and the noise PSD are dealt with in Section 2.1.4.

After estimating the clean speech spectrum $\hat{S}_{k,\ell}$, the time-domain signal needs to be resynthesized. For this, the enhanced spectra are transformed back to the time-domain using the inverse STFT as

$$\hat{s}_t = \frac{1}{K}\sum_{\ell}\sum_{k=0}^{K-1}\omega_{\mathrm{s}}(t-\ell R)\hat{S}_{k,\ell}e^{j2\pi(t-\ell R)k/K}. \tag{2.8}$$

Here, $\omega_{\mathrm{s}}(\cdot)$ is the synthesis window which is chosen such that a perfect reconstruction of the input signal is possible. Block artifacts caused by circular convolution [44] can be reduced with specific choices for the analysis and synthesis window pair. The circular convolution of the filter function and the noisy observation in the time domain is caused by the multiplication of the noisy observation $Y_{k,\ell}$ with the gain function $G_{k,\ell}$ in the frequency domain. These artifacts are usually most dominant at the edges of the time-domain segment and, hence, can be effectively suppressed by applying a tapered synthesis window. If the segments overlap by 50 %, this reduction can be achieved by using a square-root Hann window as spectral analysis and synthesis window.

The segment length K is chosen such that it corresponds to a time period between 10 ms and 40 ms, for which speech is assumed quasi-stationary [44]. This means that the speech sounds change only slightly in these time periods. The segment shift is chosen such that the segments overlap by 50 % or 75 %. Using a sufficiently long segment length, the spectral fine structure of speech, i.e., the speech fundamental frequency and its harmonics, can be resolved in the STFT domain. As a consequence, it is possible to reduce the noise between spectral harmonics. If the specific application requires low latency processing, the segment length may be reduced to lower values, e.g., 8 ms and below. This, however, comes at the cost of a lower spectral resolution. Another option is to use asymmetric window functions [222] where spectral windows that are short in time are used for the synthesis to reduce the algorithmic latency. Temporally long spectral analysis windows are retained to provide a high-resolution frequency representation.

A potential disadvantage of STFT-based enhancement algorithms is that the enhancement process may introduce artifacts in the processed signal such as musical tones. To reduce such artifacts, a lower limit is commonly enforced [223]. The corresponding modified gain function is given by

$$\tilde{G}_{k,\ell} = \max(G_{k,\ell}, G_{\min}). \tag{2.9}$$

Here, $G_{\min}$ is the lower limit. Higher values result in less artifacts but reduce the noise suppression capability. The opposite is true, if $G_{\min}$ is reduced.

2.1.2. Statistical Models for Spectral Speech Enhancement

In this section, we focus on Bayesian approaches for speech enhancement in the spectral domain. The considered approaches interpret the spectral coefficients of speech $S_{k,\ell}$ and noise $N_{k,\ell}$ as random variables. For this, most methods assume that the PDF of the respective spectral coefficients is known except for the parameters. Further, it is assumed that the speech and the noise spectral coefficients are zero mean and uncorrelated, i.e., $\mathbb{E}\{S_{k,\ell}N^*_{k,\ell}\} = 0$, where $\cdot^*$ indicates a conjugate complex quantity.

One type of clean speech estimators considered in this thesis minimizes the MSE between the estimated complex speech coefficients $\hat{S}_{k,\ell}$ and the true complex speech coefficients $S_{k,\ell}$. The error function is commonly expressed in terms of the expected value as

$$\hat{S}_{k,\ell} = \arg\min_{\tilde{S}_{k,\ell}} \mathbb{E}\left\{\left|S_{k,\ell} - \tilde{S}_{k,\ell}(Y_{k,\ell})\right|^2\right\}. \tag{2.10}$$

The function $\tilde{S}_{k,\ell}(\cdot)$ of the noisy input $Y_{k,\ell}$ that minimizes the MSE can be any function and is potentially non-linear. The MSE optimal estimator of the clean speech coefficients $S_{k,\ell}$ can also be derived by solving [224, Section 5.2]

$$\hat{S}_{k,\ell} = \mathbb{E}\{S_{k,\ell}|Y_{k,\ell}\}, \tag{2.11}$$

i.e., by determining the mean of the posterior PDF $f(S_{k,\ell}|Y_{k,\ell})$. As a consequence, the result of (2.10) and (2.11), respectively, depends on the statistical models $f(S_{k,\ell})$ and $f(N_{k,\ell})$. A generally different type of estimator is obtained if the MSE optimal estimator for the clean speech amplitude $A_{k,\ell} = |S_{k,\ell}|$ is derived. In contrast to the estimator of the complex speech coefficients $S_{k,\ell}$ as in (2.10), often a compression function $c(\cdot)$ is incorporated as

$$\hat{A}_{k,\ell} = \arg\min_{\tilde{A}_{k,\ell}} c^{-1}\left(\mathbb{E}\left\{\left|c(A_{k,\ell}) - c(\tilde{A}_{k,\ell}(Y_{k,\ell}))\right|^2\right\}\right). \tag{2.12}$$

The inverse $c^{-1}(\cdot)$ is used to undo the compression for the estimate, i.e., to turn the estimate $c(A_{k,\ell})$ into an estimate of $A_{k,\ell}$. Similarly, the estimator can also be expressed in terms of the posterior distribution $f(A_{k,\ell}|Y_{k,\ell})$

$$\hat{A}_{k,\ell} = c^{-1}(\mathbb{E}\{c(A_{k,\ell})|Y_{k,\ell}\}). \tag{2.13}$$

In addition to the PDFs used to model the speech and noise coefficients, speech amplitude estimators also depend on the choice of the compression function $c(\cdot)$. Further variations of clean speech estimators are obtained if the mixing model in (2.3) is changed. This is the case in Chapter 6 and Chapter 7 where approximations of (2.3) are considered to simplify the derivation of MSE optimal estimators in the log-spectral domain.

The expected values given above depend on the PDFs of the complex speech and noise coefficients. Correspondingly, these models play an important role for the clean speech estimation in such a statistical framework. For the noise spectral coefficients $N_{k,\ell}$, a common choice is the complex Gaussian distribution, i.e.,

$$f(N_{k,\ell}) = \mathcal{N}_{\mathbb{C}}(N_{k,\ell}|0, \Lambda^n_{k,\ell}) = \frac{1}{\pi\Lambda^n_{k,\ell}} \exp\left(-\frac{|N_{k,\ell}|^2}{\Lambda^n_{k,\ell}}\right). \tag{2.14}$$

In the notation $\mathcal{N}_{\mathbb{C}}(\cdot|\cdot,\cdot)$ the two latter parameters are the mean and the variance, respectively. Here, only the distribution of a single frequency band is given as it is commonly assumed that the time-frequency points are independent. Further, it is implicitly assumed that the complex coefficients are zero-mean and consequently the only parameter of the distribution is the noise variance $\Lambda^n_{k,\ell}$. This quantity $\Lambda^n_{k,\ell}$ is also often referred to as noise PSD. The Gaussian assumption is often justified by the Fourier sum and the central limit theorem. In practice, if the time-domain samples are sufficiently uncorrelated, the distribution of the spectral coefficients approximately approaches a Gaussian distribution [47], [50]. Even though speech is highly correlated in the time-domain, effectively making the Gaussian assumption inappropriate, the complex Gaussian PDF has also been used to model speech [12], [28], [49]. Using the Gaussian assumption for both the speech coefficients $S_{k,\ell}$ and the noise coefficients $N_{k,\ell}$, the MSE optimal estimator of the complex speech coefficients $S_{k,\ell}$ results in the well-known Wiener filter [44, Section 11.4.3]

$$\hat{S}_{k,\ell} = \underbrace{\frac{\xi_{k,\ell}}{\xi_{k,\ell}+1}}_{G_{k,\ell}} Y_{k,\ell}. \tag{2.15}$$

Following the definition in (2.4), the gain function $G_{k,\ell}$ corresponds to the fraction $Y_{k,\ell}$ is multiplied with. Similar to the noise PDF, also the Gaussian speech PDF depends only on the speech variance or PSD $\Lambda^s_{k,\ell}$. Further variants of Gaussian clean speech estimators have been derived by considering the speech amplitude $A_{k,\ell}$ instead of the complex coefficients $S_{k,\ell}$. With the Gaussian assumption for speech $S_{k,\ell}$ and noise $N_{k,\ell}$, the MSE optimal estimator of the speech amplitude $A_{k,\ell}$ is the short-term spectral amplitude estimator (STSA) [12]. Further, using the compression $c(A_{k,\ell}) = \log(A_{k,\ell})$ results in the log-spectral amplitude estimator (LSA) [28].

As stated above, the Gaussian PDF is not an appropriate model to describe the speech spectral coefficients due to the strong correlations in the time domain. Studies on the distribution of spectral speech coefficients conclude that the distribution is better represented by super-Gaussian models, e.g., [30], [46], [50]. Such distributions can be formulated using representations in polar coordinates, i.e., in terms of the speech magnitude $A_{k,\ell}$ and the speech phase $\Phi^s_{k,\ell}$. In [9], [30]–[32] the speech amplitude $A_{k,\ell}$ and the speech phase $\Phi^s_{k,\ell}$ are assumed to be independent, i.e., $f(A_{k,\ell}, \Phi^s_{k,\ell}) = f(A_{k,\ell})f(\Phi^s_{k,\ell})$. Further, a uniform distribution between $-\pi$ and π is employed for $\Phi^s_{k,\ell}$. A super-Gaussian PDF is obtained if the speech magnitude $A_{k,\ell}$ is model by a χ-distribution, as [9], [30]–[32]

$$f(A_{k,\ell}) = \frac{2}{\Gamma(\nu)}\left(\frac{\nu}{\Lambda^s_{k,\ell}}\right)^{\nu} A^{2\nu-1}_{k,\ell} \exp\left(-\frac{\nu A^2_{k,\ell}}{\Lambda^s_{k,\ell}}\right), \tag{2.16}$$

where $\Gamma(\cdot)$ denotes the gamma function and ν is the shape parameter. The case $\nu = 1$ for $f(A_{k,\ell})$ combined with the uniformly distributed phase is equivalent to the assumption that the complex speech coefficients $S_{k,\ell}$ follow a circular-symmetric Gaussian distribution with variance $\Lambda^s_{k,\ell}$. If $0 < \nu < 1$, (2.16) represents the distribution of the speech amplitudes $A_{k,\ell}$ under the assumption that the complex speech coefficients $S_{k,\ell}$ are super-Gaussian distributed. Using a change of variables, $f(A_{k,\ell}, \Phi^s_{k,\ell})$ can be converted to a distribution $f(S_{k,\ell})$ depending only on the complex value $S_{k,\ell}$. The amplitude estimator derived in [31] is based on the speech model in (2.16), the noise model in (2.14) and the compression function $c(A_{k,\ell}) = |A_{k,\ell}|^\beta$, where $0 < \beta \leq 1$. Depending on the parameters chosen for ν and β, this estimator generalizes various other clean speech estimators, e.g., [12], [28], [30], [32], [49], [53].

In [30], [46], [55], other PDFs have been considered to model speech as a super-Gaussian distributed random variable. In [30], the more flexible generalized gamma distribution is used which generalizes the χ-distribution above. In [46], [55], it is assumed that the real and imaginary part are independent instead of assuming independence between magnitude and phase. The real and imaginary parts are modeled by a Laplace or gamma distributions. Those models lead to different estimators which have been analytically derived in the respective papers.

2.1.3. Log-Spectral Models and Their Relation to Spectral Models

Log-spectral models and their relation to the spectral domain are the fundamental theoretical concept of the algorithms presented in Chapter 6 and Chapter 7. Both topics are covered in this section.

In this thesis, the log-spectral coefficients are defined as the logarithm of the periodogram. For the noisy speech spectral coefficients $Y_{k,\ell}$, the corresponding log-spectrum $y^{(\log)}_{k,\ell}$ is given by

$$y^{(\log)}_{k,\ell} = \log(|Y_{k,\ell}|^2). \tag{2.17}$$

In a similar way, also the clean speech log-spectrum $s^{(\log)}_{k,\ell}$ and the noise log-spectrum $n^{(\log)}_{k,\ell}$ are defined. Assuming a super-Gaussian distribution in the spectral domain as in Section 2.1.2, the respective log-spectral coefficients follow an exp-gamma distribution [225]. However, many contributions where speech is processed in the log-spectral domain, e.g., [26], [100], [101], model the log-spectral coefficients by Gaussian distributions, i.e.,

$$f(s^{(\log)}_{k,\ell}) = \mathcal{N}(s^{(\log)}_{k,\ell} | \mu^s_{k,\ell}, \lambda^s_{k,\ell}) = \frac{1}{\sqrt{2\pi\lambda^s_{k,\ell}}} \exp\left(-\frac{(s^{(\log)}_{k,\ell} - \mu^s_{k,\ell})^2}{2\lambda^s_{k,\ell}}\right), \tag{2.18}$$

$$f(n^{(\log)}_{k,\ell}) = \mathcal{N}(n^{(\log)}_{k,\ell} | \mu^n_{k,\ell}, \lambda^n_{k,\ell}) = \frac{1}{\sqrt{2\pi\lambda^n_{k,\ell}}} \exp\left(-\frac{(n^{(\log)}_{k,\ell} - \mu^n_{k,\ell})^2}{2\lambda^n_{k,\ell}}\right). \tag{2.19}$$

Here, $\mu^s_{k,\ell} = \mathbb{E}\{s^{(\log)}_{k,\ell}\}$ and $\lambda^s_{k,\ell} = \mathbb{E}\{(s^{(\log)}_{k,\ell} - \mu^s_{k,\ell})^2\}$ are the mean and the variance of the speech log-spectral coefficients, respectively. Similarly, the parameters of the noise, which are denoted by $\mu^n_{k,\ell}$ and $\lambda^n_{k,\ell}$, are defined.

Given the super-Gaussian model for the spectral speech coefficients $S_{k,\ell}$ based on (2.16) and the uniformly distributed phase, the following relations between the spectral PSDs and the statistical parameters of the log-spectral coefficients have been established [83], [226]. As shown in [83], the mean of the log-spectral speech coefficients $\mu^s_{k,\ell}$ is related to the spectral quantities via

$$\mu^s_{k,\ell} = \log(\Lambda^s_{k,\ell}) + \psi(\nu) - \log(\nu). \tag{2.20}$$

Here, $\psi(\cdot)$ denotes the digamma function [227, (8.360.1)]. Following [83], the relation of super-Gaussian distributed spectral coefficients to the variance of the log-spectral coefficients $\lambda^s_{k,\ell}$ is given by

$$\lambda^s_{k,\ell} = \psi_1(\nu). \tag{2.21}$$

Here, $\psi_1(\cdot)$ is the trigamma function [228, Chapter 6.4]. As the super-Gaussian distribution generalizes the complex Gaussian distribution, the equations (2.20) and (2.21) can also be applied for computing the parameters of the noise coefficients, for which often a Gaussian assumption is made in the spectral domain. For this, the spectral noise PSD $\Lambda^n_{k,\ell}$ needs to be employed in (2.20) and the shape parameter has to be set to $\nu = 1$ in (2.20) and (2.21). This relationship gives an interpretation of the means and variances of speech and noise models that are directly trained in the log-spectral domain, e.g., [100]–[102]. Considering (2.21), the log-spectral variance depends only the shape ν. Hence, if the log-spectral variance, e.g., $\lambda^s_{k,\ell}$, has been trained as a parameter of an ML-based model, it can be associated with a specific shape of the spectral coefficients. Correspondingly, the log-spectral mean in (2.20) is mainly related to the spectral PSD.

2.1.4. Non-ML Estimation of the Spectral Speech and Noise PSD

To use the estimators presented in Section 2.1.2 and Section 2.1.3, the speech PSD $\Lambda^s_{k,\ell}$ and the noise PSD $\Lambda^n_{k,\ell}$ need to be estimated from the noisy observation. An overview over noise and speech PSD estimators has been given in Section 1.2. In the following paragraphs, the non-ML-based estimation methods that are commonly used in this thesis are described.

First, the noise PSD estimator proposed in [70], [71] is considered. This algorithm exploits the uncertainty of speech presence. Given the hypothesis $\mathcal{H}_1$, i.e., speech is present, the observed signal is modeled as $Y_{k,\ell} = S_{k,\ell} + N_{k,\ell}$. Under $\mathcal{H}_0$, i.e., speech is absent, the observed signal comprises only noise as $Y_{k,\ell} = N_{k,\ell}$. As for the Wiener filter, the speech coefficients $S_{k,\ell}$ and the noise coefficients $N_{k,\ell}$ are assumed to follow a complex circular-symmetric Gaussian distribution. Accordingly, the likelihoods under the hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$, i.e., $f(Y_{k,\ell}|\mathcal{H}_0)$ and $f(Y_{k,\ell}|\mathcal{H}_1)$, are also modeled using Gaussian distributions.

The posterior probability, i.e., the SPP $P(\mathcal{H}_1|Y_{k,\ell})$, can be derived using Bayes' theorem as [70], [71]

$$P(\mathcal{H}_1|Y_{k,\ell}) = \left(1 + (1+\xi_{\mathcal{H}_1})\exp\left(-\frac{|Y_{k,\ell}|^2}{\hat{\Lambda}^n_{k,\ell-1}}\frac{\xi_{\mathcal{H}_1}}{\xi_{\mathcal{H}_1}+1}\right)\right)^{-1}. \tag{2.22}$$

Here, it is assumed that the prior probability is $P(\mathcal{H}_1) = 0.5$, i.e., $P(\mathcal{H}_1) = 1 - P(\mathcal{H}_1) = P(\mathcal{H}_0)$. A fixed SNR $\xi_{\mathcal{H}_1}$ is used which is interpreted as the local SNR that is expected if the hypothesis $\mathcal{H}_1$ holds [70], [71]. The likelihood models $f(Y_{k,\ell}|\mathcal{H}_0)$ and $f(Y_{k,\ell}|\mathcal{H}_1)$ have been used to formulate a speech detection problem in [71]. By minimizing the total risk of error, the optimal value $\xi_{\mathcal{H}_1} = -15$ dB has been found [71].

In [70], [71], the posterior probability $P(\mathcal{H}_1|Y_{k,\ell})$ is used to estimate the noise periodogram as

$$|\hat{N}_{k,\ell}|^2 = P(\mathcal{H}_0|Y_{k,\ell})|Y_{k,\ell}|^2 + P(\mathcal{H}_1|Y_{k,\ell})\hat{\Lambda}^n_{k,\ell-1}. \tag{2.23}$$

Here, $P(\mathcal{H}_0|Y_{k,\ell}) = 1 - P(\mathcal{H}_1|Y_{k,\ell})$ is the speech absence probability. The estimated noise periodogram is smoothed over time to obtain the noise PSD estimate $\hat{N}_{k,\ell}$ as

$$\hat{\Lambda}^n_{k,\ell} = (1 - \alpha^{(\text{fix})}_{\text{SPP}})|Y_{k,\ell}|^2 + \alpha^{(\text{fix})}_{\text{SPP}}\hat{\Lambda}^n_{k,\ell-1}. \tag{2.24}$$

Here, $\alpha^{(\text{fix})}_{\text{SPP}}$ is a fixed smoothing constant. This estimator can be implemented in speech enhancement framework by evaluating (2.22), (2.23) and (2.24) for each new observation $Y_{k,\ell}$. If the noise PSD is strongly underestimated, the SPP in (2.22) is overestimated, i.e., it is close to 1. As a result, the noise periodogram in (2.23) may no longer be updated. To avoid such stagnations, the SPP is set to a lower value if it has been stuck at 1 for a longer period of time [70], [71].

As shown in Fig. 2.2, the speech PSD is estimated based on the noise PSD and the noisy observation $Y_{k,\ell}$. Assuming a Gaussian prior for the speech and the noise coefficients allows the derivation of a maximum-likelihood estimator of the speech PSD $\Lambda^s_{k,\ell}$. Under a Gaussian model, the PDF of the noisy coefficients $Y_{k,\ell}$ can be written as

$$f(Y_{k,\ell}|\Lambda^s_{k,\ell}, \Lambda^n_{k,\ell}) = \frac{1}{\pi(\Lambda^s_{k,\ell} + \Lambda^n_{k,\ell})}\exp\left(-\frac{|Y_{k,\ell}|^2}{\Lambda^s_{k,\ell} + \Lambda^n_{k,\ell}}\right). \tag{2.25}$$

Calculating the derivative of (2.25) with respect to $\Lambda^s_{k,\ell}$, setting the result to zero and solving for $\Lambda^s_{k,\ell}$ results in the maximum-likelihood estimator. Dividing the result by $\Lambda^n_{k,\ell}$ allows the maximum-likelihood estimator to be expressed in terms of the *a priori* SNR as

$$\hat{\xi}_{k,\ell} = \max\left(\frac{|Y_{k,\ell}|^2}{\hat{\Lambda}^n_{k,\ell}} - 1, \xi^{\text{ml}}_{\text{min}}\right). \tag{2.26}$$

The noise PSD estimate $\hat{\Lambda}^n_{k,\ell}$ is used to compute the maximum-likelihood estimator in practical applications. Further, to avoid numerical issues in the enhancement and a negative speech PSD, the maximum of $|Y_{k,\ell}|^2/\hat{\Lambda}^n_{k,\ell} - 1$ and $\xi^{\mathrm{ml}}_{\min}$ is used. The maximum-likelihood estimator itself is rarely used in practical applications due to the many artifacts which are induced in the enhanced signal. The artifacts are caused by local overestimations of the speech PSD such that single time-frequency points are less suppressed which generates tone like noises. As the estimates of the maximum-likelihood approach are highly variant, such overestimations occur frequently.

An alternative is the decision-directed approach which has been proposed in [12]. Here, the maximum-likelihood estimate and the last segment's estimated clean speech coefficients are combined as

$$\hat{\xi}_{k,\ell} = \alpha_{\mathrm{DD}} \frac{|\hat{S}_{k,\ell-1}|^2}{\hat{\Lambda}^n_{k,\ell-1}} + (1 - \alpha_{\mathrm{DD}}) \max\left(\frac{|Y_{k,\ell}|^2}{\hat{\Lambda}^n_{k,\ell}} - 1, \xi^{\mathrm{ml}}_{\min} \right). \tag{2.27}$$

The rationale behind this combination is that the expected value of both components, i.e., $\mathbb{E}\{|\hat{S}_{k,\ell-1}|^2/\Lambda^n_{k,\ell-1}\}$ and $\mathbb{E}\{|Y_{k,\ell}|^2/\Lambda^n_{k,\ell} - 1\}$, respectively, yields the *a priori* SNR $\xi_{k,\ell}$. Here, $0 < \alpha_{\mathrm{DD}} < 1$ is a smoothing constant. For high values of α_{DD}, the estimator generally generates less artifacts compared to the maximum-likelihood estimator in (2.26). However, the estimate $\hat{\xi}_{k,\ell}$ becomes delayed and, as a result, speech onsets may be suppressed and distorted. For low values, the weight on the maximum-likelihood estimator part is increased, which, again, results in stronger musical tone artifacts, but reduces the estimation delay. Hence, the choice of α_{DD} is a compromise of speech quality and background noise quality.

An algorithm which allows the estimation of the speech PSD $\Lambda^s_{k,\ell}$ without musical tone artifacts and high tracking speed has been proposed in [82], [83] and is referred to as temporal cepstrum smoothing (TCS). Here, the maximum-likelihood estimate as in (2.26) is transformed to the cepstral domain by taking the logarithm and applying the inverse discrete Fourier transform (IDFT) as

$$\hat{\Lambda}^{s,\mathrm{ml}}_{o,\ell} = \frac{1}{K} \sum_{k=0}^{K-1} \log(\hat{\Lambda}^{s,\mathrm{ml}}_{k,\ell}) e^{j\frac{2\pi o k}{K}}. \tag{2.28}$$

The index o denotes the quefrency. In the cepstral domain, speech can be represented by using only a few coefficients: The speech spectral envelope, which reflects the impact of the vocal tract filter, is represented by the lower coefficients with $o < 2.5$ ms whereas the speech spectral fine structure, i.e., the fundamental frequency and its harmonics, is approximated by a single peak among the high cepstral coefficients. This peak is also referred to as pitch peak. The compact representation of speech is exploited by the TCS approach by using a quefrency and time dependent smoothing constant $\alpha_{o,\ell}$ to smooth $\hat{\Lambda}^{s,\mathrm{ml}}_{o,\ell}$ as

$$\hat{\Lambda}^{s}_{o,\ell} = (1 - \alpha_{o,\ell}) \hat{\Lambda}^{s,\mathrm{ml}}_{o,\ell} + \alpha_{o,\ell} \hat{\Lambda}^{s}_{o,\ell-1}. \tag{2.29}$$

For the cepstral coefficients that are associated with speech only little smoothing is applied while the remaining cepstral coefficients are strongly smoothed. Accordingly, $\alpha_{o,\ell}$ is set close to 0 for the lower cepstral coefficients and close to 1 for the high coefficients. In voiced segments, the $\alpha_{o,\ell}$ in close vicinity to the cepstral pitch peak are changed to values close to 0.

The cepstrally smoothed speech PSD $\hat{\Lambda}^{s}_{o,\ell}$ is transformed back to the spectral domain as

$$\hat{\Lambda}^{s}_{k,\ell} = \exp(\mathrm{DFT}\{\hat{\Lambda}^{s}_{o,\ell}\} + \varkappa). \tag{2.30}$$

As the smoothing in the cepstral domain results in a biased estimate [83], the correction term $\varkappa$ is added. In [82], it has been argued that the bias of computing the expected value of a spectral quantity following a Gaussian distribution in the logarithmic domain amounts to the Euler constant. Due to the smoothing, the estimate in the cepstral domain is between an instantaneous value and the expected value. Hence, $\varkappa$ is set 1/2 of the Euler constant, i.e., $\varkappa \approx 0.5 \cdot 0.5772\ldots$ is used. A more rigorous analysis of the bias is given in [83].

2.2. INSTRUMENTAL MEASURES

In this section, instrumental measures are presented that are used to assess the accuracy of noise PSD estimation algorithms, estimate the quality of the enhanced signal as well as the speech distortion and noise reduction induced by the enhancement algorithm.

2.2.1. Estimation Accuracy of Noise PSD Estimators

The task of noise PSD estimation algorithms is to determine the variance $\Lambda^{n}_{k,\ell}$ of the spectral noise coefficients $N_{k,\ell}$. To assess the accuracy of such estimation algorithms, various measures have been utilized. In [29], [75], [78], averages of the relative error between the true noise PSD $\Lambda^{n}_{k,\ell}$ and the estimate $\hat{\Lambda}^{n}_{k,\ell}$ have been considered. In [78], it has been found that the relative error is more sensitive to overestimations than to underestimations and, therefore, a more symmetric error measure has been proposed. This measure is referred to as log-error distortion and has been employed in many studies, e.g., [70], [71], [73], [78], [80], [229]. Accordingly, the log-error distortion is used also here to assess the accuracy of noise PSD estimation algorithms, e.g., in Chapter 3 and Chapter 4.

As in [70], [71], the log-error distortion is split into an overestimation and an underestimation error

$$\mathrm{LogErr} = \mathrm{LogErr}_{\uparrow} + \mathrm{LogErr}_{\downarrow}. \tag{2.31}$$

Here, $\mathrm{LogErr}_{\uparrow}$ denotes the contributions of the overestimation while $\mathrm{LogErr}_{\downarrow}$ are the contributions of the underestimation. These two quantities are computed using the estimated noise PSD $\hat{\Lambda}^{n}_{k,\ell}$ and a reference PSD $\Lambda^{n}_{k,\ell}$ as

$$\mathrm{LogErr}_{\downarrow} = -\frac{1}{LK}\sum_{\ell=0}^{L-1}\sum_{k=0}^{K-1}\min\left(0, 10\log_{10}\left(\frac{\hat{\Lambda}^{n}_{k,\ell}}{\Lambda^{n}_{k,\ell}}\right)\right), \tag{2.32}$$

$$\mathrm{LogErr}_{\uparrow} = \frac{1}{LK}\sum_{\ell=0}^{L-1}\sum_{k=0}^{K-1}\max\left(0, 10\log_{10}\left(\frac{\hat{\Lambda}^{n}_{k,\ell}}{\Lambda^{n}_{k,\ell}}\right)\right). \tag{2.33}$$

Overestimation errors, as indicated by $\mathrm{LogErr}_{\uparrow}$, result in stronger attenuations of the respective time-frequency points. As a consequence, the clean speech signal may potentially be suppressed which results in distortions and a reduced perceived quality. Contrarily, underestimation errors, which is indicated by $\mathrm{LogErr}_{\downarrow}$, result in less attenuation of the background noise. Accordingly, more potentially disturbing residual noise components remain after the enhancement. Especially if strong underestimations occur locally, i.e., the period were the underestimation occurs is short in time, annoying artifacts such as musical tones may degrade the enhanced signal.

For evaluating (2.32) and (2.33) in practical applications, the reference noise PSD $\Lambda^{n}_{k,\ell}$ needs to be known. For stationary noise types, e.g., white Gaussian noise or pink noise, the reference noise PSD $\Lambda^{n}_{k,\ell}$ can be easily determined by averaging the noise periodogram $|N_{k,\ell}|^2$ over several minutes of audio. However, most real-world background noise types are non-stationary, i.e., the underlying statistical moments change over time. As a consequence, a temporal average over a long time interval cannot represent the time-varying noise PSD adequately. In [70], [71], [80], it is proposed to use the noise periodogram $|N_{k,\ell}|^2$ as reference noise PSD. However, as the periodogram is a highly variant estimator, other studies, e.g., [78], [229], smooth the periodograms $|N_{k,\ell}|^2$ temporarily using a short smoothing filter to reduce the variations. If smoothing is applied, exponential smoothing filters such as

$$\Lambda^{n}_{k,\ell} = (1-\alpha^{(\mathrm{fix})}_{\mathrm{LogErr}})|N_{k,\ell}|^2 + \alpha^{(\mathrm{fix})}_{\mathrm{LogErr}}\Lambda^{n}_{k,\ell-1} \tag{2.34}$$

are commonly employed. Here, $0 < \alpha^{(\mathrm{fix})}_{\mathrm{LogErr}} < 1$ is the smoothing constant, which controls the amount of smoothing. On the one hand, using values close to 0, only a slight reduction of the estimation variance can be achieved but changes in the noise PSD can be tracked quickly. On the other hand, values close to 1 strongly reduce the variance but the tracking is considerably slower. For the evaluations, in this thesis, the exponential smoothing approach given in (2.34) is used to determine the reference PSD. Here, $\alpha^{(\mathrm{fix})}_{\mathrm{LogErr}}$ is chosen such that the exponential smoothing window, which results from applying (2.34), achieves the same variance reduction as an equivalent rectangular window of 50 ms. This value is chosen as a compromise between variance reduction and tracking speed.

2.2.2. Instrumental Measures of the Perceived Signal Quality

In this section, instrumental measures are presented that are used to assess the perceived quality of the enhanced signal, distortions of the speech signal and the noise reduction.

The overall perceived quality is reflected by improvements of the segmental SNR (SegSNR) which is an intrusive measure, i.e., a reference of an optimally enhanced signal is required. For this, the clean speech signal s_t is used. The SegSNR of the noisy signal y_t is defined as

$$\mathrm{SegSNR}(y_t) = \frac{1}{|\mathbb{L}^{(s)}|} \sum_{\ell \in \mathbb{L}^{(s)}} 10 \log_{10} \left(\frac{\sum_{t=0}^{M-1} s_{\ell M+t}^2}{\sum_{t=0}^{M-1} (y_{\ell M+t} - s_{\ell M+t})^2} \right). \tag{2.35}$$

Here, $\mathbb{L}^{(s)}$ denotes the set of segments, in which speech is active. The speech active segments are determined on the clean speech input signal and segments that are at most 45 dB lower in power compared to the segment with the highest power are marked as speech active. Note that the segment length and segment shift used to compute the SegSNR are generally different to the segment length and segment shift used for the speech enhancement. As in [71], a segment length of 10 ms is used and the segments do not overlap. Similar to (2.35), the SegSNR of the enhanced speech signal $\mathrm{SegSNR}(\hat{s}_t)$ can be computed. From this, the segmental SNR improvement ΔSegSNR is obtained as

$$\Delta\mathrm{SegSNR} = \mathrm{SegSNR}(\hat{s}_t) - \mathrm{SegSNR}(y_t). \tag{2.36}$$

Higher values of the segmental SNR improvements ΔSegSNR generally indicate a better performance of the enhancement algorithm.

Further, Perceptual Evaluation of Speech Quality (PESQ) [230] scores are used as an instrumental measure of the perceived signal quality. Similar to the SegSNR, also PESQ is an intrusive measure and requires a reference signal, i.e., the clean speech signal. This measures gives values in the range of -0.5 and 4.5, where -0.5 is the lowest quality score and 4.5 indicates a very high quality. Again, the improvements in PESQ, denoted by ΔPESQ, are considered. Similar to the SegSNR improvements, these are obtained by subtracting the raw PESQ score of the noisy speech signal y_t from the raw PESQ score of the enhanced signal $\hat{s}_t$.

The speech quality of the enhanced signal and the amount of noise reduction are evaluated using the segmental speech SNR (SegSSNR) and the segmental noise reduction (SegNR), respectively. For computing the SegSSNR and the SegNR, a master-slave filtering approach is employed. This approach is depicted in Fig. 2.3. Here, the speech signal s_t and the noise signal n_t are mixed at a given SNR and the noisy signal y_t is enhanced using the enhancement scheme under test. In parallel streams, the filter coefficients of the enhancement filter, i.e., the coefficients of the gain function $G_{k,\ell}$, are also applied to the clean speech signal and the background noise. For this, the same STFT framework is used as for the enhancement algorithm. This yields the processed clean speech signal $\tilde{s}_t$ and the processed noise signal $\tilde{n}_t$ which allow the effects of the enhancement scheme in terms of speech distortion and noise reduction to be measured separately. In [52], [71], the

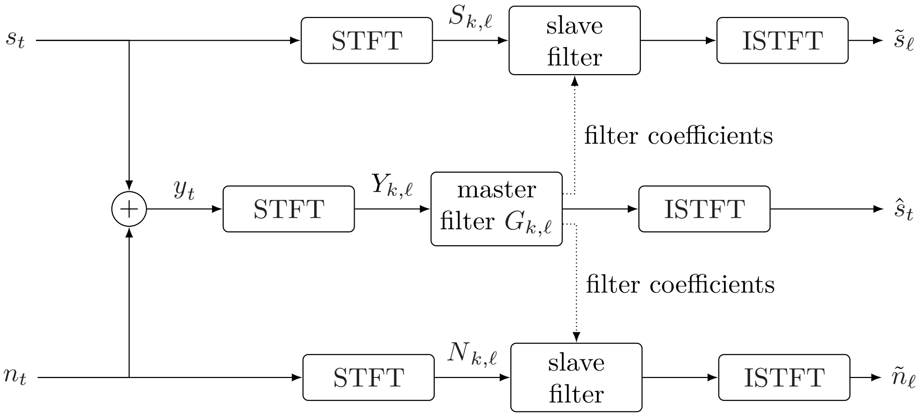

Fig. 2.3.: Block scheme of the master-slave filtering approach used to compute the SegSSNR and SegNR.

SegSSNR and the SegNR are defined as

$$\text{SegSSNR} = \frac{1}{|\mathbb{L}^{(s)}|} \sum_{\ell \in \mathbb{L}^{(s)}} 10 \log_{10} \left(\frac{\sum_{t=0}^{M-1} s_{M\ell+t}^2}{\sum_{t=0}^{M-1} (s_{M\ell+t} - \tilde{s}_{M\ell+t})^2} \right), \tag{2.37}$$

$$\text{SegNR} = \frac{1}{|\mathbb{L}^{(s)}|} \sum_{\ell \in \mathbb{L}^{(s)}} 10 \log_{10} \left(\frac{\sum_{t=0}^{M-1} n_{M\ell+t}^2}{\sum_{t=0}^{M-1} \tilde{n}_{M\ell+t}^2} \right). \tag{2.38}$$

High values of the measures indicate low speech distortion and high noise reduction which are both desirable properties of speech enhancement algorithms.

Part II.

Bias of Adaptive Smoothing Based Noise PSD Estimators

CHAPTER 3

ON THE BIAS OF ADAPTIVE SMOOTHING BASED NOISE PSD ESTIMATION

The topic of Part II of this thesis is the estimation accuracy of single-channel noise PSD estimators. The task of such algorithms is equivalent to finding the mean of the noise periodogram, i.e., $\mathbb{E}\{|N_{k,\ell}|^2\}$, given a noisy observation. In this part, noise PSD estimators are considered whose structure is similar to first-order recursive smoothing filters. Because of the low computational complexity and the low memory demand, first-order recursive smoothing is a commonly applied technique to track the mean of non-stationary random variables. It is equivalent to a moving average where an exponentially decaying smoothing window is employed. A stronger weight is put on the more recent samples allowing these filters to track changes of the mean value over time.

In [215], it has been shown that the noise PSD estimators presented in [81, Section 14.1.3] and [70] are implicitly or explicitly based on a first-order recursive structure as shown in Fig. 3.1. However, in many applications, such as the noise PSD estimators in [70], [81], an adaptive smoothing factor $\alpha(y_\ell, \overline{y}_{\ell-1})$ is employed as

$$\overline{y}_\ell = [1 - \alpha(y_\ell, \overline{y}_{\ell-1})]y_\ell + \alpha(y_\ell, \overline{y}_{\ell-1})\overline{y}_{\ell-1}, \tag{3.1}$$

where $\alpha(y_\ell, \overline{y}_{\ell-1})$ is a function of both y_ℓ and $\overline{y}_{\ell-1}$. The quantity y_ℓ is the observation of the random process describing the periodogram of the input signal at time ℓ while $\overline{y}_\ell$ denotes the estimated mean, i.e., the estimated noise PSD. Similar to nonadaptive first-order smoothing, the smoothing factor $0 \leq \alpha(y_\ell, \overline{y}_{\ell-1}) \leq 1$ controls the tracking speed and the variance of the estimate. The noise PSD estimators in [81, Section 14.1.3] and [70] employ adaptive smoothing factors to avoid speech leakage. The algorithm described in [81, Section 14.1.3] switches between two fixed smoothing constants where a larger one is used if the energy of the noisy periodogram is higher than the background noise PSD, i.e., for large *a posteriori* SNRs $\gamma_{k,\ell}$. In [70], the value of the adaptive smoothing factor is implicitly adapted using the SPP and also grows with an increasing *a posteriori* SNR $\gamma_{k,\ell}$. In contrast to the noise PSD estimator in [81, Section 14.1.3], this results in a soft transition.

This chapter is partly based on:

[215] R. Rehr and T. Gerkmann, “On the bias of adaptive first-order recursive smoothing,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2015, © 2015 IEEE.

[216] R. Rehr and T. Gerkmann, “An analysis of adaptive recursive smoothing with applications to noise PSD estimation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 2, pp. 397–408, Feb. 2017, © 2017 IEEE.

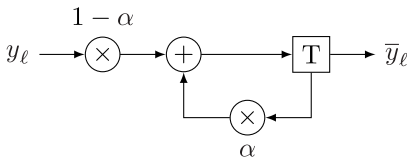

Fig. 3.1.: Block diagram of a first-order recursive filter structure.

A disadvantage of the application of adaptive smoothing is that the estimate of the mean is in general biased as shown in [215]. In this chapter, we analyze this bias and derive an algorithm to compensate for it. The proposed algorithm adds only a low amount of computational complexity to the existing noise PSD estimators as only a computation of a term similar to the Wiener filter and a multiplication with this factor is required. In this chapter, the approach introduced in [215] is summarized and its extension in [216] is presented. In [215], the case of speech presence for the correction has not been explicitly considered which is included in [216]. This helps to prevent overestimations in high SNR regions. Further, a novel method based on the transition density $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ between two successive smoothed filter outputs is presented. Experiments are conducted on real world signals showing that the reduction of the bias leads to a reduced log-error distortion [231] and increases the quality in terms of PESQ scores [230]. Additional experiments are conducted where the influence of signal correlations on the bias is explicitly considered. Throughout the evaluation, we use the noise PSD estimators [70], [81] as examples taken from single-channel noise PSD estimation. The methods presented in this chapter are generally iterative and require multiple steps to determine the correction factor. In Chapter 4, [217], another method for correcting the bias is presented where the correction factor can be computed in a single step.

This chapter is organized as follows: First, we introduce basic properties of adaptive smoothing in Section 3.1. These are used to derive a fixed correction factor to compensate for the bias caused by adaptive smoothing. In Section 3.2 and Section 3.3 two different methods are proposed to estimate the fixed correction factor. After that, we apply the bias compensation methods to speech enhancement frameworks. For this, we describe the signal model and explain the relationship between components of the model and the quantities of adaptive smoothing in Section 3.4. In the same section, we also introduce the noise PSD estimators given in [81, Section 14.1.3] and [70] in the context of adaptive smoothing. For the application of noise PSD estimation, we extend the correction method to account for the additional energy of the speech signal in Section 3.5. The evaluation of the proposed methods follows in Section 3.6 while Section 3.7 concludes this chapter.

3.1. BASIC PROPERTIES AND BIAS COMPENSATION

In this section, we present basic properties of adaptive first-order recursive smoothing: first, adaptive smoothing as defined in (3.1) does not alter the properties of the input signal y_ℓ in terms of stationarity and ergodicity. Second, the adaptive smoothing functions are scale-invariant if $\alpha(y_\ell, \overline{y}_{\ell-1})$ depends only on the ratio $y_\ell/\overline{y}_{\ell-1}$. Scale-invariance describes the property that if the input y_ℓ is scaled by a factor $r > 0$, the resulting output $\overline{y}_\ell$ is scaled by the same factor r. These two properties allow the bias to be simply compensated by a multiplicative factor.

3.1.1. Stationarity and Ergodicity

The propositions 6.6 and 6.31 in [232] state that a process defined by

$$\tilde{y}_\ell = \phi(y_\ell, y_{\ell-1}, \dots) \tag{3.2}$$

is stationary and ergodic, if the process given by $y_\ell, y_{\ell-1}, \dots$ is stationary and ergodic. Here, $\phi(\cdot)$ is a function of the current and the past elements of the random process, e.g., the adaptive first-order smoothing as in (3.1). The propositions, however, implicitly assume that the output process $\tilde{y}_\ell$ exists meaning that the process $\tilde{y}_\ell$ does not diverge. As the adaptive smoothing factors $\alpha(y_\ell, \overline{y}_{\ell-1})$ are limited to values between zero and one, the filter function in (3.1) is stable in the sense that a bounded input results in a bounded output. Thus, considering a finite stationary and ergodic input y_ℓ, it follows that also the filter output $\overline{y}_\ell$ is ergodic and stationary.

3.1.2. Scale-Invariance

The process of adaptive recursive smoothing (3.1) is scale-invariant if the adaptive smoothing function depends only on the ratio $y_\ell/\overline{y}_{\ell-1}$. In particular, if the input y_ℓ is scaled by a factor r, the output $\overline{y}_\ell$ is scaled by the same factor r. This property is of particular relevance for the noise PSD estimators considered in Section 3.4 as their respective adaptive smoothing function depends only on the ratio $y_\ell/\overline{y}_{\ell-1}$.

The statement can be proven using the method of induction. For linear first-order recursive smoothing filters, it is often assumed that the system is initially at rest, i.e., $\overline{y}_\ell = 0$ for $\ell < 0$. As all of the considered adaptive smoothing functions depend on the ratio $y_\ell/\overline{y}_{\ell-1}$, this assumption is not applicable because of the division by zero. Thus, we assume that the first filter output $\overline{y}_0$ is equal to the first filter input y_0. From the assumption that $\overline{y}_0 = y_0$ it follows that a scaling of y_ℓ by r leads to $r\overline{y}_0 = ry_0$. Hence, it can be shown for the following samples of (3.1) that

$$\left[1-\alpha\left(\frac{ry_\ell}{r\overline{y}_{\ell-1}}\right)\right]ry_\ell+\alpha\left(\frac{ry_\ell}{r\overline{y}_{\ell-1}}\right)r\overline{y}_{\ell-1} \tag{3.3}$$

$$=r\left(\left[1-\alpha\left(\frac{y_\ell}{\overline{y}_{\ell-1}}\right)\right]y_\ell+\alpha\left(\frac{y_\ell}{\overline{y}_{\ell-1}}\right)\overline{y}_{\ell-1}\right) \tag{3.4}$$

$$=r\overline{y}_\ell. \tag{3.5}$$

This shows that the adaptive smoothing procedure is scale-invariant if the smoothing function depends only on the ratio $y_\ell/\overline{y}_{\ell-1}$.

3.1.3. Bias Compensation

In this part, we describe how the bias caused by adaptive smoothing can be compensated. For the derivation, we assume that the filter input y_ℓ can be described by a stationary and ergodic random process. Note that the presence of a speech signal in a speech enhancement context will explicitly be taken into account in Section 3.5. With the stationarity, the ergodicity, and the scale invariance described in the sections 3.1.1 and 3.1.2, the bias can be corrected by multiplying the filter output $\overline{y}_\ell$ by a fixed correction factor $\mathcal{C}$ as

$$\check{\overline{y}}_\ell = \mathcal{C}\overline{y}_\ell. \tag{3.6}$$

Here, $\check{\overline{y}}_\ell$ denotes the corrected filter output. For obtaining an unbiased estimate $\mathbb{E}\{\check{\overline{y}}_\ell\} = \mathbb{E}\{y_\ell\}$, the factor has to be set to

$$\mathcal{C} = \mathbb{E}\{y_\ell\}/\mathbb{E}\{\overline{y}_\ell\}. \tag{3.7}$$

As this factor does not depend on the scaling of y_ℓ or $\overline{y}_\ell$, it is sufficient to determine this quantity for a given mean of the input signal, e.g., $\mathbb{E}\{y_\ell\} = 1$. With the assumption of stationarity, the fixed factor $\mathcal{C}$ does also not depend on time. Consequently, $\mathcal{C}$ can be determined before any processing takes place. The factor $\mathcal{C}$ can be considered the bias between filter input and output after convergence. Despite the assumption of stationarity, we show that the bias reflected by $\mathcal{C}$ is also applicable to nonstationary signals in the evaluation, i.e., Section 3.6. Methods that can be employed to determine the fixed correction factor $\mathcal{C}$ are presented in Section 3.2 and Section 3.3.

3.2. ITERATIVE BIAS COMPENSATION

In this section, we revise the method for determining the fixed correction factor $\mathcal{C}$ that we propose in [215]. If the adaptive smoothing function depends only on the unsmoothed input y_ℓ but not on the smoothed output $\overline{y}_{\ell-1}$, the bias caused by adaptive smoothing can be determined by analytically deriving the expected value of $\overline{y}_\ell$. Based on the solution obtained for the analytically solvable case, an iterative method has been presented in [215] that can be used to approximately determine the bias for the more complicated case where the adaptive smoothing function also depends on the estimated mean $\overline{y}_{\ell-1}$. The method estimates the fixed correction factor $\mathcal{C}$ quite accurately as shown in our evaluations.

First, we consider adaptive smoothing factors $\alpha(y_\ell, \overline{y}_{\ell-1})$ that are independent of the previous filter output $\overline{y}_{\ell-1}$. With this assumption, (3.1) simplifies to

$$\overline{y}_\ell = [1 - \alpha(y_\ell)]\, y_\ell + \alpha(y_\ell)\overline{y}_{\ell-1}. \tag{3.8}$$

For the derivations, we assume that all y_ℓ are identically distributed and uncorrelated. Further, using the stationarity property described in Section 3.1.1, we can assume that a stationary input y_ℓ results in a stationary output $\overline{y}_\ell$. From this, it follows that $\mathbb{E}\{\overline{y}_i\} = \mathbb{E}\{\overline{y}_j\}$ where $i \neq j$ are two different time instances. With the first assumption, the expected value $\mathbb{E}\{y_\ell \overline{y}_{\ell-1}\}$ can be written as $\mathbb{E}\{y_\ell\}\mathbb{E}\{\overline{y}_{\ell-1}\}$. Consequently, applying $\mathbb{E}\{\cdot\}$ to (3.8) and rearranging the terms, results in [215]

$$\mathbb{E}\{\overline{y}_\ell\} = \frac{\mathbb{E}\{y_\ell\} - \mathbb{E}\{y_\ell \alpha(y_\ell)\}}{1 - \mathbb{E}\{\alpha(y_\ell)\}}. \tag{3.9}$$

The obtained expression depends only on the adaptive function $\alpha(y_\ell)$ and the PDF of y_ℓ.

In the remainder, we consider the case where $\alpha(y_\ell, \overline{y}_{\ell-1})$ depends also on the recursively estimated mean $\overline{y}_{\ell-1}$. This case is more challenging because the quantity $\overline{y}_{\ell-1}$ influences the behavior of the adaptive smoothing factor which, in turn, influences the estimation of $\overline{y}_\ell$. This type of adaptation is, however, the most relevant for noise PSD estimators, e.g., for the approaches [70], [81] considered in Section 3.4.

Deriving $\mathbb{E}\{\overline{y}_\ell\}$ while taking into account the dependence on $\overline{y}_{\ell-1}$ is difficult because $\overline{y}_{\ell-1}$ appears in a generally non-linear function $\alpha(y_\ell, \overline{y}_{\ell-1})$ and is a random variable itself as it emerges from the combination of all past y_ℓ. Consequently, $\overline{y}_{\ell-1}$ is also correlated with the previous estimates $\overline{y}_{\ell-2}, \overline{y}_{\ell-3}, \cdots$. Hence, the problem is simplified in [215] by replacing $\overline{y}_{\ell-1}$ in the adaptive function by a fixed value $\overline{y}^{(\mathrm{fix})}$. With that, the bias can be determined iteratively based on the result given in (3.9) as

$$\overline{y}_i^{(\mathrm{fix})} = \frac{\mathbb{E}\{y_\ell\} - \mathbb{E}\{y_\ell \alpha(y_\ell, \overline{y}_{i-1}^{(\mathrm{fix})})\}}{1 - \mathbb{E}\{\alpha(y_\ell, \overline{y}_{i-1}^{(\mathrm{fix})})\}}. \tag{3.10}$$

Here, $\overline{y}_i^{(\mathrm{fix})}$ is the estimate of $\mathbb{E}\{\overline{y}_\ell\}$ obtained for the ith iteration step where the initial condition is denoted by $\overline{y}_0^{(\mathrm{fix})}$. This approach is motivated by the recursive update of $\overline{y}_\ell$ in (3.1), which is performed sample by sample. In each step of (3.10), however, all samples over an infinite time period are considered. To determine the final estimate of $\mathbb{E}\{\overline{y}_\ell\}$, the iteration is continued until it converges. With the converged $\overline{y}_i^{(\mathrm{fix})}$, the estimated correction factor can be determined as $\mathcal{C} = \mathbb{E}\{y_\ell\}/\overline{y}_i^{(\mathrm{fix})}$. For the adaptive smoothing factors used in [70], [81], we will show that the parameter $\overline{y}_0^{(\mathrm{fix})}$ does not influence the convergence of the iterative approach. This procedure is summarized in Algorithm 1.

Algorithm 1 Iterative estimation of the fixed correction factor $\mathcal{C}$ for adaptive functions depending on $\overline{y}_{\ell-1}$ proposed in Section 3.2. Here, we refer to the solutions for the specific noise PSD estimators [70], [81] where appropriate.

```
$i \leftarrow 0$, $\overline{y}_0^{(\mathrm{fix})} \leftarrow 1$, $\Lambda_{k,\ell}^{y} \leftarrow 1$.
while convergence criterion for $\overline{y}_i^{(\mathrm{fix})}$ is not met do
    Obtain $\overline{y}_{i+1}^{(\mathrm{fix})}$ using (3.10). The solutions for the adaptive functions in [70], [81] are
    given in (A.1) and (A.2) of Appendix A.1 if $y_\ell$ is exponentially distributed.
    $i \leftarrow i + 1$
end while
Compute compensation factor: $\mathcal{C} = \Lambda_{k,\ell}^{y} / \overline{y}_i^{(\mathrm{fix})}$.
```

3.3. ESTIMATING THE BIAS USING TRANSITION DENSITIES

In this section, we propose a novel method for determining the fixed correction factor $\mathcal{C}$. For this, we use the transition density $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ which can be considered a description that explains how the smoothing factor $\alpha(y_\ell, \overline{y}_{\ell-1})$ influences the filter output $\overline{y}_\ell$ and vice versa.

If the input samples y_ℓ are assumed to be independent and identically distributed, i.e., stationary and ergodic, the random process of the filter output can be described by the transition density $f(\overline{y}_\ell|\overline{y}_{\ell-1})$. The conditional density $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ is a function that depends on the smoothing function $\alpha(y_\ell, \overline{y}_{\ell-1})$ and the distribution of the input variable y_ℓ as we will show later in this section. It can be considered the link that describes how the previous filter output $\overline{y}_{\ell-1}$ affects the behavior of the smoothing function, $\alpha(y_\ell, \overline{y}_{\ell-1})$, and vice versa. In other words, the interaction between $\overline{y}_{\ell-1}$ and $\alpha(y_\ell, \overline{y}_{\ell-1})$ is included in the PDF $f(\overline{y}_\ell|\overline{y}_{\ell-1})$. We use this conditional PDF to optimize the parameters $\boldsymbol{\theta}$ of a known model PDF $\tilde{f}(\overline{y}_\ell|\boldsymbol{\theta})$ such that it matches the PDF of the filter output samples, i.e., $f(\overline{y}_\ell)$, as close as possible. To determine the parameters $\boldsymbol{\theta}$, we exploit the stationarity from which it follows that $f(\overline{y}_\ell) = f(\overline{y}_{\ell-1}) = \cdots$. According to that, there is a PDF such that marginalizing $f(\overline{y}_\ell|\overline{y}_{\ell-1})f(\overline{y}_{\ell-1})$ over $\overline{y}_{\ell-1}$ results in the same PDF for $\overline{y}_{\ell-1}$ as for $\overline{y}_\ell$, i.e., $f(\overline{y}_{\ell-1}) = f(\overline{y}_\ell)$. Therefore, we propose to optimize the parameters $\boldsymbol{\theta}$ of a model PDF $\tilde{f}(\overline{y}_\ell|\boldsymbol{\theta})$ such that the PDF $\tilde{g}(\overline{y}_\ell|\boldsymbol{\theta})$ obtained by the marginalization

$$\tilde{g}(\overline{y}_\ell|\boldsymbol{\theta}) = \int_{-\infty}^{\infty} f(\overline{y}_\ell|\overline{y}_{\ell-1})\tilde{f}(\overline{y}_{\ell-1}|\boldsymbol{\theta})d\overline{y}_{\ell-1} \tag{3.11}$$

resembles the originally used model $\tilde{f}(\overline{y}_\ell|\boldsymbol{\theta})$ as closely as possible. The similarity between $\tilde{g}(\overline{y}_\ell|\boldsymbol{\theta})$ and $\tilde{f}(\overline{y}_\ell|\boldsymbol{\theta})$ is quantified by the Bhattacharyya distance [233]

$$\mathcal{B}\left(\tilde{f}, \tilde{g}\right) = -\ln\left[\eta\left(\tilde{f}, \tilde{g}\right)\right]. \tag{3.12}$$

Algorithm 2 Estimation of the fixed correction factor $\mathcal{C}$ by maximizing the self-similarity with respect to the transition density $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ as proposed in Section 3.3. Here, we refer to the solutions of the specific noise PSD estimators [70], [81] where appropriate.

```
1: Choose a PDF $f(y_\ell)$ that describes the filter input, e.g., (3.16).
2: Determine $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ using (3.15).
   For the adaptive functions in [70], [81], the analytical solutions for $\mathcal{F}^{-1}(\cdot)$ and $\mathcal{F}'(\cdot)$ are given in Appendix A.2 .
3: Select a model PDF $\tilde{f}(\overline{y}_\ell|\boldsymbol{\theta})$, e.g, (3.22).
4: Minimize (3.14) to obtain $\hat{\boldsymbol{\theta}}$, e.g., using [236].
5: Compute the correction factor: $\mathcal{C} = \mathbb{E}\{y_\ell\}/m(\hat{\boldsymbol{\theta}})$.
```

Here, $\eta(\cdot)$ is the Bhattacharyya coefficient which is given by

$$\eta\left(\tilde{f}, \tilde{g}\right) = \int_{-\infty}^{\infty} \sqrt{\tilde{f}(x|\boldsymbol{\theta})\tilde{g}(x|\boldsymbol{\theta})}dx \tag{3.13}$$

for continuous PDFs [234]. The Bhattacharyya coefficient takes values between zero and one where a result of one means that both PDFs are identical. Consequently, the optimal parameters $\hat{\boldsymbol{\theta}}$ are defined as those that minimize the Bhattacharyya distance as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{B}\left(\tilde{f}, \tilde{g}\right). \tag{3.14}$$

As the analytic solution for the integrals in (3.11) and (3.13) are unknown, we solve these expressions using numerical integration methods. This also motivates the usage of the Bhattacharyya distance which is numerically easier to handle than other distance measures, e.g., the Kullback-Leibler divergence [235]. After the optimization, the optimal parameters $\hat{\boldsymbol{\theta}}$ are used to determine the expected value $\mathbb{E}\{\overline{y}_\ell\}$. For this, we assume that $m(\boldsymbol{\theta})$ is a function that returns the mean of the model distribution $\tilde{f}(\overline{y}_\ell|\boldsymbol{\theta})$ for the parameters $\boldsymbol{\theta}$. With that, the fixed correction factor is determined as $\mathcal{C} = \mathbb{E}\{y_\ell\}/m(\hat{\boldsymbol{\theta}})$. This procedure can be described as self-similarity maximization with respect to the transition density $f(\overline{y}_\ell|\overline{y}_{\ell-1})$.

The transition density function $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ can be derived given a model for the PDF of the input $f(y_\ell)$ and (3.1). As $\overline{y}_{\ell-1}$ is the given variable in the conditional PDF, $\overline{y}_{\ell-1}$ can be thought of as a fixed quantity and (3.1) can be treated as a function $\overline{y}_\ell = \mathcal{F}(y_\ell)$ of the random variable y_ℓ. Then, for a piecewise monotonic function $\mathcal{F}(\cdot)$, the conditional density function $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ can be determined using a change of variables as described in [237, Chapter 5]. The solution is given by

$$f(\overline{y}_\ell|\overline{y}_{\ell-1}) = \sum_{j=1}^{L} \frac{f_{y_\ell}(\mathcal{F}_j^{-1}(\overline{y}_\ell))}{|\mathcal{F}'(\mathcal{F}_j^{-1}(\overline{y}_\ell))|} \tag{3.15}$$

where $\mathcal{F}_j^{-1}(\cdot)$ denotes the inverse of the jth monotonic segment of the function $\mathcal{F}(\cdot)$ while L denotes the number of monotonic segments of the considered function. Furthermore, $\mathcal{F}'(\cdot)$

is the first derivative of the function $\mathcal{F}(\cdot)$. In contrast to the iterative method in Section 3.2, the conditional PDF $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ is easily derived for any PDF of the input signal y_ℓ as it is only required to exchange $f_{y_\ell}(\cdot)$ in (3.15). The whole process of determining the correction factor $\mathcal{C}$ is summarized in Algorithm 2.

3.4. NOISE PSD ESTIMATORS IN THE CONTEXT OF ADAPTIVE SMOOTHING

In this section, we consider the noise PSD estimators [70], [81] in the context of adaptive smoothing. For this, we introduce the employed signal model in a speech enhancement context and illustrate the relationship between the model components and the quantities of adaptive smoothing in (3.1). After that, a brief overview over the considered noise PSD estimators is given.

3.4.1. Signal Model

The considered smoothing functions are employed in noise PSD estimators that operate in the STFT domain as detailed in Section 2.1.1. We follow the common assumption that the periodogram of the noisy input $|Y_{k,\ell}|^2$ follows an exponential distribution which is given by

$$f(|Y_{k,\ell}|^2) = \begin{cases} (1/\Lambda^y_{k,\ell}) \exp\left(-|Y_{k,\ell}|^2/\Lambda^y_{k,\ell}\right), & \text{if } |Y_{k,\ell}|^2 \geq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{3.16}$$

where $\Lambda^y_{k,\ell} = \mathbb{E}\{|Y_{k,\ell}|^2\}$. This model strictly holds if the speech coefficients $S_{k,\ell}$ and the noise coefficients $N_{k,\ell}$ follow circular-complex Gaussian distributions as given in Section 2.1.2. The considered noise PSD estimators [70], [81] are based on an adaptive recursive smoothing of the noisy periodogram such that the input to the recursive smoother y_ℓ in Fig. 3.1 is given by $|Y_{k,\ell}|^2$ while the output $\overline{y}_\ell$ resembles an estimate of the noise PSD $\hat{\Lambda}^n_{k,\ell}$.

3.4.2. Two Different Smoothing Factors Based on Thresholding

In [81, Section 14.1.3], a simple approach for estimating the background noise PSD from a noisy periodogram has been proposed. Based on a threshold value, one out of two fixed smoothing constants is selected. A larger smoothing constant is used if the input periodogram is larger than the noise PSD which has been estimated for the previous segment. For the other case, a smaller smoothing constant is used. In other words, the tracking speed is reduced if the *a posteriori* SNR $\gamma_{k,\ell}$ is larger than one. The goal is to reduce the speech leakage if the speech signal is likely to be present. The adaptive smoothing function is given by

$$\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1}) = \begin{cases} \alpha^\uparrow, & \text{if } y_\ell/\overline{y}_{\ell-1} > 1 \\ \alpha^\downarrow, & \text{otherwise.} \end{cases} \tag{3.17}$$

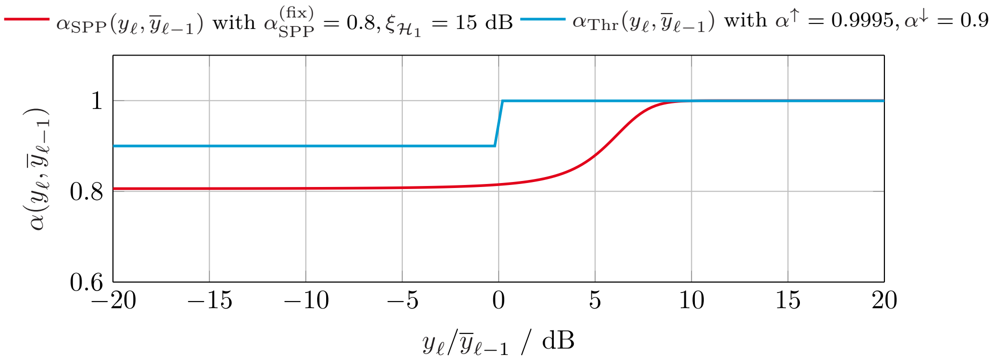

Fig. 3.2.: Value of the adaptive smoothing factors $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$, [81, Section 14.1.3] and $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$, [70] as functions of the *a posteriori* SNR $y_\ell/\overline{y}_{\ell-1} = |Y_{k,\ell}|^2/\hat{\Lambda}^n_{k,\ell-1} = \gamma_{k,\ell}$.

Both, $\alpha^{\uparrow}$ and $\alpha^{\downarrow}$ are fixed smoothing constants chosen between zero and one where $\alpha^{\uparrow}$ is chosen larger than $\alpha^{\downarrow}$.

Under the assumption that y_ℓ is exponentially distributed, the analytic solution to (3.10) is given in Appendix A.1. For the self-similarity optimization described in Section 3.3, analytic solutions to the inverse function $\mathcal{F}^{-1}(\cdot)$ and the derivative $\mathcal{F}'(\cdot)$ are given in Appendix A.2. A sketch of $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$ is given in Figure 3.2 for the proposed parameter values $\alpha^{\uparrow} = 0.9995$ and $\alpha^{\downarrow} = 0.9$ given in [81].

3.4.3. Speech Presence Probability Based Noise PSD Estimation

The noise PSD estimator described in [70] employs an estimate of the SPP to avoid speech leakage. Even though the noise PSD estimator has not been explicitly derived as an adaptive smoothing factor, we show here that the algorithm can be rewritten as such a function. The algorithm has been described in Section 2.1.4 and the main part of the algorithm is described by (2.22), (2.23) and (2.24). By combining (2.22), (2.23) and (2.24), the SPP-based noise PSD estimator can be described as an adaptive smoothing function:

$$\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1}) = \alpha_{\mathrm{SPP}}^{(\mathrm{fix})} + \frac{1 - \alpha_{\mathrm{SPP}}^{(\mathrm{fix})}}{1 + (1 + \xi_{\mathcal{H}_1}) e^{-y_\ell \xi_{\mathcal{H}_1}/[\overline{y}_{\ell-1}(1+\xi_{\mathcal{H}_1})]}}. \tag{3.18}$$

We omit the heuristic that has been proposed in [70] to avoid stagnation of the estimation if the noise PSD is underestimated. This is done because this procedure is not trivial to include in the derivations and its effect on the bias can be considered negligible because it is only activated in degenerate cases. The behavior of the adaptive smoothing function is similar to the one proposed by [81, Section 14.1.3] in that the function approaches one for large *a posteriori* SNRs and is close to the fixed smoothing constant $\alpha_{\mathrm{SPP}}^{(\mathrm{fix})}$ if the *a posteriori* SNR is close to zero.

Again, analytic solutions are given in Appendix A.1 and Appendix A.2 for the iterative estimation method and the self-similarity optimization, respectively. The function $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$ is sketched in Figure 3.2 where the proposed parameter values $\alpha_{\mathrm{SPP}}^{(\mathrm{fix})} = 0.8$ and $\xi_{\mathcal{H}_1} = 15$ dB from [70] have been employed.

3.5. BIAS COMPENSATION FOR NOISE PSD ESTIMATION

In Section 3.1.3, a bias compensation method has been presented that can be employed to compensate for the bias caused by adaptive smoothing. However, the composition of the input signal y_ℓ, i.e., whether it contains speech, noise or both, is not taken into consideration. Hence, regarding the application of noise PSD estimation, this correction may overcompensate for the bias in speech presence. To prevent such overcompensations, a time-varying correction factor is derived in this section.

For noise PSD estimation, the input signal comprises two components, namely speech and noise. If speech is present and assumed to be uncorrelated to the noise component, the expected value $\mathbb{E}\{y_\ell\}$ is equal to the sum of the speech PSD $\Lambda_{k,\ell}^{s}$ and the noise PSD $\Lambda_{k,\ell}^{n}$. If adaptive smoothing is employed on such a noisy signal, with (3.7) the mean of the filter output converges towards

$$\mathbb{E}\{\overline{y}_\ell\} = \frac{\mathbb{E}\{y_\ell\}}{\mathcal{C}} = \frac{\Lambda_{k,\ell}^{s} + \Lambda_{k,\ell}^{n}}{\mathcal{C}}. \tag{3.19}$$

Applying the fixed factor $\mathcal{C}$ removes the bias of the filter output mean $\mathbb{E}\{\overline{y}_\ell\}$ from the input mean $\mathbb{E}\{y_\ell\}$. As a consequence, the output converges towards the noisy PSD, i.e., $\Lambda_{k,\ell}^{s} + \Lambda_{k,\ell}^{n}$, but not towards the noise PSD. The rate of convergence depends on the additional inertia imposed by the adaptive smoothing factor $\alpha(y_\ell, \overline{y}_{\ell-1})$ which is increased in speech presence by the considered noise PSD estimators [70], [81]. Still, applying $\mathcal{C}$ directly may potentially overestimate the noise PSD $\hat{\Lambda}_{k,\ell}^{n}$ which, as a consequence, may result in speech distortions in the speech enhancement context. To take the speech energy into account, we propose to modify the correction such that the filter output is corrected towards the noise PSD $\Lambda_{k,\ell}^{n}$. For this, a time-varying correction term $\mathcal{G}_{k,\ell}$ is introduced which is set such that $\mathcal{G}_{k,\ell}\mathbb{E}\{\overline{y}_\ell\} = \Lambda_{k,\ell}^{n}$ holds. With (3.19), $\mathcal{G}_{k,\ell}$ can be derived as follows:

$$\mathcal{G}_{k,\ell}\mathbb{E}\{\overline{y}_\ell\} = \mathcal{G}_{k,\ell}\frac{\Lambda_{k,\ell}^{s} + \Lambda_{k,\ell}^{n}}{\mathcal{C}} \overset{!}{=} \Lambda_{k,\ell}^{n} \tag{3.20}$$

which can be rearranged to

$$\mathcal{G}_{k,\ell} = \mathcal{C}\frac{\Lambda_{k,\ell}^{n}}{\Lambda_{k,\ell}^{s} + \Lambda_{k,\ell}^{n}}. \tag{3.21}$$

Algorithm 3 Proposed algorithm for the bias compensation which is aware of the speech level.

```
$\mathcal{C}$ is obtained using Algorithm 1 or Algorithm 2.
Initialize algorithm: $\overline{y}_0 \leftarrow y_0$.
Compensate bias: $\check{\overline{y}}_0 \leftarrow \mathcal{G}_{k,\ell}\overline{y}_0$.
for all remaining observations $y_\ell$ do
    Perform smoothing:
    $\overline{y}_\ell = [1 - \alpha(y_\ell, \overline{y}_{\ell-1})]y_\ell + \alpha(y_\ell, \overline{y}_{\ell-1})\overline{y}_{\ell-1}$.
    Compensate bias: $\check{\overline{y}}_\ell = \mathcal{G}_{k,\ell}\overline{y}_\ell$.
end for
```

The time-varying term $\mathcal{G}_{k,\ell}$ can be split into the fixed correction factor $\mathcal{C}$ and a Wiener-like term $\Lambda^n_{k,\ell}/(\Lambda^s_{k,\ell} + \Lambda^n_{k,\ell})$. Consequently, the fixed correction factor $\mathcal{C}$ is reduced such that overestimations in speech presence are avoided. In Section 3.6, we discuss how the speech and the noise PSD in (3.21) can be estimated in practical applications.

The proposed bias correction is summarized in Algorithm 3 where $\check{\overline{y}}_\ell$ denotes the corrected filter output. The additional computational complexity of the proposed correction is given by the computation of the complete correction term $\mathcal{G}_{k,\ell}$ and its application. As discussed in Section 3.1.3, it is possible to determine the factor $\mathcal{C}$ before the processing starts. Thus, the additional computational cost for the bias correction can be considered low.

3.6. EVALUATION

In the first part of this section, we verify that the fixed correction factor $\mathcal{C}$ estimated with the methods described in Section 3.2 and Section 3.3 matches the true underlying bias. For this, we use Monte-Carlo simulations where the input signal consists of artificially generated uncorrelated noise samples that follow an exponential distribution. These experiments also give insights into how large the bias in the considered noise PSD estimators is. Further, we also include an analysis on how signal correlations affect the bias.

In the second part of this section, the behavior of adaptive smoothing is analyzed in a speech enhancement context using real world signals. We show that correcting the bias leads to an improved estimation of the noise PSD in terms of the log-error distortion measure [231] and also in an improved or similar speech quality as predicted by PESQ [230].

Within our evaluation, the noise PSD estimators given in (3.17) and (3.18) are used. In the evaluation, we mainly focus on the default parameters which were proposed in the literature [70], [81]. In accordance with [81, Section 14.1.3], $\alpha^{\uparrow}$ and $\alpha^{\downarrow}$ are set to 0.9995 and 0.9, respectively in (3.17). In accordance with [70], for the SPP-based noise estimator, $\xi_{\mathcal{H}_1}$ is set to 15 dB while a value of 0.8 is used for the fixed smoothing constant $\alpha^{(\text{fix})}_{\text{SPP}}$ in (3.18).

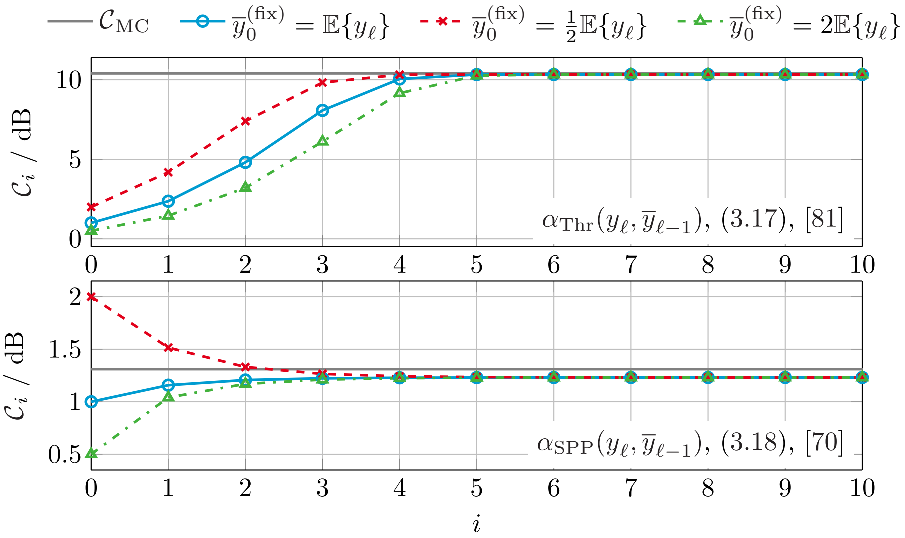

Fig. 3.3.: Bias correction factor $\mathcal{C}_i = \mathbb{E}\{y_\ell\}/\overline{y}_i^{(\mathrm{fix})}$ computed for each iteration step in Algorithm 1 given the adaptive smoothing functions used in [70], [81] and the true bias correction term $\mathcal{C}_{\mathrm{MC}}$ obtained from Monte-Carlo simulations with 10^6 realizations.

3.6.1. Verification of the Bias Estimation Methods

Here, we analyze how well the methods proposed in Section 3.2 and Section 3.3 determine the bias. To obtain the ground-truth, we use Monte-Carlo simulations. For this, 10^6 random numbers y_ℓ are generated that are independently sampled and follow an exponential distribution (3.16) with fixed parameter $\Lambda^y_{k,\ell} = \mathbb{E}\{y_\ell\}$. The generated random numbers are employed as the input signal of the respective adaptive smoothing filters. As the evaluated algorithms preserve the ergodicity and stationarity of the filter input, the expected value $\mathbb{E}\{\overline{y}_\ell\}$ can be estimated by computing the temporal average of the filter output. With this, a Monte-Carlo estimate $\mathcal{C}_{\mathrm{MC}}$ of the fixed correction factor $\mathcal{C} = \mathbb{E}\{y_\ell\}/\mathbb{E}\{\overline{y}_\ell\}$ is obtained.

First, the iterative procedure described in Section 3.2 is covered. Fig. 3.3 shows the estimated fixed correction factor $\mathcal{C}_i = \mathbb{E}\{y_\ell\}/\overline{y}_i^{(\mathrm{fix})}$, i.e., the outcome for each iteration step of Algorithm 1. The initial $\overline{y}_0^{(\mathrm{fix})}$ is set to three different values to show that the iteration converges to the same value. Additionally, the true correction factor obtained from Monte-Carlo simulations is included. The results show that the iteration converges for all considered smoothing functions after 10 to 15 steps and that the value obtained after convergence is independent of the initial condition $\overline{y}_0^{(\mathrm{fix})}$. For the parameters of the adaptive smoothing functions given in [70], [81], the iteratively determined bias corresponds well with the Monte-Carlo simulations. For the smoothing $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$ proposed in [81, Section 14.1.3], the iteratively determined fixed correction factor $\mathcal{C}$ is nearly identical to the ground truth obtained from Monte-Carlo simulations while for the SPP-based smoothing, i.e., $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$, the bias is underestimated by 0.27 dB (see Table 3.1). This deviation from the correct result is because $\overline{y}_{\ell-1}$ is replaced by a fixed constant $\overline{y}^{(\mathrm{fix})}$, and is not considered as a random variable.

The second method proposed for estimating the bias is described in Section 3.3. Here, a model PDF $\tilde{f}(\overline{y}_\ell|\boldsymbol{\theta})$ is required for the optimization. It is known that after recursive smoothing, an exponentially distributed random process approximately follows a χ^2 distribution with an increased shape parameter [66], [83]. The shape of the resulting PDF can also be approximated by a generalized gamma distribution or a log-normal distribution. In our experiments, we obtained the best results using the log-normal distribution which is consequently employed in the evaluations. The PDF is given by

$$f(x) = \frac{1}{x\sqrt{2\pi\lambda_{\log(\mathcal{N})}}} \exp\left(-\frac{(\log(x) - \mu_{\log(\mathcal{N})})^2}{2\lambda_{\log(\mathcal{N})}}\right). \tag{3.22}$$

It assumes that the PDF that results after taking the logarithm of the random variable y is a normal distribution. Consequently, $\mu_{\log(\mathcal{N})}$ and $\lambda_{\log(\mathcal{N})}$ denote the mean and the variance of the normal distribution in the logarithmic domain, respectively. The mean of this distribution can be computed using its parameters as

$$m(\mu_{\log(\mathcal{N})}, \lambda_{\log(\mathcal{N})}) = \exp\left(\mu_{\log(\mathcal{N})} + \frac{\lambda_{\log(\mathcal{N})}}{2}\right). \tag{3.23}$$

For the minimization of the cost function given in (3.14), we use the downhill simplex method proposed by [236].

Fig. 3.4 shows the PDF of the model $\tilde{f}(\overline{y}_\ell|\hat{\boldsymbol{\theta}})$ with optimized parameters $\hat{\boldsymbol{\theta}}$, which are determined using the method described in Section 3.3, and the PDF $\tilde{g}(\overline{y}_\ell|\hat{\boldsymbol{\theta}})$ which is the PDF that results after computing (3.11) with the optimized parameters $\hat{\boldsymbol{\theta}}$. Finally, the plots also include an estimate of the true PDF of the filter output that has been estimated from Monte-Carlo simulations. Though slight deviations between the true PDF and the optimized log-normal distribution can be observed, the optimized model PDF $\tilde{f}(\overline{y}_\ell|\hat{\boldsymbol{\theta}})$ approximates the distribution of the filter output reasonably well. Furthermore, Fig. 3.4 shows that the optimized model distribution $\tilde{f}(\overline{y}_\ell|\hat{\boldsymbol{\theta}})$ and the marginalized PDF $\tilde{g}(\overline{y}_\ell|\hat{\boldsymbol{\theta}})$ are nearly identical from which we conclude that our approach to finding the bias in Section 3.3 is reasonable.

In Table 3.1, the Monte-Carlo ground-truth of the correction factor $\mathcal{C}$ is given along with the estimates of the iterative method of Section 3.2 and the self-similarity optimization of Section 3.3. It can be seen that the self-similarity optimization of Section 3.3 outperforms the iterative method of Section 3.2. Using the self-similarity optimization, for the smoothing with $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$ the ground-truth is matched, for the SPP-based smoothing the difference to the ground-truth is only 0.07 dB.

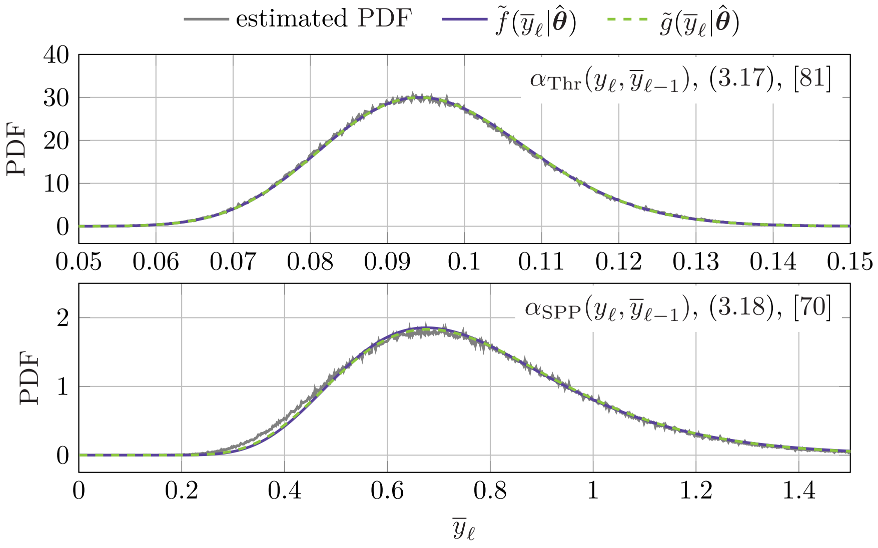

Fig. 3.4.: Shape of the fitted model distribution $\tilde{f}(\overline{y}_\ell|\hat{\boldsymbol{\theta}})$, the marginalized distribution $\tilde{g}(\overline{y}_\ell|\hat{\boldsymbol{\theta}})$ obtained by using the optimized model in (3.11), and the true PDF of the filter output obtained from Monte-Carlo simulations with 10^6 samples for the adaptive smoothing factors used in [70], [81].

Table 3.1.: Correction factor $\mathcal{C} = \mathbb{E}\{y_\ell\}/\mathbb{E}\{\overline{y}_\ell\}$ for the adaptive smoothing functions in (3.17) and (3.18) without replacement of $\overline{y}_{\ell-1}$.

Smoothing factor	Monte-Carlo	Sec. 3.2 / Alg. 1	Sec. 3.3 / Alg. 2
α_{Thr}, (3.17), [81]	10.18 dB	10.14 dB	10.18 dB
α_{SPP}, (3.18), [70]	1.17 dB	0.90 dB	1.10 dB

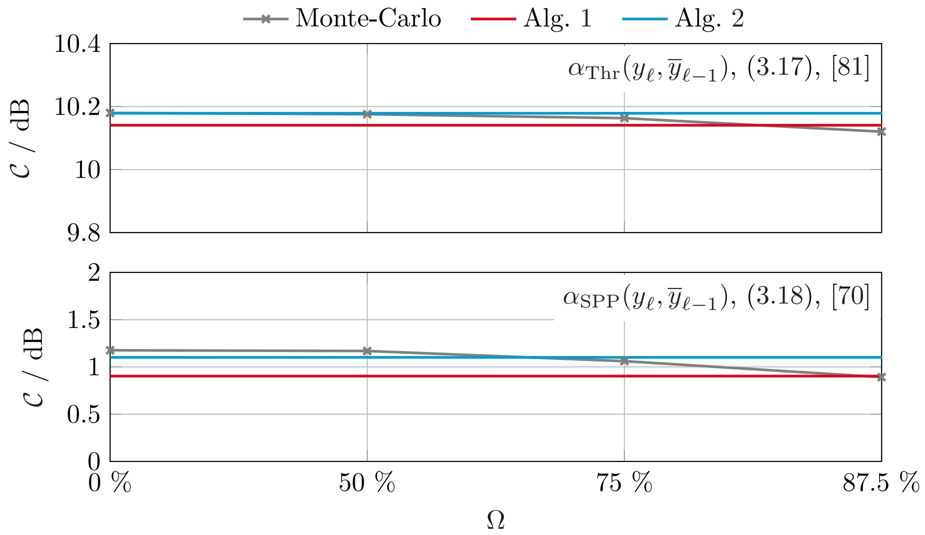

Fig. 3.5.: Correction factor $\mathcal{C}$ determined using Monte-Carlo simulations on white noise for different overlaps Ω in the STFT domain and the adaptive smoothing functions used in [70], [81]. Additionally shown: the correction factors reported in Table 3.1.

Also note that the bias obtained for the SPP-based estimation method is only 1.17 dB and, thus, rather small. In contrast, the method in [81, Section 14.1.3] yields a bias of 10.2 dB which is rather large. The reason for this appears to be the choice of the parameter $\alpha^{\uparrow}$. As it is very close to one, the adaptive smoothing is forced to considerably smaller values resulting in the observed bias. Further, this result only covers the case where only noise is present. In the presence of speech, the underestimation is less severe as shown in Section 3.6.2.

From further experiments we conclude that our proposed algorithms 1 and 2 work also well for other choices of the parameters $\alpha^{\uparrow}$, $\alpha^{\downarrow}$, $\alpha_{\mathrm{SPP}}^{(\mathrm{fix})}$, and $\xi_{\mathcal{H}_1}$. Considering $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$ and both algorithms, the deviation of the estimated bias from the true bias is smaller than 1 dB for a wide range of combinations of $\alpha^{\uparrow}$ and $\alpha^{\downarrow}$. Determining the bias for $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$ is, however, more challenging. Still, a low deviation of 1 dB from the true mean or less is obtained for the parameter ranges $0.4 \leq \alpha_{\mathrm{SPP}}^{(\mathrm{fix})} \leq 0.9$ and $7.5\ \mathrm{dB} \leq \xi_{\mathcal{H}_1} \leq 20\ \mathrm{dB}$ for Algorithm 1 and Algorithm 2. In general, the estimation method proposed in Algorithm 2 has the potential to estimate the bias with very high accuracy as no approximations were used in the derivations. For the practical application, however, an appropriate model PDF $\tilde{f}(\overline{y}_\ell|\boldsymbol{\theta})$ has to be employed and the numerical optimization may converge to local optima leading to unsatisfactory results. In contrast to that, the estimation method in Algorithm 1 is more robust but results only in approximate estimates of the bias due to the used approximations used in the derivation.

Finally, we analyze how the fixed correction factor $\mathcal{C}$ is influenced if the samples of the input signal y_ℓ are correlated over time, e.g., due to the overlap in the STFT framework. For this, Monte-Carlo simulations are employed again. Also here, $\mathbb{E}\{\overline{y}_\ell\}$ can be estimated using temporal averaging, as no further restrictions have to be imposed on the random process except for ergodicity which is also fulfilled for correlated input samples. Under the assumption that the sampling rate is 16 kHz, we generate a white Gaussian noise signal with a length of 360 s in the time-domain. After that, we transform the signal to the STFT domain where a Hann-window is employed. The segment and window lengths are set to 32 ms. These STFT parameters are chosen because they allow the results to be easily related to typical single-channel speech enhancement frameworks, e.g., [12], [29], [70]. For this experimental design, the results are also valid if shorter or longer window lengths are used or a different underlying sampling rate is assumed.

To obtain Fourier coefficients with different degrees of correlation, we vary the overlap Ω of the STFT analysis segments, where $\Omega = (\text{segment length} - \text{segment shift})/\text{segment length}$. The adaptive smoothing functions are applied to the magnitude squared coefficients in each frequency band which can be assumed to follow an exponential distribution (3.16). Finally, the mean over all time-frequency points is computed, where the 0 Hz bin and the Nyquist bin are omitted because the assumption that the coefficients follow an exponential distribution is not fulfilled here. Additionally, we leave out the first 500 segments to account for the adaptation of the adaptive smoothing filters. In Fig. 3.5, we show the fixed correction factor $\mathcal{C}$ as a function of the overlap Ω. In general, it is observed that the bias becomes smaller with increasing overlap — and, thus, also with an increasing amount of correlation. For $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$, the bias is reduced by 0.06 dB in absolute value if the overlap is increased from 0 % to 87.5 %. Correspondingly, the correlation has a negligible influence on the absolute bias of 10.2 dB. For the SPP-based smoothing $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$, the bias is reduced by 0.29 dB for the same increase of correlation. As the absolute bias for $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$ is with 1.2 dB much smaller than the bias of $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$, this difference indicates that the influence of the correlation is much stronger here. Thus, the higher overlap leads to a notable reduction of the absolute bias. However, for the typical choice of 50 % overlap, the bias hardly changes. As a consequence, the proposed correction methods are directly applicable in practice.

3.6.2. Applications to Speech-Enhancement

In this section, we consider the practical implications of the bias caused by adaptive smoothing for noise PSD estimation in a speech enhancement framework. We show that the logarithmic estimation error [231] between the true and the resulting noise PSD is reduced if the bias is corrected. Additionally, we use PESQ scores [230] to give an instrumental prediction of the change in signal quality. Even though PESQ has been developed for the evaluation of speech coding algorithms, it has been shown that it also correlates with the quality of enhanced speech [238]. We show that the log-error distortion and also PESQ scores can be improved for the noise PSD estimators proposed in [70], [81]. For the log-error distortion, we additionally consider a special case where noise only signals are used as input.

For the evaluation, we employ a variety of synthetic and natural noise types. Among these noise types are a pink and a babble noise taken from the Noisex-92 database [239]. Additionally, a traffic noise is employed which comprises an acoustic scene with passing cars. For the experiments that include speech, we use 1120 sentences from the TIMIT corpus [240]. The sentences are corrupted at SNRs ranging from −10 dB to 30 dB in 5 dB steps. Each sentence is embedded in a different segment of the respective background noise. All signals have a sampling rate of 16 kHz.

The speech enhancement framework, in which the considered noise PSD estimators [70], [81] are embedded, operates in the STFT domain as in Section 2.1.1. For this, a segment length of 32 ms with 50 % overlap is used. This parameter combination is often used for speech enhancement, e.g., [12], [29], [70], as speech signals are assumed to be stationary only for a short time period similar to the chosen segment length [44, Section 5.10]. Further, a square-root Hann window is employed for spectral analysis. For estimating the *a priori* SNR, the decision-directed approach (2.27) on page 36 with a smoothing factor of 0.98 is used [12]. The clean speech signal is estimated using the Wiener filter where a lower limit of −12 dB is enforced. For resynthesizing the signal, again, a square-root Hann window is employed.

The time-varying correction term $\mathcal{G}_{k,\ell}$ has to be determined at the beginning of a new segment ℓ. At this point, there is no updated estimate of the speech PSD $\hat{\Lambda}^s_{k,\ell}$ and the noise PSD $\hat{\Lambda}^n_{k,\ell}$ available. As the noise PSD is the first quantity estimated from a new noisy observation, we propose to use the estimates from the previous segment. Correspondingly, $\hat{\Lambda}^n_{k,\ell-1}$ is used instead of $\Lambda^n_{k,\ell}$ in (3.21) for the practical evaluation. Further, a slightly modified version of the decision-directed approach (2.27) is used to obtain an estimate of the clean speech

$$\hat{\xi}_{k,\ell} = \alpha_{\mathrm{DD}} \frac{|\hat{S}_{k,\ell-1}|^2}{\hat{\Lambda}^n_{k,\ell-1}} + (1 - \alpha_{\mathrm{DD}}) \max\left(\frac{|Y_{k,\ell}|^2}{\hat{\Lambda}^n_{k,\ell-1}}, 0 \right). \tag{3.24}$$

The main difference to (2.27) lies in the second term on the right hand side, where the noise PSD estimate of the previous segment $\hat{\Lambda}^n_{k,\ell-1}$ is used for normalization. The modified estimate of the *a priori* SNR $\hat{\xi}_{k,\ell}$ from (3.24) is used to obtain an estimate of the speech PSD by multiplying with $\hat{\Lambda}^n_{k,\ell-1}$. In the evaluation, we consider only the correction parameters obtained by Algorithm 2 as both methods yield similar values for $\mathcal{C}$ such that for the considered practical application very similar outcomes would be obtained. We use the values for $\mathcal{C}$ obtained using Algorithm 2 as it performs slightly better than Algorithm 1. To avoid stagnations of the noise PSD estimation which may be caused by the time-varying correction factor $\mathcal{G}_{k,\ell}$, we apply a lower limit to $\mathcal{G}_{k,\ell}$ which is set to −20 dB.

As described in Section 2.2.1, we use a separated version of the log-error distortion which is computed for each speech signal. Here, only segments after a five seconds initialization

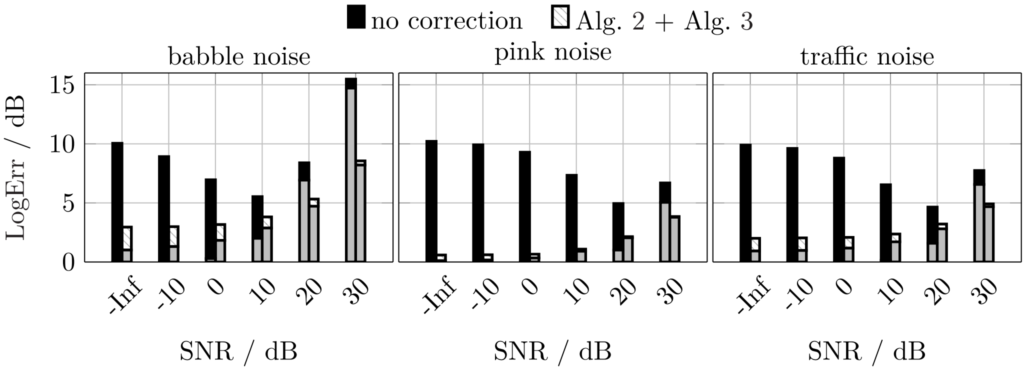

Fig. 3.6.: Log-error distortion of the adaptive smoothing function $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$ described in (3.17), [81, Section 14.1.3] with and without the proposed correction method for speech in noise at different SNRs. The lower part (gray) of the bars represents the overestimation $\mathrm{LogErr}_\uparrow$, whereas the upper part (black / white) is the underestimation $\mathrm{LogErr}_\downarrow$.

period, which only includes noise, are considered in the evaluation. During this initialization phase, the noise PSD estimators can adapt to the background noise. The goal is to exclude initialization artifacts from the evaluation which may result in an erroneous estimate of the performance. Even though in real applications, such an initialization period is not available, this poses only a minor problem as the algorithms recover from an erroneous initializations after a short processing time, e.g., during speech pauses. As the correction factors were determined based on the assumption that the periodogram is exponentially distributed, we exclude the coefficient at 0 Hz and the Nyquist frequency also here. The measure is computed for each speech signal separately and averaged over all speech signals afterwards. For the noise only case, the log-error distortion is computed using a long excerpt of about four minutes from the respective noise signal.

The results for the two noise PSD estimators are shown in Fig. 3.6 and Fig. 3.7, respectively. Here, an SNR of -Inf denotes the noise only case. For the adaptive smoothing function $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$ proposed in [81, Section 14.1.3], the results in Fig. 3.6 show that the uncorrected version of the noise PSD estimator tends to underestimate the background noise PSD in low SNR regions while it overestimates the noise PSD for high SNRs. The observed overestimation at high SNRs is caused by the fact that this estimator always allows the input periodogram to be tracked, albeit slowly, even if the *a posteriori* SNR is high. Thus, the speech leakage, which is reflected in the overestimation, increases with increasing SNR. The underestimation at low SNRs is mainly caused by the adaptive smoothing. If the proposed correction is applied, the noise PSD log-error distortion can be considerably reduced for all considered SNRs and noise types. As the fixed correction factor $\mathcal{C}$ required for this noise PSD estimator is rather large, the total estimation error

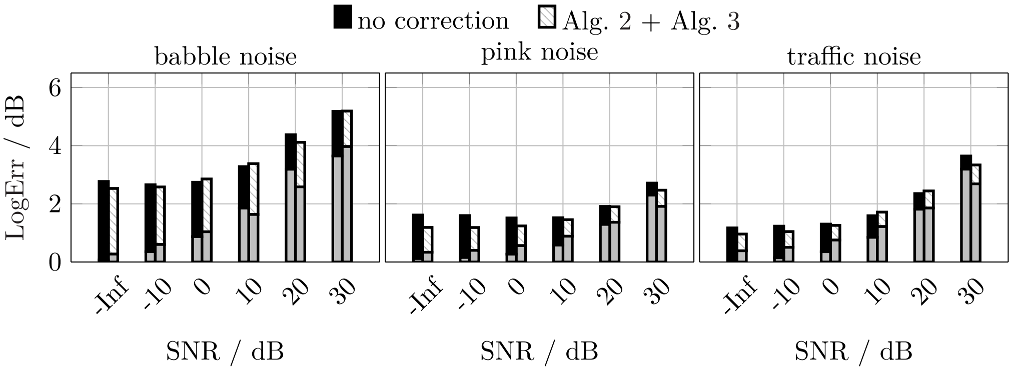

Fig. 3.7.: Same as Fig. 3.6, but for the adaptive smoothing function $\alpha_{\text{SPP}}(y_\ell, \overline{y}_{\ell-1})$ described in (3.18), [70].

is often dominated by the overestimation if the correction is applied. The total log-error distortion, however, is in general smaller. Especially, if either noise or speech is dominant, i.e., for low SNRs and high SNRs, lower estimation errors are obtained.

Similar tendencies are also observed for the SPP-based noise estimator $\alpha_{\text{SPP}}(y_\ell, \overline{y}_{\ell-1})$ as shown in Fig. 3.7. For both cases, i.e., with and without correction, the overestimation increases also for this noise PSD estimator with increasing SNR. For an SNR range around 0 dB and 10 dB, the proposed correction increases the log-error distortion slightly. For high SNRs and low SNRs, however, a slight reduction of the log-error distortion is observed. In general, the benefits of the correction are expected to be smaller as the bias of this algorithm is rather low as shown in Table 3.1.

Fig. 3.8 shows the PESQ improvement scores which are obtained if the considered adaptive smoothing functions are used as noise PSD estimators in a simple enhancement scheme. Again, the adaptive smoothing functions are employed with and without correction to show the change in performance. For $\alpha_{\text{SPP}}(y_\ell, \overline{y}_{\ell-1})$, the corrected and the uncorrected version of the noise PSD lead to virtually the same result. In general, the measure indicates a slight reduction of the quality if the proposed correction is applied. Considering the log-error distortions in Fig. 3.7, the result is not unexpected as the differences between the corrected and uncorrected version are small. Contrarily, the PESQ scores can be considerably improved for the smoothing function $\alpha_{\text{Thr}}(y_\ell, \overline{y}_{\ell-1})$, [81, Section 14.1.3]. After applying the correction, the PESQ scores are increased by up to 0.2 points where the largest gains are obtained for SNRs between 0 dB and 10 dB. The predicted quality of the corrected version of $\alpha_{\text{Thr}}(y_\ell, \overline{y}_{\ell-1})$ is comparable to the SPP-based noise PSD estimator. These improvements can be attributed to the reduction of the strong underestimation in low SNR regions and the prevention of overestimation in speech presence. These results are also confirmed in informal listening tests.

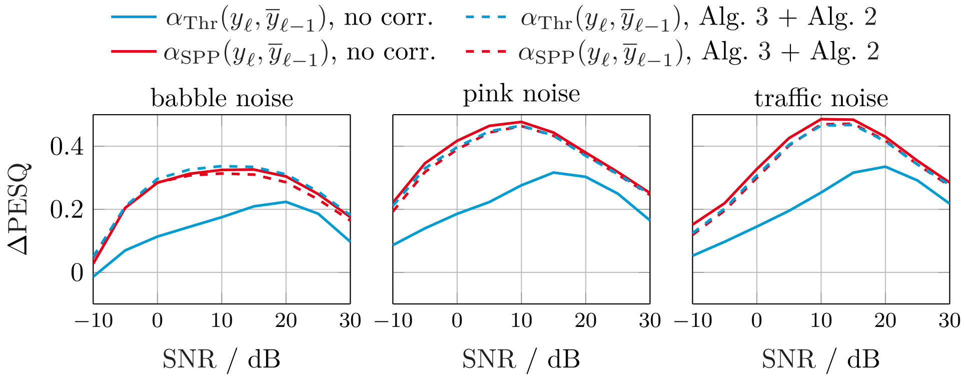

Fig. 3.8.: PESQ improvement scores for a simple speech enhancement framework where the adaptive smoothing functions $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$ and $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$ are used as noise PSD estimators with and without the correction proposed in Algorithm 3. The fixed correction factor $\mathcal{C}$ was estimated using Algorithm 2.

3.7. SUMMARY

In this chapter, we analyzed the bias of adaptive first-order recursive smoothing filters which play a central role, e.g., in the noise PSD estimators presented in [70], [81]. From our analysis, it followed that due to the used adaptive smoothing, both algorithms generally underestimate the noise PSD. We could show that the bias is scale-invariant and that the bias from the input signal mean $\mathbb{E}\{y_\ell\}$ caused by adaptive smoothing can be compensated using a single fixed correction factor $\mathcal{C}$. For the application of noise PSD estimation, we extended the correction method which resulted in a time-varying correction factor to avoid overestimation by accounting for the speech energy. This led to the proposed correction method shown in Algorithm 3. The fixed correction factor $\mathcal{C}$ can be determined using the proposed algorithms 1 and 2. Algorithm 1 employs an iterative method which is based on the analytically solvable case where the adaptive smoothing factor does not depend on the previous filter output $\overline{y}_{\ell-1}$. Algorithm 2 determines the factor $\mathcal{C}$ by maximizing the self-similarity of a model PDF with respect to the transition density $f(\overline{y}_\ell|\overline{y}_{\ell-1})$. In the evaluation, we could demonstrate that Algorithm 2 estimates the correction factor $\mathcal{C}$ with a higher accuracy than the iterative method, i.e., Algorithm 1. If the estimation error of the adaptive smoothing filter is sufficiently large, the proposed correction method yields considerable improvements in terms of the log-error distortion and PESQ.

CHAPTER 4

NON-ITERATIVE BIAS COMPENSATION FOR ADAPTIVE SMOOTHING BASED NOISE PSD ESTIMATION

In this chapter, another method is proposed to correct the bias of adaptive smoothing based noise PSD estimators. It is based on [217] and, similar to the methods in Chapter 3, makes use of a correction factor which is used to scale one of the quantities involved in the adaptive recursive smoothing. The correction method is however generally different from the correction method considered in Chapter 3 and results in a correction scheme which is not based on the scaling of the input or the output of the smoothing filter. As a result, also the required correction factor is different from the correction factor considered in Chapter 3. We present a method that allows the approximate determination of the alternative correction factor. Although the method is related to the iterative method presented in Section 3.2, we will show that the desired correction factor can be determined within a single iteration.

Further, the work in [217] is extended in this chapter. Similar to the correction method considered in Chapter 3, the alternative correction method originally proposed in [217] is not aware of the speech signal that is present in the noisy mixture. As the considered noise PSD estimators generally tend to underestimate the noise PSD, the estimation is corrected by increasing the power of the input signal. In speech presence, however, this may result in overestimations which possibly cause distortions of the speech signal. To resolve this issue, we present a method which allows the incorporation of the speech energy in the correction method similar to the approach in Section 3.5.

The methods proposed in this chapter are based on the theory about adaptive smoothing presented in Chapter 3. The contributions of this chapter are structured as follows: In Section 4.1, we start by recapitulating the alternative correction and methods to determine the corresponding correction factor previously presented in [217]. After this, the extensions that are required to include the speech energy in the estimation are considered in Section 4.2. In Section 4.3, the correction method proposed here is evaluated and compared against the correction method in Chapter 3 in terms of log-error distortion and PESQ improvement scores.

This chapter is partly based on:

[217] R. Rehr and T. Gerkmann, “Bias correction methods for adaptive recursive smoothing with applications in noise PSD estimation,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 206–210, © 2016 IEEE.

4.1. NON-ITERATIVE BIAS COMPENSATION

In this section, the alternative correction method is introduced mathematically. Further, we present a method to estimate the correction factor that is required to apply this method.

First, we recapitulate the correction method in Chapter 3 to highlight the differences to the correction method presented here. To correct the bias caused by adaptive smoothing, the input or the output of the adaptive smoothing filter is scaled by a correction factor in Chapter 3. For now, we consider the case that only a fixed correction factor is employed, i.e., $\mathcal{C}$ is used instead of $\mathcal{G}_{k,\ell}$. Although this does not allow the speech signal in the noisy mixture to be taken into account, it simplifies the following explanations. If the correction factor $\mathcal{C}$ is applied to the input y_ℓ, the adaptive smoothing filter in (3.1) changes to

$$\overline{y}_\ell = (1 - \alpha(\mathcal{C}y_\ell, \overline{y}_{\ell-1}))\mathcal{C}y_\ell + \alpha(\mathcal{C}y_\ell, \overline{y}_{\ell-1})\overline{y}_{\ell-1}, \tag{4.1}$$

which is obtained by replacing y_ℓ by $\mathcal{C}y_\ell$ in (3.1). The main difference of the alternative bias correction method in this chapter to the method in (4.1), i.e., in Chapter 3, is that the correction factor is only applied to the filter input y_ℓ that does not occur in the adaptive smoothing function $\alpha(y_\ell, \overline{y}_{\ell-1})$, i.e.,

$$\overline{y}_\ell = (1 - \alpha(y_\ell, \overline{y}_{\ell-1}))\mathcal{C}^{(\mathrm{a})}y_\ell + \alpha(y_\ell, \overline{y}_{\ell-1})\overline{y}_{\ell-1}. \tag{4.2}$$

Here, $\mathcal{C}^{(\mathrm{a})}$ denotes the correction factor required for the correction considered in this chapter. This factor is generally different from the correction factor $\mathcal{C}$ considered in Chapter 3 which is why a different symbol is employed. If $\mathcal{C}^{(\mathrm{a})}$ in (4.2) is set to the value that corrects the bias of the respective adaptive smoothing $\alpha(y_\ell, \overline{y}_{\ell-1})$, then only the corrected output is used to determine the adaptive smoothing factor while the input y_ℓ in $\alpha(y_\ell, \overline{y}_{\ell-1})$ remains unscaled. For adaptive smoothing functions that depend only on the ratio $y_\ell/\overline{y}_{\ell-1}$, such as noise PSD estimators, the correction factor does not cancel out in the computation of $\alpha(y_\ell, \overline{y}_{\ell-1})$ as in Algorithm 3. As a consequence, the smoothing factors change if this correction method is applied. Considering the SPP-based noise PSD estimator in [70], this also leads to a correction of the estimated SPP. Note that introducing the correction factor $\mathcal{C}^{(\mathrm{a})}$ into the adaptive smoothing equation as in (4.2) does not change the scale-invariance property. This means that scaling the filter input y_ℓ results in the same scaling of the filter output $\overline{y}_\ell$. This can be verified by performing the same induction steps given in Section 3.1.2 for (4.2).

In [217], a method has been proposed to determine the correction factor $\mathcal{C}^{(\mathrm{a})}$ which is recapitulated in the remainder of this section. It is based on the iterative method proposed in Section 3.2, [215]. As for the method in Section 3.2, it allows the approximative determination the correction factor $\mathcal{C}^{(\mathrm{a})}$. It has been derived in [217], by repeating the same steps as in Section 3.2, [215] for the modified filter function in (4.2). Accordingly, the expected value of the adaptive smoothing filter in (4.2) is considered without the dependence on the previous filter output $\overline{y}_{\ell-1}$ first. After this, the dependence on $\overline{y}_{\ell-1}$ is replaced by a deterministic value which is updated in an iterative procedure. In contrast to Section 3.2, we will show that the correction factor $\mathcal{C}^{(\mathrm{a})}$ is obtained after a single iteration, i.e., that no iteration is required.

The first step is to consider (4.2) without the dependence on the previous filter output $\overline{y}_{\ell-1}$. With that, the filter equation in (4.2) simplifies to

$$\overline{y}_\ell = [1-\alpha(y_\ell)]\,\mathcal{C}^{(\mathrm{a})} y_\ell + \alpha(y_\ell)\overline{y}_{\ell-1}. \tag{4.3}$$

For computing the expected value, we assume that all y_ℓ are identically distributed and uncorrelated. Additionally, we assume that the filter output $\overline{y}_\ell$ will remain stationary if the filter input y_ℓ is stationary. Experiments indicate that this property is sufficiently fulfilled. From this it follows that $\mathbb{E}\{\overline{y}_i\} = \mathbb{E}\{\overline{y}_j\}$ where $i \neq j$. With the first assumption, the expected value $\mathbb{E}\{y_\ell \overline{y}_{\ell-1}\}$ can be written as $\mathbb{E}\{y_\ell\}\mathbb{E}\{\overline{y}_{\ell-1}\}$. Thus, by applying $\mathbb{E}\{\cdot\}$ to (4.3) and rearranging the terms,

$$\mathbb{E}\{\overline{y}_\ell\} = \mathcal{C}^{(\mathrm{a})}\,\frac{\mathbb{E}\{y_\ell\} - \mathbb{E}\{y_\ell\alpha(y_\ell)\}}{1-\mathbb{E}\{\alpha(y_\ell)\}} \tag{4.4}$$

is obtained. It is very similar to the solution obtained in (3.9) of Section 3.2. The main difference is the additional multiplication with $\mathcal{C}^{(\mathrm{a})}$. The result given in (4.4) depends only on the adaptive function $\alpha(y_\ell)$ and the probability density function of y_ℓ.

The second step is to derive the iterative procedure to find an approximate estimate of the bias of the adaptive recursive smoothing for the case where the dependence on $\overline{y}_{\ell-1}$ is included. For this, the adaptive smoothing factors are simplified by replacing $\overline{y}_{\ell-1}$ by a fixed value $\overset{\circ}{\overline{y}}^{(\mathrm{fix})}$. Here, $\overset{\circ}{\overline{y}}^{(\mathrm{fix})}$ is used as symbol for the fixed value to distinguish it from the fixed values and iteration results used in Section 3.2. Then, the equation for the iteration is given by

$$\overset{\circ}{\overline{y}}{}_i^{(\mathrm{fix})} = \mathcal{C}^{(\mathrm{a})}\mathcal{Z}(\overset{\circ}{\overline{y}}{}_{i-1}^{(\mathrm{fix})}), \tag{4.5}$$

with

$$\mathcal{Z}(\overset{\circ}{\overline{y}}{}_i^{(\mathrm{fix})}) = \frac{\mathbb{E}\{y_\ell\} - \mathbb{E}\{y_\ell\alpha(y_\ell, \overset{\circ}{\overline{y}}{}_{i-1}^{(\mathrm{fix})})\}}{1-\mathbb{E}\{\alpha(y_\ell, \overset{\circ}{\overline{y}}{}_{i-1}^{(\mathrm{fix})})\}}. \tag{4.6}$$

Similar to Section 3.2, $\overset{\circ}{\overline{y}}{}_i^{(\mathrm{fix})}$ is the estimate of $\mathbb{E}\{\overline{y}_\ell\}$ obtained for the ith iteration step where the initial condition is denoted by $\overset{\circ}{\overline{y}}{}_0^{(\mathrm{fix})}$. For determining the final estimate of $\mathbb{E}\{\overline{y}_\ell\}$, in Section 3.2, [215], the iteration is continued until it converges. Here, however, the correction factor needs to be determined by choosing the constant $\mathcal{C}^{(\mathrm{a})}$ such that the iteration converges to $\mathbb{E}\{y_\ell\}$ to obtain an unbiased estimate. To achieve this, $\overset{\circ}{\overline{y}}{}_0^{(\mathrm{fix})}$ is set to the mean of the input signal $\mathbb{E}\{y_\ell\}$. With that, the correction factor $\mathcal{C}^{(\mathrm{a})}$ is determined as

$$\mathcal{C}^{(\mathrm{a})} = \frac{\mathbb{E}\{y_\ell\}}{\mathcal{Z}(\mathbb{E}\{y_\ell\})}, \tag{4.7}$$

which is obtained by solving the following expression for $\mathcal{C}^{(\mathrm{a})}$

$$\overset{\circ}{\overline{y}}{}_1^{(\mathrm{fix})} = \mathcal{C}^{(\mathrm{a})}\mathcal{Z}(\mathbb{E}\{y_\ell\}) \overset{!}{=} \mathbb{E}\{y_\ell\}. \tag{4.8}$$

Using the resulting $\mathcal{C}^{(\mathrm{a})}$ and $\overset{\circ}{\overline{y}}{}_0^{(\mathrm{fix})} = \mathbb{E}\{y_\ell\}$ in (4.5), it can be seen that this correction factor enforces the same result, namely $\mathbb{E}\{y_\ell\}$, for each iteration step $\overset{\circ}{\overline{y}}{}_i^{(\mathrm{fix})}$. Similar to the iteration in Section 3.2, [215], it can be shown experimentally that for other initializations $\overset{\circ}{\overline{y}}{}_0^{(\mathrm{fix})}$ the iteration in (4.5) converges to the same value, i.e., $\mathbb{E}\{y_\ell\}$. This indicates that the determined value for $\mathcal{C}^{(\mathrm{a})}$, which can be obtained without an iteration, compensates the bias. Due to the scale-invariance, this procedure leads to the same $\mathcal{C}^{(\mathrm{a})}$ for any given mean $\mathbb{E}\{y_\ell\}$. A convenient choice is to use $\mathbb{E}\{y_\ell\} = \overset{\circ}{\overline{y}}{}_0^{(\mathrm{fix})} = 1$, but the same correction factor could as well be obtained by using, e.g., $\mathbb{E}\{y_\ell\} = \overset{\circ}{\overline{y}}{}_0^{(\mathrm{fix})} = 5$.

4.2. BIAS COMPENSATION FOR NOISE PSD ESTIMATION

In this chapter, the bias compensation is, again, mainly used for noise PSD estimators. Thus, the relationship between noise PSD estimators and adaptive smoothing are recapitulated in this section. Further, a method is presented that allows the incorporation of the speech energy in the estimation of the correction factor.

Similar as in Section 3.4, the filter input y_ℓ is again considered as the noisy periodogram $|Y_{k,\ell}|^2$. As in (3.16) of Section 3.4, it is assumed here that $|Y_{k,\ell}|^2$ follows an exponential distribution with mean $\Lambda^y_{k,\ell} = \Lambda^s_{k,\ell} + \Lambda^n_{k,\ell}$. The filter output $\overline{y}_\ell$ corresponds to the estimated noise PSD $\hat{\Lambda}^n_{k,\ell}$. Further, the noise PSD estimators that have been described in Section 3.4 are considered again as adaptive smoothing functions. Due to the relationship between the estimation methods proposed in Section 3.2 and Section 4.1, the analytic solutions in Appendix A.1 can also be used for computing the expected value in (4.6).

The bias compensation method presented in Section 4.1 allows the removal of the bias induced by adaptive smoothing. However, the approach is not aware of the energy of additional signals in the input mixture, e.g., the speech signal. Similar to the reasons given in Section 3.2, using a fixed correction factor $\mathcal{C}^{(\mathrm{a})}$ without further considerations of the speech signal may overcompensate the bias in speech presence. In the context of noise PSD estimation, this is very important as overestimations of the background noise may result in severe distortions of the speech signal. In the remainder of this section, we show how to turn the fixed correction factor $\mathcal{C}^{(\mathrm{a})}$ into a time-variant correction that incorporates the speech energy in the signal.

If speech is present, the expected value of the input is $\mathbb{E}\{y_\ell\} = \Lambda^y_{k,\ell} = \Lambda^s_{k,\ell} + \Lambda^n_{k,\ell}$. In this case, using the correction factor given in (4.7), will correct the filter output such that $\mathbb{E}\{\overline{y}_\ell\} = \Lambda^s_{k,\ell} + \Lambda^n_{k,\ell}$. However, the goal of noise PSD estimators is to obtain an estimate of the noise PSD, i.e., $\mathbb{E}\{\overline{y}_\ell\} \overset{!}{=} \Lambda^n_{k,\ell}$. The corresponding correction factor can be obtained

Algorithm 4 Estimation of the time-varying correction factor $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$ for the alternative correction method considered in Chapter 4.

```
$\overset{\circ}{\bar{y}}_0^{(\mathrm{fix})} \leftarrow \Lambda_{k,\ell}^n$, $\Lambda_{k,\ell}^y \leftarrow \Lambda_{k,\ell}^s + \Lambda_{k,\ell}^n$.
Obtain $\mathcal{Z}\{\overset{\circ}{\bar{y}}_0^{(\mathrm{fix})}\}$ given in (4.6). The solutions for the adaptive functions in [70], [81]
    are given in (A.1) and (A.2).
Compute the compensation factor $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$ using (4.10).
```

by small modifications to (4.8) and solving for $\mathcal{C}^{(\mathrm{a})}$ again. For this, the iteration in (4.5) is used again as a starting point, but the initialization is set to $\overset{\circ}{\bar{y}}_0^{(\mathrm{fix})} = \Lambda_{k,\ell}^n$. To find the correction factor that leads to the estimate of the noise PSD $\Lambda_{k,\ell}^n$, the result of the next iteration step is set equal to the noise PSD as

$$\overset{\circ}{\bar{y}}_1^{(\mathrm{fix})} = \mathcal{G}_{k,\ell}^{(\mathrm{a})}\ \mathcal{Z}(\Lambda_{k,\ell}^n)\big|_{\Lambda_{k,\ell}^s+\Lambda_{k,\ell}^n} \overset{!}{=} \Lambda_{k,\ell}^n. \tag{4.9}$$

Analgous to $\mathcal{G}_{k,\ell}$, $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$ is the time and frequency dependent correction factor for the correction considered in this chapter. Further, $\mathcal{Z}(\cdot)\big|_{\Lambda_{k,\ell}^s+\Lambda_{k,\ell}^n}$ indicates that $\mathcal{Z}(\cdot)$ is computed under the assumption that $\mathbb{E}\{y_\ell\} = \Lambda_{k,\ell}^s + \Lambda_{k,\ell}^n$. Similar to (4.7), the solution is

$$\mathcal{G}_{k,\ell}^{(\mathrm{a})} = \frac{\Lambda_{k,\ell}^n}{\mathcal{Z}\{\Lambda_{k,\ell}^n\}\big|_{\Lambda_{k,\ell}^s+\Lambda_{k,\ell}^n}}. \tag{4.10}$$

Using $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$ from (4.10) to compute the iteration steps in (4.5) results in $\overset{\circ}{\bar{y}}_i^{(\mathrm{fix})} = \Lambda_{k,\ell}^n$ for all iterations i. Again, using a different initialization, i.e., $\overset{\circ}{\bar{y}}_0^{(\mathrm{fix})} \neq \Lambda_{k,\ell}^n$, the iteration converges back to the noise PSD $\Lambda_{k,\ell}^n$. Similar to Section 3.5, the time-varying correction factor $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$ also depends on the speech PSD $\Lambda_{k,\ell}^s$ and the noise PSD $\Lambda_{k,\ell}^n$. Again, the estimates from previous segments are used to allow the time-varying correction in (4.10) to be used in practical applications. The specific solution used here is discussed in Section 4.3. Algorithm 4 summarizes the method to compute the time-varying correction factor $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$.

The alternative correction method proposed in this chapter including the estimation of the time-varying factor $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$ is summarized in Algorithm 5. In contrast to the solution in (3.21) of Section 3.5, the solution here cannot be easily expressed as a multiplication of the fixed correction $\mathcal{C}^{(\mathrm{a})}$ and a time-varying factor. As a consequence, determining $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$ is computationally more complex than the correction method in Chapter 3. However, due to the scale invariance, it is sufficient to compute the result for various *a priori* SNRs $\xi_{k,\ell}$, i.e., the correction factors can be easily tabulated. This can be done by assuming that $\Lambda_{k,\ell}^y = \Lambda_{k,\ell}^s + \Lambda_{k,\ell}^n = 1$, i.e., that the energy of the noisy signal is constant. From this it can be deduced that $\Lambda_{k,\ell}^n = 1/(\xi_{k,\ell} + 1)$ which can be used in (4.10) to compute $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$ depending on the SNR $\xi_{k,\ell}$.

Algorithm 5 Bias compensation for adpative smoothing filters where the bias is corrected by scaling only the y_ℓ not occurring in $\alpha(y_\ell, \overline{y}_{\ell-1})$ by $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$.

```
Initialize algorithm and compensate bias:
$\breve{\overline{y}}_0 \leftarrow y_0$
**for all** all remaining segments $\ell$ **do**
    Obtain correction factor $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$ using Algorithm 4.
    Perform smoothing and correct bias:
    $\breve{\overline{y}}_\ell = (1 - \alpha(y_\ell, \breve{\overline{y}}_{\ell-1}))\mathcal{G}^{(\mathrm{a})}_{k,\ell} y_\ell + \alpha(y_\ell, \breve{\overline{y}}_{\ell-1})\breve{\overline{y}}_{\ell-1}$
**end for**
```

4.3. EVALUATION

In this section, we evaluate the proposed correction method and compare it with the method proposed in Chapter 3. First, it is verified that the proposed estimation method described in Section 4.1 and Algorithm 4 yields accurate results for both the static correction factor $\mathcal{C}^{(\mathrm{a})}$ and the time-varying version $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$. For this, we employ Monte-Carlo simulations where the samples of the input signal y_ℓ are artificially generated. In the second part of the section, it is analyzed how the proposed correction method performs against the method in Chapter 3. For this, the accuracy of the noise PSD estimation is evaluated in terms of the log-error distortion measure given in Section 2.2.1. Further, the quality of the enhanced signals is compared using PESQ [230] improvement scores as described in Section 2.2.2.

In the evaluations, we use the default values of the noise PSD estimators as described in the literature. Correspondingly, $\alpha^{\uparrow}$ and $\alpha^{\downarrow}$ are set to 0.9995 and 0.9, respectively [81, Section 14.1.3]. For the SPP-based noise estimator $\xi_{\mathcal{H}_1} = 15$ dB and $\alpha^{(\mathrm{fix})}_{\mathrm{SPP}} = 0.8$ are used [70].

4.3.1. Verification of the Estimation Methods

First, the estimation of the static correction factor $\mathcal{C}^{(\mathrm{a})}$ is considered. For this, we estimate the correction factor once using (4.7) for both noise PSD estimators. The results are compared to Monte-Carlo simulations where 10^6 independent realizations of the input signal y_ℓ are generated. Assuming the same signal model as in Section 3.4, we also employ an exponential distribution as given in (3.16) here. As the mean of the input samples is set to a fixed value, the input samples are stationary. The Monte-Carlo reference $\mathcal{C}^{(\mathrm{a})}_{\mathrm{MC}}$ for the factor $\mathcal{C}^{(\mathrm{a})}$ is determined as the factor that minimizes the difference between the mean of the input signal $\mathbb{E}\{y_\ell\}$ and the mean of the corrected filter output $\hat{\mathbb{E}}\{\breve{\overline{y}}_\ell\}$, i.e.,

$$\mathcal{C}^{(\mathrm{a})}_{\mathrm{MC}} = \arg \min_{\mathcal{C}^{(\mathrm{a})}} |\mathbb{E}\{y_\ell\} - \hat{\mathbb{E}}\{\breve{\overline{y}}_\ell\}|. \tag{4.11}$$

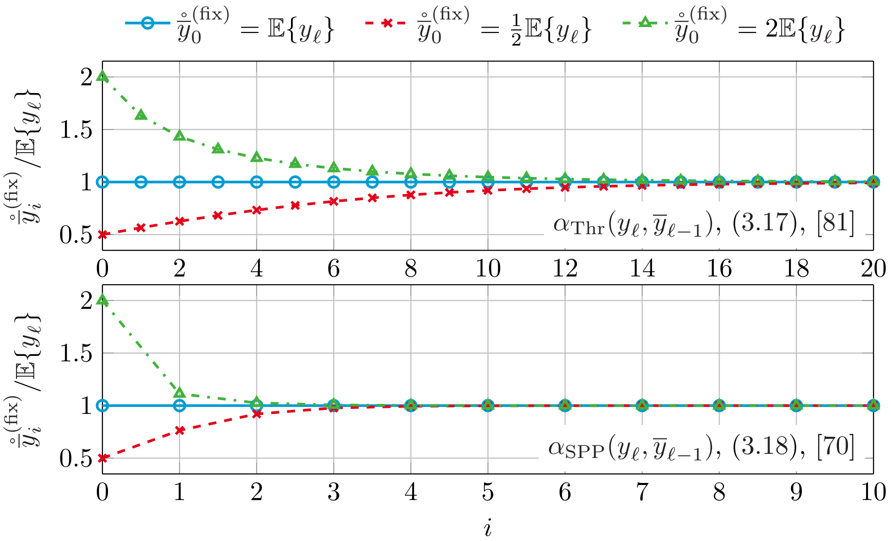

Fig. 4.1.: Iteration steps of (4.5) normalized to the expected value of the filter input, i.e., $\overset{\circ}{\overline{y}}_i^{(\mathrm{fix})}/\mathbb{E}\{y_\ell\}$, for the noise PSD estimators in (3.17) and (3.18) described in Section 3.4.

Here, $\hat{\mathbb{E}}\{\cdot\}$ denotes the estimate of the expected value that is obtained by temporal averaging. The optimal $\mathcal{C}_{\mathrm{MC}}^{(\mathrm{a})}$ in (4.11) is obtained using the Simplex-Downhill method proposed in [236].

First, the claims in Section 4.1 are verified, i.e., that the iteration yields the same value if initialized as proposed and $\mathcal{C}^{(\mathrm{a})}$ is computed as in (4.7). Further, it is shown that the iteration converges to the same value if a different value is used for the initialization. Fig. 4.1 shows the results if the iteration steps in (4.5) are computed. This experiment verifies that using the computed correction factor, the iteration always yields the same value, namely the sought expected value $\mathbb{E}\{y_\ell\}$, if the initial value is set to $\overset{\circ}{\overline{y}}_0^{(\mathrm{fix})} = \mathbb{E}\{y_\ell\}$. Further, the iteration converges to the same value, i.e., the mean of the filter input $\mathbb{E}\{y_\ell\}$, if different values are used for $\overset{\circ}{\overline{y}}_0^{(\mathrm{fix})}$.

Table 4.1 shows the correction factors $\mathcal{C}^{(\mathrm{a})}$ obtained with Algorithm 4. The factors $\mathcal{C}^{(\mathrm{a})}$ determined with Algorithm 4 are smaller than the compensation factors obtained from the Monte-Carlo simulations. The error which remains by using the estimated correction factors instead of the ground truth has also been evaluated using Monte-Carlo simulations. By averaging the output of the corrected adaptive smoothing filter given the $\mathcal{C}^{(\mathrm{a})}$ obtained from Algorithm 4, an underestimation of 0.6 dB and 0.2 dB remains for the smoothing factors $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$, (3.17) [81, Section 14.1.3] and $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$, (3.18), [70], respectively.

Smoothing factor	Monte-Carlo	Sec. 4.1 / Alg. 4
$\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$, (3.17), [81]	2.44	2.37
$\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$, (3.18), [70]	1.20	1.16

Table 4.1.: Estimation of the correction factor $\mathcal{C}^{(\mathrm{a})}$ by Monte-Carlo simulations and Algorithm 4 for the adaptive smoothing functions in (3.17) and (3.18) of Section 3.4. The correction factors are shown as linear quantities.

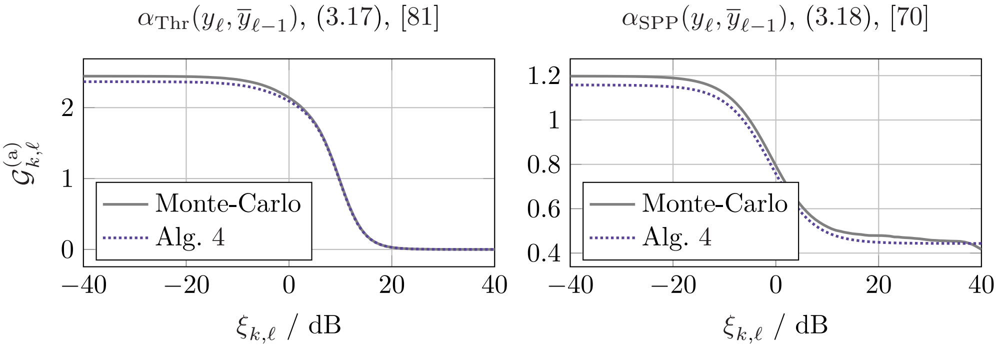

Fig. 4.2.: Estimation of the SNR depended correction $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$ obtained via Monte-Carlo simulations and Algorithm 4 for the noise PSD estimators (3.17) and (3.18) described in Section 3.4.

However, as the remaining underestimation is relatively small, it can be concluded that a good approximation of the true correction factors can be achieved using the method described in Section 4.1. Further, comparing the values of $\mathcal{C}^{(\mathrm{a})}$ to the correction factor $\mathcal{C}$ given in Table 3.1, the respective values of $\mathcal{C}^{(\mathrm{a})}$ are smaller. The difference results from the more indirect influence of Algorithm 5 on the recursion.

Until now, the dependence on the SNR has not been considered for estimating the correction factor $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$. The results in Table 4.1 are equivalent to the case of the *a priori* SNR being equal to $\xi_{k,\ell} = -\infty$ dB. Fig. 4.2 shows how the correction factor $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$ changes with the *a priori* SNR $\xi_{k,\ell}$. For this experiment, the number of samples for the Monte-Carlo simulations results has been reduced to $4 \cdot 10^5$ because (4.11) has to be solved for various SNRs which is computationally rather expensive. The results in Fig. 4.2 show that, in general, the value of the correction factor decreases with increasing SNR. For high SNRs, the input signal is no longer boosted and damped instead to make the adaptive smoothing converge to the sought noise PSD $\Lambda^n_{k,\ell}$. Interestingly, for the SPP-based noise PSD estimator the correction factor $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$ does not approach very small values for high SNRs. Instead $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$ converges to values slightly above 0.4 if the SNR is high. The results obtained from Algorithm 4 are compared again to the Monte-Carlo simulations. Similar to the results shown in Table 4.1, the method in Algorithm 4 slightly underestimates the correction factor. In general, however, the difference is about the same or smaller than in Table 4.1. From this it can be concluded that the proposed approximative method for estimating $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$ also delivers sufficiently accurate results for the SNR dependent case.

4.3.2. Applications to Speech Enhancement

In this section, the correction method considered in this chapter is embedded in a speech enhancement framework. It is compared to the correction method in Chapter 3 and the uncorrected versions in terms of the noise PSD estimation accuracy and speech quality. For the former, the log-error distortion measure is employed as described in Section 2.2.1 while for the latter PESQ [230] improvement scores are considered.

We use 1120 sentences from the TIMIT test corpus [240] which are artificially corrupted by background noises at SNRs ranging from -10 dB to 30 dB in 5 dB steps. Various synthetic and natural noise types are employed namely pink and babble noise taken from the Noisex-92 corpus [239] and a passing car noise. The TIMIT sentences are embedded in random excerpts of the background noise. All signals have a sampling rate of 16 kHz.

The speech enhancement framework, in which the noise PSD estimators are embedded, matches the descriptions in Section 2.1.1. For the STFT, we use 32 ms segments with an overlap of 50 % and a square-root Hann window for spectral analysis and synthesis. The speech PSD is estimated using the decision-directed approach [12] given in (2.27) of Section 2.1.4. The smoothing factor is set to $\alpha_{\mathrm{DD}} = 0.98$. The clean speech coefficients are estimated using the Wiener filter, i.e., (2.15) where the maximum attenuation is limited at a maximum of 12 dB.

For computing the time-varying correction factor $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$, a similar approach as in Section 3.6 is used. For estimating the speech PSD at the beginning of a new segment ℓ, the estimated noise PSD from the previous segment, i.e., $\hat{\Lambda}_{k,\ell-1}^{n}$, is used with the previously estimated speech coefficients $|\hat{S}_{k,\ell-1}|^2$ in the decision-directed approach. This pre-estimate is used to determine the time-varying correction factor $\mathcal{G}_{k,\ell}^{(\mathrm{a})}$ where, again, a smoothing constant of $\alpha_{\mathrm{DD}} = 0.98$ is used. Similar to Chapter 3, a limit is imposed on the time-varying correction factor to avoid stagnation of the estimation process. This is achieved by limiting the pre-estimated SNR to values equal or below 20 dB.

For the correction method in Chapter 3, which is used for comparisons here, the same parameters are used as in Section 3.6. Correspondingly, the fixed correction factor $\mathcal{C}$ is estimated using Algorithm 2 and the time-varying extensions presented in Section 3.5 is used for the bias correction as in Algorithm 3. The same practical methods that have been used to compute $\mathcal{G}_{k,\ell}$ in Section 3.6 are also employed here.

For the evaluation of log-error distortion, we ensure that the first 5 s are excluded to allow the noise estimators to adapt to the background noise. Further, we exclude the 0 Hz

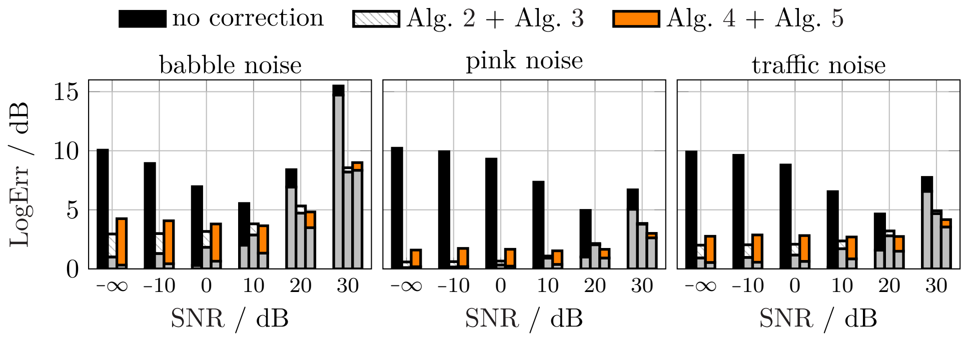

Fig. 4.3.: Comparison of the proposed correction methods in terms of the log-error distortion with respect to the noise PSD estimators $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$ proposed in (3.17), [81] for noisy speech. The lower part of the bars (gray) is the overestimation whereas the upper part (colored) is the underestimation.

bin and the Nyquist frequency as the periodogram cannot be considered exponentially distributed here. As the pink noise is stationary, the reference noise PSD $\Lambda^n_{k,\ell}$ is obtained by temporal averaging of the noise periodogram. For the nonstationary noises, we employ a slightly smoothed version of the noise periodogram, where a fixed smoothing constant of $\alpha^{(\mathrm{fix})}_{\mathrm{LogErr}} = 0.73$ is used. The results are shown in Fig. 4.3 and Fig. 4.4 for the noise PSD estimators in (3.17), [81, Section 14.1.3], and (3.18), [70], respectively.

Considering the results in Fig. 4.3, the log-error distortions show that the correction methods proposed in this chapter and in Chapter 3 generally reduce the estimation error. For the noise only case, i.e., SNR $= -\infty$, both correction methods lead to lower log-error distortions at the cost of a slightly increased noise overestimation in comparison to the case where no correction is applied. As a consequence, the total log-error distortion is considerably smaller compared to the case where no correction is employed. The method proposed here often results in a higher total error for low SNRs in comparison to the method in Chapter 2 while the total error is lower for high SNRs. In general, the alternative correction method considered in this chapter tends to underestimate the noise PSD more strongly. As the noise PSD estimator proposed in [81, Section 14.1.3] always allows the noise PSD to be tracked, the noise PSD is severely overestimated in high SNRs. Similar to the correction method in Chapter 3, the alternative correction method proposed here also reduces the overestimation through a time-varying adaptation of the correction factor.

A somewhat similar picture is shown by the results in Fig. 4.4, which depicts the log-error distortion for the SPPs-based noise PSD estimator. Again, the alternative correction factor generally tends to result in stronger underestimations for low SNRs. As a consequence, the total log-error is generally higher for this approach compared to the correction proposed in Chapter 3 in low SNRs. In very high SNRs, the method presented here again leads

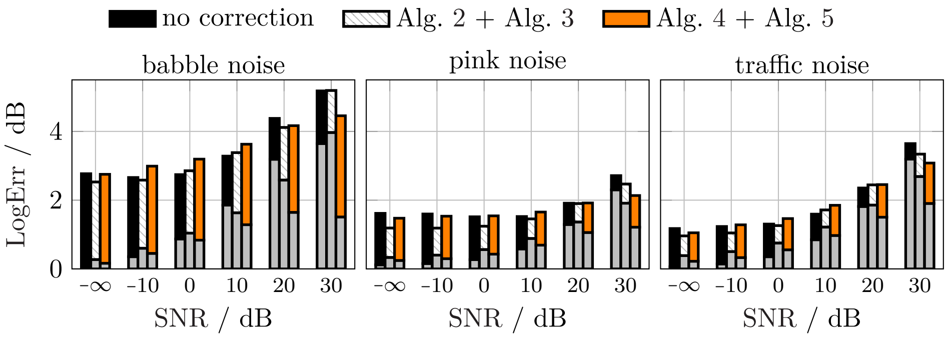

Fig. 4.4.: Same as in Fig. 4.3 but for the noise PSD estimator $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$ proposed in (3.18), [70].

to slightly lower total log-error distortions. However, as mentioned in Section 3.6, the estimation error of the SPP-based noise PSD estimator given in (3.18), [70] is generally low. Hence, the corrected versions and the uncorrected version of this algorithm yield virtually the same results in terms of the log-error distortion.

In the last part of the evaluation, PESQ [230] scores are considered. In Fig. 4.5, the results of the corrected and uncorrected versions of the noise PSD estimators considered in Section 3.4 are shown. It compares the correction methods presented in this chapter to the method presented in Chapter 3. Additionally, also the results of the uncorrected noise PSD estimators are considered. The results are similar to the PESQ scores shown in Fig. 3.8. Considering the SPP-based noise PSD estimator given in (3.18), [70], both correction methods virtually yield no difference to the uncorrected version of this estimator. However, correcting the bias for the noise PSD estimator given in (3.17), [81, Section 14.1.3], the PESQ scores are considerably improved. Similar to the correction method considered in Chapter 3, the improvements can be up to 0.2 points in PESQ. Again, these increases can be attributed to the reduction of the underestimation in low SNRs and the reduction of overestimation in high SNRs, which are also obtained by this alternative correction method presented in this chapter. Comparing the results for the correction method in this chapter with the correction method considered in Chapter 3, the PESQ scores are virtually identical.

4.4. SUMMARY

In this chapter, an alternative correction method for compensating the bias of adaptive smoothing filters has been presented. The proposed estimation method for determining the fixed correction factor is related to the iterative procedure presented in Section 3.2, [215]. In contrast to the method in Chapter 3, [215], the correction factor for the compensation method considered here can be determined without any iteration. Further, the correction

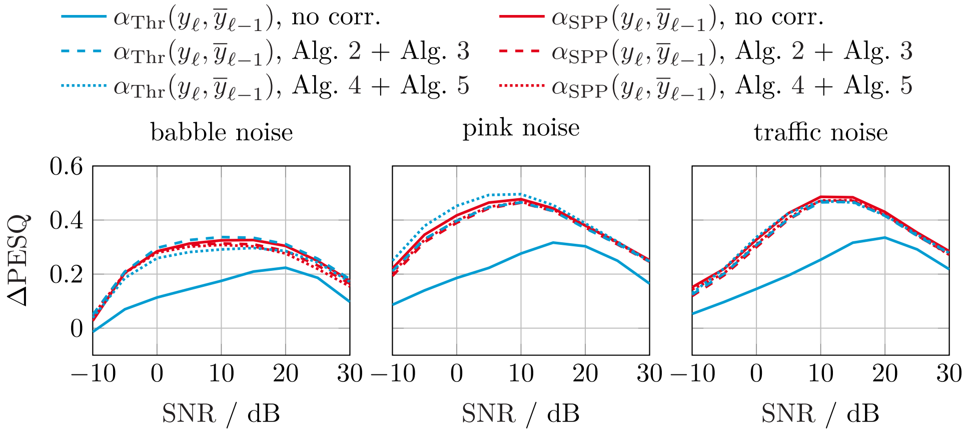

Fig. 4.5.: PESQ improvement scores depending on the adaptive smoothing functions embedded in the speech enhancement framework. Here, the adaptive smoothing functions $\alpha_{\mathrm{Thr}}(y_\ell, \overline{y}_{\ell-1})$ and $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$ are used as noise PSD estimators with and without the correction proposed in Algorithm 5. The correction factor $\mathcal{G}^{(\mathrm{a})}_{k,\ell}$ is estimated using Algorithm 4.

method was extended such that the compensation is aware of the speech signal which prevents overestimations in speech presence. Using Monte-Carlo simulations, we analyzed the proposed procedure to determine the correction factor and showed that the factors can be estimated with sufficient accuracy. Further, the correction method proposed here has been evaluated using the log-error distortion measures. For the SPP-based noise PSD estimator, the correction method proposed here has little effect while the error induced by the method proposed in [81, Section 14.1.3] can be considerably reduced. Comparisons with the compenstation method used in Chapter 3 show that the correction method proposed here generally tends to lead to underestimations of the noise PSD. This results in a smaller total log-error distortion for high SNRs but higher distortions for low SNRs. Further, PESQ scores have been considered by embedding the corrected and uncorrected versions of the noise PSD estimators in a speech enhancement framework. Similar to the results obtained for the log-error distortion, the correction method yields virtually the same results if the SPP-based noise PSD estimator is considered. For the noise PSD estimator in [81, Section 14.1.3], however, considerable improvements can be obtained. The PESQ improvements obtained by this compensation method are virtually the same as for the compensation method presented in Chapter 3.

Part III.

Improvements for Envelope Based Enhancement Schemes

CHAPTER 5

EFFECTS OF SUPER-GAUSSIAN PRIORS ON ENVELOPE BASED SPEECH ENHANCEMENT

To apply statistically motivated clean speech estimators as described in Section 2.1.2, the clean speech PSD and the noise PSD need to be estimated. In this chapter, we consider ML-based methods where the structure of the speech and possibly also the noise PSD are learned before the processing takes place. Specifically, we focus on a type of ML-based algorithm, where the learned speech models only represent the spectral envelope, e.g., [19], [20], [26], [27], [91], [101], [104]. The spectral envelope of speech corresponds to the vocal tract, which acts as a filter on the excitation signal and allows humans to utter different phonemes. However, the excitation signal, which has a harmonic structure for voiced sounds due to the vibrating vocal cords, is not reflected by the spectral envelope. Enhancement schemes where only the spectral envelope of speech is modeled using an ML algorithm, is referred to as machine-learning spectral envelope (MLSE)-based enhancement schemes in this chapter. Such approaches increase the generalization and also reduce the computational complexity, as well as, the amount of data required for training. Often, these methods belong to the category of HMM, GMM or codebook based methods which have been considered in Section 1.3.1.

These methods are distinguished from non-MLSE-based estimation schemes considered in Section 2.1, where the speech and noise PSDs are estimated blindly without any pre-training. While MLSE approaches exploit prior knowledge about typical speech spectral structures, the envelope representation also limits the quality of the enhanced signal. Due to the coarse representation of speech, residual noise may remain between spectral harmonics. To reduce the undesired residual noise between harmonics, different solutions have been proposed. In [104], a harmonic model has been used to attenuate the remaining noise component between harmonics. In [26], [27], the speech presence probability is estimated, which is used to attain a suppression of the residual noise. Other approaches, e.g., [241], incorporate the excitation, e.g., the harmonic structures of voiced excitations, explicitly in a statistical model to avoid this problem.

In this chapter, we show that if super-Gaussian clean speech estimators are used, post-processing as in [26], [27], [104] is not necessary. For this, we consider the parameterized

This chapter is partly based on:

[218] R. Rehr and T. Gerkmann, “On the importance of super-Gaussian speech priors for machine-learning based speech enhancement,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 357–366, Feb. 2018, © 2018 IEEE.

clean speech estimator proposed in [31], which leverages the statistical models presented in Section 2.1.2. An analysis of this estimator shows that, under a super-Gaussian speech model, the background noise can be reduced even if the speech PSD is overestimated, e.g., between spectral harmonics when modeling only the envelope. Furthermore, the estimator in [31] is employed in two MLSE-based enhancement schemes. Both methods serve as examples and can be considered as variants of previously proposed methods in the literature. The first one is a DNN-based scheme similar to [26] which is chosen due to its similarities to other MLSE-based enhancement methods, e.g., [19], [20], [27], [91], [101], [104]. To demonstrate the effectiveness of super-Gaussian estimators also for other MLSE-based enhancement methods, the estimator in [31] is additionally embedded in a supervised, sparse NMF enhancement scheme based on [122], [136]. Here, a low amount of basis vectors is employed such that mainly spectral envelopes are represented by the NMF basis vectors. We show that for the used non-MLSE-based enhancement scheme, which is capable of estimating the spectral fine structure of speech, the super-Gaussian speech model yields only small improvements. However, for the MLSE-based enhancement schemes, which only employ a model of the speech envelope, the super-Gaussian model has a very beneficial effect, because it allows the removal of disturbing residual noises. Besides the MLSE approaches addressed here, also MLSE approaches with log-max, also known as MixMax, mixing models benefit from this effect [219] which will be discussed in Chapter 6. Super-Gaussian speech models have also been previously employed in ML-based speech enhancement algorithms, e.g., [96], [97], [99]. However, none of the papers provides an explicit analysis of the obtained improvements over Gaussian estimators in terms of the gain functions that result under super-Gaussian speech models. Furthermore, the advantages of these estimators in combination with spectral speech envelope models have not been highlighted.

This chapter is structured as follows: First, we recapitulate the clean speech estimator proposed in [31] in Section 5.1. After that, we describe the considered MLSE-based enhancement schemes in Section 5.2 and Section 5.3. In Section 5.4 and Section 5.5, an analysis of the super-Gaussian estimator [31] and, respectively, a comparison of clean speech estimators employed in different enhancement schemes is presented. In Section 5.6, the results of the subjective evaluation test are reported.

5.1. SPEECH ESTIMATORS

In this section, we revisit the clean speech estimator [31]. This estimator is parameterized such that various known estimators, e.g., [12], [28], [32], [49], [53], result as special cases. In particular, it allows the incorporation of super-Gaussian speech models and the estimation of compressed amplitudes. As in [242], we use the name (M)MSE estimation with (o)ptimizable (s)peech (m)odel and (i)nhomogeneous (e)rror criterion (MOSIE) for the estimator in [31].

In this chapter, the STFT-based enhancement scheme described in Section 2.1.1 is shared among all algorithms. Here, the employed input signals have a sampling rate of 16 kHz.

Table 5.1.: List of clean speech estimators that MOSIE [31] generalizes.

ν	β	Related estimator
1	1	Gaussian STSA [12]
1	$\beta \to 0$	Gaussian LSA [28]
$\nu < 1$	1	super-Gaussian STSA [30], [53]
$\nu < 1$	$\beta \to 0$	super-Gaussian LSA [32]

The enhancement takes place in the STFT domain where the segment length of the STFT is set to 32 ms and a segment overlap of 50 % is employed. The estimate of the clean speech spectral coefficients $\hat{S}_{k,\ell}$ is obtained from the noisy observation $Y_{k,\ell}$ using [31]. For the analysis and the synthesis a square-root Hann window is used.

MOSIE [31] is a statistically optimal estimator of the speech amplitude $A_{k,\ell}$. It results from the minimization of the expected value in (2.13). MOSIE is derived under the assumption that the noisy spectra $Y_{k,\ell}$ result from the addition of speech $S_{k,\ell}$ and $N_{k,\ell}$ as in (2.3). Further, the compression function $c(\cdot)$ is set to $|A_{k,\ell}|^\beta$ where β denotes a compression factor. In [31], the complex noise coefficients $N_{k,\ell}$ are assumed to follow a circular-symmetric complex Gaussian distribution as given in (2.14). Further, a parametrizable circular-symmetric possibly heavy-tailed super-Gaussian distribution is employed to describe $S_{k,\ell}$ in [31]. It is the same model that results from using the χ-distribution (2.16) for the speech magnitude $A_{k,\ell}$ and a uniform distribution for the phase $\Phi^s_{k,\ell}$ as described in Section 2.1.2. Given the mixing model and the statistical assumptions about the noise and speech coefficients, the estimate of the amplitude $\hat{A}_{k,\ell}$ is given by [31]

$$\hat{A}_{k,\ell} = \sqrt{\frac{\Lambda^n_{k,\ell}\xi_{k,\ell}}{\xi_{k,\ell}+\nu}} \left[\frac{\Gamma(\nu+\beta/2)}{\Gamma(\nu)} \frac{\mathcal{M}(\nu+\beta/2, 1; \zeta_{k,\ell})}{\mathcal{M}(\nu, 1; \zeta_{k,\ell})} \right]^{\frac{1}{\beta}}. \tag{5.1}$$

Here, $\zeta_{k,\ell}$ is given by $\gamma_{k,\ell}\xi_{k,\ell}/(\nu+\xi_{k,\ell})$ and the symbol $\mathcal{M}(\cdot,\cdot;\cdot)$ represents the confluent hypergeometric function [227, Section 9.21]. The remaining symbols are explained in Section 2.1 and can be found in the glossary. As MOSIE estimates only the clean speech magnitudes $\hat{A}_{k,\ell}$, the estimated amplitude in (5.1) is combined with the noisy phase Φ^y as in (2.5).

It is interesting to note that MOSIE [31], generalizes existing clean speech estimators. For example, if $\beta = 1$ and $\nu = 1$, MOSIE [31] is equivalent to Ephraim and Malah's STSA [12] and, for very small values of β and $\nu = 1$, the LSA [28] is approximated. Super-Gaussian estimators are obtained for $\nu < 1$. Table 5.1 gives an overview over the related estimators.

To evaluate the expression in (5.1), estimates of the speech PSD $\Lambda^s_{k,\ell}$ and the noise PSD $\Lambda^n_{k,\ell}$ are required. These can be obtained from non-MLSE-based speech PSD and noise PSD estimators. In this chapter, the noise PSD $\Lambda^n_{k,\ell}$ is estimated using the SPP-based

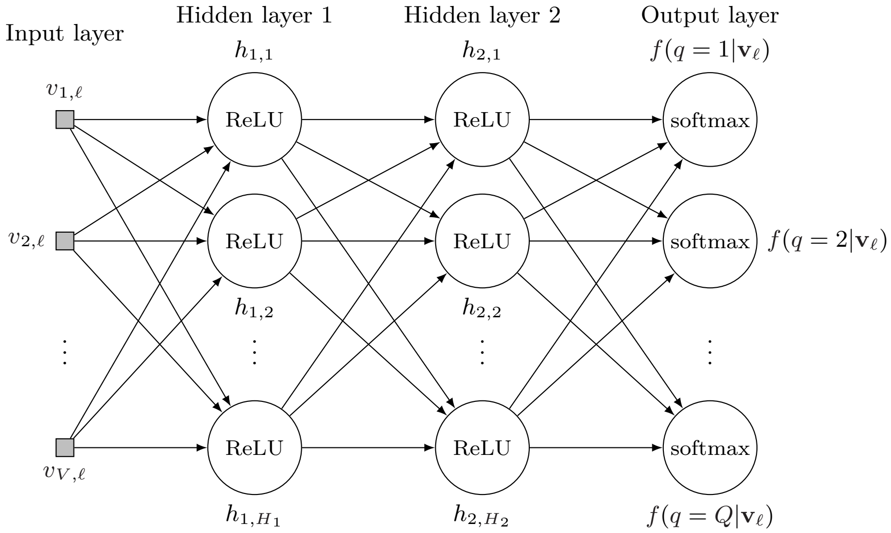

Fig. 5.1.: Architecture of the employed DNN.

noise estimator [70]. The speech PSD of the non-MLSE-based enhancement scheme is estimated using TCS as proposed in [82]. More information about these algorithms can be found in Section 2.1.4. The enhancement scheme that results from using these speech and noise PSD estimators in MOSIE is referred to as non-MLSE-based enhancement scheme throughout this chapter. However, also ML-based estimators of the clean speech and the noise PSD can be employed which are considered next.

5.2. DNN-BASED SPEECH ENHANCEMENT SCHEME

As the first example of an MLSE enhancement scheme, a method using a DNN-based phoneme recognizer similar to [26] is considered. Similarly, MLSE models have also been used for enhancement schemes in [19], [20], [27], [91], [101], [104]. In [26], a two step procedure is used for speech enhancement. First, the spoken phoneme is identified from the noisy observation. After that, a learned speech PSD corresponding to the recognized phoneme is used in a clean speech estimator, e.g., MOSIE [31], to enhance the noisy observation. As speech is modeled on a phoneme level, the speech spectral fine structures, e.g., the spectral harmonics, are not resolved.

For the phoneme recognition, a DNN is used with the architecture shown in Fig. 5.1. The DNN's input is given by 13 MFCCs including the Δ and $\Delta\Delta$ accelerations which are extracted for each segment ℓ. To these features, a context is added by including the features of the three previous and three future segments which results in the feature vector $\mathbf{v}_\ell = [v_{1,\ell}, \ldots, v_{V,\ell}]^T$ with dimensionality $V = 273$. Here, $v_{i,\ell}$ denote the elements of the

Algorithm 6 DNN-based enhancement scheme.

```
Require: Trained DNN and offline computed $\Lambda_k^{s|q_\ell}$.
Require: Noisy observations $Y_{k,\ell}$ of a complete utterance.
Extract MFCCs $\mathbf{v}_\ell$ from $Y_{k,\ell}$ for complete utterance and add context.
Apply CMVN over complete utterance to give $\mathbf{v}_\ell$.
for all segments $\ell$ do
    Estimate noise PSD $\hat{\Lambda}_{k,\ell}^n$ using [70].
    Obtain $f(q|\mathbf{v}_\ell)$ from the DNN.
    for all phonemes $q$ do
        Obtain clean speech estimate $\hat{S}_{k,\ell}^{(q)}$ for phoneme $q$.
        For this, $\Lambda_k^{s|q_\ell}$ and $\hat{\Lambda}_{k,\ell}^n$ are employed in (5.1).
    end for
    Obtain final clean speech estimate $\hat{S}_{k,\ell}$ using (5.2).
end for
```

feature vector $\mathbf{v}_\ell$. Further, $\cdot^T$ denotes the vector and matrix transpose. For the employed segment length and segment shift, the context is approximately 100 ms. To improve the robustness of the recognition in noisy environments, the feature vectors are normalized using cepstral mean and variance normalization (CMVN) [243] before they are employed for training or testing [26]. The CMVN is applied per utterance.

The features are passed through two hidden layers to finally obtain a score $f(q|\mathbf{v}_\ell)$ for each phoneme $q \in \{1, \dots, Q\}$. We base the number of phonemes on the annotation given in the TIMIT database [240] which distinguishes between $Q = 61$ classes including pauses and non-speech events. The hidden layers of the DNN consist of H_1 and H_2 outputs, where $H_1 = H_2 = 512$ is used. Similar to [26], [244], [245], rectified linear units (ReLUs) are employed as transfer functions of these two layers. For the output layer, a softmax transfer function is used which is interpreted as the posterior probability $f(q|\mathbf{v}_\ell)$ that phoneme q was spoken given the features $\mathbf{v}_\ell$.

For the enhancement, MLSE-based clean speech PSDs $\Lambda_k^{s|q_\ell}$ are employed where each $\Lambda_k^{s|q_\ell}$ represents the speech PSD of a specific phoneme q. During processing, each $\Lambda_k^{s|q_\ell}$ is used in (5.1) via $\xi_{k,\ell} = \Lambda_k^{s|q_\ell}/\Lambda_{k,\ell}^n$, which yields the phoneme specific clean speech estimates $\hat{S}_{k,\ell}^{(q)}$. For this, the noise PSD $\Lambda_{k,\ell}^n$ is estimated using [70]. Similar to [26], the estimates $\hat{S}_{k,\ell}^{(q)}$ are averaged based on the recognition scores $f(q|\mathbf{v}_\ell)$ to give a final estimate $\hat{S}_{k,\ell}$. More specifically, the clean speech coefficients are obtained by

$$\hat{S}_{k,\ell} = \sum_{j=1}^{Q} f(q = j|\mathbf{v}_\ell)\hat{S}_{k,\ell}^{(q)}. \tag{5.2}$$

The steps required to enhance the noisy observations $Y_{k,\ell}$ using the DNN-based enhancement scheme are summarized in Algorithm 6.

For the training of the DNN-based MLSE system, we employ 1196 gender and phonetically balanced sentences from the TIMIT training set. As in [26], the DNN is trained only using clean speech to ensure that the phoneme recognition does not depend on the background noise type. The target vectors for the training are given by a one-hot encoding of the TIMIT phoneme annotation [240]. The error function is given by the cross-entropy which is minimized using scaled conjugate gradient back-propagation [246]. Before back-propagation, the weights of the DNN's two hidden layers are initialized using the Glorot method [160]. The weights of the output layer are initialized using the Nguyen-Widrow method [247].

Similar to the non-MLSE-based enhancement scheme, the noise PSD $\Lambda^n_{k,\ell}$ is estimated using [70]. The speech PSDs $\Lambda^{s|q_\ell}_k$ that are linked to the phonemes q are obtained as

$$\Lambda^{s|q_\ell}_k = \frac{1}{|\mathbb{L}^{(q)}|} \sum_{\ell \in \mathbb{L}^{(q)}} |S_{k,\ell}|^2, \tag{5.3}$$

where $\mathbb{L}^{(q)}$ denotes the set that contains the segments that belong to the phoneme q in the training data. As (5.3) is scale-dependent, we normalize the time-domain clean speech input signal both in training and testing such that all sentences have the same peak value. During training, the clean speech data is available, while during testing, oracle knowledge is provided. This normalization is also employed for the other enhancement schemes, i.e., for the non-MLSE-based and the NMF-based enhancement scheme given in Section 5.3. Here, however, the normalization has no influence as these approaches are scale-independent.

5.3. NMF-BASED SPEECH ENHANCEMENT SCHEME

In this part, the MLSE-based enhancement scheme that employs NMF is described. It serves as a second example for MLSE-based enhancement schemes. NMF approximates a non-negative matrix $\mathbf{Y}$ as $\mathbf{Y} \approx \mathbf{BW}$, where $\mathbf{B}$ and $\mathbf{W}$ are also non-negative matrices. The columns of $\mathbf{B}$ are referred to as basis vectors and the columns of $\mathbf{W}$ as activation vectors. An overview of NMF-based enhancement schemes has been given in Section 1.3.2.

Here, a simple, supervised, sparse NMF approach is used which employs the Itakura-Saito (IS) divergence as the cost function [122], [136]. As argued in [122], if the noisy spectral coefficients $Y_{k,\ell}$ are independent and follow a circular-symmetric Gaussian distribution, minimizing the IS divergence for approximating the noisy periodogram as $\left[|Y_{k,\ell}|^2\right] = \mathbf{Y} \approx \mathbf{BW}$ allows the elements of the product $\mathbf{BW}$ to be interpreted as the noisy PSD $\Lambda^y_{k,\ell}$. The IS cost function including the sparsity constraint is given by [136]

$$J = \delta|\mathbf{W}|_1 + \sum_{i,j} \frac{(\mathbf{Y})_{i,j}}{(\mathbf{BW})_{i,j}} + \log\left(\frac{(\mathbf{Y})_{i,j}}{(\mathbf{BW})_{i,j}}\right) - 1, \tag{5.4}$$

Algorithm 7 NMF-based enhancement scheme.

```
Require: Speech and noise basis matrix $\mathbf{B}^{(s)}$, $\mathbf{B}^{(n)}$.
Set $\mathbf{B} = [\mathbf{B}^{(s)}, \mathbf{B}^{(n)}]$.
for all segments $\ell$ do
    Create vector $\mathbf{y}_\ell = |Y_{k,\ell}|^2$ and add context.
    Initialize $\mathbf{W}$ with positive random numbers.
    repeat
        Update $\mathbf{W}$ with the update rule in [136, (4)].
    until convergence or maximum iterations reached
end for
Obtain $\hat{\Lambda}^s_{k,\ell}$ and $\hat{\Lambda}^n_{k,\ell}$ using (5.5) and (5.6).
Use estimated PSDs in (5.1) to obtain $\hat{S}_{k,\ell}$.
```

where $(\cdot)_{i,j}$ denotes element of the respective matrix, $|\cdot|_1$ the L_1-norm, and δ is the factor that controls the sparsity. This cost function can be optimized using the multiplicative update rules in [136].

For estimating the speech and the noise PSD, it is assumed that the basis matrix $\mathbf{B}$ is given by the concatenation of a speech basis matrix $\mathbf{B}^{(s)}$ and a noise basis matrix $\mathbf{B}^{(n)}$ as $\mathbf{B} = [\mathbf{B}^{(s)}, \mathbf{B}^{(n)}]$. The speech and noise basis matrices are learned prior to the processing and are held fixed during processing. This means that only the activation matrices are updated. For obtaining an estimate of $\Lambda^s_{k,\ell}$ and $\Lambda^n_{k,\ell}$, also the activation matrix $\mathbf{W}$ is split into a speech and noise dependent part as $\mathbf{W} = \left[(\mathbf{W}^{(s)})^T, (\mathbf{W}^{(n)})^T\right]^T$ such that $\mathbf{Y} \approx \mathbf{BW} = [\mathbf{B}^{(s)}, \mathbf{B}^{(n)}]\left[(\mathbf{W}^{(s)})^T, (\mathbf{W}^{(n)})^T\right]^T$. With this, the speech and the noise PSD can be obtained as

$$\hat{\Lambda}^s_{k,\ell} = \sum_{i=1}^{I^{(s)}} (\mathbf{B}^{(s)})_{k,i} (\mathbf{W}^{(s)})_{i,\ell} \tag{5.5}$$

$$\hat{\Lambda}^n_{k,\ell} = \sum_{i=1}^{I^{(n)}} (\mathbf{B}^{(n)})_{k,i} (\mathbf{W}^{(n)})_{i,\ell}, \tag{5.6}$$

where $I^{(s)}$ is the number of speech bases while $I^{(n)}$ denotes the number of noise bases. The steps for enhancing the noisy observations are summarized in Algorithm 7.

For the NMF-based enhancement scheme, the same speech audio material is employed for training as for the DNN-based enhancement scheme. Also here, a context of 7 segments is employed, i.e., three past and three future segments are appended to the noisy input vectors. As a consequence, the number of rows of the basis matrices is increased and the speech PSD and the noise PSD are reconstructed with a context. For the enhancement, however, only the elements corresponding to the current segment are employed. We use 30 bases in the speech basis matrix $\mathbf{B}^{(s)}$ and the noise basis matrix $\mathbf{B}^{(n)}$ while the sparsity weight in (5.4) is set to $\delta = 10$. The low amount speech basis vectors forces the NMF algorithm to learn a dictionary that represents only the spectral envelope of speech. It is used to demonstrate the effects of super-Gaussian estimators for MLSE approaches. For better performance, larger dictionaries are generally used for NMF approaches which allow the dictionary to resolve the spectral fine structure.

The noise basis matrices $\mathbf{B}^{(n)}$ are trained for a set of specific background noise types. The used types are babble noise, factory 1 noise, and pink noise taken from the NOISEX-92 database [239]. Further, an amplitude modulated version of the pink noise similar to [70] and a traffic noise taken from [248] are included. These noise types are also used later in the evaluation in Section 5.5. To ensure that different audio material is used in the evaluation, only the first two minutes of the respective noise type are used for training. This corresponds to a partitioning where 50 % of the background noise material is used for training and 50 % for testing. For training and testing, a maximum of 200 iterations are performed for the multiplicative updates in [136]. For testing, the noise matrix appropriate for the respective noise type is chosen in the evaluation, i.e., the background noise type is assumed to be known. The employed non-MLSE-based and the DNN-based enhancement scheme do not require such prior knowledge. However, as discussed in [21], [141], such a supervised approach may be appropriate for some applications, e.g., where the environment can be identified using an environment classifier.

5.4. IMPORTANCE OF SUPER-GAUSSIANITY FOR MLSE-BASED SPEECH ENHANCEMENT

In this section, we analyze the effect of the super-Gaussian speech estimators on non-MLSE-based and MLSE-based speech enhancement schemes. Before that, we analyze how the shape ν and the compression β influence the behavior of MOSIE [31].

5.4.1. Analysis of the Gain Functions

In this part, we analyze the behavior of the clean speech estimator MOSIE [31]. For this, the gain function $G_{k,\ell}$ as defined in (2.4) is considered. As MOSIE [31] combines an estimate of the clean speech magnitude $\hat{A}_{k,\ell}$ with the noisy phase Φ^y, the gain is a real-valued. Hence, it describes by how much a spectral coefficient is boosted or attenuated depending on the speech PSD $\Lambda^s_{k,\ell}$, the noise PSD $\Lambda^n_{k,\ell}$, and the noisy input $Y_{k,\ell}$.

Fig. 5.2 shows the gain $G_{k,\ell}$ of MOSIE [31] over the *a posteriori* SNR $\gamma_{k,\ell}$ for two *a priori* SNRs: $\xi_{k,\ell} = -5$ dB is shown in the upper row and $\xi_{k,\ell} = 10$ dB in the lower row. The compression parameter β is varied and the shape ν is kept fixed in the left panel and vice versa in the right panel. It is well known that super-Gaussian estimators ($\nu < 1$) preserve speech better than Gaussian estimators ($\nu = 1$) for large *a posteriori* SNRs [46]. However, in the context of MLSE-based speech enhancement, it is of particular interest to observe in Fig. 5.2 that with decreasing shape ν, a stronger attenuation is applied to the input

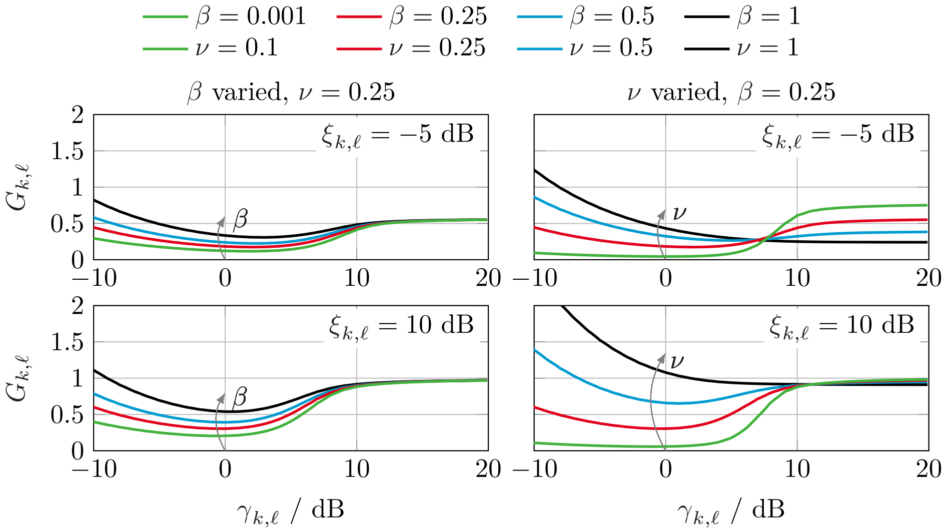

Fig. 5.2.: Gain function $G_{k,\ell}$ of MOSIE [31] over the *a posteriori* SNR $\gamma_{k,\ell}$ for different values of shape ν and compression β. The upper row shows the results for an *a priori* SNR of -5 dB and the lower for an *a priori* SNR of 10 dB. See Table 5.1 for related estimators for the values of ν and β.

coefficients for low *a posteriori* SNRs $\gamma_{k,\ell}$ even if the *a priori* SNR $\xi_{k,\ell}$ is large. A similar effect is observed if a stronger compression, i.e., smaller values for β, are employed.

These observations are supported by Fig. 5.3 where the gain function $G_{k,\ell}$ is shown over the *a priori* SNR $\xi_{k,\ell}$. Here, the two rows show the behavior for two *a posteriori* SNRs $\gamma_{k,\ell} = 0$ dB and $\gamma_{k,\ell} = 10$ dB. For the Gaussian case ($\nu = 1$), Fig. 5.3 shows that the gain $G_{k,\ell}$ mainly depends on the *a priori* SNR $\xi_{k,\ell}$. If the *a posteriori* SNR $\gamma_{k,\ell}$ is close to 0 dB and low values for β and ν are employed, i.e., the super-Gaussian case is considered, the attenuation remains low over a wide range of *a priori* SNRs $\xi_{k,\ell}$. Hence, for MLSE-based speech enhancement schemes, the residual noise can be suppressed even for large overestimations of the *a priori* SNR $\xi_{k,\ell}$. This occurs, e.g., between speech spectral harmonics which are not resolved by spectral envelope models.

5.4.2. Effects of Super-Gaussian Estimators on the Enhancement

In this part, we analyze how the behavior of MOSIE [31] influences the considered enhancement schemes. For this, a speech signal taken from the TIMIT test set is corrupted by stationary pink noise at an SNR of 5 dB. The spectrogram of the used signal is shown in Fig. 5.4. This signal is processed by the non-MLSE-based enhancement scheme and the two MLSE-based enhancement schemes.

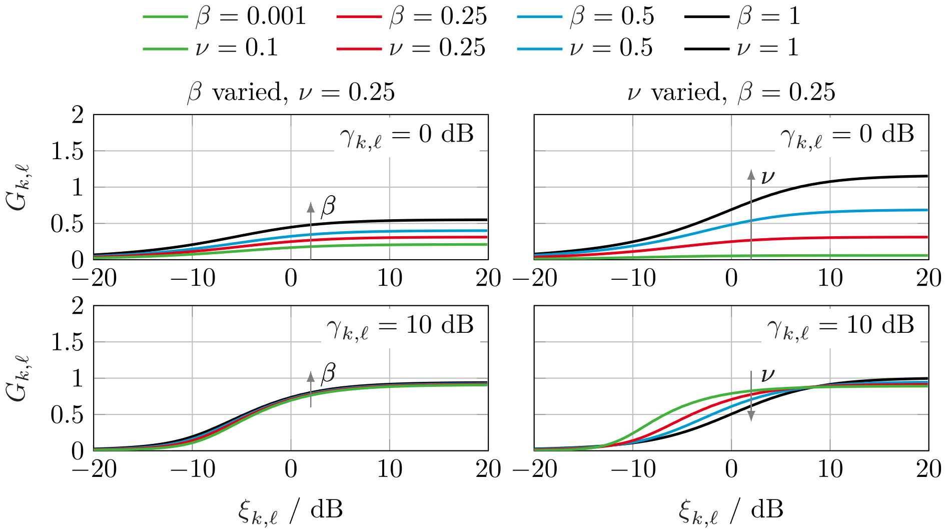

Fig. 5.3.: Same as Fig. 5.2 but over the *a priori* SNR $\xi_{k,\ell}$ and for two fixed *a posteriori* SNRs $\gamma_{k,\ell} = 0$ dB and $\gamma_{k,\ell} = 10$ dB.

In Fig. 5.5, we depict the resulting *a priori* SNRs $\xi_{k,\ell}$. For the DNN-based enhancement scheme, the *a priori* SNR of the phoneme that is most likely to be present is shown for each segment. Note that this selection is only performed for the visualization in Fig. 5.5. Otherwise, $\hat{S}_{k,\ell}$ is estimated as in (5.2). In Fig. 5.5, the estimated *a priori* SNRs $\xi_{k,\ell}$ obtained from the non-MLSE-based enhancement scheme shows a fine structure which is similar to the speech structure visible in Fig. 5.4. Contrarily, the structure of the *a priori* SNRs $\xi_{k,\ell}$ estimated by the MLSE-based enhancement schemes is very coarse and reveals no or only little of the harmonic fine structure shown in Fig. 5.4. Using these envelope models for the speech component leads to an overestimation of the *a priori* SNR $\xi_{k,\ell}$ between spectral harmonics.

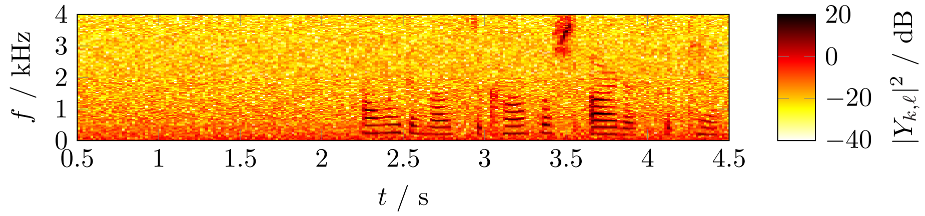

Fig. 5.4.: Spectrogram of the example speech signal in stationary pink noise at at 5 dB SNR. Here, f denotes frequency and t time.

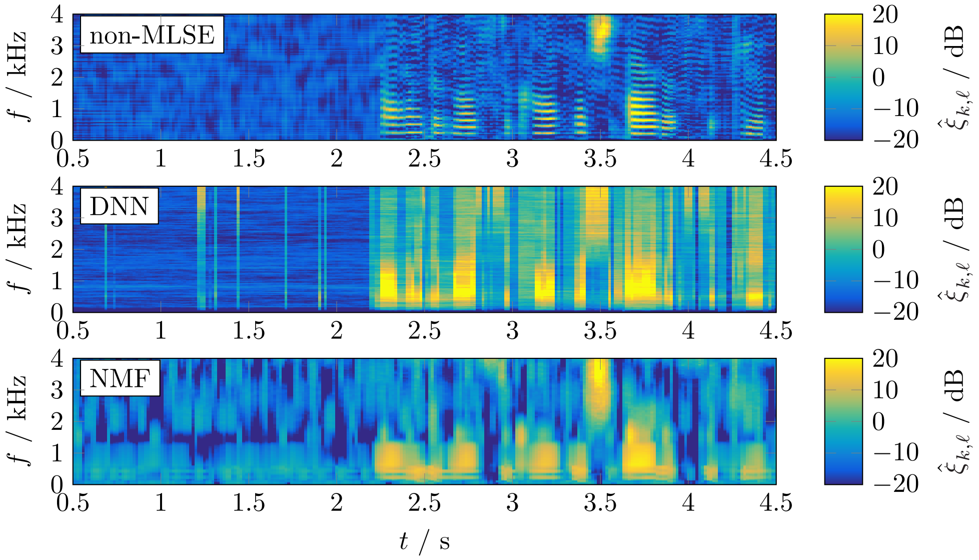

Fig. 5.5.: *A priori* SNR $\hat{\xi}_{k,\ell}$ estimated using different enhancement schemes for the excerpt shown in Fig. 5.4. Here, f denotes frequency and t time.

Next, the gain $G_{k,\ell}$ as defined in (2.4) is considered. For this example, we use MOSIE [31] with two different parameter setups. First, a setup is used where the clean speech coefficients $S_{k,\ell}$ are assumed to follow a complex circular-symmetric Gaussian distribution. For this, the parameters of MOSIE [31] are set to $\nu = 1$ and $\beta = 0.001$, which approximates the Gaussian LSA [28]. For the second setup, the shape is reduced to $\nu = 0.2$, i.e., a super-Gaussian LSA is employed. To limit speech distortions, the gain is limited such that attenuations larger than 12 dB are prevented. This limit is applied throughout the chapter if not stated otherwise. The applied gains for the Gaussian and super-Gaussian case are shown in Fig. 5.6.

The upper row in Fig. 5.6 shows that the overestimations of the *a priori* SNR $\xi_{k,\ell}$, e.g., between spectral harmonics, result in a poor suppression for the MLSE-based enhancement schemes when using a Gaussian estimator ($\nu = 1$). The non-MLSE-based enhancement scheme is, however, not affected and achieves high suppression values between harmonics. As discussed in Section 5.4, this behavior can be explained from Fig. 5.3. For $\nu = 1$, the attenuation is mainly controlled by the *a priori* SNR $\xi_{k,\ell}$ where lower *a priori* SNRs $\xi_{k,\ell}$ lead to higher suppression values. From this it follows that an overestimation of $\xi_{k,\ell}$ results in lower attenuations as observed for the MLSE-based enhancement schemes. As a consequence, using Gaussian clean speech estimators (see Table 5.1) for MLSE-based enhancement schemes results in audible artifacts.

Interestingly, the lower row in Fig. 5.6 shows that the issues observed for $\nu = 1$ can be

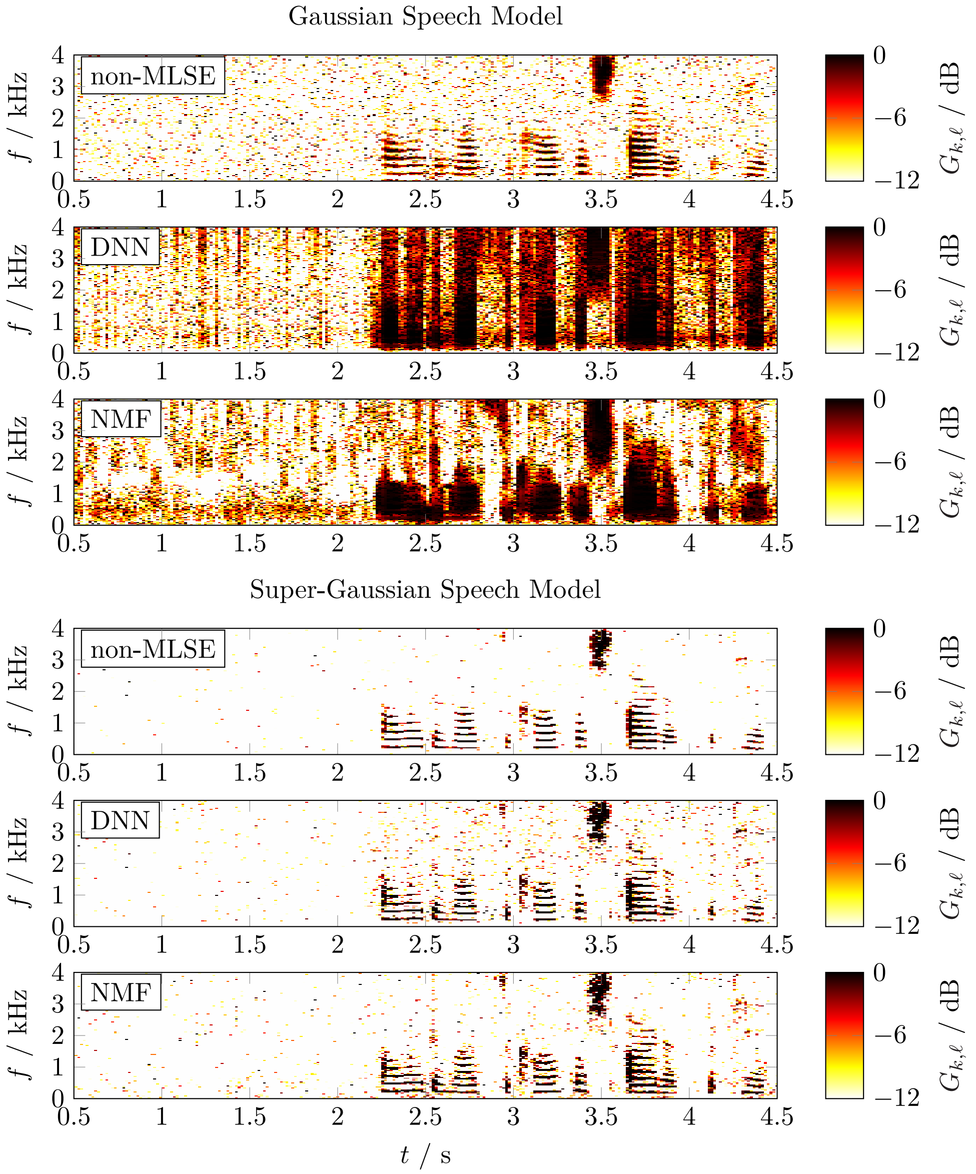

Fig. 5.6.: Gain applied to the noisy input coefficients $Y_{k,\ell}$ by MOSIE [31] for different MLSE-based enhancement schemes for the excerpt shown in Fig. 5.4. In the upper rows, $\nu = 1$ and $\beta = 0.001$ which approximates the Gaussian LSA proposed in [28] as shown in Table 5.1. In the lower rows, $\nu = 0.2$ and $\beta = 0.001$ is used which corresponds to a super-Gaussian LSA. Here, f denotes frequency and t time.

reduced if a super-Gaussian estimator ($\nu < 1$) is employed. In contrast to Fig. 5.6, noise is suppressed also between harmonics. Further, also higher attenuations are applied to the noise only segments. Considering Fig. 5.2 and Fig. 5.3, the behavior can be explained by the fact that lower shape values cause more suppression for low *a posteriori* SNRs $\gamma_{k,\ell}$. Hence, our key conclusion is that using super-Gaussian clean speech estimators, the background noise can be suppressed also when MLSE-based approaches are employed.

5.5. INSTRUMENTAL EVALUATION

We evaluate the performance of the different speech estimators using instrumental measures such as PESQ improvement scores [230] and SegSNR improvements [50], [71]. As described in Section 2.2.2, the improvements are based on the noisy signal, i.e., they are computed as the difference between the raw scores of the enhanced signal and the noisy signal. Additionally, the SegSSNR and the SegNR [50] are employed to quantify the speech distortions and noise suppression, respectively. Information about these two measures can also be found in Section 2.2.2.

For this evaluation, we use 128 sentences from the TIMIT core set. Again, it is ensured that the amount of audio material is balanced between genders. The clean speech signals are artificially corrupted by the same noise types used for training the NMF-based enhancement scheme. The SNRs are ranging from -5 dB to 20 dB in 5 dB steps. For each sentence, the segment of the noise signal where the speech signals are embedded in is randomly chosen. The instrumental measures are only evaluated after a two second initialization period to avoid initialization artifacts that may bias the results. Similarly, also the SNRs used for the artificial mixing are determined based on the signal powers in speech presence. Further, the noise segments that were used for training the NMF-based enhancement scheme are excluded in the evaluation for all enhancement schemes, i.e., also for the non-MLSE-based and the DNN-based enhancement schemes. This is done to make the enhancement schemes more easily comparable.

5.5.1. Performance Impact of MOSIE's Parameters

In this section, we analyze how the choice of the shape and the compression parameter influences the performance of clean speech estimators if used for the MLSE-based enhancement schemes.

Fig. 5.7 shows the PESQ improvement scores for MOSIE [31] as a function of the shape parameter ν and the compression parameter β. The graphs depict the average over all considered noise types and speech files for two different input SNRs. For the non-MLSE-based enhancement scheme, increasing super-Gaussianity ($\nu < 1$) and compression ($\beta < 1$) slightly improves the speech quality predicted by PESQ. However, the key message is that for the MLSE-based enhancement schemes, increasing super-Gaussianity ($\nu < 1$) and compression ($\beta < 1$) improves the signal quality predicted by PESQ considerably stronger.

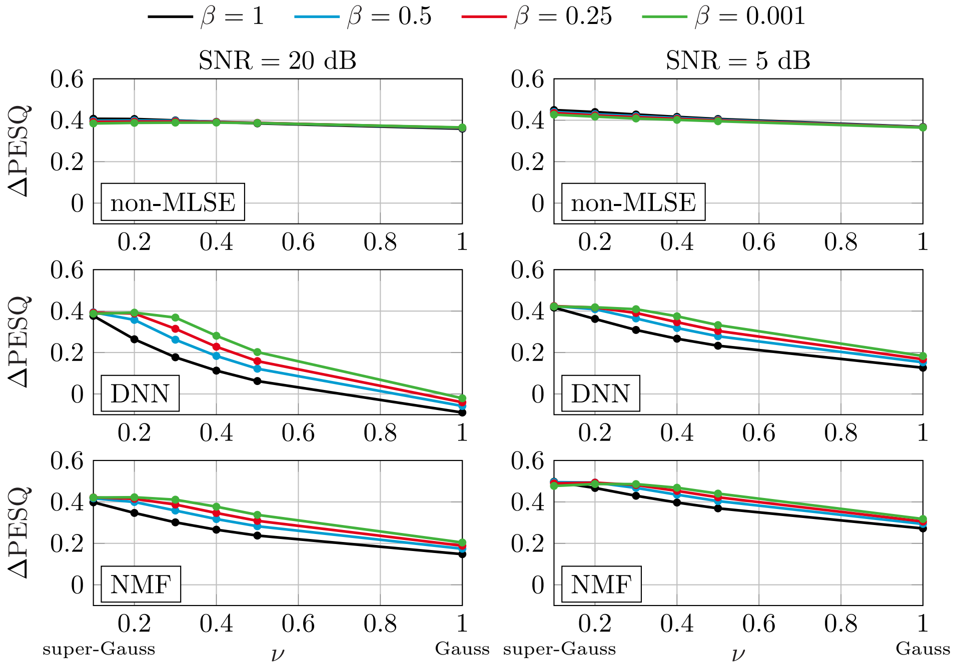

Fig. 5.7.: PESQ improvement scores of MOSIE [31] for all considered enhancement schemes in dependence of the shape ν and compression β. For relations to other clean speech estimators, see Table 5.1.

5.5.2. Comparison with Common Enhancement Schemes

In this final part of the evaluation section, we compare the super-Gaussian estimators, i.e., MOSIE [31] to Gaussian approaches. To demonstrate that super-Gaussian estimators considerably improve the performance of MLSE-based methods, we use the following two parameter settings for MOSIE [31]: $\beta = 0.001, \nu = 0.2$ and $\beta = 1, \nu = 0.2$. The parameters are chosen as a compromise such that all MLSE-based enhancement schemes yield satisfying results.

Fig. 5.8 shows PESQ improvement scores and segmental SNR measures for the considered enhancement schemes. The results again show that for the non-MLSE-based enhancement scheme, a super-Gaussian estimator only slightly improves the performance. Contrarily, the super-Gaussian setup for MOSIE [31] performs considerably better than the Gaussian clean speech estimator, i.e., the Gaussian STSA [12] and the Gaussian LSA [28], if the MLSE-based estimators are considered. As shown in Section 5.4, the suppression capability of the Gaussian approaches is mainly controlled by the *a priori* SNR resulting in low suppressions between harmonics for the MLSE-based enhancement schemes where the *a*

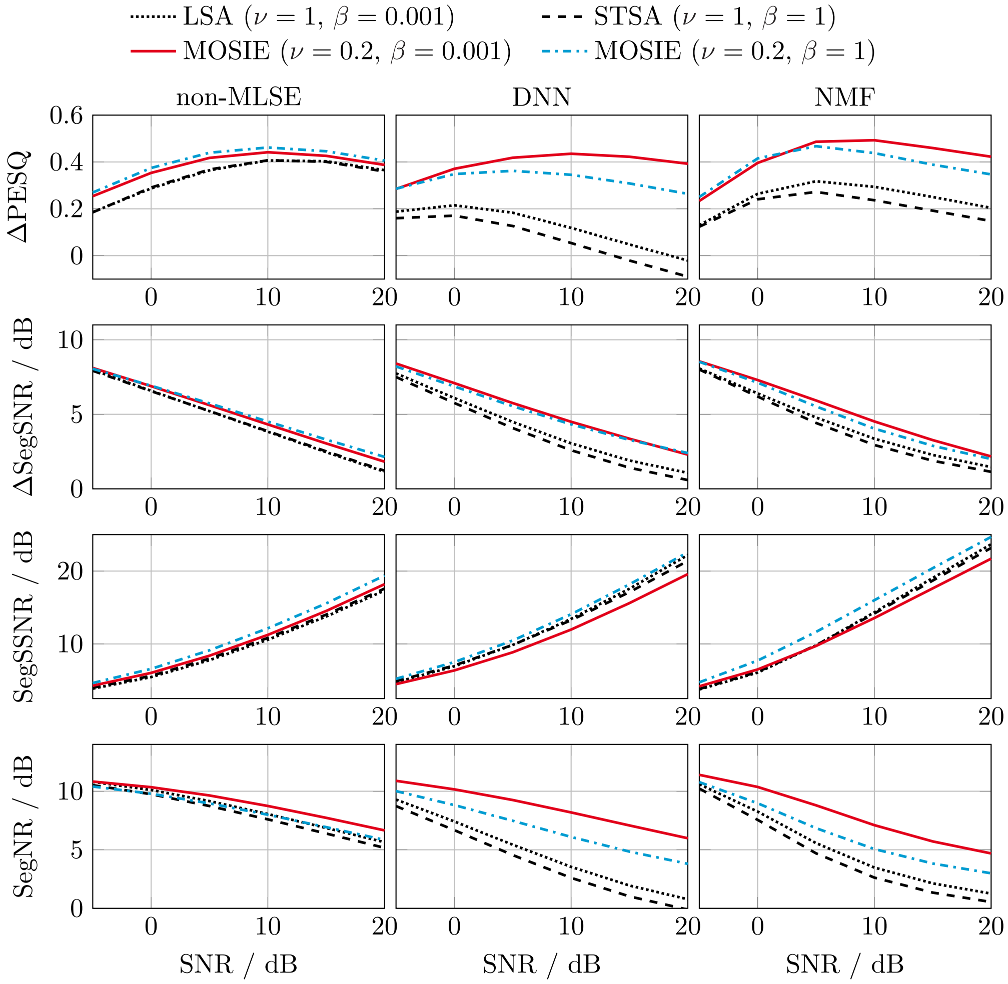

Fig. 5.8.: PESQ improvement scores and segmental SNR measures for different clean speech estimators employed in the non-MLSE-based, the DNN-based, and the NMF-based enhancement scheme. While LSA and STSA employ Gaussian speech priors, MOSIE ($\nu = 0.2, \beta = 0.001$) and MOSIE ($\nu = 0.2, \beta = 1$) represent modern super-Gaussian speech estimators (see Table 5.1).

priori SNR is overestimated. Here, this is reflected by the low segmental noise reduction values observed for the DNN-based and the NMF-based approach if the Gaussian STSA [12] or the Gaussian LSA [28] are employed. However, for the super-Gaussian estimators MOSIE ($\nu = 0.2, \beta = 0.001$) and MOSIE ($\nu = 0.2, \beta = 1$) the noise reduction is strongly increased and the residual noise, e.g., the noise between harmonics, is reduced. This comes with a slight increase in speech distortion for MOSIE ($\nu = 0.2, \beta = 0.001$) as visible in a decrease in SegSSNR. For MOSIE ($\nu = 0.2, \beta = 1$), the SegSSNR remains unchanged or is even slightly increased. Overall, the behavior of the super-Gaussian estimators helps to improve the quality predicted by PESQ and to improve the SegSNR.

5.6. SUBJECTIVE EVALUATION

As the results of instrumental measures cannot perfectly represent the impressions of human listeners, we verify the results using a subjective listening test. For this, we employ a multi-stimulus test with hidden reference and anchor (MUSHRA) [249]. In the experiment, two different acoustic scenarios are tested: traffic noise and babble noise both at an SNR of 5 dB. For both acoustic scenarios, an utterance of a male and a female speaker taken from the TIMIT test set are used. These signals are processed by the non-MLSE-based enhancement scheme, the DNN-based enhancement scheme, and the NMF-based enhancement schemes. For all enhancement schemes, a Gaussian STSA ($\nu = 1, \beta = 1$) and a super-Gaussian STSA ($\nu = 0.2, \beta = 1$) are compared (see Table 5.1). Even though MOSIE with $\nu = 0.2$ and $\beta = 0.001$ achieves the highest scores in most instrumental measures, we use MOSIE with $\beta = 1$ in the subjective listening test as this configuration produces less musical artifacts.

In each trial, four signals are presented to the listeners: the noisy signals processed by the Gaussian and the super-Gaussian estimator, an anchor, and a hidden reference. The trials are repeated over all combinations of acoustic conditions, speakers and enhancement schemes. The reference signal is a noisy signal with an SNR of 17 dB. Finally, for the anchor, the clean speech utterance is filtered using a low-pass filter at a cutoff frequency of 4 kHz and mixed at an SNR of -5 dB. This signal is processed using a non-MLSE-based enhancement scheme where the noise PSD is estimated using [70] and the speech PSD is obtained using the decision-directed approach [12] with a smoothing constant set to 0.9. A Wiener filter with a minimum gain of -20 dB is employed to obtain the anchor. The sound examples used in the experiment are also available at https://uhh.de/inf-sp-tasl2018a.

A total of 13 subjects have participated in the MUSHRA. The test took place in a quiet office and the subjects listened to diotic signals played back through headphones (Beyerdynamic DT-770 Pro 250 Ohm) through a RME Fireface UFX+ sound card. The test was conducted in two phases. In the first phase, the subjects were asked to listen to a subset of the files used in test such that they can familiarize themselves with the different signals. During this training phase, the listeners were also asked to set the level of the headphones to a comfortable level. In the second phase, the listener's task was to judge the overall quality of the signals on a scale ranging from 0 to 100, where 0 was labeled with

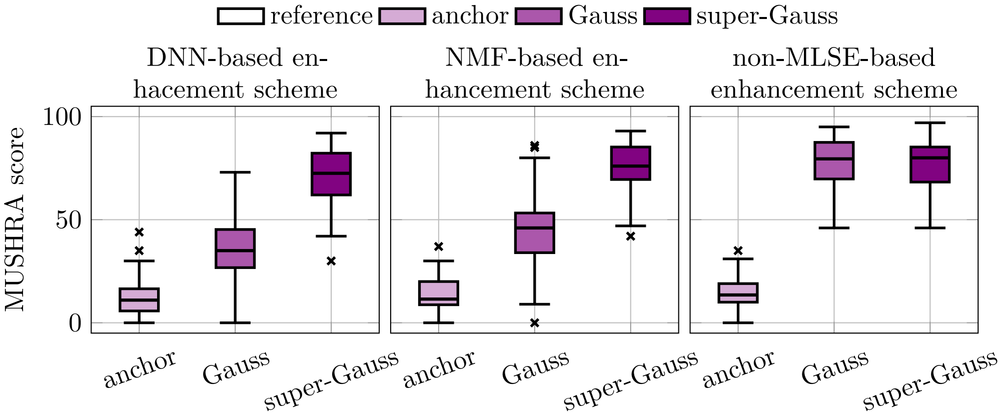

Fig. 5.9.: Box plot of the subjective ratings for different enhancement schemes.

“bad” and 100 with “excellent”. The order of presentations of algorithms and conditions were randomized between all subjects.

The obtained MUSHRA scores are summarized in Fig. 5.9 using box plots. The upper and the lower edge of the box show the upper and lower quartile while the bar within the box is the median. The upper whisker reaches to the largest data point that is smaller than the upper quartile plus 1.5 times the interquartile range. The lower whisker is defined analogously. The crosses denote outliers that do not fall in the range spanned by both whiskers. For each box plot, the results of all acoustic conditions and speakers are pooled, which yields 52 data points. The result show that all participants were able to detect the hidden reference, which had to be rated with 100, and that the anchor was consistently given the lowest scores. Further, the results clearly confirm that for the DNN-based and the NMF-based enhancement scheme, the sound quality of the super-Gaussian estimator is considered better than the Gaussian estimator. For the non-MLSE-based estimator, however, the MUSHRA scores of the Gaussian and the super-Gaussian estimator are nearly the same.

Finally, a brief statistical analysis of the results confirms that the differences in MUSHRA scores between the Gaussian and super-Gaussian estimators are statistically significant for the MLSE-based enhancement schemes. For the used statistical tests, a significance level of 5 % is employed. We apply a Wilcoxon signed-rank test to test for the difference in medians between the MUSHRA scores of the Gaussian and super-Gaussian estimators. This test is employed as the Shapiro-Wilk test indicates that the data is not Gaussian distributed for all conditions. The different enhancement schemes, i.e., the MLSE-based approaches and the non-MLSE-based approach, are treated separately. Considering the difference between the Gaussian and super-Gaussian clean speech estimators for the MLSE-based approaches, the differences are significant in both cases (DNN: $p < 0.001$, NMF: $p < 0.001$). Comparing the estimators for the non-MLSE-based algorithm reveals no significant difference ($p = 0.55$). Hence, the subjective listening tests confirm the previously obtained results of the instrumental measures.

5.7. SUMMARY

In this chapter, super-Gaussian clean speech estimators have been analyzed in the context of ML-based speech enhancement approaches that employ spectral envelope models. We refer to these approaches as MLSE. In the analysis part, we showed that the usage of envelope models results in an overestimation of the *a priori* SNR, e.g., between speech spectral harmonics. As a consequence, using Gaussian estimators, noise between harmonic structures cannot be reduced such that residual noises remain after the enhancement. However, in this chapter, we show that employing super-Gaussian clean speech estimators, such as MOSIE [31], leads to a reduction of the undesired residual noise. This interesting result stems from the higher attenuation that is applied by the super-Gaussian estimators if the *a posteriori* SNRs are low. This allows the estimators to compensate for the overestimated *a priori* SNRs without any further post-processing steps. As a consequence, we showed via theoretical analysis and experimental evaluation that for MLSE-based enhancement schemes, super-Gaussian estimators have a much larger effect on improving the enhancement performance than for classic non-MLSE-based enhancement schemes. Sound examples of the considered algorithms are given at https://uhh.de/inf-sp-tasl2018a.

CHAPTER 6

SUPER-GAUSSIAN MLSE-BASED SPEECH ENHANCEMENT UNDER THE MIXMAX MODEL

This chapter is an extension to the work presented in Chapter 5. Here, we show that the additional noise suppression of super-Gaussian speech priors is not restricted to estimators that have been derived under an additive signal model in the spectral domain. It can also be observed for MLSE-based speech enhancement algorithms that operate in the log-spectral domain, e.g., [26], [101], [102], [112], [250]. Often, approximations of the additive mixing model in Section 2.1.2 are used to simplify statistical inference for the speech log-spectral coefficients. In this chapter, such an approximation has been employed where the noisy log-spectral coefficients are modeled as the maximum of the speech and noise coefficients. This is referred to as MixMax model [112] or log-max approximation [102]. In [112], it has been motivated by the empirical finding that the approximation yields spectral representations which are visually similar to the results that are obtained if the additive mixing model is used in the time domain. The validity of this approximation has been further supported in [251] where it has been shown that the MixMax model is the MSE optimal estimator of the *noisy* log-spectral coefficients if the phase of the complex speech and noise coefficients is uniformly distributed. Further, it is argued in [102] that the error of the MixMax approximation has only a considerable influence if two sources have the same energy in time and frequency. Consequently, as speech has a sparse spectral representation and is uncorrelated to the noise, time-frequency points are often dominated by either speech or noise. From this, the practical expedience is concluded in [102].

The MixMax model is commonly used in combination with ML-based approaches where speech and noise are modeled using Gaussian distributions in the log-spectral domain [26], [101], [102], [112], [250] as in Section 2.1.3. In [112], it has been used to adapt the clean speech models to the background noise for robust speech recognition. In the context of speech enhancement, it has been used to infer the log-spectrum of the target speech from noisy observations [26], [101] or mixtures of multiple speakers [102], [250] in an MSE optimal way. In this chapter, we show that the MixMax based clean speech estimator can be interpreted as a super-Gaussian LSA similar to [31], [32]. For this, the relationship between spectral and log-spectral coefficients described in Section 2.1.3 [83] is exploited. This relationship also allows the combination of ML-based enhancement schemes based on

This chapter is partly based on:

[219] R. Rehr and T. Gerkmann, “MixMax approximation as a super-Gaussian log-spectral amplitude estimator for speech enhancement,” in *Interspeech*, Stockholm, Sweden, Aug. 2017, © 2017 ISCA.

the MixMax model and non-ML-based speech and noise PSD estimators such as [12], [29], [70], [82]. As in Chapter 5, we employ the MixMax model in an MLSE speech enhancement scheme similar to [26]. We show that the MixMax based speech estimator [112] causes less artifacts in the background noise compared to super-Gaussian LSA [31], [32] without degrading the speech quality in comparison to the method in Chapter 5.

First, we recapitulate the MixMax based clean speech estimator in Section 6.1. In Section 6.2, we analyze the gain functions that result for the MixMax based clean speech estimator using the relationships in Section 2.1.3 [83]. In Section 6.3, the MixMax based estimator is compared to the super-Gaussian LSA [31], [32] within an MLSE-based enhancement scheme and Section 6.4 summarizes the chapter.

6.1. MIXMAX BASED SPEECH ESTIMATOR

In this section, we recapitulate the MSE optimal estimator of the log-spectral speech coefficients that results from the MixMax model. This estimator operates on the STFT of the input signal. The MixMax model considers the log-spectra of the noisy input $y_{k,\ell}^{(\log)}$ which are defined in (2.17) on page 33. The MixMax signal mixing model [112], also known as log-max approximation [102], is given by

$$y_{k,\ell}^{(\log)} = \max(s_{k,\ell}^{(\log)}, n_{k,\ell}^{(\log)}). \tag{6.1}$$

Under the model in (6.1), the distribution of the noisy log-spectral coefficients $y_{k,\ell}^{(\log)}$ is given by [112]

$$f_y(y_{k,\ell}^{(\log)}) = f_s(y_{k,\ell}^{(\log)})F_n(y_{k,\ell}^{(\log)}) + f_n(y_{k,\ell}^{(\log)})F_s(y_{k,\ell}^{(\log)}). \tag{6.2}$$

Here, $f_s(\cdot)$ and $F_s(\cdot)$ denote the PDF and the cumulative distribution function (CDF) of the speech log-spectral coefficients $s_{k,\ell}^{(\log)}$, respectively. Similarly, $f_n(\cdot)$ and $F_n(\cdot)$ denote the PDF and the CDF of the background noise. In [26], [101], [112], $f_s(\cdot)$ and $f_n(\cdot)$ are set to a Gaussian distribution as in (2.18) and (2.19) on page 33, respectively. A Gaussian model in the log-spectral domain differs from modeling the complex Fourier coefficients by a Gaussian distribution. If a Gaussian distribution is used to model the spectral Fourier coefficients, it can be shown that the log-spectral coefficients follow an exp-gamma distribution [225] (see also (7.23) in Chapter 7). Correspondingly, the use of a Gaussian distribution in the log-spectral domain results in a non-Gaussian distribution in the complex Fourier domain. If $s_{k,\ell}^{(\log)}$ and $n_{k,\ell}^{(\log)}$ are modeled by a Gaussian PDF and the mixing model in (6.1) is used, the MSE optimal estimator of the speech log-spectral speech coefficients, i.e., $\hat{s}_{k,\ell}^{(\log)} = \mathbb{E}\{s_{k,\ell}^{(\log)}|y_{k,\ell}^{(\log)}\}$, is [26], [112]

$$\hat{s}_{k,\ell}^{(\log)} = b_{k,\ell}y_{k,\ell}^{(\log)} + (1 - b_{k,\ell})\left(\mu_{k,\ell}^{s} - \lambda_{k,\ell}^{s}\frac{f_s(y_{k,\ell}^{(\log)})}{F_s(y_{k,\ell}^{(\log)})}\right). \tag{6.3}$$

In (6.3), $b_{k,\ell}$ is given by $b_{k,\ell} = f_s(y_{k,\ell}^{(\log)}) F_n(y_{k,\ell}^{(\log)}) / f_y(y_{k,\ell}^{(\log)})$ [26], [112]. For obtaining an estimate of the spectral clean speech coefficients $\hat{S}_{k,\ell}$, the log-spectral transformation in (2.17) is reverted and the result is combined with the noisy phase $\Phi_{k,\ell}^{y}$ as in (2.5) on page 29. For obtaining the time-domain representation of the enhanced signal, the inverse STFT is performed as in (2.8) on page 30.

The parameters of $f_s(\cdot)$ and $f_n(\cdot)$, i.e., the means $\mu_{k,\ell}^{s}$ and $\mu_{k,\ell}^{n}$, as well as, the variances $\lambda_{k,\ell}^{s}$ and $\lambda_{k,\ell}^{n}$, can be directly trained in the log-spectral domain [26], [101]. Here, however, we propagate these quantities from spectral estimates of the speech PSD $\Lambda_{k,\ell}^{s}$ and the noise PSD $\Lambda_{k,\ell}^{n}$ using the relationship described in Section 2.1.3 [83]. Accordingly, the same super-Gaussian distribution as in Chapter 5 is employed for the spectral speech coefficients $S_{k,\ell}$, i.e., the magnitudes are assumed to follow a χ-distribution as in (2.16) while the phase is assumed to be uniformly between $-\pi$ and π. The spectral noise coefficients $N_{k,\ell}$ are assumed to follow a circular-symmetric complex Gaussian distribution as in (2.14) on page 32. Consequently, the means $\mu_{k,\ell}^{s}$ and $\mu_{k,\ell}^{n}$ can be determined using (2.20) on page 34. The variances $\lambda_{k,\ell}^{s}$ and $\lambda_{k,\ell}^{n}$ are given by (2.21) on page 34. As the spectral noise coefficients $N_{k,\ell}$ are assumed to follow a Gaussian distribution, $\nu = 1$ has to be used in (2.20) and (2.21), respectively. To reflect super-Gaussian distributed spectral coefficients in the log-spectral domain, $0 < \nu < 1$ has to be used to determine $\mu_{k,\ell}^{s}$ and $\lambda_{k,\ell}^{s}$, respectively.

The relationship between spectral and log-spectral coefficients in (2.20) and (2.21), allows the spectral speech PSDs $\Lambda_{k,\ell}^{s}$ and noise PSDs $\Lambda_{k,\ell}^{n}$ to be used in combination with the MixMax based clean speech estimator in (6.3). As a result, ML-based algorithms using log-spectral representations can easily be used in combination with speech and noise spectral PSD estimators, e.g., [12], [70], [82]. Hence, the advantages of both domains can be exploited: on the one hand, many different non-ML approaches are available for spectral PSD estimation [12], [29], [82] while, on the other hand, log-spectral representations are better suited for constructing generalizing pre-trained speech models.

6.2. ANALYSIS OF THE MIXMAX GAIN FUNCTIONS

The relationship between the spectral parameters and the log-spectral ones, i.e,. (2.20) and (2.21) on page 34, allows the estimator given by (6.3) to be interpreted as a real-valued spectral gain function $G_{k,\ell}$ that depends on the spectral *a priori* SNR $\xi_{k,\ell} = \Lambda_{k,\ell}^{s} / \Lambda_{k,\ell}^{n}$ and *a posteriori* SNR $\gamma_{k,\ell} = |Y_{k,\ell}|^2 / \Lambda_{k,\ell}^{n}$. Such an interpretation is usually reserved for MSE optimal estimators that have been defined in the spectral domain, e.g., [12], [28], [31]–[33]. In Figure 6.1, the gain function of the MixMax based estimator is shown that results if the relationship in (2.20) and (2.21) is exploited. It is compared to the super-Gaussian LSA [31], [32] which is based on an additive mixing model in the time domain. This estimator is implemented using MOSIE proposed in [31] which generalizes [32] if small values are employed for the compression parameter which is set to $\beta = 0.001$ here. This estimator is chosen as it is also an estimator of the log-spectral amplitudes, i.e., the $\hat{S}_{k,\ell}$ is

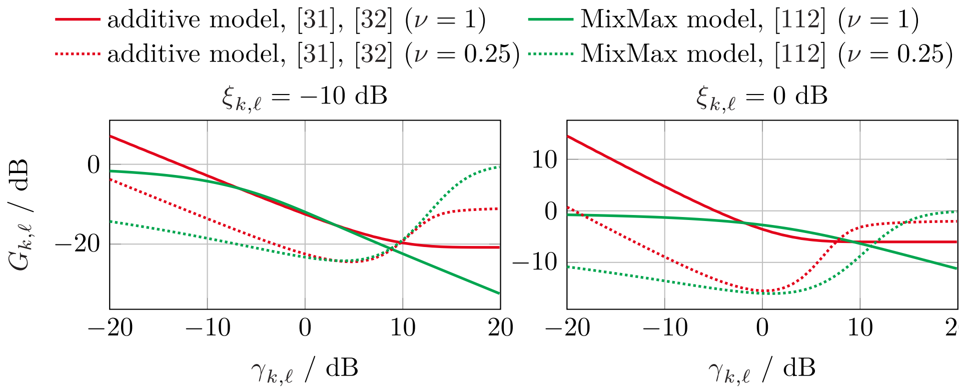

Fig. 6.1.: Gain functions of the super-Gaussian LSA [31], [32] derived under an additive model in the spectral domain and the MixMax based clean speech estimator.

estimated which minimizes $\mathbb{E}\{\log(|S_{k,\ell}|) - \log(|\hat{S}_{k,\ell}|)\}$. Additionally, the same statistical model for the spectral coefficients is used as in the derivation of (2.20) and (2.21).

For $\nu = 1$, the suppression of both estimators mainly depends on the *a priori* SNR $\xi_{k,\ell}$. With increasing *a priori* SNR, the applied suppression decreases. Differences can be observed for very high and low *a posteriori* SNRs $\gamma_{k,\ell}$ where the MixMax model results in lower gains. Reducing ν, i.e., assuming a super-Gaussian distribution for $S_{k,\ell}$, has a similar effect for both gain functions. In both cases, a higher suppression is applied if the *a posteriori* SNR $\gamma_{k,\ell}$ is close to 0 dB. In Chapter 5, [218], it has been shown that this behavior is beneficial if pre-trained speech models are employed that only represent the spectral speech envelope. In this case, it allows the suppression of the noise between harmonics which are not represented by the speech models. Little suppression is applied if the *a posteriori* SNR $\gamma_{k,\ell}$ is high, which results in lower speech distortions. This behavior is characteristic for super-Gaussian estimators.

6.3. EVALUATION

In this section, the MixMax based estimator and the super-Gaussian LSA [32], again realized as in Section 6.2 using MOSIE, [31], are embedded in an MLSE-based enhancement scheme similar to [26]. The used enhancement scheme is identical to the DNN method described in Section 5.2. We show that the MixMax based estimator yields similar results in comparison to a super-Gaussian LSA in terms of speech quality. For evaluating the speech quality PESQ improvement scores are used [230] as described in Section 2.2.2. However, the MixMax approach results in less musical tones as indicated by a modified version of the log-kurtosis ratio [252]. First, the parameters and evaluation setup are considered, and the results are presented afterwards.

6.3.1. Evaluation Setup

The speech signals processed by the enhancement scheme are sampled at a rate of 16 kHz. For the STFT, 32 ms segments with 50 % overlap are employed and a square-root Hann window is used for spectral analysis and synthesis.

The MLSE-based algorithm considered in this chapter is the same as the DNN-based approach described in Section 5.2. Further, the same setup is used, i.e., the input features, the hidden layers and the hidden units therein are configured in the same way as in Section 5.2. As in Chapter 5, the weights of the DNN are optimized prior to the processing using 1196 sentences taken from the training set of the TIMIT database [240]. Again, it has been ensured that the training sentences are gender and phonetically balanced. As in [26], only clean speech data is used for training to avoid noise specific adaptations of the DNN. The targets of the DNN are given by the TIMIT annotation which are represented by one-hot encoded target vectors. For each phoneme q, the phoneme dependent speech PSD $\Lambda_k^{s|q_\ell}$ is determined by averaging all speech periodograms $|S_{k,\ell}|^2$ labeled as the corresponding phoneme q in the TIMIT annotation as in (5.3) on page 82.

Due to the averaging of phonemes, only spectral envelopes can be represented by the pre-trained speech PSDs $\Lambda_k^{s|q_\ell}$. Similar to Chapter 5, [218], we show that using the relationship in Section 2.1.3 and modeling the spectral speech coefficients $S_{k,\ell}$ using a super-Gaussian distribution also allows the MixMax based estimator to suppress noise between spectral harmonics if MLSE speech models are employed. Hence, we show results for the Gaussian case, i.e., $\nu = 1$, and for the super-Gaussian case, i.e., $\nu = 0.25$. The considered gain functions $G_{k,\ell}$ are limited such that a time-frequency bin may not be suppressed by more than 15 dB.

We use PESQ [230] improvement scores as instrumental measure for the speech quality and a modified version of the log-kurtosis ratio proposed in [252] to evaluate the noise quality in terms of musical tones. Similar to [252], we define the log-kurtosis ratio as

$$\Delta\kappa^{(\log)} = \log\left(\frac{\kappa_{\tilde{n}}}{\kappa_n}\right), \tag{6.4}$$

where $\kappa_{\tilde{n}}$ is the empirical kurtosis of the processed noise whereas κ_n denotes the empirical kurtosis of the unprocessed noise. The kurtosis can be considered a measure of outliers and a positive log-kurtosis ratio $\Delta\kappa^{(\log)}$ is expected if the processed signals contains musical tones. Instead of estimating the kurtosis for each segment ℓ and using the average along time as κ_n and $\kappa_{\tilde{n}}$, we estimate the kurtosis per frequency band as

$$\kappa_n[k] = \frac{\frac{1}{|\mathbb{L}_k^{(n)}|}\sum_{\ell\in\mathbb{L}_k^{(n)}}\left[|N_{k,\ell}|^2 - \overline{|N|_k^2}\right]^4}{\left(\frac{1}{|\mathbb{L}_k^{(n)}|}\sum_{\ell\in\mathbb{L}_k^{(n)}}\left[|N_{k,\ell}|^2 - \overline{|N|_k^2}\right]^2\right)^2}. \tag{6.5}$$

In (6.5), the set $\mathbb{L}_k^{(n)}$ contains only segments in the kth frequency band where the back-

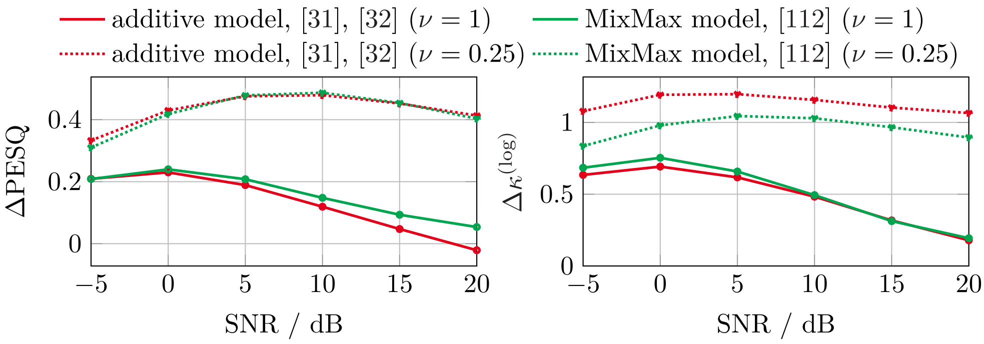

Fig. 6.2.: PESQ improvement score (left panel) and log-kurtosis ratio (right panel) of the super-Gaussian LSA [31], [32] and the MixMax based estimator averaged over all noise types.

ground noise is dominant as

$$\mathbb{L}_k^{(n)} = \{\ell |\, 10 \log_{10} \left(|S_{k,\ell}|^2 / |N_{k,\ell}|^2 \right) < \tau\}. \tag{6.6}$$

Here, τ is a threshold value which is set to -10 dB in this evaluation. The cardinality of $\mathbb{L}_k^{(n)}$ is denoted by $|\mathbb{L}_k^{(n)}|$ and $\overline{|N|}_k^2$ is given by $\overline{|N|}_k^2 = \sum_{\ell \in \mathbb{L}_k^{(n)}} |N_{k,\ell}|^2 / |\mathbb{L}_k^{(n)}|$. Finally, κ_n is given by $\kappa_n = \sum_{k=0}^{K-1} \kappa_n[k]/K$, where K denotes the number Fourier coefficients. Similarly, the kurtosis of the processed noise periodogram $|\tilde{N}_{k,\ell}|^2$ is determined.

For testing, 128 sentences taken from the TIMIT test corpus [240] are used where, again, a gender balanced set is used. The clean speech sentences are corrupted by babble noise, factory 1 noise and pink noise taken from the NOISEX-92 database [239] at SNRs ranging from -5 dB to 20 dB. Additionally, a modulated version of the pink noise similar to [70] and a traffic noise taken from https://www.freesound.org/s/75375/ is used.

6.3.2. Results

Fig. 6.2 depicts the PESQ improvement scores and log-kurtosis ratio obtained for the used variant of the super-Gaussian LSA [31], [32] and the MixMax based clean speech estimator. The results are averaged over all noise types and the left panel of Fig. 6.2 shows the PESQ improvement scores. As in Chapter 5, using super-Gaussian speech models, i.e., $\nu < 1$, results in considerably higher PESQ improvements than a Gaussian assumption ($\nu = 1$). Again, this effect can be explained by the higher suppression that is achieved using super-Gaussian models as indicated by the gain function $G_{k,\ell}$ depicted in Fig. 6.1. The log-kurtosis in the right panel of Figure 6.2 shows low values if Gaussian models are employed, i.e., $\nu = 1$, and rises for super-Gaussian models ($\nu = 0.25$). As expected, the log-kurtosis is generally higher for super-Gaussian estimators ($\nu = 0.25$) than for

for Gaussian estimators ($\nu = 1$). Of the two super-Gaussian estimators, however, the MixMax based approach introduced in this chapter achieves the lower log-kurtosis ratio, indicating less spectral outliers such as musical noise. We note that if the babble noise and the factory noise are considered separately, the log-kurtosis ratio is higher for the MixMax based estimator. In informal listening tests, however, no disturbing musical tones could be noticed and both clean speech estimators have been found to sound very similar in these highly non-stationary noise types. Part of the reason may be that estimating the fourth-order moments in the kurtosis metric is rather difficult for these noise types. This possibly renders the log-kurtosis ratio unreliable for non-stationary noises. However, for other noise types, such as pink noise and traffic noise, it is clearly audible that the MixMax based estimator causes less artifacts. Hence, the overall averaged log-kurtosis ratio in Figure 6.2 adequately reflects the trend that the MixMax based estimator results in less musical tones if super-Gaussian speech models are employed. This is confirmed in informal listening tests. Further, this is achieved while maintaining the same PESQ scores as the super-Gaussian LSA [31], [32]. Audio examples can be found at https://www.inf.uni-hamburg.de/en/inst/ab/sp/publications/interspeech2017.html.

6.4. SUMMARY

In this chapter, we showed that the MixMax based estimator used in [26], [101] can be interpreted as a super-Gaussian LSA. For this, the relationship described in [83] (see also Section 2.1.3) is exploited. Additionally, this allows the combination of pre-trained log-spectral models with spectral speech and noise PSD estimators for speech enhancement. Further, the MixMax based speech estimator is compared to the super-Gaussian LSA proposed in [31], [32] using an MLSE-based speech enhancement scheme. The instrumental measures indicate that the speech quality of both estimators is nearly identical in the super-Gaussian case while the MixMax based speech estimator causes less musical artifacts in the residual background noise.

CHAPTER 7

COMBINATION OF NON-ML AND ML-BASED ALGORITHMS FOR ENVELOPE BASED SPEECH ENHANCEMENT

This chapter presents another method for suppressing the background noise between speech spectral harmonics if MLSE models are employed. In contrast to the approaches presented in Chapter 5 and Chapter 6, where super-Gaussian models have been exploited, a combination of a non-MLSE and an MLSE-based speech enhancement approach is proposed. The combination is embedded in a statistical framework where the enhancement algorithms are represented by different statistical models. The models describe the likelihood of the noisy observations allowing the method which is best suited for enhancing a noisy time-frequency point to be identified. This results in a soft mixing of the estimated clean speech spectra obtained from the combined enhancement schemes.

Instead of using a DNN-based phoneme recognizer or an NMF-based approach, the MLSE approach in this chapter is GMM-based. It is similar to the feature enhancement methods presented in [109], [253]–[255], where the VTS approximation is used to incorporate spectral noise PSD estimates in cepstral or log-spectral feature vectors for noise robust speech recognition. In [103]–[105], the VTS approach has been used to develop ML-based speech enhancement algorithms. These algorithms model the log-spectral speech coefficients and possibly also the log-spectral noise coefficients using a GMM. The clean speech coefficients are, hence, inferred in the log-spectral domain. For this, a VTS is used to approximate the additive mixing model such that statistical inference becomes feasible in the log-spectral domain. For such GMM-based speech enhancement approaches, the number of mixtures determines the resolution of the learned speech model, i.e., whether it describes only the spectral envelope or if it also includes the fine structure. To obtain a high-resolution estimate of the clean speech, i.e., an estimate which includes the vocal tract shape as well as the pitch, a large amount of mixtures is employed in [103]. This, however, increases the demands with respect to memory and computational complexity. In [104], a reduced amount of mixtures is employed resulting in an MLSE speech model which may only represent the speech spectral envelopes, but typically not the spectral fine structure. As a consequence, noise between the speech spectral harmonics is not reduced. This problem is mitigated in [104] by applying a post-filter based on a harmonic model in voiced speech. In [26], [27], the residual noise is suppressed by applying an estimate of the

This chapter is partly based on:

[220] R. Rehr and T. Gerkmann, “A combination of pre-trained approaches and generic methods for an improved speech enhancement,” in *ITG Conference on Speech Communication*, Paderborn, Germany, Oct. 2016, pp. 51–55, © 2016 VDE Verlag.

speech presence probability as a post-filter. In [241], an explicit model of the excitation has been incorporated to avoid this issue.

In this chapter, we show that combining non-MLSE with MLSE-based approaches reduces the noise between spectral harmonics which cannot be reduced using only MLSE-based approaches. As a consequence, the sound quality in terms of PESQ scores improves in comparison to the sole application of a pure MLSE-based enhancement method. Further, it is shown that the combination also outperforms a pure non-MLSE estimator based on the LSA [28] and a harmonic model based post-filter applied to the output of an MLSE-based estimator similar to [104].

This chapter is structured as follows: First, the employed signal model and the statistical relations are described in Section 7.1. In Section 7.2 and Section 7.3, the used MLSE-based speech enhancement and non-MLSE-based enhancement methods are summarized. Section 7.4 presents the proposed combination which is evaluated in Section 7.5. Finally, the chapter is summarized in Section 7.6.

7.1. SIGNAL MODEL AND STATISTICAL MODELS

The considered MLSE-based enhancement method learns the statistics of the clean speech coefficients in the log-spectral domain. Together with cepstral domain models, this representation is often preferred for learning approaches, e.g., in automatic speech recognition [173], [256]. The noisy input signal is processed in the log-spectral domain similar to Chapter 6. For this, the STFT framework given in Section 2.1.1 is employed and the definition of the log-spectrum given in (2.17) on page 33 is used. The estimate of the log-spectral clean speech coefficients $\hat{s}_{k,\ell}^{(\log)}$ allows the spectral gain function to be computed as

$$G_{k,\ell} = \exp([\hat{s}_{k,\ell}^{(\log)} - y_{k,\ell}^{(\log)}]/2). \tag{7.1}$$

As in (2.4) on page 29, this function is applied to the noisy input spectrum to obtain an estimate of the complex clean speech coefficients $\hat{S}_{k,\ell}$. As the phase information is removed with the transformation to the log-spectral domain in (2.17), this is equivalent to combining the estimated clean speech magnitude with the noisy phase. The time-domain representation of the enhanced signal is obtained using the overlap-add method after applying a synthesis window as in (2.8) on page 30.

In contrast to the MLSE-based enhancement scheme, non-MLSE-based enhancement schemes are generally derived in the spectral domain. Correspondingly, the statistical quantities, such as the PSDs, are estimated in the spectral domain. To be able to combine such spectral approaches with the considered MLSE enhancement approach, the spectral estimates are propagated to the log-spectral domain similar to Chapter 6. This allows the application of state-of-the-art noise PSD and speech PSD estimators, e.g., [70], [82], in combination with MLSE-based speech enhancement schemes operating in the log-spectral domain. The propagation is based on the equations given in Section 2.1.3, [226], [257]. In this chapter, specifically, the non-MLSE-based algorithms are based on the assumption that the complex DFT coefficients of speech, noise and the noisy observation, i.e., $S_{k,\ell}$, $N_{k,\ell}$, and $Y_{k,\ell}$, respectively, follow a zero-mean circular-symmetric Gaussian distribution. The respective variances are denoted by $\Lambda^s_{k,\ell}$, $\Lambda^n_{k,\ell}$, and $\Lambda^y_{k,\ell}$. Further, we assume that speech and noise are uncorrelated, i.e., $\Lambda^y_{k,\ell}$ can be expressed as $\Lambda^y_{k,\ell} = \Lambda^s_{k,\ell} + \Lambda^n_{k,\ell}$. Under the assumption that the spectral coefficients of speech and noise are both Gaussian distributed, the means in the log-spectral domain, i.e., $\mu^s_{k,\ell}$, $\mu^n_{k,\ell}$, and $\mu^y_{k,\ell}$, can be determined by using $\nu = 1$ and the respective spectral variance, i.e., $\Lambda^s_{k,\ell}$, $\Lambda^n_{k,\ell}$, or $\Lambda^y_{k,\ell}$, in (2.20) on page 34. Similarly, the log-spectral variances, which only depend on the shape parameter ν, can be determined by using $\nu = 1$ in (2.21). Because the Gaussian assumption ($\nu = 1$) is used for all signal components in the non-MLSE-based approaches, the propagated log-spectral variance is the same for all components. In [226], it has been derived analytically to be $\pi^2/6$, i.e., $\lambda^y_{k,\ell} = \lambda^s_{k,\ell} = \lambda^n_{k,\ell} = \pi^2/6$.

Furthermore, the log-spectral cross-covariance $\lambda^{sy}_{k,\ell}$ is used by one of the estimators considered in this chapter. It depends on the magnitude squared correlation coefficient $\rho^2_{k,\ell}$ between the spectral coefficients of noisy speech $Y_{k,\ell}$ and clean speech $S_{k,\ell}$, i.e.,

$$\rho^2_{k,\ell} = \frac{|\mathbb{E}\{S_{k,\ell} Y^*_{k,\ell}\}|^2}{\mathbb{E}\{|S_{k,\ell}|^2\}\mathbb{E}\{|Y_{k,\ell}|^2\}}. \tag{7.2}$$

In [226], [257], it has been shown that this quantity is related to the Wiener filter in the spectral domain

$$\rho^2_{k,\ell} = \frac{\Lambda^s_{k,\ell}}{\Lambda^s_{k,\ell} + \Lambda^n_{k,\ell}}. \tag{7.3}$$

With this and the Gaussian assumption for speech an noise in the spectral domain, the log-spectral cross-covariance $\lambda^{sy}_{k,\ell}$ can be determined using [226]

$$\lambda^{sy}_{k,\ell} = \sum_{i=1}^{\infty} \frac{\left(\rho^2_{k,\ell}\right)^i}{i^2}. \tag{7.4}$$

7.2. MLSE-BASED SPEECH ENHANCEMENT

In this section, the MLSE part of the proposed combination is presented. It is based on the work in [104], [253], [254]. It is assumed that the joint distribution of the log-spectral speech coefficients can be described by a GMM as

$$f(\mathbf{s}^{(\log)}_\ell | \mathbf{z}_\ell = z^{\mathrm{MLSE}}) = \sum_{q_\ell=1}^{Q} f(q_\ell) \left(\prod_{k=0}^{K/2} f(s^{(\log)}_{k,\ell} | q_\ell, z_{k,\ell} = z^{\mathrm{MLSE}}) \right) \tag{7.5}$$

$$= \sum_{q_\ell=1}^{Q} f(q_\ell) \left(\prod_{k=0}^{K/2} \mathcal{N}\left(s^{(\log)}_{k,\ell} | \mu_k^{s|q_\ell, z^{\mathrm{MLSE}}}, \lambda_k^{s|q_\ell, z^{\mathrm{MLSE}}} \right) \right). \tag{7.6}$$

Here, $\mathbf{s}_\ell^{(\log)} = \left[s_{0,\ell}^{(\log)}, s_{1,\ell}^{(\log)}, \ldots, s_{K/2,\ell}^{(\log)}\right]^T$ is a vector which comprises the frequency components of the speech log-spectrum at segment ℓ. Further, $\mathbf{z}_\ell = \left[z_{0,\ell}, z_{1,\ell}, \ldots, z_{K/2,\ell}\right]^T$ is a vector that contains a state indicator for each frequency bin of a segment ℓ. The state indicator is a latent random variable which is used later in Section 7.4 for the combination. As the MLSE-based approach is considered here, the state indicator is assumed to be z^{MLSE}, i.e., the state of the MLSE approach, for all frequency bins. Each mixture component, which are indexed by q_ℓ, is given by a Gaussian distribution which is denoted by $\mathcal{N}(\cdot)$. The parameters of the mixture components are the mean $\mu_k^{s|q_\ell,z^{\text{MLSE}}}$ and variance $\lambda_k^{s|q_\ell,z^{\text{MLSE}}}$. The log-spectral coefficients are assumed to be independent across frequency allowing each mixture component to be represented by a multiplication over all frequency bins. The probability $f(q_\ell)$ is the prior of the q_ℓth mixture component and Q denotes the number of mixtures. During training, which is performed prior to the application of this algorithm, the parameters $\mu_k^{s|q_\ell,z^{\text{MLSE}}}$ and $\lambda_k^{s|q_\ell,z^{\text{MLSE}}}$ and the prior probabilities $f(q_\ell)$ are determined. For this, the expectation maximization algorithm [87] is employed. By using the EM algorithm, the states q_ℓ are not directly related to phonemes as in Chapter 5 and Chapter 6. However, as the number of GMM components is chosen relatively low, the GMM will only be able to represent phoneme-like structures.

The linear relationship between speech components and noise components in the spectral domain in (2.3) on page 28 is in general non-linear in the log-spectral domain. Here, similar to [104], [253], [254], the relationship in the log-spectral domain (2.3) is approximated using a first-order VTS. Commonly, the phase information is omitted as originally proposed in [253], so $|Y_{k,\ell}|^2$ can be written as

$$|Y_{k,\ell}|^2 \approx |S_{k,\ell}|^2 + |N_{k,\ell}|^2. \tag{7.7}$$

This approximation omits the cross-term which additionally depends on the phase difference between speech and noise. While clearly a simplification, it is often used in VTS-based enhancement approaches. Under the reasonable assumption that speech and noise are uncorrelated, the cross-term cancels out on average, i.e., at least $\mathbb{E}\{|Y_{k,\ell}|^2\} = \mathbb{E}\{|S_{k,\ell}|^2\} + \mathbb{E}\{|N_{k,\ell}|^2\}$, e.g., [109], [223]. Similar approximations are also used in other pre-trained approaches, e.g., NMF [21], [117]. A study on how these approximations affect the quality of enhancement algorithms is given in [137]. While attempts for incorporating the cross-term exist [255], [258], [259], they typically increase the computational complexity. Thus, for simplicity, we stick to the simple model in (7.7) in this work. In the log-spectral domain, the relationship in (7.7) can be rewritten as

$$y_{k,\ell}^{(\log)} = \mathcal{V}(s_{k,\ell}^{(\log)}, n_{k,\ell}^{(\log)}) = \log\left\{\exp\left(s_{k,\ell}^{(\log)}\right) + \exp\left(n_{k,\ell}^{(\log)}\right)\right\}. \tag{7.8}$$

The non-linear mixing function $\mathcal{V}\left(s_{k,\ell}^{(\log)}, n_{k,\ell}^{(\log)}\right)$ is approximated using a first-order VTS with respect to the speech and noise components $s_{k,\ell}^{(\log)}$ and $n_{k,\ell}^{(\log)}$, as

$$y_{k,\ell}^{(\log)} \approx \mathcal{V}_s^{\mathcal{P}_0}(s_{k,\ell}^{(\log)} - \mathcal{P}_0{}^{(s)}) + \mathcal{V}_n^{\mathcal{P}_0}(n_{k,\ell}^{(\log)} - \mathcal{P}_0{}^{(n)}) + \mathcal{V}^{\mathcal{P}_0}, \tag{7.9}$$

Here, ${\mathcal{P}_0}^{(s)}$ and ${\mathcal{P}_0}^{(n)}$ form the linearization point $\mathcal{P}_0$ as $\mathcal{P}_0 = [{\mathcal{P}_0}^{(s)}, {\mathcal{P}_0}^{(n)}]$ and $\mathcal{V}^{\mathcal{P}_0} = \mathcal{V}({\mathcal{P}_0}^{(s)}, {\mathcal{P}_0}^{(n)})$. The symbols $\mathcal{V}_s^{\mathcal{P}_0}$ and $\mathcal{V}_n^{\mathcal{P}_0}$ denote derivatives with respect to $s_{k,\ell}^{(\log)}$ and $n_{k,\ell}^{(\log)}$ evaluated at $\mathcal{P}_0$. The linearization point is usually given by ${\mathcal{P}_0}^{(s)} = \mu_k^{s|q_\ell,z^{\mathrm{MLSE}}}$ and ${\mathcal{P}_0}^{(n)} = \mu_{k,\ell}^n$ and therefore depends on the mixture q_ℓ. As in [104], the approximation in (7.9) is used to determine the parameters of the likelihood of $s_{k,\ell}^{(\log)}$ given the q_ℓth mixture and the state indicator $f(y_{k,\ell}^{(\log)}|s_{k,\ell}^{(\log)}, q_\ell, z^{\mathrm{MLSE}})$ which is assumed to follow a Gaussian distribution. The mean and the variance of $f(y_{k,\ell}^{(\log)}|s_{k,\ell}^{(\log)}, q_\ell, z^{\mathrm{MLSE}})$ are obtained by determining the expected values $\mu_{k,\ell}^{y|s,q_\ell,z^{\mathrm{MLSE}}} = \mathbb{E}\{y_{k,\ell}^{(\log)}\}$ and $\lambda_{k,\ell}^{y|s,q_\ell,z^{\mathrm{MLSE}}} = \mathbb{E}\{(y_{k,\ell}^{(\log)} - \mu_{k,\ell}^{y|s,q_\ell})^2\}$ using the simplified $y_{k,\ell}^{(\log)}$ in (7.9). As the speech component $s_{k,\ell}^{(\log)}$ is given, the only remaining random variable is the noise $n_{k,\ell}^{(\log)}$. Thus, the mean and the variance are given by

$$\mu_{k,\ell}^{y|s,q_\ell,z^{\mathrm{MLSE}}} = \mathcal{V}_s^{\mathcal{P}_0}(s_{k,\ell}^{(\log)} - {\mathcal{P}_0}^{(s)}) + \mathcal{V}_n^{\mathcal{P}_0}(\mu_{k,\ell}^n - {\mathcal{P}_0}^{(n)}) + \mathcal{V}^{\mathcal{P}_0}, \tag{7.10}$$

$$\lambda_{k,\ell}^{y|s,q_\ell,z^{\mathrm{MLSE}}} = (\mathcal{V}_n^{\mathcal{P}_0})^2 \lambda_{k,\ell}^n. \tag{7.11}$$

With the model used for $f(y_{k,\ell}^{(\log)}|s_{k,\ell}^{(\log)}, q_\ell, z^{\mathrm{MLSE}})$, also the likelihood of the q_ℓth mixture given the state $f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{MLSE}})$ and the posterior of $s_{k,\ell}^{(\log)}$ given the q_ℓth mixture and the state $f(s_{k,\ell}^{(\log)}|y_{k,\ell}^{(\log)}, q_\ell, z^{\mathrm{MLSE}})$ can be determined. Also these probability density functions follow Gaussian distributions due to the Gaussian assumption for $f(y_{k,\ell}^{(\log)}|s_{k,\ell}^{(\log)}, q_\ell, z^{\mathrm{MLSE}})$ and for the speech mixtures in (7.5). The mean and variance of $f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{MLSE}})$ are given by

$$\mu_{k,\ell}^{y|q_\ell,z^{\mathrm{MLSE}}} = \mathcal{V}_s^{\mathcal{P}_0}(\mu_k^{s|q_\ell,z^{\mathrm{MLSE}}} - {\mathcal{P}_0}^{(s)}) + \mathcal{V}_n^{\mathcal{P}_0}(\mu_{k,\ell}^n - {\mathcal{P}_0}^{(n)}) + \mathcal{V}^{\mathcal{P}_0}, \tag{7.12}$$

$$\lambda_{k,\ell}^{y|q_\ell,z^{\mathrm{MLSE}}} = (\mathcal{V}_s^{\mathcal{P}_0})^2 \lambda_k^{s|q_\ell,z^{\mathrm{MLSE}}} + (\mathcal{V}_n^{\mathcal{P}_0})^2 \lambda_{k,\ell}^n, \tag{7.13}$$

while the mean of the posterior $f(s_{k,\ell}^{(\log)}|y_{k,\ell}^{(\log)}, q_\ell, z^{\mathrm{MLSE}})$ is given by

$$\mu_{k,\ell}^{s|y,q_\ell,z^{\mathrm{MLSE}}} = \mu_k^{s|q_\ell,z^{\mathrm{MLSE}}} + \frac{\lambda_k^{s|q_\ell,z^{\mathrm{MLSE}}} \mathcal{V}_s^{\mathcal{P}_0}}{\lambda_{k,\ell}^{y|q_\ell,z^{\mathrm{MLSE}}}} \left(y_{k,\ell}^{(\log)} - \mu_{k,\ell}^{y|q_\ell,z^{\mathrm{MLSE}}} \right). \tag{7.14}$$

With this, the MSE estimator of the log-spectral clean speech coefficients is determined. The estimator is given by the mean of $f(\mathbf{s}_\ell^{(\log)}|\mathbf{y}_\ell^{(\log)}, z^{\mathrm{MLSE}})$, which can be computed for each frequency bin k as

$$\mu_{k,\ell}^{s|y,z^{\mathrm{MLSE}}} = \sum_{q_\ell=1}^{Q} f(q_\ell|\mathbf{y}_\ell^{(\log)}, z^{\mathrm{MLSE}}) \mu_{k,\ell}^{s|y,q_\ell,z^{\mathrm{MLSE}}} \tag{7.15}$$

By setting $\hat{s}_{k,\ell}^{(\log)} = \mu_{k,\ell}^{s|y,z^{\mathrm{MLSE}}}$ in (7.1), the gain function $G_{k,\ell}$ can be determined, which is then used to to enhance the noisy spectrum $Y_{k,\ell}$ as $\hat{S}_{k,\ell} = G_{k,\ell} Y_{k,\ell}$. The probability $f(q_\ell|\mathbf{y}_\ell^{(\log)}, z^{\mathrm{MLSE}})$ can be obtained using Bayes' rule as

$$f(q_\ell|\mathbf{y}_\ell^{(\log)}, z^{\mathrm{MLSE}}) = \frac{f(\mathbf{y}_\ell^{(\log)}|q_\ell, z^{\mathrm{MLSE}}) f(q_\ell)}{\sum_{q'_\ell=1}^{Q} f(\mathbf{y}_\ell^{(\log)}|q'_\ell, z^{\mathrm{MLSE}}) f(q'_\ell)}, \tag{7.16}$$

where $f(\mathbf{y}_\ell^{(\log)}|q_\ell, z^{\mathrm{MLSE}})$ is given by the product of the frequency dependent PDF $f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{MLSE}})$. Furthermore, for computing the posterior, an estimate of the log-spectral noise mean $\mu_{k,\ell}^{n}$ and the log-spectral noise variance $\lambda_{k,\ell}^{n}$ is required. For obtaining these values, a spectral noise tracking algorithm, e.g., [29], [70], [80], is employed to determine the spectral noise variance $\Lambda_{k,\ell}^{n}$. More details about this algorithm are given in Section 2.1.4. In contrast to the static speech model, this estimate is time-variant. Using the equations for propagation given in Section 2.1.3 and Section 7.1, the log-spectral quantities $\mu_{k,\ell}^{n}$ and $\lambda_{k,\ell}^{n}$ can be obtained from $\Lambda_{k,\ell}^{n}$.

7.3. NON-MLSE-BASED SPEECH ENHANCEMENT

This section gives an overview over the non-MLSE clean speech estimators that are combined with the MLSE enhancement method described in Section 7.2. Here, we consider a linear log-spectral estimator related to the linear cepstrum estimator in [226] and the LSA [28]. The non-MLSE approaches model the log-spectral speech coefficients using distributions which are independent of the GMM and its mixture components. The parameters are obtained from the noisy spectrum $Y_{k,\ell}$ using a spectral speech PSD tracker for which TCS [12], [82] is used. The noise variance is obtained in a similar way as for the MLSE method, i.e., it is also estimated from the noisy observation using the SPP-based noise estimator [70], [71]. Both algorithms are described in more detail in Section 2.1.4. Additionally, in this section, the underlying likelihood models are given as they form the basis of the combination.

7.3.1. Linear Log-Spectral Filter

The linear log-spectral filter is closely related to the linear cepstrum estimator presented in [226]. In [226], it is shown that the linearly constrained MSE estimator of the clean speech cepstral coefficients has an equivalent representation in the log-spectral domain. It results from assuming that the log-spectral coefficients of speech and noisy speech are jointly Gaussian distributed as

$$f(y_{k,\ell}^{(\log)}, s_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{WF}}) = \mathcal{N}\left(\begin{bmatrix} y_{k,\ell}^{(\log)} \\ s_{k,\ell}^{(\log)} \end{bmatrix} \middle| \begin{bmatrix} \mu_{k,\ell}^{y|z^{\mathrm{WF}}} \\ \mu_{k,\ell}^{s|z^{\mathrm{WF}}} \end{bmatrix}, \begin{bmatrix} \lambda_{k,\ell}^{y|z^{\mathrm{WF}}} & \lambda_{k,\ell}^{sy|z^{\mathrm{WF}}} \\ \lambda_{k,\ell}^{sy|z^{\mathrm{WF}}} & \lambda_{k,\ell}^{s|z^{\mathrm{WF}}} \end{bmatrix} \right). \tag{7.17}$$

The joint distribution $f(y_{k,\ell}^{(\log)}, s_{k,\ell}^{(\log)}|q_\ell, z_{k,\ell})$ generally depends on the mixture q_ℓ. However, q_ℓ has no influence on the distribution if $z_{k,\ell} = z^{\mathrm{WF}}$ which reflects the assumption above that the non-ML approaches are independent of the GMM. The symbol z^{WF} is a state indicator for the model assumed in this section and is used for the combination in Section 7.4. The required means and variances are obtained by propagating the spectral estimates of the speech PSD $\Lambda_{k,\ell}^{s}$ and the noisy speech PSD $\Lambda_{k,\ell}^{y}$ to the log-spectral domain. The PSD of noisy speech is obtained by exploiting the assumption that the spectral speech and noise coefficients are uncorrelated, i.e., $\Lambda_{k,\ell}^{y} = \Lambda_{k,\ell}^{s} + \Lambda_{k,\ell}^{n}$. For the propagation, the methods described in Section 2.1.3 and Section 7.1 are used. By marginalizing (7.17) over $s_{k,\ell}^{(\log)}$, the likelihood of this non-ML-based approach can be obtained, which is given by

$$f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{WF}}) = f(y_{k,\ell}^{(\log)}|z^{\mathrm{WF}}) = \mathcal{N}(y_{k,\ell}^{(\log)}|\mu_{k,\ell}^{y|z^{\mathrm{WF}}}, \lambda_{k,\ell}^{y|z^{\mathrm{WF}}}). \tag{7.18}$$

Under the joint distribution given in (7.17), the MSE optimal estimator of the log-spectral clean speech coefficients is

$$\mu_{k,\ell}^{s|y,q_\ell,z^{\mathrm{WF}}} = \mu_{k,\ell}^{s|y,z^{\mathrm{WF}}} = \mu_{k,\ell}^{s|z^{\mathrm{WF}}} + \frac{\lambda_{k,\ell}^{sy|z^{\mathrm{WF}}}}{\lambda_{k,\ell}^{y|z^{\mathrm{WF}}}} \left(y_{k,\ell}^{(\log)} - \mu_{k,\ell}^{y|z^{\mathrm{WF}}} \right). \tag{7.19}$$

As the speech and noisy speech coefficients are independent of q_ℓ if $z_{k,\ell} = z^{\mathrm{WF}}$, the resulting estimator is also independent of q_ℓ.

7.3.2. Log-Spectral Amplitude Estimator

The second enhancement method that can be used in combination with the considered MLSE enhancement approach is the LSA estimator [28]. Here, it is assumed that the spectral coefficients of speech and noise follow a complex normal distribution. As discussed in Section 2.1.2, it is also assumed here that the spectral speech and noise coefficients are uncorrelated and are zero mean. With these assumptions, the joint distribution of the complex spectral speech and noisy speech coefficients can be written similar as in [257]

$$f(Y_{k,\ell}, S_{k,\ell}|q_\ell, z^{\mathrm{LSA}}) = \mathcal{N}_{\mathbb{C}}\left(\begin{bmatrix} Y_{k,\ell} \\ S_{k,\ell} \end{bmatrix} \middle| \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Lambda_{k,\ell}^{y|z^{\mathrm{LSA}}} & \Lambda_{k,\ell}^{s|z^{\mathrm{LSA}}} \\ \Lambda_{k,\ell}^{s|z^{\mathrm{LSA}}} & \Lambda_{k,\ell}^{s|z^{\mathrm{LSA}}} \end{bmatrix} \right). \tag{7.20}$$

Here, z^{LSA} is the indicator for the model used for the LSA [28]. Further, the variances $\Lambda_{k,\ell}^{s|z^{\mathrm{LSA}}}$ and $\Lambda_{k,\ell}^{y|z^{\mathrm{LSA}}}$ are identical to the speech and noisy speech PSD, i.e., $\Lambda_{k,\ell}^{s|z^{\mathrm{LSA}}} = \Lambda_{k,\ell}^{s}$ and $\Lambda_{k,\ell}^{y|z^{\mathrm{LSA}}} = \Lambda_{k,\ell}^{y}$. As both signal components, i.e., speech and noise, are assumed to follow a zero-mean Gaussian distribution, their joint distribution is given by a multivariate zero-mean Gaussian distribution. Because speech and noise are uncorrelated, the variance of the noisy components can also be expressed as $\Lambda_{k,\ell}^{y} = \Lambda_{k,\ell}^{s} + \Lambda_{k,\ell}^{n}$. For the same reason, the off-diagonal elements of the covariance matrix are equal to the variance of speech $\Lambda_{k,\ell}^{s}$, since $\mathbb{E}\{Y_{k,\ell} S_{k,\ell}^{*}\} = \Lambda_{k,\ell}^{s}$. In [28], the MSE optimal estimator of the speech log-spectral coefficients has been derived under the model in (7.20), i.e., $\mathbb{E}(\log(|S_{k,\ell}|)|Y_{k,\ell})$. Correspondingly, this method is the MSE optimal estimator of $\log(|S_{k,\ell}|) = 1/2 \cdot s_{k,\ell}^{(\log)}$. The result of the MSE estimator given in [28] can be rewritten as an estimator of the log-spectral speech coefficients $s_{k,\ell}^{(\log)}$ as

$$\mu_{k,\ell}^{s|y,q_\ell,z^{\mathrm{LSA}}} = \mu_{k,\ell}^{s|y,z^{\mathrm{LSA}}} = 2\log\left[\frac{\Lambda_{k,\ell}^{s}}{\Lambda_{k,\ell}^{s} + \Lambda_{k,\ell}^{n}}\right] + y_{k,\ell}^{(\log)} + \int_{\zeta_{k,\ell}}^{\infty} \frac{e^{-t}}{t} dt, \tag{7.21}$$

where

$$\zeta_{k,\ell} = \frac{\Lambda_{k,\ell}^{s}}{\Lambda_{k,\ell}^{n} + \Lambda_{k,\ell}^{s}} \frac{\exp(y_{k,\ell}^{(\log)})}{\Lambda_{k,\ell}^{n}}. \tag{7.22}$$

For this approach, no propagation of the statistics from the spectral domain is required. The likelihood under this model can be derived as

$$f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{LSA}}) = f(y_{k,\ell}^{(\log)}|z^{\mathrm{LSA}}) = \frac{1}{\Lambda_{k,\ell}^{y}} \exp\left(-\frac{e^{y_{k,\ell}^{(\log)}}}{\Lambda_{k,\ell}^{y}} + y_{k,\ell}^{(\log)}\right). \tag{7.23}$$

7.4. PROPOSED COMBINATION

In this section, we describe the proposed method for combining the MLSE approach from Section 7.2 and the non-MLSE enhancement methods given in Section 7.3.

For the combination, we exploit the fact that each enhancement method exhibits a different underlying likelihood model. Therefore, we define the likelihood of the state $z_{k,\ell}$ given the q_ℓth mixture $f(y_{k,\ell}^{(\log)}|z_{k,\ell}, q_\ell)$ as

$$f(y_{k,\ell}^{(\log)}|q_\ell, z_{k,\ell}) = \begin{cases} f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{MLSE}}), & z_{k,\ell} = z^{\mathrm{MLSE}}, \\ f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{WF}}), & z_{k,\ell} = z^{\mathrm{WF}}, \\ f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{LSA}}), & z_{k,\ell} = z^{\mathrm{LSA}}. \end{cases} \tag{7.24}$$

The different enhancement approaches are distinguished by the discrete state variable $z_{k,\ell}$ which can take the values z^{MLSE}, z^{LSA}, and z^{WF} for the MLSE approach, the non-MLSE LSA, and the non-MLSE linear log-spectral estimator, respectively. The state $z_{k,\ell}$ is allowed to be different for each frequency k and segment ℓ. The likelihoods $f(y_{k,\ell}^{(\log)}|z^{\mathrm{WF}})$ and $f(y_{k,\ell}^{(\log)}|z^{\mathrm{LSA}})$ are given in (7.18) and (7.23). These two likelihoods are independent of the mixtures q_ℓ, such that for all mixtures q_ℓ the equalities $f(y_{k,\ell}^{(\log)}|z^{\mathrm{WF}}) = f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{WF}})$ and $f(y_{k,\ell}^{(\log)}|z^{\mathrm{LSA}}) = f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{LSA}})$ hold. For the pre-trained approach, $f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{MLSE}})$ is equivalent to

$$f(y_{k,\ell}^{(\log)}|q_\ell, z^{\mathrm{MLSE}}) = \mathcal{N}(y_{k,\ell}^{(\log)}|\mu_{k,\ell}^{y|q_\ell,z^{\mathrm{MLSE}}}, \lambda_{k,\ell}^{y|q_\ell,z^{\mathrm{MLSE}}}) \tag{7.25}$$

with parameters given in (7.12) and (7.13). With Bayes' rule, it can be determined which of these states can be considered the most appropriate one for the noisy observation

$$f(z_{k,\ell}|q_\ell, y_{k,\ell}^{(\log)}) = \frac{f(y_{k,\ell}^{(\log)}|q_\ell, z_{k,\ell})f(z_{k,\ell})}{\sum\limits_{z'_{k,\ell}} f(y_{k,\ell}^{(\log)}|q_\ell, z'_{k,\ell})f(z'_{k,\ell})}. \tag{7.26}$$

The state prior probability $f(z_{k,\ell})$ can be used to control the mixing of the combined algorithms such that a specific method may be preferred over the others. The posterior probability in (7.26) can be included in the calculation of the MSE estimate of the clean speech log-periodogram. This leads to a weighted combination of all combined enhancement methods as

$$\mu_{k,\ell}^{s|y,q_\ell} = \sum_{z_{k,\ell}} f(z_{k,\ell}|y_{k,\ell}^{(\log)}, q_\ell)\mu_{k,\ell}^{s|y,q_\ell,z}. \tag{7.27}$$

For the MLSE trained enhancement method, the mean $\mu_{k,\ell}^{s|y,q_\ell,z^{\text{MLSE}}}$ is given in (7.14). For the non-MLSE enhancement methods, the means are $\mu_{k,\ell}^{s|y,q_\ell,z^{\text{WF}}} = \mu_{k,\ell}^{s|y,z^{\text{WF}}}$ and $\mu_{k,\ell}^{s|y,q_\ell,z^{\text{LSA}}} = \mu_{k,\ell}^{s|y,z^{\text{LSA}}}$ which are given in (7.19) and (7.21), respectively. As the non-MLSE enhancement methods are independent of the mixture q_ℓ, the values have to be computed only once and can be reused for each q_ℓ in (7.27). The $\mu_{k,\ell}^{s|y,q_\ell}$ have to be marginalized over the mixtures q_ℓ similar to (7.15) to obtain a final estimate of the clean speech. For this, the probability $f(q_\ell|\mathbf{y}_\ell^{(\log)})$ is required instead of $f(q_\ell|\mathbf{y}_\ell^{(\log)}, z^{\text{MLSE}})$. In our experiments, however, we found that using $f(q_\ell|\mathbf{y}_\ell^{(\log)}, z^{\text{MLSE}})$ yields better results, i.e., we compute $\mu_{k,\ell}^{s|y}$ as

$$\mu_{k,\ell}^{s|y} = \sum_{q_\ell} f(q_\ell|\mathbf{y}_\ell^{(\log)}, z^{\text{MLSE}})\mu_{k,\ell}^{s|y,q_\ell}. \tag{7.28}$$

The reason for this is that $f(q_\ell|\mathbf{y}_\ell^{(\log)})$ tends more towards the prior $f(q_\ell)$ which only reflects the global distribution of the mixtures and is inappropriate for describing the local distribution, i.e., for a new observation $\mathbf{y}_\ell^{(\log)}$. This happens because for computing $f(q_\ell|\mathbf{y}_\ell^{(\log)})$, $f(y_{k,\ell}^{(\log)}|q_\ell, z_{k,\ell})$ has to be marginalized over $z_{k,\ell}$ as

$$f(y_{k,\ell}^{(\log)}|q_\ell) = \sum_{z_{k,\ell}} f(y_{k,\ell}^{(\log)}|q_\ell, z_{k,\ell})f(z_{k,\ell}), \tag{7.29}$$

where the joint distribution $f(\mathbf{y}_\ell^{(\log)}|q_\ell)$ is given by the product of $f(y_{k,\ell}^{(\log)}|q_\ell)$ over all k. The sum in (7.29) includes $f(y_{k,\ell}^{(\log)}|q_\ell, z^{\text{WF}})$ or $f(y_{k,\ell}^{(\log)}|q_\ell, z^{\text{LSA}})$ which are both independent of q_ℓ. If the prior $f(z_{k,\ell})$ for the non-MLSE approach, i.e., $z_{k,\ell} = z^{\text{WF}}$ or $z_{k,\ell} = z^{\text{LSA}}$, is set to 1, for example, then $f(q_\ell|\mathbf{y}_\ell^{(\log)}) = f(q_\ell)$ will result after applying Bayes rule. This does not reflect the actual distribution of the states in the trained speech model and can be avoided by using $f(\mathbf{y}_\ell^{(\log)}|q_\ell, z^{\text{MLSE}})$ as in (7.28).

While here we focus on the combination of MLSE and non-MLSE enhancement approaches, it is interesting to note that this method allows different combinations of algorithms, e.g., it is possible to combine the pre-trained enhancement method with either the linear estimator or the LSA estimator or both non-MLSE enhancement algorithms. The overall procedure is summarized in Algorithm 8.

7.5. EVALUATION

In this section, we evaluate the proposed combination and compare it to the non-MLSE algorithm based on the LSA [28] and an MLSE-based approach with a harmonic post-filter similar to [104, Section 3]. The algorithm are compared by means of PESQ [230] improvement scores as described in Section 2.2.2.

For the evaluation, we use 128 sentences taken from the test set of the TIMIT database [240] where we ensure that the amount of sentences spoken by male and female speakers is balanced. The speech signals are artificially corrupted by different background noises with SNRs ranging from -5 dB to 20 dB. Here, we employ babble noise, pink noise, which are taken from the NOISEX-92 database [260], and a non-stationary traffic noise. Additionally, we also include an amplitude modulated version of the pink noise as in [70], [71]. In our evaluation, the sampling rate of all signals is 16 kHz.

The corrupted signals are processed in 32 ms blocks with an overlap of 50 %. For spectral analysis and synthesis, a square-root Hann window is used. The speech model used in the MLSE approach consists of 128 mixtures. The parameters are trained off-line on the log-spectra of 784 gender balanced uncorrupted sentences from the TIMIT training corpus using the expectation maximization algorithm [87]. The speech presence probability based harmonic post-filter is implemented according to the description in [104, Section 3]. In [104], the post-filter is only applied to voiced segments. Therefore, we determine the voiced probability for each segment using [261]. If the probability exceeds 50 %, the harmonic post filter is applied. The noise PSD is obtained using the estimator described in [70]. The speech PSD is determined using temporal cepstrum smoothing as described in [82]. These two methods are considered in Section 2.1.4 in more detail. For the MLSE method in Section 7.2 and the linear log-spectrum estimator in Section 7.3.1, these estimates are propagated to the log-spectral domain using the method described in Section 2.1.3 and Section 7.1. For all enhancement methods, we ensure that the noisy input spectrum is attenuated by a maximum of 12 dB. Further, the VTS approximation may give values larger than the noisy observation such that the input signal may be boosted. We prevent this by setting an upper limit to the amplitude of the estimated clean speech spectrum which is given by the amplitude of the noisy observation. This limit is applied for all algorithms in the comparison.

The results are shown in Figure 7.1. Here, “LSA” denotes the non-MLSE-based speech enhancement algorithm which uses the gain function from [28] but no pre-trained speech models. In the legend, “MLSE” indicates the speech enhancement approach in Section 7.2.

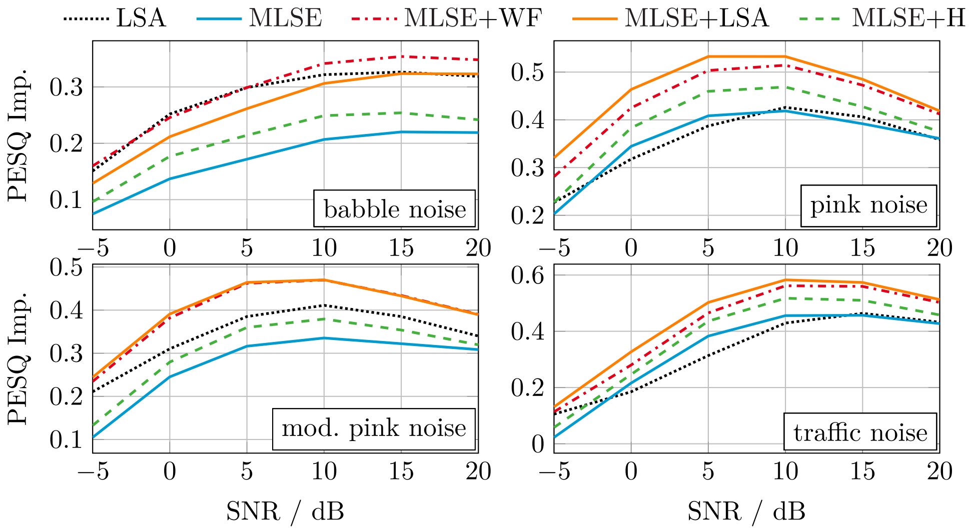

Fig. 7.1.: PESQ improvements for four different noise types over different SNRs. LSA: [28], MLSE: MLSE method, no combination, MLSE+WF: MLSE method with linear log-spectral filter, MLSE+LSA: MLSE method with LSA, MLSE+H: MLSE method with harmonic filter (based on [104]).

The combinations are denoted by MLSE+additional method, where the additional methods are given by the linearly constrained log-spectral filter (WF), the LSA (LSA), and the harmonic model based speech presence probability mask (H) [104]. Combinations with more than two algorithms are not analyzed in this chapter. For the combinations with a non-MLSE-based enhancement method, the prior $f(z_{k,\ell}) = 0.5$ is used in (7.26), i.e., there is no preference of one algorithm over another.

The results show that the sole application of the MLSE-based enhancement is comparable to the LSA in pink noise and traffic noise while lower PESQ scores are obtained for the modulated pink noise and babble noise. Especially babble noise appears to be a challenging situation for the employed MLSE-based speech enhancement method. Here, the performance is usually lower compared to the non-MLSE-based LSA. Only in combination with the linear log-spectral filter, the performance of the MLSE-based approach is comparable to the LSA in terms of PESQ scores in babble noise. The results for the remaining noise types, however, show an improvement of the proposed combination in contrast to the sole application of either the LSA or the MLSE-based approach. Furthermore, the proposed combination also outperforms the competing method MLSE+H that employs a harmonic post-filter [104].

In our experiments, the MLSE-based approaches showed a tendency to preserve weak speech components more than the compared non-MLSE estimators. This more conservative behaviour, however, has the effect that outliers in the noise sometimes remain unsuppressed. As a result, these enhancement methods generate more audible processing artifacts and noise activations during speech activity. These issues are, on the one hand, related to the speech models which mainly represent the envelope of speech, but, on the other hand, are also linked to the noise PSD estimator which is not able to follow very fast changes in the background noise, e.g., speech bursts in babble. The combination with the non-MLSE approaches reduces these artifacts. Informal listening showed that this reduction of artifacts is largest for the MLSE+WF approach.

7.6. SUMMARY

In this chapter, a combination of an MLSE-based speech enhancement method and a non-MLSE single-channel speech enhancement algorithms has been introduced. The proposed combination is employed to reduce processing artifacts of an MLSE and VTS-based enhancement method, which occur, when only a small number of mixtures is available. The proposed combinations outperform the stand-alone MLSE-based enhancement scheme for all noise types under investigation. Highly non-stationary noise types such as babble noise are the most challenging noise types for the stand-alone MLSE-based enhancement scheme which is clearly outperformed by the non-MLSE baseline in such environments. In this noise type, the proposed combination boosts the performance in terms of PESQ scores to the non-MLSE baseline. In all remaining noise types, the combinations achieve the highest PESQ scores and outperform the non-MLSE-based baseline and the already existing combination with a harmonic filter.

Algorithm 8 Algorithm for combining MLSE based and non-MLSE based estimators using Bayesian inference. In the algorithm, it is assumed that the MLSE approach is either combined with the log-spectral Wiener Filter (Section 7.3.1) or the LSA (Section 7.3.2).

```
Require: Parameters of the GMM: $f(q_\ell)$, $\mu_k^{s|q_\ell}$, $\lambda_k^{s|q_\ell}$
Require: $f(z_{k,\ell} = z^{\mathrm{MLSE}})$ and $f(z_{k,\ell} = z^{\mathrm{WF}})$ or $f(z_{k,\ell} = z^{\mathrm{LSA}})$
Ensure: $\sum_{z_{k,\ell}} f(z_{k,\ell}) = 1$
for all frames $\ell$ do
    Estimate the noise PSD $\Lambda_{k,\ell}^{n}$ using [70], [71].

    Estimate likelihood and speech coefficients for each state of the MLSE approach.
    for all states $q_\ell$ in the GMM do
        Compute $f(y_{k,\ell}^{(\log)}|q_\ell, z_{k,\ell} = z^{\mathrm{MLSE}}) = \mathcal{N}(y_{k,\ell}^{(\log)}|\mu_{k,\ell}^{y|q_\ell,z^{\mathrm{MLSE}}}, \lambda_{k,\ell}^{y|q_\ell,z^{\mathrm{MLSE}}})$.
        The solution for $\mu_{k,\ell}^{y|q_\ell,z^{\mathrm{MLSE}}}$ and $\lambda_{k,\ell}^{y|q_\ell,z^{\mathrm{MLSE}}}$ are given in (7.12) and (7.13).
        Compute $\mu_{k,\ell}^{s|y,q_\ell,z^{\mathrm{MLSE}}}$ using (7.14).
    end for

    Estimate the likelihood and speech coefficients of the non-MLSE approach. This
    approach does not depend on $q_\ell$ and hence, no loop over $q_\ell$ is required.
    Estimate the speech PSD $\Lambda_{k,\ell}^{s}$ using [82], [83].
    Compute $f(y_{k,\ell}^{(\log)}|q_\ell, z_{k,\ell})$ for $z_{k,\ell} = z^{\mathrm{WF}}$ (7.18) or $z_{k,\ell} = z^{\mathrm{LSA}}$ (7.23).
    Compute $\mu_{k,\ell}^{s|y,q_\ell,z}$ for $z = z^{\mathrm{WF}}$ (7.19) or $z = z^{\mathrm{LSA}}$ (7.21).

    Combine estimated speech coefficients of the non-MLSE approach with the estimates
    of each state in the MLSE approach.
    for all states $q_\ell$ in the GMM do
        Compute $f(z_{k,\ell}|q_\ell, y_{k,\ell}^{(\log)})$ using (7.26).
        Use $f(z_{k,\ell}|q_\ell, y_{k,\ell}^{(\log)})$ to obtain $\mu_{k,\ell}^{s|y,q_\ell}$ as in (7.27).
    end for
    Obtain final estimate $\hat{s}_{k,\ell}^{(\log)} = \mu_{k,\ell}^{s|y}$ using (7.28).
end for
```

Part IV.

Generalization of ML-Based Enhancement

CHAPTER 8

NORMALIZED FEATURES FOR IMPROVING THE GENERALIZATION OF DNN-BASED SPEECH ENHANCEMENT

As described in Section 2.1, STFT-based non-ML enhancement schemes such as [12], [28], [29], [70], [71], [82] use filter functions, which are derived in a statistical framework, to suppress noise. The required parameters, i.e., the speech PSD and the noise PSD, are estimated blindly from the noisy observation. Generally, the algorithms used for this estimation are based on the assumption that the background noise changes more slowly than the speech signal. This makes these algorithms applicable to many noise types but transient sounds such as the cutlery in a restaurant environment are generally not suppressed by non-ML-based enhancement schemes. Contrarily, studies on ML enhancement approaches, e.g., [23], [24], [167], show that DNN-based approaches in principle have the ability to reduce highly non-stationary noise, which cannot be easily tracked using non-ML-based estimators. As a consequence, these methods appear to be a promising approach to improve single-channel speech enhancement. However, one of the major concerns towards ML-based approaches is their generalization to noise types that have not been seen during training.

In [23], [167], the issue of generalization is encountered with large and diverse training data, where hundreds or even thousands of different noise types are included to allow the DNN-based enhancement schemes to generalize to unseen noise conditions. Even though large training sets increase the generalization, a huge number of noise types may still be inappropriate as in real scenarios virtually infinitely many noise types can possibly occur as argued in [26], [210]. Contrarily, non-ML approaches have been proven to generalize well to many different acoustical environments and provide good results in moderately varying noise types. To improve the robustness in unseen noise conditions, DNN-based enhancement approaches incorporate estimates of non-ML-based noise PSD estimators, e.g., [23], [208]–[210]. More specifically, an estimate of the noise PSD is appended to the noisy input features which is referred to as noise aware training. In [23], [208], a fixed noise PSD estimate is used which has been obtained from the first segments of the noisy input signal. In [209], [210], this idea has been advanced by employing a dynamic, i.e., time-varying noise PSD estimate, obtained from a non-ML-based estimator. However, the results in [209], [210] show only small improvements over the approaches that are not

This chapter is partly based on:

[221] R. Rehr and T. Gerkmann, “Robust DNN-based speech enhancement with limited training data,” in *ITG Conference on Speech Communication*, Oldenburg, Germany, Oct. 2018, © 2018 VDE Verlag.

aware of the background noise if a non-ML approach is used to estimate the noise PSD. Noise aware training has been pursued further in [211], where various improvements over the methods in [209] are presented.

In this chapter, we propose another powerful method to increase the generalization of DNN-based approaches to unseen noise conditions. More specifically, we propose to employ the *a priori* SNR $\xi_{k,\ell}$ and the *a posteriori* SNR $\gamma_{k,\ell}$ as features (see (2.6) and (2.7) in Section 2.1.1). Thus, instead of appending the noise PSD to the input features extracted from the noise observation as in [23], [208]–[210], here, the noise PSD estimate is used for normalization. The usage of the *a priori* SNR and the *a posteriori* SNR is motivated by non-linear clean speech estimators, e.g., [12], [28] where these quantities result from the derivation of Bayesian estimators. We show that the proposed features outperform features where the noise PSD estimate is appended to the noisy input vector. Further, the proposed SNR-based features have the advantage that the enhancement system is independent of the scaling of the input signal, i.e., the overall level has no effect on the enhancement.

These claims are confirmed in the evaluation using instrumental measures. For this, PESQ [230] scores and the short-time objective intelligibility (STOI) [262] are evaluated in a cross-validation based experimental setup, where different sets of noise types for training and testing are used. Further, the instrumental evaluation is supported by subjective evaluations. First, we describe the employed algorithms in Section 8.1 and Section 8.2. The results of the instrumental evaluation is given in Section 8.3 while the subjective evaluation is described in Section 8.4.

8.1. NON-ML-BASED ENHANCEMENT ALGORITHMS

This section gives an overview over the non-ML-based enhancement algorithms which form the basis of the proposed features in Section 8.2. It is very similar to the enhancement framework described in Section 2.1. For estimating the clean speech coefficients, the non-ML-based clean speech estimators makes use of the STFT. The physically plausible assumption in (2.3) is used that the speech signal and the noise signal mix additively. The speech coefficients are estimated using the Wiener filter gain function (2.15), which is applied as in (2.4). The clean speech estimates $\hat{S}_{k,\ell}$ are transformed back to the time-domain, which are used to reconstruct the enhanced time-domain signal using an overlap-add method.

The SPP-based noise PSD estimator presented in [70], [71] is used to estimate the noise PSD. This estimator allows moderate changes in the background noise, such as passing cars, to be tracked. However, it cannot track transient disturbances. For estimating the speech PSD $\Lambda^s_{k,\ell}$, the TCS approach described in [82] is employed. In contrast to the commonly used decision-directed approach [12], this approach causes less isolated estimation estimation errors, which may be perceived as annoying musical tones. Both algorithms are summarized in Section 2.1.4.

8.2. ML-BASED ENHANCEMENT ALGORITHMS

In this section, the ML-based algorithms used in this chapter are presented. First, the enhancement framework is described and, after that, the employed input features are considered. Note that the algorithms share the same ML-based enhancement framework but differ in the input features.

8.2.1. ML-Based Enhancement Framework

The architecture of the used framework resembles the approaches that have been proposed in [23], [24]. Similar to the non-ML-based enhancement scheme, also the ML-based approaches operate in the STFT domain. A feed-forward DNN is used to predict an IRM from the input features extracted from the noisy input signal. The features considered in this chapter are described in Section 8.2.2 and Section 8.2.3 in detail. The IRM has been proposed in [169] and is similar to the Wiener filter gain function shown in (2.15) with the difference that the speech periodogram $|S_{k,\ell}|^2$ and the noise periodogram $|N_{k,\ell}|^2$ are employed instead of the respective PSDs as

$$\mathrm{IRM}(k,\ell) = \frac{|S_{k,\ell}|^2}{|S_{k,\ell}|^2 + |N_{k,\ell}|^2}. \tag{8.1}$$

Similar to the Wiener filter, the predicted IRM obtained from the DNN is used to estimate the clean speech coefficients $\hat{S}_{k,\ell}$ as

$$\hat{S}_{k,\ell} = \max\left(\widehat{\mathrm{IRM}}(k,\ell), G_{\min}\right) Y_{k,\ell}, \tag{8.2}$$

where $\widehat{\mathrm{IRM}}(\cdot)$ denotes the IRM estimated by the DNN. We enforce a lower limit $G_{\min}$ as in (2.9) on page 30 and the time-domain signal is reconstructed using an overlap-add method.

8.2.2. Non-Normalized Features

In this part, the non-normalized feature inputs of the DNN are presented. The first representative of the non-normalized features is the logarithmized noisy periodogram, i.e.,

$$y^{(\log)}_{k,\ell} = \log\left(|Y_{k,\ell}|^2\right), \tag{8.3}$$

which has also been employed in [23], [209]. All spectral coefficients of a segment ℓ, i.e., $y^{(\log)}_{k,\ell}$ for all frequency bins k for a given segment ℓ, are stacked in a feature vector.

Given only the log-spectral coefficients, the DNN needs to learn how to distinguish between speech and noise using the training data. This is a potentially challenging task, as a large amount of different acoustic scenarios is required for training to match real conditions. Hence, the approaches in [209], [210] sought to improve the robustness to unseen noise environments using non-ML-based noise PSD estimators. For this, the noisy log-spectral features given above have been extended by appending an estimate of the noise PSD [209], [210], which is also known as noise aware training [208]. Similar to (8.3), the logarithmized estimate of the noise PSD is given by

$$\hat{\Lambda}^{n,(\log)}_{k,\ell} = \log\left(\hat{\Lambda}^{n}_{k,\ell}\right). \tag{8.4}$$

As a result, the feature vector for this set has twice the dimensionality as using only the log-spectral features. In our experiments, the noise PSD is estimated using the SPP-based noise PSD estimator proposed in [70], [71], which is described in Section 2.1.4. For both features sets, a context of three past segments is added to this vector by appending the respective feature vectors to the end of the vector. We do not add context from future segments to keep the algorithmic latency as low as for the non-ML-based enhancement scheme.

8.2.3. Proposed Normalized Features

The main goal of the proposed normalized features is also to increase the robustness of DNN-based enhancement schemes to unseen noise conditions. However, instead of appending the noise PSDs to the noisy input features, we incorporate the generalization of non-ML-based enhancement schemes by using the estimated noise PSD as normalization term. More specifically, we employ the logarithmized *a priori* SNR $\xi^{(\log)}_{k,\ell} = \log(\xi_{k,\ell})$ and *a posteriori* SNR $\gamma^{(\log)}_{k,\ell} = \log(\gamma_{k,\ell})$ as input features. The *a priori* SNR and *a posteriori* SNR are defined as

$$\xi_{k,\ell} = \frac{\Lambda^{s}_{k,\ell}}{\Lambda^{n}_{k,\ell}} \tag{8.5}$$

$$\gamma_{k,\ell} = \frac{|Y_{k,\ell}|^2}{\Lambda^{n}_{k,\ell}}. \tag{8.6}$$

The usage of the *a priori* and *a posteriori* SNRs is motivated by non-ML-based clean speech estimators, e.g., [9], [12], [28], [31], where the quantities appear in the analytical solutions derived in a statistical framework. The speech PSD $\Lambda^{s}_{k,\ell}$ and the noise PSD $\Lambda^{n}_{k,\ell}$ are estimated blindly from the noisy observation using the SPP noise PSD [70] estimator and TCS [82], which are summarized in Section 2.1.4. Both SNRs can be used separately or in combination by concatenating both in a single vector. Note that the dimensionality of the features is the same as the noisy log-spectra if one of the SNRs is used as input. In all considered cases, a temporal context of three previous segments is appended to the feature vectors.

In contrast to the non-normalized features in Section 8.2.2, the *a priori* and the *a posteriori* SNR exhibit the advantage that these features are scale-invariant. As their value does not depend on the overall level, differently scaled training data results in identical normalized features as when the scaling is not varied. To make the scale-invariance also available to a DNN using non-normalized features, e.g., Section 8.2.2, the training data has to reflect these gain variations. This increases the variations in the training examples such that learning potentially becomes challenging. Such gain variations do not increase the variability for the normalized features, which may improve the enhancement.

8.3. INSTRUMENTAL EVALUATION

In this section, the algorithms described in Section 8.1 and Section 8.2 are compared using instrumental measures. Further, the optimally modified log-spectral amplitude estimator (OMLSA) proposed in [75], [263] is used as a reference. First, the audio material, the used parameters and instrumental measures are considered. Afterwards, the results of the experimental analysis are presented which compares the properties of the normalized and the non-normalized input features in the DNN framework. Further, the computational complexity and the training convergence speed are considered. In the last part of the instrumental evaluation, the performance of the evaluated algorithms is compared.

8.3.1. Audio Material, Parameters and Instrumental Measures

For all algorithms, the STFT uses 32 ms segments which overlap by 50 %. For the analysis step as well as the synthesis step a square-root Hann window is employed. All signals have a sampling rate of 16 kHz. For computing the features, the mirror spectrum is omitted such that the resulting dimension of the spectra is 257. Correspondingly, the feature dimensionality of the noisy log-spectra $y_{k,\ell}^{(\log)}$, the *a priori* SNR $\xi_{k,\ell}^{(\log)}$ and the *a posteriori* SNR $\gamma_{k,\ell}^{(\log)}$ is $257 \times (3+1) = 1028$ including the context. The dimensionality of the input features doubles to 2056 for the combination of the *a priori* SNR $\xi_{k,\ell}$ and the *a posteriori* SNR, as well as, for the combination of the noisy log-spectra $y_{k,\ell}^{(\log)}$ and the logarithmized noise PSD $\hat{\Lambda}_{k,\ell}^{n,(\log)}$ as employed in [209], [210]. The DNN's architecture comprises three hidden layers with ReLUs [161] as non-linearities and an output layer with sigmoidal activation functions. The number of units in each hidden layer amounts to 1024 for both DNN-based approaches. For the evaluations in this section, the minimum gain is set to $G_{\min} = -20$ dB for all employed enhancement schemes. For the non-ML algorithms in Section 8.1 the parameters in the respective publication [70], [82] are used.

The employed background noises are taken from a fixed pool of nine noise types. It includes the babble noise and the factory 1 noise taken from the NOISEX-92 database [239]. Further, a modulated version of NOISEX-92's pink and white noise are included as in [70]. Additional noise types are taken from the freesound database http://www.freesound.org. Among them are the sounds of an overpassing propeller plane (https://freesound.org/s/115387/), the interior of a passenger jet during flight (https://freesound.org/s/188810/), a vacuum cleaner (https://freesound.org/s/67421/) and a traffic noise (https://freesound.org/s/75375/). Further, a two-talker babble noise is included which is generated using two read out stories taken from https://www.vorleser.net. The two stories are read by a male and a female speaker, respectively, and are mixed at an SNR of 0 dB after speech pauses have been removed. The noise types are used to conduct cross-validation experiments where all noise types except one are included in the training set. The training data of each cross-validation set are augmented by additionally including a highly non-stationary noise type which is generated from the noise snippets collected by [264]. The noise excerpts in this database are generally short and are, hence, concatenated multiple times in various orders to give a continuous noise signal. Long noise excerpts are split into roughly 2 second long snippets during this generation. This noise type is referred to as concatenated short noise excerpts (CSNE). The remaining unseen noise type is used for testing in the evaluations.

The speech material for training is taken from the TIMIT database [240]. For the training of the DNN-based enhancement schemes, a set of 1196 female and 1196 male sentences taken from the TIMIT training set is employed. All sentences are embedded once in each noise type used for training at a random temporal position. For the employed noise PSD estimator, a two second initialization period is added at the beginning of each sentence to avoid initialization artifacts during feature extraction. This period is removed from the final features used for training. However, a noise only period which amounts to 15 % of the utterance length is included for each sentence in the training data. To allow the DNN to learn the effect of different SNRs, the sentences are embedded in the background noise at SNRs ranging from −10 dB to 15 dB. The SNR is randomly chosen for each sentence and also the scaling is randomly varied for each sentence by adjusting the peak level of the speech signal from −26 dB and −3 dB. These variations are included in the training data, to allow the DNN-based on the non-normalized features to learn a scale-independent function of the IRM.

The parameters of the DNN are adapted by minimizing the following optimization criterion

$$J = \sum_{\ell} \sum_{k} \left| \log \left(\widehat{\mathrm{IRM}}(k, \ell) + \epsilon \right) - \log \left(\mathrm{IRM}(k, \ell) + \epsilon \right) \right|^2 . \tag{8.7}$$

Here, the squared error of the logarithmized quantities is minimized which is motivated by the human loudness perception which approximately follows a logarithmic law. Further, ϵ is a bias term which is used to avoid that extremely low gains of the target IRM are overly penalized by the cost function. Here, $\epsilon = 0.1$ is employed such that differences between the target IRM and the DNN output are treated as irrelevant if the target IRM is below −20 dB. The weights and biases of the layers are initialized using the Glorot method [160]. After the initialization, the weights are optimized using the AdaGrad approach [265] where the learning rate has been set to 0.005 while a batch size of 128 samples has been used. The order of the training observations is randomized. To avoid overfitting of the network, an early stopping scheme is employed where the training procedure is stopped if the error J is not reduced by more than 1 % over 10 iterations on a validation set. The validation set is constructed by randomly selecting 15 % of the training set.

For testing, 128 sentences, 64 female and 64 male, are taken from the TIMIT test set.

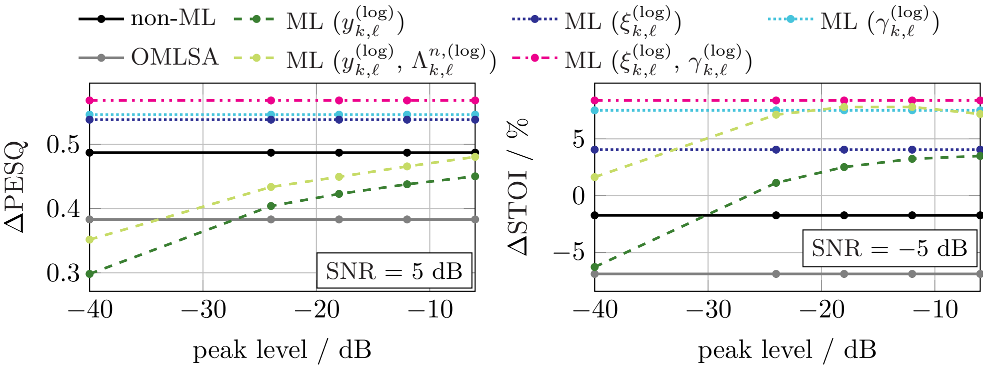

Fig. 8.1.: PESQ and STOI improvements for the considered enhancement algorithms in dependence of the peak level of the clean speech signal averaged over all noise types at an input SNR of 0 dB.

Similar to the training, the clean speech sentences are embedded at random positions in the background. All sentences are mixed at SNRs ranging from −5 dB to 20 dB in 5 dB steps. Furthermore, also here, an initialization period of two seconds is added to avoid initialization artifacts of the employed noise PSD estimator [70]. This period is omitted during the evaluation, i.e., the instrumental measures are only evaluated on the part that contains the embedded sentence.

For the comparison, PESQ [230] is used as an instrumental measure to evaluate the quality of the enhanced signals. Generally, PESQ improvement scores (ΔPESQ) are shown which are obtained by computing the difference between the PESQ score of the enhanced and the noisy signal. Further, STOI [262] is used to instrumentally predict the speech intelligibility of the enhanced signals. In this evaluation, the STOI scores are mapped to actual intelligibility scores, i.e., the percentage of words a human would correctly identify in a listening experiment. As no mapping is available for the TIMIT database, the mapping function given for the IEEE sentences in [262] is used. Also here, improvements are computed (ΔSTOI) which are obtained by subtracting the mapped speech intelligibility of the enhanced and the noisy signal.

8.3.2. *Analysis*

In this part, we give an analysis on the features proposed in Section 8.2.2 and Section 8.2.3. We demonstrate that the proposed normalized features in Section 8.2.3 are independent of scaling of the input signal. Further, we show that the DNN converges more quickly if the proposed features are employed. In the last part of this section, the computational complexity of the various approaches is considered.

To demonstrate the scale-invariance of the proposed features, the considered enhancement

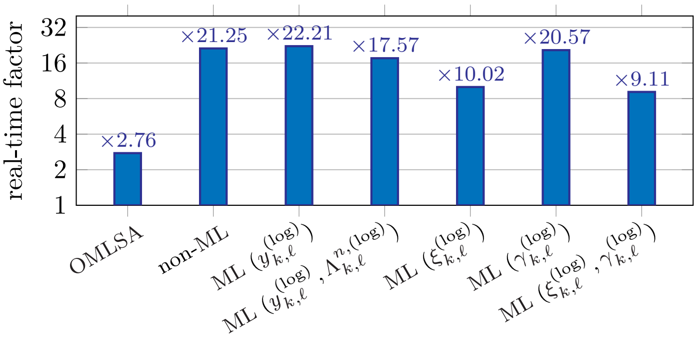

Fig. 8.2.: Computational complexity of the considered algorithms in terms of the real-time factor. This factor describes how many seconds of the audio material can be processed within a second in the real world. The number on top of the bar shows the actual value of the real-time factor.

approaches are evaluated on speech material where the peak level of the speech utterances is varied systematically. For this, we set the peak level of the speech utterances to −6 dB, −12 dB, −18 dB, −24 dB and −40 dB. The −40 dB peak level has not been seen during training and can be considered an extreme case whereas the remaining levels are within the range of variations included in the training data. For this evaluation, the SNR of the input signals is fixed at −5 dB for STOI and 5 dB for PESQ. A lower SNR is used for STOI because the speech intelligibility reduces only considerably for SNRs lower than 0 dB. The results in terms of PESQ and STOI improvements are depicted in Fig. 8.1. For this, the averages over all noise types excluding the CSNE [264] are computed. The results show that the non-ML-based speech enhancement algorithms and the ML-based approaches based on the normalized features yield the same outcome independent of the scaling of the input signal. Contrarily, the performance of the ML-based enhancement scheme using noisy log-spectra varies over the peak level of the input signal. The same can be observed for the combination with the estimated noise PSD. This indicates that by using the normalized features, the ML-based algorithm does not depend on the overall level. Contrarily, despite the efforts taken to increase the scale-independence during the training process, the non-normalized features result in scale-dependent results.

The convergence speed of the proposed features is measured using the number of epochs that have been required until the validation error converges. Due to the cross-validation setup, nine models are trained for each feature type, which allows the number of epochs required to train each model to be averaged. About 28 to 29 epochs are required on average if the non-normalized features are employed, whereas only 20 to 23 iterations are required for the normalized features. This result provides evidence that the proposed normalized features simplify the training of the respective DNNs.

Last, the computational complexity of the considered algorithms is considered. Fig. 8.2, shows the processing speed of the various algorithms in terms of the real-time factor. The factor describes how many seconds of the audio signal can be processed within a second in the real world. Correspondingly, if the factor is larger than one, the algorithm processes the signals faster than real-time and if the factor is smaller than one, the processing is slower than real-time. The algorithms have been evaluated on the CPU (Intel Core i7-5930K) of a desktop PC. For this, their respective Python or Matlab implementations have been used. Fig. 8.2 shows that the OMLSA runs slowest while the non-ML described in Section 8.1 and the ML-based algorithms where the noisy log-spectra $y_{k,\ell}^{(\log)}$ or the *a posteriori* SNR $\gamma_{k,\ell}$ are used as input feature run fastest. On the used hardware, the quickest algorithms run roughly 20 times faster than real-time. The OMLSA is only about three times faster than real-time which may be explained by the fact that the Matlab implementation is run through Python which potentially introduces further processing overhead. Using the *a priori* SNR $\xi_{k,\ell}^{(\log)}$ instead of the *a posteriori* $\gamma_{k,\ell}^{(\log)}$ reduces the real-time factor to 10. This is because, in comparison to $\gamma_{k,\ell}^{(\log)}$, the TCS needs to be additionally computed to obtain $\xi_{k,\ell}^{(\log)}$. Using both, the *a priori* SNR $\xi_{k,\ell}^{(\log)}$ and the *a posteriori* SNR $\gamma_{k,\ell}^{(\log)}$, as input features, the real-time factor further drops to 9. Concatenating $y_{k,\ell}^{(\log)}$ and $\Lambda_{k,\ell}^{n,(\log)}$ is computationally more complex than using the $Y_{k,\ell}$ normalized by $\Lambda_{k,\ell}^{n}$, i.e., $\gamma_{k,\ell}^{(\log)}$. For the concatenated features, the input dimensionality is twice as large as for the *a posteriori* SNR $\gamma_{k,\ell}^{(\log)}$ which results in the additional computational complexity. From this, it is followed that the inclusion of the noise PSD generally increases the computational complexity as expected. Using $\gamma_{k,\ell}^{(\log)}$, the increase is only small whereas including the *a priori* SNR considerably increases the complexity.

8.3.3. Comparisons

The following results show the outcome of the cross-validation procedure and are used to compare the enhancement algorithms used in this chapter. For these experiments, the peak level of the 128 TIMIT sentences used for testing is randomly varied between -6 dB and -26 dB which is similar to the range used for training. Fig. 8.3 and Fig. 8.4 depict the results.

For both instrumental measures, first the performance of the non-ML-based approach is considered. For the aircraft interior noise, the OMLSA achieves higher PESQ scores in low SNRs. This is, however, an exception as for the remaining noise types, especially the nonstationary ones such as babble noise or the amplitude modulated versions of the pink and white noise, the performance of the employed non-ML enhancement approach is higher than for the OMLSA. In terms of the speech intelligibility predicted by STOI, both non-ML approaches have either little effect or reduce the intelligibility. In all cases, however, ΔSTOI is higher for the approach described in Section 8.1 than for the OMLSA. Consequently, the algorithm described in Section 8.1 generally outperforms the OMLSA [75], [263].

The speech intelligibility predicted by STOI is generally higher for the ML-based algorithms

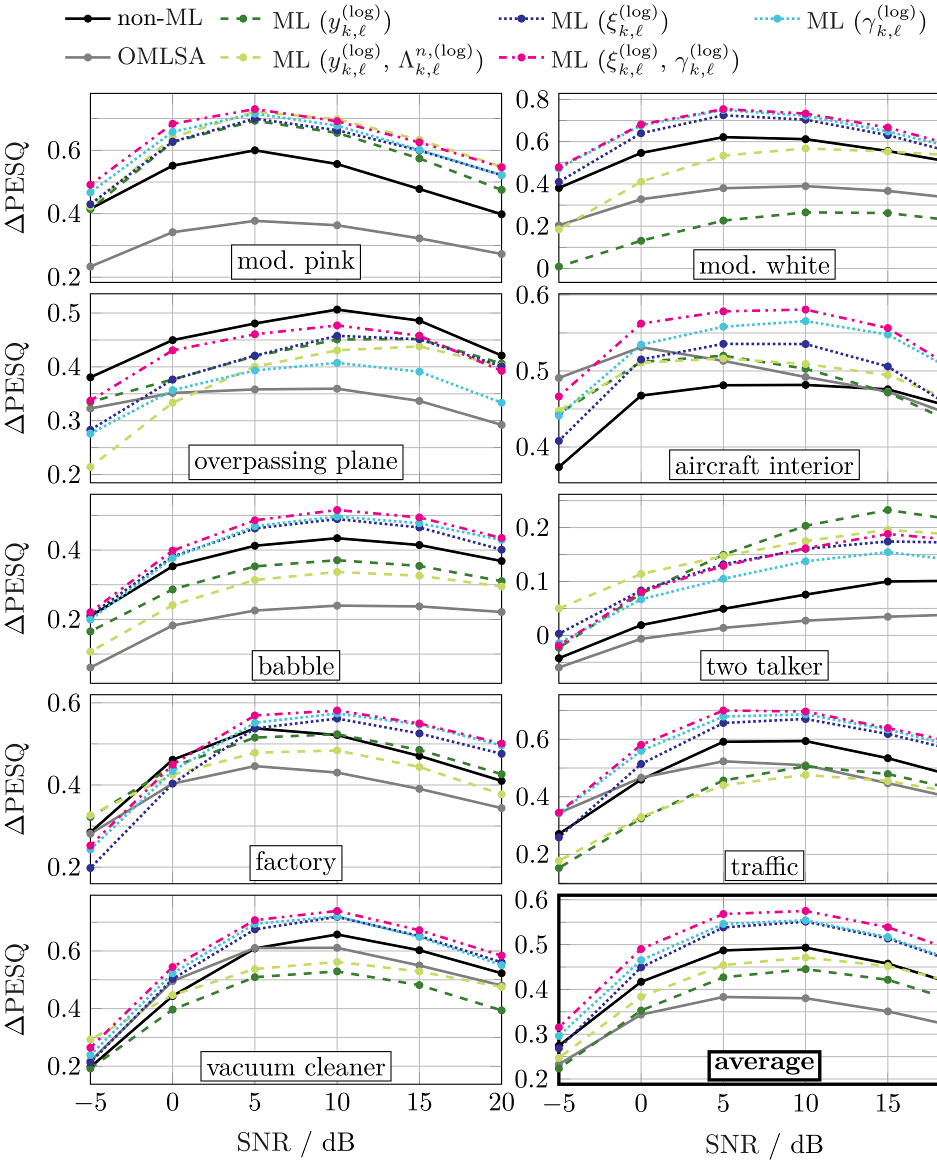

Fig. 8.3.: PESQ improvements that results for the considered enhancement algorithms in dependence of the background noise type and the SNR. The ML-based algorithms are always trained on CSNE [264] and the noise types not given in the respective plot, i.e., the background noise is unseen in all cases.

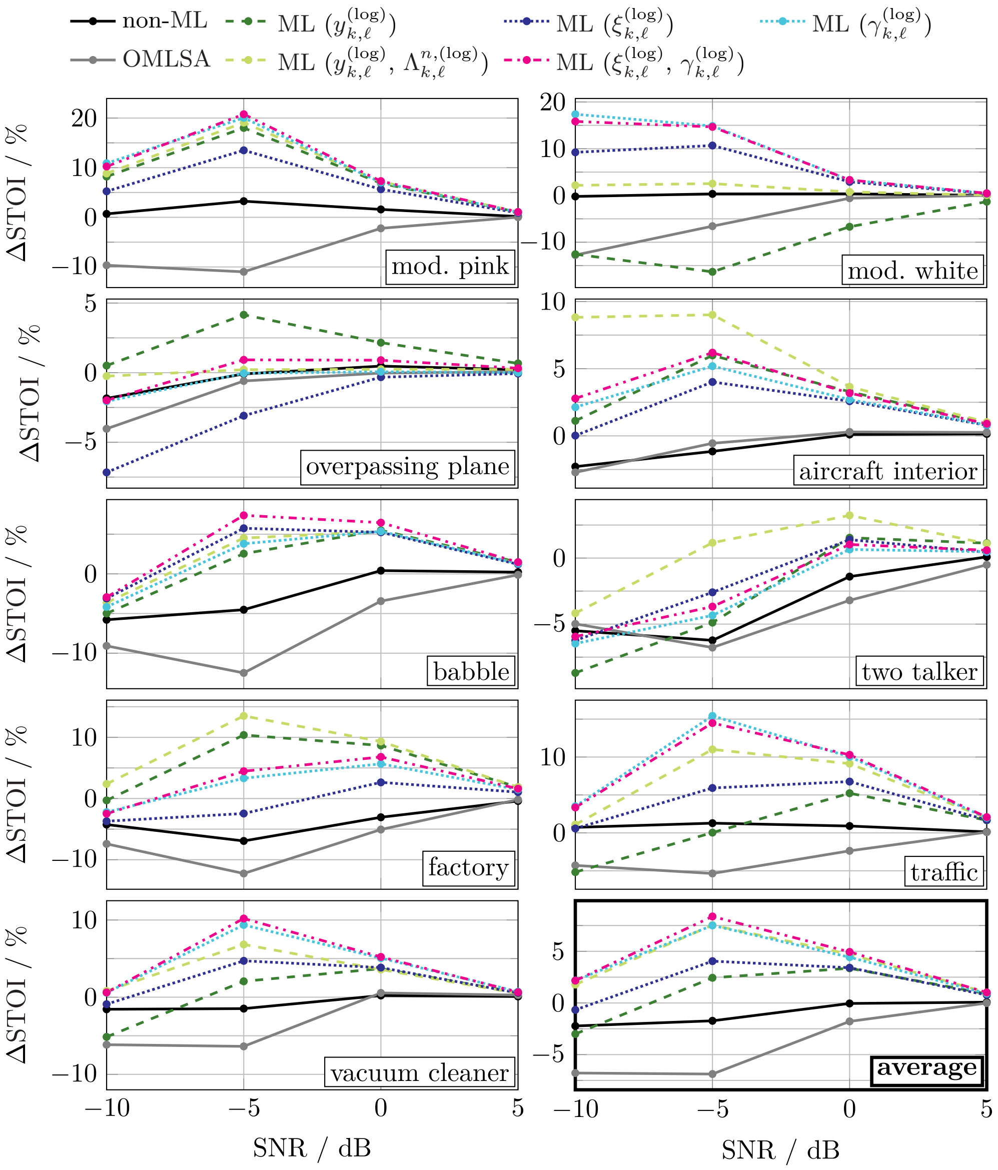

Fig. 8.4.: Same as Fig. 8.3 but for STOI.

than for the non-ML approaches. In factory noise and the aircraft interior noise, STOI predicts a higher speech intelligibility for the non-normalized features. The same is true for the overpassing plane and the two-talker noise. Here, however, ΔSTOI is generally smaller compared to the other noise types. Among the non-normalized features, ΔSTOI is generally higher for the combination of the noisy log-spectra $y_{k,\ell}^{(\log)}$ and the estimated noise PSD $\Lambda_{k,\ell}^{n,(\log)}$. For the remaining noise types, however, the proposed normalized features yield similar or higher STOI improvements than the non-normalized features. Comparing the normalized feature sets amongst each other shows that using only the *a priori* SNR $\xi_{k,\ell}^{(\log)}$ often results in the lowest scores. Contrarily, the combination of the *a priori* SNR $\xi_{k,\ell}^{(\log)}$ and the *a posteriori* SNR $\gamma_{k,\ell}^{(\log)}$ generally yields the highest scores. In many cases, using only the *a posteriori* SNR $\gamma_{k,\ell}^{(\log)}$ yields scores similar to the combination. For cases, where the computational complexity plays an important role this feature type is thus a considerable alternative.

The PESQ improvements for the ML-based algorithms indicate a clear preference for the proposed normalized features. Only for the two talker noise, the PESQ improvements obtained for the non-normalized features are higher than for the normalized features. However, as basically all the considered enhancement algorithms struggle in this noise type, the gains of 0.05 points are rather small and therefore negligible. For most of the remaining noise types, the performance of the non-normalized features predicted by PESQ is between the OMLSA and the non-ML approach described in Section 8.1. Except for the modulated white noise, PESQ does not indicate considerable advantages if an estimate of the noise PSD $\Lambda_{k,\ell}^{n,(\log)}$ is appended to the noisy log-spectra $y_{k,\ell}^{(\log)}$. This changes if the normalized features are used. Using these features, the performance of the ML approach is more robust and, often, both non-ML approaches are outperformed. Again, the combination of the *a priori* SNR $\xi_{k,\ell}^{(\log)}$ and *a posteriori* SNR $\gamma_{k,\ell}^{(\log)}$ yields the highest scores in most noise types. Also here, using the *a posteriori* SNR $\gamma_{k,\ell}^{(\log)}$ without the *a priori* SNR $\xi_{k,\ell}^{(\log)}$ yields similar results as the combination of both. Consequently, it is possible to benefit from the advantages of the normalized features without severely increasing the computational complexity. Further, as this feature type has the same dimensionality as the noisy log-spectra, this demonstrates the importance of the normalized features on the generalization of ML-based enhancement schemes.

8.4. SUBJECTIVE EVALUATION

Instrumental measures such as PESQ give an indication on how the quality of the processed signals would be judged by humans. Still, as such measures cannot perfectly model human perception, we verify the instrumental results in Section 8.3 using subjective evaluation tests. Here, a MUSHRA [249] is employed to compare the algorithms described in Section 8.1 and Section 8.2. First, the audio material, parameters and evaluation are explained and, after that, the results are discussed.

8.4.1. Audio Material, Parameters and Setup

For this experiment, a sentence of a male and a female speaker is embedded in factory 1 noise and traffic noise at an SNR of 5 dB. The noisy signals are processed by the speech enhancement schemes described in Section 8.1 and Section 8.2. The ML-based algorithm is included once using the noisy log-spectra as features $y_{k,\ell}^{(\log)}$ and once using the combination of *a priori* SNR $\xi_{k,\ell}^{(\log)}$ and *a posteriori* SNR $\gamma_{k,\ell}^{(\log)}$. For this experiment, the CSNE [264] and the two talker noise are excluded from the noise type pool such that eight noise types remain. We train the DNN once on a set which includes mod. pink noise, mod. white noise, factory 1 noise and traffic noise. Note that this includes the traffic and factory noise which is also used for testing, i.e., this corresponds to a seen condition. For this condition, it is ensured that the noise realizations used for the training are not reused for testing. Therefore, only the first 120 s of the noise types are used while the last 120 s are used to embed the sentences for the listening experiment. The algorithms have also been evaluated in an unseen condition where all noise types are included in the training set except the one used for evaluation. Here, the full length of the training noise is utilized. For each sentence embedded in the training noise type, the peak level is varied between -26 dB and -6 dB, while the SNR is chosen between -5 dB and 15 dB. The minimum gain is set to $G_{\min} = -15$ dB in this experiment.

In each trial of the experiment, the participants compared six stimuli. In addition to the processed signals, the noisy signal is included and a reference signal is presented where the speech signal and the background noise are mixed at an SNR of 20 dB. Lastly, a low quality anchor is added where the speech signal is low pass filtered at 2 kHz and mixed at an SNR of -5 dB. This signal is enhanced using a non-ML-based enhancement algorithm where the noise PSD is estimated using [70] while the speech PSD is obtained using the decision-directed approach [12]. The smoothing constant is set to 0.9 and the signal is enhanced using the Wiener filter where a more aggressive lower limit of -20 dB is employed. This results in an anchor signal with very poor quality due to many musical tone artifacts and strong speech distortions. The audio examples used for the listening experiment are available under https://www.inf.uni-hamburg.de/en/inst/ab/sp/publications/tasl2017-dnn-rr.

A total of 11 subjects with age in the range of 24 to 38 years who are not familiar with single-channel signal processing have participated in the MUSHRA. The experiment took place in a quiet office. The diotic signals were presented via Beyerdynamic DT-770 Pro 250 Ohm headphones attached to an RME Fireface UFX+ sound card. All signals were normalized in amplitude. The test consisted of two phases. First, the participants were asked to complete a training phase to familiarize themselves with the presented sounds and to adjust the volume to a comfortable level. For this, a subset of the processed signals was presented. In the second part of the experiment, the participants were asked to rate the signals according to their overall preference on a scale from 0 to 100, where 0 was labeled with “bad” and 100 with “excellent”. The order of the presentation of algorithms and conditions were randomized between all subjects.

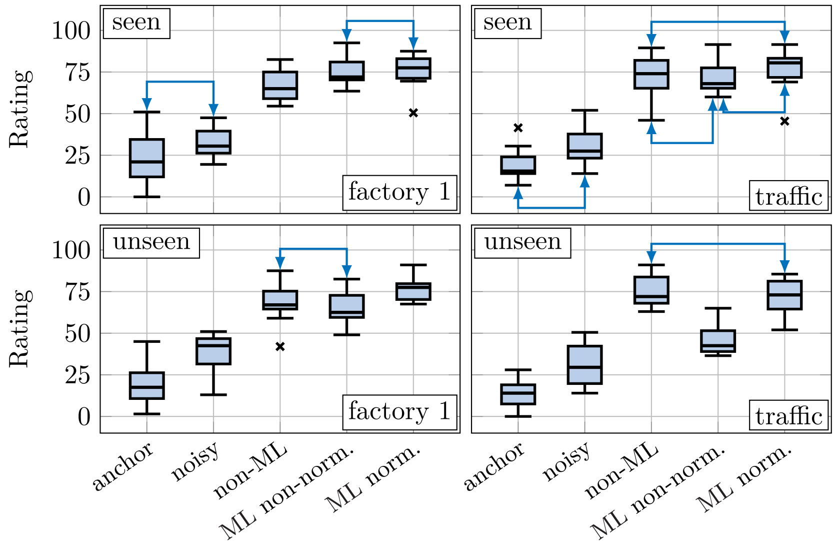

Fig. 8.5.: Box plots for the subjective rating of different enhancement schemes. The left column shows the results for factory 1 noise and the right column for traffic noise as test signals. The rows show different training strategies. The linking lines show pairings that are *not* identified as statistically significant by the post-hoc tests.

8.4.2. Results

For the evaluation, we average the ratings over the two speakers for each tested scenario. Further, the results are validated using a statistical analysis. For each acoustic scenario, a repeated measures analysis of variance (ANOVA) [266] is performed to test if the factor "enhancement algorithm" has a significant effect on the participants' rating. For this, we employ a significance level of 5 % for all statistical tests. For each acoustic scenario, we validated that the residuals of the general linear model fitted during the process of the repeated measures ANOVA are normally distributed using the Shapiro-Wilk test [267]. The sphericity assumption has been validated using Mauchly's test [268] and a Greenhouse-Geisser correction [269] is employed in cases where it has been violated. In all acoustic scenarios, the enhancement algorithms have a statistically significant effect on ratings. Hence, post-hoc tests are used to identify the sources of significance. For this, matched pair t-tests with a Bonferroni-Holm [270] correction are employed to account for the error inflation. The results are shown in Fig. 8.5 where the ratings that are statistically *not* significantly different are indicated by linking lines.

All listeners were able to correctly identify the hidden reference and assigned the highest score to it. The anchor signal and the noisy signal were assigned the lowest scores in most of the cases. For the seen conditions, all enhancement schemes have been rated similar in traffic noise, while in factory noise, both ML-based speech enhancement schemes yield slightly better results than the non-ML-based algorithm. For the unseen conditions, the ratings for the ML-based approach only using the non-normalized noisy log-spectra as features drop while the ratings for the proposed normalized features remain high. Additionally, the proposed features show slightly higher ratings in comparison to the non-ML enhancement scheme in factory noise. The statistical evaluation confirms that the highlighted differences are statistically significant.

8.5. SUMMARY

In this chapter, we propose features for ML-based speech enhancement which incorporate non-ML-based estimates of the speech and noise PSD. The goal is to improve the robustness of ML-based enhancement scheme towards unseen noise conditions. In contrast to the already existing noise aware training [23], [208]–[210], the noise PSD is not appended but used as a normalizing term. This results in the *a priori* SNR and the *a posteriori* SNR which exhibit the advantageous property of being scale-invariant. For the noisy log-spectra, the performance of the ML-based enhancement scheme in terms of PESQ is low in unseen noise conditions. Appending an estimate of the noise PSD has only a little impact on the performance in PESQ while the intelligibility predicted by STOI increases. Using the proposed normalized features, however, the performance of the ML-based enhancement scheme is generally higher as for the compared algorithms in both instrumental measures. This is supported by the MUSHRA-based listening experiments where, in unseen noise conditions, the proposed combination was significantly preferred over the ML-based enhancement scheme using only the log-spectra of the noisy observations. Audio examples are available under https://www.inf.uni-hamburg.de/en/inst/ab/sp/publications/tasl2017-dnn-rr. Feed-forward networks clearly benefit from the proposed normalized features, but their effect on other architectures such as recurrent neural networks or convolutional networks remains a question for future research.

Part V.

Conclusions and Further Research

CHAPTER 9

CONCLUSIONS AND FURTHER RESEARCH

This chapter summarizes the main contributions of the thesis and provides directions for further research.

9.1. CONCLUSIONS

In many speech communication based applications, undesired background noises may be captured by the microphones in addition to the desired speech signal. This generally degrades speech such that the perceived quality and potentially also the intelligibility is reduced. This does not only affect human perception but also affects automatic speech recognition systems. This thesis dealt with robust approaches to reduce background noise such that the quality and, if possible, also the speech intelligibility are improved. Single-channel algorithms have been considered that either process the input of a single microphone or the output of a spatial filtering algorithm. The main objectives of this thesis was to improve non-ML-based and ML-based single-channel speech enhancement algorithms by exploiting synergies of both approaches. On the one hand, ML-based approaches show potential for suppressing highly non-stationary noise types, which non-ML approaches fail to suppress. On the other hand, non-ML approaches are more robust in unseen noise conditions. In this thesis, various aspects of the respective approaches and their combinations have been highlighted. All proposed methods have been evaluated using instrumental measures and selected methods have been validated by subjective listening tests. In the following paragraphs, the respective contributions are summarized.

In Chapter 3 and Chapter 4, methods for estimating and correcting a bias occurring in noise PSD estimation algorithms were proposed. The methods allow the estimation bias of noise PSD estimators based on first-order recursive smoothing filters with adaptively changing smoothing factors to be compensated. The concept of first-order adaptive recursive smoothing filters and their basic properties were introduced in Chapter 3. If the adaptive smoothing factor depends only on the ratio of the input and the previous output, which is generally the case for noise PSD estimators, the filter is scale-invariant. For signals that contain only noise, this allows the compensation of the bias by scaling the input or the output of the filter with a fixed correction factor. For determining the correction factor, two methods were proposed in Chapter 3. The first approach seeks to estimate the expected value of the filter output for which an iterative procedure was proposed. The second approach leverages the transition densities of the respective adaptive smoothing factors which model the distribution of the filter output given the previous filter output. To estimate the bias, the parameters of a candidate distribution were optimized such that the filter output distributions of two adjacent segments become as similar as possible. In the evaluation, both bias estimation methods were applied to quantify the bias of the SPP-based noise PSD estimator proposed in [70], [71] and a threshold based noise PSD estimator given in [81, Section 14.1.3]. In Monte-Carlo simulations, the accuracy of the proposed bias estimation methods was validated for stationary inputs that correspond to noise only signals. The SPP-based noise PSD estimator underestimates the noise PSD by 1.2 dB while the threshold based method underestimates the PSD by 10.2 dB. The bias estimation algorithms generally underestimate the error where the deviation from the reference values has been found to be less than 0.3 dB. The proposed estimation methods only estimate a fixed correction factor which implicitly assumes that the input comprises only noise. To avoid overestimations in speech presence, a time-varying correction factor was proposed which is given by the multiplication of the fixed correction factor and a time-varying Wiener like term. In a speech enhancement framework, the used PESQ measure indicates that using the time-varying correction factor yields virtually no difference for the SPP-based noise PSD estimator [70], [71]. For the threshold based estimation algorithm [81, Section 14.1.3], for which the bias is considerably larger, the performance in PESQ is boosted by up to 0.2 points.

In Chapter 4, an alternative method was proposed for correcting the bias caused by adaptive recursive smoothing filters. Again, a fixed multiplicative correction factor was employed which allows the compensation of the bias for stationary signals comprising only noise. However, the correction factor is used at a different point in the adaptive recursive smoothing equations and does not scale the filter input or output. Correspondingly, the fixed correction factor is generally different to the method proposed in Chapter 3. For estimating the fixed correction factor, we proposed an approach based on the iterative method in Chapter 3 which results in an algorithm, which allows the determination of the bias within a single iteration. In the evaluation, the SPP-based noise PSD estimator [70], [71] and the threshold based noise PSD estimator [81, Section 14.1.3] were considered again. The accuracy of the proposed estimation method was validated using Monte-Carlo simulations which reflect the estimation error of the noise PSD estimators for stationary signals containing only noise. The results show that the proposed method for estimating the correction factor slightly underestimates its value which results in a remaining underestimation of 0.6 dB for the threshold based noise PSD estimator and of 0.2 dB for the SPP-based noise PSD estimator. Similar to Chapter 3, the fixed correction factor was replaced by a time-varying version to cope with signals where speech is present. The alternative correction method has a small influence on the PESQ scores of the SPP-based noise PSD estimator, whereas PESQ is boosted by 0.2 points if the threshold based noise PSD estimator is considered.

In Chapter 5 and Chapter 6, we proposed to use a super-Gaussian clean speech estimator to improve the enhancement quality of MLSE-based enhancement schemes. Such enhancement methods leverage ML algorithms to model speech only by means of its spectral envelope. As a consequence, MLSE-based enhancement schemes overestimate the speech PSD between spectral speech harmonics. Our analysis of super-Gaussian clean speech estimators showed that these estimators are able to reduce the background noise when the speech PSD is overestimated. For the evaluation, we used super-Gaussian estimators in two exemplary MLSE-based enhancement algorithms. Instrumental measures such as PESQ and SegSNR show that the super-Gaussian clean speech estimators improve the performance over Gaussian estimators in MLSE-based approaches. Interestingly, the improvement achieved by the super-Gaussian clean speech estimator is much larger for MLSE approaches than for non-ML approaches that resolve the spectral harmonics. This indicates that the performance benefit comes mainly from the increased suppression between the spectral harmonics. The beneficial effect of the super-Gaussian estimators was further validated using subjective listening experiments. The MLSE-based approaches that employ super-Gaussian estimators are rated significantly better than the Gaussian based MLSE approaches.

Chapter 6 showed that the beneficial effect of super-Gaussian clean speech estimators on MLSE-based enhancement schemes can also be observed for estimators that have not been derived under an additive mixing model. Using the relations in [83], the parameters of the spectral super-Gaussian distribution used in Chapter 5 were mapped to the mean and variance in the log-spectral domain. These quantities were used in an MSE optimal clean speech estimator based on the MixMax model [26], [101]. The gain function that results by using the propagated means and variances in the MixMax based clean speech estimator shows similar properties as the super-Gaussian estimators derived under the additive mixing model in the spectral domain. Correspondingly, the super-Gaussian speech model also allows the MixMax based estimator to suppress the background noise if the speech PSD is overestimated. In terms of speech quality measured by PESQ, the estimator based on the MixMax model and the estimator based on the spectral additive mixing model yield similar results. Comparing both approaches using the log-kurtosis ratio, which was used as a measure of the musical tone artifacts, reveals that the MixMax based estimator significantly reduces musical tone artifacts.

MLSE-based speech enhancement algorithms have also been the topic of Chapter 7. But instead of using super-Gaussian models, a combination of ML and non-ML-based estimators was proposed, which is embedded in a statistical framework. The combination is realized by defining a likelihood model for each enhancement approach used in the combination and leveraging Bayesian statistics. Combining the MLSE-based enhancement approach with the non-ML enhancement scheme allows the suppression of the noise bursts during speech activity which results in higher PESQ scores. Further, the combination outperforms an already existing approach, which employs a harmonic model, and the compared non-ML approach. The PESQ scores of the combination are about 0.1 points higher than for the non-ML-based approach in the majority of the considered noise types.

In Chapter 8, normalized non-ML-based input features were proposed to improve the generalization of a DNN-based enhancement scheme. Instead of using non-normalized features such as noise aware training [23], [208]–[210], we proposed a novel set of features based on the *a priori* SNR and the *a posteriori* SNR. We showed that the proposed normalized features have various advantages over noise aware training and the noisy log-spectra, which were additionally included in the comparisons. The proposed features are independent of the overall level while the performance of the enhancement schemes varies with the signal level if the log-spectra and noise-aware features are used. Here, the variations can be up to 0.2 points in PESQ and up to 10 % in absolute error in STOI. Further, the proposed features generally yield higher scores in instrumental measures such as PESQ and STOI for unseen acoustic conditions. Again, the proposed features yield scores that are on average 0.2 points higher in PESQ than the compared non-normalized features. Further, the DNN-based enhancement scheme outperforms the compared non-ML enhancement schemes by 0.1 points in PESQ and by 5 to 10 % in STOI if the proposed features are employed. Interestingly, using only the *a posteriori* SNR yields similar results as the combination of the *a priori* and *a posteriori* SNR in most acoustic environments. This feature can be implemented by adding only little additional computational complexity to the baseline DNN using noisy log-spectra as input features. Further, it shows that the performance advantages of the proposed features are not the result of an increased feature dimensionality, but are rather caused by the employed normalization. The results of the instrumental measures were confirmed by subjective listening tests, which verify that the proposed combination of *a priori* SNR and *a posteriori* SNR is perceived significantly better than using only the noisy log-spectra in unseen noise conditions.

9.2. SUGGESTIONS TO FURTHER RESEARCH

In this thesis, ML-based speech enhancement, non-ML-based speech enhancement and possible synergies of both approaches were discussed. A large part of this thesis dealt with MLSE-based speech models, their shortcomings with respect to speech enhancement and possible solutions. A general disadvantage of methods considered in this thesis is that the corresponding speech model has to be inferred from the noisy environments. In very low SNR conditions, identifying the correct model becomes increasingly difficult, which inherently limits the performance of such approaches. A potentially promising approach to overcome this limitation is to include further modalities, e.g., visual cues, to improve the recognition. Recent research has shown that features extracted from lip movements allow the spoken phoneme to be inferred from video material. In addition, these features have been used for speech recognition, speech enhancement and synthesis of intelligible speech. As a consequence, if reliable cues are available from a visual modality, speech can be enhanced in very low SNR conditions. Using features extracted from the shape of the lips will most likely only allow the extraction of information about the speech spectral envelope. Hence, the research in this thesis on methods that allow noisy speech signals to be enhanced using spectral envelope models with low quality losses complements this approach. Additionally, visual cues may not only be used to identify the spoken phoneme but potentially reveal further properties of the speaker, e.g., the gender and emotions, and the acoustic environment which can be included in the enhancement process.

Even though simplified speech models, e.g., MLSE models, have advantages in terms of computational complexity and tractability for deriving solutions, simplifying assumptions, such as modeling only the envelope or ignoring correlations over time and frequency, are limiting factors of the performance for single-channel speech enhancement algorithms. Accordingly, using more powerful models to describe the target signal of speech enhancement algorithms appears to be a natural approach to increase the performance of speech enhancement algorithms. Learning a powerful speech model appears to be a feasible task compared to learning a general background noise model as speech exhibits a rather specific structure while any arbitrary sound may compose the background noise. More powerful speech models may be realized using dynamic NMF approaches, e.g., [141], [149], which are able to describe speech with high-resolution and scale-invariant frequency domain models that additionally include temporal dependencies. Recent research further shows that modern ML-based approaches, e.g., DNNs, make it possible to learn generative models where long-term temporal correlations are extracted from training data. The WaveNet approach proposed in [202] learns the conditional PDF of the current speech time domain sample given a large number of previous ones. This allows the generation of high quality speech signals in the time domain by sampling from the learned distribution. However, while NMF-based models often ignore correlations along frequency, WaveNet is limited to the time domain. Finding ways to obtain powerful and efficient models of speech that include temporal and spectral dependencies in a time-frequency representation and are suitable for speech enhancement remains an interesting topic for future research.

Besides using improved speech models, single-channel speech enhancement will also benefit from improving the tracking capabilities of noise PSD estimation algorithms. Indirectly, also the noise PSD estimation would benefit from the availability of more powerful speech models. The improved speech models might facilitate the discrimination of speech and non-speech sounds such that the tracking of highly non-stationary sounds can be improved. Further, combinations, where slow changes are tracked by non-ML approaches and fast, non-stationary changes are captured by ML-based methods, may increase the tracking speed of noise PSD estimators.

In this thesis, reduction of additive background noise has been the main topic. In many practical applications, however, speech may not only be degraded by additive background noise but also by reverberation. In [271], [272], a single-channel non-ML-based algorithm has been proposed to reduce reverberation. For this, the reverberant energy of the speech signal is estimated blindly from the noisy and reverberant observation. In the Reverb Challenge [18], this algorithm has been rated highest in terms of quality in comparison to other single-channel approaches. However, also more ML-based speech dereverberation methods have been considered in [152], [273], [274]. Correspondingly, using non-ML-based estimators in combination with ML enhancement schemes to improve joint denoising and dereverberation is another field for future research.

APPENDIX A

ANALYTIC SOLUTIONS FOR BIAS ESTIMATION

A.1. ANALYTICAL RESULTS FOR THE ITERATIVE BIAS ESTIMATION

Here, we present the analytic expression of the expected value in (3.9) that are obtained for the considered adaptive smoothing functions in (3.17) and (3.18). Here, we employ the simplification described in Section 3.2 again, i.e., $\overline{y}_{\ell-1}$ is replaced by the deterministic $\overline{y}^{(\mathrm{fix})}$. The following equations were derived under the assumption that y_ℓ follows an exponential distribution (3.16).

For the noise PSD estimator proposed in [81, Section 14.1.3], the expected value $\mathbb{E}\{\overline{y}_\ell\}$, i.e., the solution to (3.9) given (3.17), results in

$$\mathbb{E}\{\overline{y}_\ell\} = \Lambda^{y}_{k,\ell} \frac{(\alpha^{\downarrow}-1)\exp(\mathcal{R}) + (\alpha^{\uparrow}-\alpha^{\downarrow})(1+\mathcal{R})}{(\alpha^{\downarrow}-1)\exp(\mathcal{R}) + \alpha^{\uparrow}-\alpha^{\downarrow}}, \tag{A.1}$$

with $\mathcal{R} = \overline{y}^{(\mathrm{fix})}/\Lambda^{y}_{k,\ell}$.

The expected value $\mathbb{E}\{\overline{y}_\ell\}$ for the expression in (3.18) can be derived using the property of the geometric series [227, p. 1.112.1] and the analytic continuation property of the hypergeometric series [227, p. 9.130]. The result is

$$\mathbb{E}\{\overline{y}_\ell\} = \Lambda^{y}_{k,\ell} \frac{1 - {}_3F_2\left[1,\mathcal{T},\mathcal{T};\mathcal{T}+1,\mathcal{T}+1;-(1+\xi_{\mathcal{H}_1})\right]}{1 - {}_2F_1\left[1,\mathcal{T};\mathcal{T}+1;-(1+\xi_{\mathcal{H}_1})\right]}, \tag{A.2}$$

where ${}_pF_q$ is the generalized hypergeometric function with

$$\mathcal{T} = \mathcal{R}\frac{\xi_{\mathcal{H}_1}+1}{\xi_{\mathcal{H}_1}}. \tag{A.3}$$

A.2. ANALYTIC SOLUTIONS FOR THE SELF-SIMILARITY OPTIMIZATION

Here, we derive the analytic expressions of the inverse function $\mathcal{F}^{-1}(\cdot)$ and the derivative $\mathcal{F}'(\cdot)$ for the considered adaptive smoothing functions. Using these results, the conditional PDF $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ can be obtained with (3.15). For the derivations, we assume that $\mathcal{F}(\cdot)$ is given by the expression in (3.1).

For the adaptive smoothing function $\alpha_{\mathrm{Thr}}(y_\ell,\overline{y}_{\ell-1})$ in (3.17), [81, Section 14.1.3], the existence of an inverse $\mathcal{F}^{-1}(\cdot)$ depends on the relationship between the updated filter output $\overline{y}_\ell$ and the previous filter output $\overline{y}_{\ell-1}$. Under the assumption that $y_\ell \geq 0$, the adaptive smoothing given in (3.1) can be inverted if $\alpha^{\downarrow}\overline{y}_{\ell-1} \leq \overline{y}_{\ell} \leq \overline{y}_{\ell-1}$ or if $\overline{y}_{\ell} > \overline{y}_{\ell-1}$. For the first condition, the inverse is given by

$$\mathcal{F}_1^{-1}(\overline{y}_\ell) = \frac{\overline{y}_\ell - \alpha^{\downarrow}\overline{y}_{\ell-1}}{1-\alpha^{\downarrow}} \tag{A.4}$$

and the denominator of (3.15) is given by

$$\mathcal{F}'(\mathcal{F}_1^{-1}(\overline{y}_\ell)) = 1-\alpha^{\downarrow}. \tag{A.5}$$

For the case that $\overline{y}_\ell > \overline{y}_{\ell-1}$, the filter function in (3.1) can be inverted as

$$\mathcal{F}_2^{-1}(\overline{y}_\ell) = \frac{\overline{y}_\ell - \alpha^{\uparrow}\overline{y}_{\ell-1}}{1-\alpha^{\uparrow}} \tag{A.6}$$

where the denominator of (3.15) is

$$\mathcal{F}'(\mathcal{F}_2^{-1}(\overline{y}_\ell)) = 1-\alpha^{\uparrow}. \tag{A.7}$$

For some values of $\overline{y}_\ell$ none of the conditions applies so that $L=0$. For these $\overline{y}_\ell$, it follows that also $f(\overline{y}_\ell|\overline{y}_{\ell-1}) = 0$.

If $\alpha_{\mathrm{SPP}}(y_\ell, \overline{y}_{\ell-1})$ from (3.18), [70] is employed in (3.1), the filter equation can be inverted if $\overline{y}_{\ell-1}(1+\alpha_{\mathrm{SPP}}^{(\mathrm{fix})}(1+\xi_{\mathcal{H}_1}))/(2+\xi_{\mathcal{H}_1}) \leq \overline{y}_\ell \leq \mathcal{L}\overline{y}_{\ell-1}$ where also the assumption is made that $y_\ell > 0$. In other words, $f(\overline{y}_\ell|\overline{y}_{\ell-1})$ is zero if this condition is not fulfilled. The quantity $\mathcal{L}$ is given by

$$\mathcal{L} = \tilde{\mathcal{L}} + (1-\tilde{\mathcal{L}})\left(\alpha_{\mathrm{SPP}}^{(\mathrm{fix})} + \frac{1-\alpha_{\mathrm{SPP}}^{(\mathrm{fix})}}{1+(1+\xi_{\mathcal{H}_1})e^{-\tilde{\mathcal{L}}\xi_{\mathcal{H}_1}/(1+\xi_{\mathcal{H}_1})}}\right) \tag{A.8}$$

with

$$\tilde{\mathcal{L}} = \frac{\xi_{\mathcal{H}_1}+1}{\xi_{\mathcal{H}_1}}\left[1+\mathcal{W}_0\left(e^{-1-\xi_{\mathcal{H}_1}/(\xi_{\mathcal{H}_1}+1)}(\xi_{\mathcal{H}_1}+1)\right)\right]+1. \tag{A.9}$$

Here, $\mathcal{W}_0(\cdot)$ denotes the main branch of the Lambert-W function [275]. This function, together with its second real branch $\mathcal{W}_{-1}(\cdot)$, constitutes the inverse of the expression $f(x) = x\exp(x)$ [275]. One inverse function of the filter in (3.1) with respect to the smoothing function in (3.18) is

$$\mathcal{F}_1^{-1}(\overline{y}_\ell) = -\overline{y}_{\ell-1}\frac{1+\xi_{\mathcal{H}_1}}{\xi_{\mathcal{H}_1}}\mathcal{W}_0(\mathcal{S}_1) + \frac{\overline{y}_\ell - \alpha_{\mathrm{SPP}}^{(\mathrm{fix})}\overline{y}_{\ell-1}}{1-\alpha_{\mathrm{SPP}}^{(\mathrm{fix})}} \tag{A.10}$$

where $\mathcal{S}_1$ is given by

$$\mathcal{S}_1 = \frac{\xi_{\mathcal{H}_1}(1-\overline{y}_\ell/\overline{y}_{\ell-1})}{(1-\alpha_{\mathrm{SPP}}^{(\mathrm{fix})})(1+\xi_{\mathcal{H}_1})^2}\exp\left(\frac{\xi_{\mathcal{H}_1}(\overline{y}_\ell - \alpha_{\mathrm{SPP}}^{(\mathrm{fix})}\overline{y}_{\ell-1})}{\overline{y}_{\ell-1}(1-\alpha_{\mathrm{SPP}}^{(\mathrm{fix})})(1+\xi_{\mathcal{H}_1})}\right). \tag{A.11}$$

If the first condition holds and, additionally, $\overline{y}_\ell$ fulfills $\overline{y}_\ell > \overline{y}_{\ell-1}$, a second inverse can be found. The result is

$$\mathcal{F}_2^{-1}(\overline{y}_\ell) = -\overline{y}_{\ell-1}\frac{1+\xi_{\mathcal{H}_1}}{\xi_{\mathcal{H}_1}}\mathcal{W}_{-1}(\mathcal{S}_1) + \frac{\overline{y}_\ell - \alpha_{\mathrm{SPP}}^{(\mathrm{fix})}\overline{y}_{\ell-1}}{1-\alpha_{\mathrm{SPP}}^{(\mathrm{fix})}}. \tag{A.12}$$

Note that the conditions for the two inverse functions are not exclusive, i.e., there are values for $\overline{y}_\ell$ where both conditions are fulfilled. For these $\overline{y}_\ell$, the number of piecewise monotonic segments L is two. Finally, the derivative is given by

$$\mathcal{F}'(y) =(1-\alpha_{\mathrm{SPP}}^{(\mathrm{fix})})\left[\frac{\xi_{\mathcal{H}_1}\mathcal{S}_2}{(1+(1+\xi_{\mathcal{H}_1})\mathcal{S}_2)^2}\left(1-\frac{y}{\overline{y}_{\ell-1}}\right)\left(1-\frac{1}{1+(1+\xi_{\mathcal{H}_1})\mathcal{S}_2}\right)\right], \tag{A.13}$$

with

$$\mathcal{S}_2 = \exp\left(-\frac{y}{\overline{y}_{\ell-1}}\frac{\xi_{\mathcal{H}_1}}{1+\xi_{\mathcal{H}_1}}\right). \tag{A.14}$$

BIBLIOGRAPHY

[1] N. R. French and J. C. Steinberg, “Factors governing the intelligibility of speech sounds,” *The Journal of the Acoustical Society of America*, vol. 19, no. 1, pp. 90–119, 1947.

[2] J. M. Festen and R. Plomp, “Effects of fluctuating noise and interfering speech on the speech-reception threshold for impaired and normal hearing,” *The Journal of the Acoustical Society of America*, vol. 88, no. 4, pp. 1725–1736, Oct. 1, 1990.

[3] M. J. Middelweerd, J. M. Festen, and R. Plomp, “Difficulties with speech intelligibility in noise in spite of a normal pure-tone audiogram,” *Audiology: Official Organ of the International Society of Audiology*, vol. 29, no. 1, pp. 1–7, 1990. pmid: 2310349.

[4] “P.11: Effect of transmission impairments,” International Telecommunication Union, ITU-T recommendation, Mar. 1993. [Online]. Available: http://www.itu.int/rec/T-REC-P.11-199303-I/en.

[5] G. M. Bidelman, “Communicating in challenging environments: Noise and reverberation,” in *The Frequency-Following Response: A Window into Human Communication*, N. Kraus, S. Anderson, T. White-Schwoch, R. R. Fay, and A. N. Popper, Eds., Springer International Publishing, 2017, pp. 193–224.

[6] J. Benesty, M. Shoji, and C. Jingdong, *Speech Enhancement*, 1st ed., ser. Signals and Communication Technology. Springer-Verlag Berlin Heidelberg, 2005.

[7] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*, 1st ed. Springer-Verlag Berlin Heidelberg, 2008.

[8] A. K. Nábělek and P. A. Dagenais, “Vowel errors in noise and in reverberation by hearing-impaired listeners,” *The Journal of the Acoustical Society of America*, vol. 80, no. 3, pp. 741–748, Sep. 1, 1986.

[9] R. C. Hendriks, T. Gerkmann, and J. Jensen, *DFT-Domain Based Single-Microphone Noise Reduction for Speech Enhancement: A Survey of the State of the Art*, ser. Synthesis Lectures on Speech and Audio Processing 1. Morgan & Claypool Publishers, 2013, vol. 9, 80 pp.

[10] “Apparatus for suppressing noise and distortion in communication signals,” 3,180,936, Apr. 27, 1965.

[11] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 2, pp. 113–120, Apr. 1979.

[12] Y. Ephraim and D. Malah, “Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, Dec. 1984.

[13] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*, ser. Digital Signal Processing. Springer Berlin Heidelberg, 2001.

[14] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*, 1st ed., ser. Springer Topics in Signal Processing. Springer-Verlag Berlin Heidelberg, 2008, vol. 1.

[15] S. Doclo, W. Kellermann, S. Makino, and S. E. Nordholm, "Multichannel signal enhancement algorithms for assisted listening devices: Exploiting spatial diversity using multiple microphones," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 18–30, Mar. 2015.

[16] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 692–730, Apr. 2017.

[17] P. A. Naylor and N. D. Gaubitch, *Speech Dereverberation*, 1st ed., ser. Signals and Communication Technology. Springer London, 2010.

[18] T. Nakatani, W. Kellermann, P. Naylor, M. Miyoshi, and B. H. Juang, "Introduction to the special issue on processing reverberant speech: Methodologies and applications," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1673–1675, Sep. 2010.

[19] Y. Ephraim, "A Bayesian estimation approach for speech enhancement using hidden Markov models," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 725–735, Apr. 1992.

[20] S. Srinivasan, J. Samuelsson, and W. B. Kleijn, "Codebook driven short-term predictor parameter estimation for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 163–176, Jan. 2006.

[21] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2140–2151, Oct. 2013.

[22] F. Deng, C. Bao, and W. B. Kleijn, "Sparse hidden Markov models for speech enhancement in non-stationary noise environments," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 1973–1987, Nov. 2015.

[23] Y. Xu, J. Du, L. R. Dai, and C. H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, Jan. 2015.

[24] M. Kolbæk, Z. H. Tan, and J. Jensen, "Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 149–163, Jan. 2017.

[25] H. Sameti, H. Sheikhzadeh, L. Deng, and R. L. Brennan, "HMM-based strategies for enhancement of speech signals embedded in nonstationary noise," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 5, pp. 445–455, Sep. 1998.

[26] S. E. Chazan, J. Goldberger, and S. Gannot, "A hybrid approach for speech enhancement using MoG model and neural network phoneme classifier," *IEEE/ACM*

Transactions on Audio, Speech, and Language Processing, vol. 24, no. 12, pp. 2516–2530, Dec. 2016.

[27] Q. He, F. Bao, and C. Bao, "Multiplicative update of auto-regressive gains for codebook-based speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 3, pp. 457–468, Mar. 2017.

[28] 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, pp. 443–445, 1985.

[29] 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, pp. 504–512, Jul. 2001.

[30] J. S. Erkelens, R. C. Hendriks, R. Heusdens, and J. Jensen, "Minimum mean-square error estimation of discrete Fourier coefficients with generalized gamma priors," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 6, pp. 1741–1752, Aug. 2007.

[31] C. Breithaupt, M. Krawczyk, and R. Martin, "Parameterized MMSE spectral magnitude estimation for the enhancement of noisy speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, NV, USA, Apr. 4, 2008, pp. 4037–4040.

[32] R. C. Hendriks, R. Heusdens, and J. Jensen, "Log-spectral magnitude MMSE estimators under super-Gaussian densities," in *Interspeech*, Brighton, United Kingdom, 2009, pp. 1319–1322.

[33] M. Krawczyk-Becker and T. Gerkmann, "On MMSE-based estimation of amplitude and complex speech spectral coefficients under phase-uncertainty," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2251–2262, Dec. 2016.

[34] Y. Ephraim and H. L. V. Trees, "A signal subspace approach for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 251–266, Jul. 1995.

[35] H. Lev-Ari and Y. Ephraim, "Extension of the signal subspace speech enhancement approach to colored noise," *IEEE Signal Processing Letters*, vol. 10, no. 4, pp. 104–106, Apr. 2003.

[36] P. C. Hansen and S. H. Jensen, "Subspace-based noise reduction for speech signals via diagonal and triangular matrix decompositions: Survey and analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, p. 092 953, Dec. 1, 2007.

[37] S. M. Nørholm, J. Benesty, J. R. Jensen, and M. G. Christensen, "Single-channel noise reduction using unified joint diagonalization and optimal filtering," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, p. 37, Mar. 26, 2014.

[38] J. R. Jensen, J. Benesty, and M. G. Christensen, "Noise reduction with optimal variable span linear filters," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 631–644, Apr. 2016.

[39] K. Paliwal and A. Basu, “A speech enhancement method based on Kalman filtering,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, TX, USA, Apr. 1987, pp. 177–180.

[40] S. Gannot, D. Burshtein, and E. Weinstein, “Iterative and sequential Kalman filter-based speech enhancement algorithms,” *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 4, pp. 373–385, Jul. 1998.

[41] V. Grancharov, J. Samuelsson, and B. Kleijn, “On causal algorithms for speech enhancement,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 3, pp. 764–773, May 2006.

[42] S. So, A. E. W. George, R. Ghosh, and K. K. Paliwal, “Kalman filter with sensitivity tuning for improved noise reduction in speech,” *Circuits, Systems, and Signal Processing*, vol. 36, no. 4, pp. 1476–1492, Apr. 1, 2017.

[43] H. Kuttruff, *Acoustics: An Introduction*, 1st ed. CRC Press, Nov. 23, 2006.

[44] P. Vary and R. Martin, *Digital Speech Transmission: Enhancement, Coding and Error Concealment*. John Wiley & Sons, Ltd, 2006.

[45] J. Lim and A. Oppenheim, “Enhancement and bandwidth compression of noisy speech,” *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[46] R. Martin, “Speech enhancement based on minimum mean-square error estimation and supergaussian priors,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 845–856, Sep. 2005.

[47] D. Brillinger, *Time Series: Data Analysis and Theory*, ser. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, Jan. 1, 2001, 556 pp.

[48] W. Pearlman and R. Gray, “Source coding of the discrete Fourier transform,” *IEEE Transactions on Information Theory*, vol. 24, no. 6, pp. 683–692, Nov. 1978.

[49] C. H. You, S. N. Koh, and S. Rahardja, “Beta-order MMSE spectral amplitude estimation for speech enhancement,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 4, pp. 475–486, Jul. 2005.

[50] T. Lotter and P. Vary, “Speech enhancement by MAP spectral amplitude estimation using a super-Gaussian speech model,” *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 7, pp. 1–17, 2005.

[51] T. Gerkmann and R. Martin, “Empirical distributions of DFT-domain speech coefficients based on estimated speech variances,” in *Internation Workshop Acoustic Echo and Noise Control (IWAENC)*, Tel Aviv, Israel, 2010.

[52] T. Lotter and P. Vary, “Noise reduction by maximum a posteriori spectral amplitude estimation with supergaussian speech modeling,” in *International Workshop on Acoustic Echo and Noise Control (IWAENC)*, Kyoto, Japan, Sep. 2003, pp. 83–86.

[53] I. Andrianakis and P. R. White, “MMSE speech spectral amplitude estimators with chi and gamma speech priors,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, May 2006, pp. 1068–1071.

[54] R. Martin, “Speech enhancement using MMSE short time spectral estimation with gamma distributed speech priors,” in *IEEE International Conference on Acoustics,*

Speech and Signal Processing (ICASSP), Orlando, Florida, USA, May 2002, pp. 253–256.

[55] R. Martin and C. Breithaupt, "Speech enhancement in the DFT domain using Laplacian speech priors," in *International Workshop on Acoustic Echo and Noise Control (IWAENC)*, Kyoto, Japan, 2003, p. 8790.

[56] D. Wang and J. Lim, "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 4, pp. 679–681, 1982.

[57] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, vol. 53, no. 4, pp. 465–494, 2011.

[58] T. Gerkmann, M. Krawczyk, and R. Rehr, "Phase estimation in speech enhancement - unimportant, important, or impossible?" In *IEEE Convention of Electrical and Electronics Engineers in Israel (IEEEI)*, Eilat, Isreal, Nov. 2012, pp. 1–5.

[59] M. Krawczyk and T. Gerkmann, "STFT phase reconstruction in voiced speech for an improved single-channel speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1931–1940, Dec. 2014.

[60] T. Gerkmann and M. Krawczyk, "MMSE-optimal spectral amplitude estimation given the STFT-phase," *IEEE Signal Processing Letters*, vol. 20, no. 2, pp. 129–132, Feb. 2013.

[61] T. Gerkmann, "Bayesian estimation of clean speech spectral coefficients given a priori knowledge of the phase," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4199–4208, Aug. 2014.

[62] T. Gerkmann, M. Krawczyk-Becker, and J. Le Roux, "Phase processing for single-channel speech enhancement: History and recent advances," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 55–66, Mar. 2015.

[63] J. Sohn and W. Sung, "A voice activity detector employing soft decision based noise spectrum adaptation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, Seattle, WA, USA, 1998, pp. 365–368.

[64] J. Ramírez, J. C. Segura, C. Benítez, Á. de la Torre, and A. Rubio, "Efficient voice activity detection algorithms using long-term speech information," *Speech Communication*, vol. 42, no. 3–4, pp. 271–287, 2004.

[65] R. Martin and D. Kolossa, "Voice activity detection, noise estimation, and adaptive filters for acoustic signal enhancement," in *Techniques for Noise Robustness in Automatic Speech Recognition*, T. Virtanen, R. Singh, and B. Raj, Eds., John Wiley & Sons, Ltd, 2012, pp. 51–85.

[66] R. Martin, "Bias compensation methods for minimum statistics noise power spectral density estimation," *Signal Processing*, vol. 86, no. 6, pp. 1215–1229, 2006.

[67] A. Chinaev and R. Haeb-Umbach, "On optimal smoothing in minimum statistics based noise tracking," in *Interspeech*, Dresden, Germany, Sep. 2015.

[68] R. Yu, "A low-complexity noise estimation algorithm based on smoothing of noise power estimation and estimation bias correction," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 4421–4424.

[69] R. C. Hendriks, R. Heusdens, and J. Jensen, "MMSE based noise PSD tracking with low complexity," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Dallas, TX, USA, Mar. 2010, pp. 4266–4269.

[70] T. Gerkmann and R. C. Hendriks, "Noise power estimation based on the probability of speech presence," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2011, pp. 145–148.

[71] ——, "Unbiased MMSEe-based noise power estimation with low complexity and low tracking delay," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1383–1393, May 2012.

[72] H. G. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Detroit, MI, USA, May 1995, pp. 153–156.

[73] A. Chinaev, J. Heymann, L. Drude, and R. Haeb-Umbach, "Noise-presence-probability-based noise PSD estimation by using DNNs," in *ITG Conference on Speech Communication*, Paderborn, Germany, Oct. 2016.

[74] I. Cohen and B. Berdugo, "Noise estimation by minima controlled recursive averaging for robust speech enhancement," *IEEE Signal Processing Letters*, vol. 9, no. 1, pp. 12–15, Jan. 2002.

[75] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 466–475, Sep. 2003.

[76] N. Fan, J. Rosca, and R. Balan, "Speech noise estimation using enhanced minima controlled recursive averaging," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, Honolulu, HI, USA, Apr. 2007, pp. 581–584.

[77] J.-M. Kum, Y. S. Park, and J. H. Chang, "Speech enhancement based on minima controlled recursive averaging incorporating conditional maximum a posteriori criterion," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 4417–4420.

[78] R. Hendriks, J. Jensen, and R. Heusdens, "Noise tracking using DFT domain subspace decompositions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 3, pp. 541–553, Mar. 2008.

[79] R. C. Hendriks, R. Heusdens, J. Jensen, and U. Kjems, "Fast noise PSD estimation with low complexity," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3881–3884.

[80] F. Heese and P. Vary, "Noise PSD estimation by logarithmic baseline tracing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, QLD, Australia, Apr. 2015, pp. 4405–4409.

[81] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*, ser. Adaptive and Learning Systems for Signal Processing, Communication and Control. Wiley & Sons, 2004.

[82] C. Breithaupt, T. Gerkmann, and R. Martin, "A novel a priori SNR estimation approach based on selective cepstro-temporal smoothing," in *IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), Las Vegas, NV, USA, Apr. 4, 2008, pp. 4897–4900.

[83] T. Gerkmann and R. Martin, "On the statistics of spectral amplitudes after variance reduction by temporal cepstrum smoothing and cepstral nulling," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4165–4174, 2009.

[84] Y. Ephraim, D. Malah, and B. H. Juang, "On the application of hidden Markov models for enhancing noisy speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 1846–1856, Dec. 1989.

[85] Y. Ephraim, "Gain-adapted hidden Markov models for recognition of clean and noisy speech," *IEEE Transactions on Signal Processing*, vol. 40, no. 6, pp. 1303–1316, Jun. 1992.

[86] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[87] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[88] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer, 2006.

[89] B.-H. Juang and L. Rabiner, "Mixture autoregressive hidden Markov models for speech signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 6, pp. 1404–1413, Dec. 1985.

[90] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Machine Learning*, vol. 29, no. 2, pp. 245–273, Nov. 1997.

[91] D. Y. Zhao and W. B. Kleijn, "HMM-based gain modeling for enhancement of speech in noise," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 882–892, Mar. 2007.

[92] D. Y. Zhao, W. B. Kleijn, A. Ypma, and B. de Vries, "Online noise estimation using stochastic-gain HMM for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 835–846, May 2008.

[93] D. M. Titterington, "Recursive parameter estimation using incomplete data," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 46, no. 2, pp. 257–267, 1984.

[94] E. Weinstein, M. Feder, and A. V. Oppenheim, "Sequential algorithms for parameter estimation based on the Kullback-Leibler information measure," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 9, pp. 1652–1654, Sep. 1990.

[95] V. Krishnamurthy and J. B. Moore, "On-line estimation of hidden Markov model parameters based on the Kullback-Leibler information measure," *IEEE Transactions on Signal Processing*, vol. 41, no. 8, pp. 2557–2573, Aug. 1993.

[96] J. Hao, T. W. Lee, and T. J. Sejnowski, "Speech enhancement using Gaussian scale mixture models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1127–1136, Aug. 2010.

[97] N. Mohammadiha, R. Martin, and A. Leijon, “Spectral domain speech enhancement using HMM state-dependent super-Gaussian priors,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 253–256, Mar. 2013.

[98] N. Mohammadiha and A. Leijon, “Nonnegative HMM for babble noise derived from speech HMM: Application to speech enhancement,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 998–1011, May 2013.

[99] A. Aroudi, H. Veisi, and H. Sameti, “Hidden Markov model-based speech enhancement using multivariate Laplace and Gaussian distributions,” *IET Signal Processing*, vol. 9, no. 2, 177–185(8), Apr. 2015.

[100] S. T. Roweis, “One microphone source separation,” in *Advances in Neural Information Processing Systems (NIPS)*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., Vancouver, BC, Canada, Dec. 2001, pp. 793–799.

[101] D. Burshtein and S. Gannot, “Speech enhancement using a mixture-maximum model,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 6, pp. 341–351, Sep. 2002.

[102] S. T. Roweis, “Factorial models and refiltering for speech separation and denoising,” in *Eurospeech*, Geneva, Switzerland, Sep. 2003.

[103] T. Kristjansson and J. Hershey, “High resolution signal reconstruction,” in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, St. Thomas, VI, USA, Nov. 2003, pp. 291–296.

[104] T. Yoshioka and T. Nakatani, “Speech enhancement based on log spectral envelope model and harmonicity-derived spectral mask, and its coupling with feature compensation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 5064–5067.

[105] J. Le Roux and J. R. Hershey, “Indirect model-based speech enhancement,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 4045–4048.

[106] H. Veisi and H. Sameti, “Cepstral-domain HMM-based speech enhancement using vector Taylor series and parallel model combination,” in *International Conference on Information Science, Signal Processing and Their Applications (ISSPA)*, Montreal, QC, Canada, Jul. 2012, pp. 298–303.

[107] H. Veisi and H. Sameti, “Speech enhancement using hidden Markov models in Mel-frequency domain,” *Speech Communication*, vol. 55, no. 2, pp. 205–220, Feb. 1, 2013.

[108] B. J. Frey, T. T. Kristjansson, L. Deng, and A. Acero, “ALGONQUIN - learning dynamic noise models from noisy speech for robust speech recognition,” in *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, BC, Canada, Dec. 2001, pp. 1165–1171.

[109] B. Frey, L. Deng, T. Kristjansson, and A. Acero, “ALGONQUIN: Iterating Laplace’s method to remove multiple types of acoustic distortion for robust speech recognition,” in *Eurospeech*, Aalborg, Denmark, Sep. 2001.

[110] V. Stouten, H. V. Hamme, and P. Wambacq, “Joint removal of additive and convolutional noise with model-based feature enhancement,” in *IEEE International*

Conference on Acoustics, Speech, and Signal Processing (ICASSP), Montreal, QC, Canada, May 2004, pp. 949–952.

[111] V. Stouten, H. Van Hamme, and P. Wambacq, "Multiple stream model-based feature enhancement for noise robust speech recognition," in *COST278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, Norwich, UK, Aug. 2004.

[112] A. Nádas, D. Nahamoo, and M. A. Picheny, "Speech recognition using noise-adaptive prototypes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 10, pp. 1495–1503, Oct. 1989.

[113] T. Yoshioka, T. Nakatani, and H. Okuno, "Noisy speech enhancement based on prior knowledge about spectral envelope and harmonic structure," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, Dallas, TX, USA, Mar. 2010, pp. 4270–4273.

[114] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[115] S. Srinivasan, J. Samuelsson, and W. Kleijn, "Codebook-based Bayesian speech enhancement for nonstationary environments," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 2, pp. 441–452, Feb. 2007.

[116] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Interspeech*, Pittsburgh, PA, USA, Sep. 2006, pp. 1652–1655.

[117] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007.

[118] N. Mohammadiha, T. Gerkmann, and A. Leijon, "A new linear MMSE filter for single channel speech enhancement based on nonnegative matrix factorization," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2011, pp. 45–48.

[119] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 21, 1999.

[120] A. Cichoki, R. Zdunek, A. H. Phan, and S.-I. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*, 1st ed. John Wiley & Sons, Ltd, Oct. 7, 2009.

[121] L. Lee and D. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems 13*, Denver, CO, USA, 2000, pp. 556–562.

[122] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, Sep. 11, 2008.

[123] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, Jun. 14, 2011.

[124] C. Févotte and A. T. Cemgil, “Nonnegative matrix factorizations as probabilistic inference in composite models,” in *European Signal Processing Conference (EUSIPCO)*, Glasgow, UK, Aug. 2009, pp. 1913–1917.

[125] A. T. Cemgil, “Bayesian inference for nonnegative matrix factorisation models,” *Computational Intelligence and Neuroscience*, vol. 2009, p. 17, 2009.

[126] M. D. Hoffman, “Poisson-uniform nonnegative matrix factorization,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 5361–5364.

[127] P. Smaragdis, B. Raj, and M. Shashanka, “A probabilistic latent variable model for acoustic modeling,” in *In Workshop on Advances in Models for Acoustic Processing at NIPS*, Whistler, BC, Canada, 2006.

[128] M. Shashanka, B. Raj, and P. Smaragdis, “Sparse overcomplete latent variable decomposition of counts data,” in *Advances in Neural Information Processing Systems (NIPS)*, Vancouver, BC, Canada, 2008, pp. 1313–1320.

[129] T. Hofmann, “Probabilistic Latent Semantic Indexing,” in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, 1999, pp. 50–57.

[130] ——, “Unsupervised learning by probabilistic latent semantic analysis,” *Machine Learning*, vol. 42, no. 1-2, pp. 177–196, Jan. 1, 2001.

[131] A. Ozerov and C. Févotte, “Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 550–563, Mar. 2010.

[132] H. Sawada, H. Kameoka, S. Araki, and N. Ueda, “Multichannel extensions of nonnegative matrix factorization with complex-valued data,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 971–982, May 2013.

[133] M. N. Schmidt and J. Larsen, “Reduction of non-stationary noise using a nonnegative latent variable decomposition,” in *IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, Cancun, Mexico, Oct. 2008, pp. 486–491.

[134] M. Sun, Y. Li, J. F. Gemmeke, and X. Zhang, “Speech enhancement under low SNR conditions via noise estimation using sparse and low-rank NMF with Kullback-Leibler divergence,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 7, pp. 1233–1242, Jul. 2015.

[135] D. Baby, T. Virtanen, J. F. Gemmeke, and H. van Hamme, “Coupled dictionaries for exemplar-based speech enhancement and automatic speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 1788–1799, Nov. 2015.

[136] J. Le Roux, F. Weninger, and J. Hershey, “Sparse NMF – half-baked or well done?” Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, TR2015-023, Mar. 2015. [Online]. Available: http://www.merl.com/publications/TR2015-023.

[137] S. Voran, “Exploration of the additivity approximation for spectral magnitudes,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2015.

[138] N. Mohammadiha and S. Doclo, “Single-channel dynamic exemplar-based speech enhancement,” in *Interspeech*, Singapore, Singapore, Sep. 2014, pp. 2690–2694.

[139] P. O. Hoyer, “Non-negative matrix factorization with sparseness constraints,” *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, Nov 2004.

[140] C. Févotte, J. le Roux, and J. R. Hershey, “Non-negative dynamical system with application to speech and audio,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 3158–3162.

[141] N. Mohammadiha, P. Smaragdis, G. Panahandeh, and S. Doclo, “A state-space approach to dynamic nonnegative matrix factorization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 949–959, Feb. 2015.

[142] G. J. Mysore and P. Smaragdis, “A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 17–20.

[143] C. Joder, F. Weninger, D. Virette, and B. Schuller, “Integrating noise estimation and factorization-based speech separation: A novel hybrid approach,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 131–135.

[144] G. J. Mysore, P. Smaragdis, and B. Raj, “Non-negative hidden Markov modeling of audio with application to source separation,” in *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, St. Malo, France, Sep. 27, 2010, pp. 140–148.

[145] C. Joder, F. Weninger, F. Eyben, D. Virette, and B. Schuller, “Real-time speech separation by semi-supervised nonnegative matrix factorization,” in *Latent Variable Analysis and Signal Separation*, ser. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, Mar. 12, 2012, pp. 322–329. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-28551-6_40 (visited on 09/22/2017).

[146] P. Smaragdis, C. Févotte, G. J. Mysore, N. Mohammadiha, and M. Hoffman, “Static and dynamic source separation using nonnegative factorizations: A unified view,” *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 66–75, May 2014.

[147] K. W. Wilson, B. Raj, and P. Smaragdis, “Regularized non-negative matrix factorization with temporal dependencies for speech denoising,” in *Interspeech*, Brisbane, QLD, Australia, Sep. 2008.

[148] C. Févotte, “Majorization-minimization algorithm for smooth Itakura-Saito nonnegative matrix factorization,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 1980–1983.

[149] N. Mohammadiha, P. Smaragdis, and A. Leijon, “Prediction based filtering and smoothing to exploit temporal dependencies in NMF,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 873–877.

[150] A. Ozerov, C. Févotte, and M. Charbit, "Factorial scaled hidden Markov model for polyphonic audio representation and source separation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2009, pp. 121–124.

[151] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2067–2080, Sep. 2011.

[152] D. Baby and H. van Hamme, "Joint denoising and dereverberation using exemplar-based sparse representations and decaying norm constraint," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 2024–2035, Oct. 2017.

[153] E. M. Grais and H. Erdogan, "Discriminative nonnegative dictionary learning using cross-coherence penalties for single channel source separation," in *Interspeech*, Lyon, France, Aug. 2013, pp. 808–812.

[154] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Supervised non-euclidean sparse NMF via bilevel optimization with applications to speech enhancement," in *Joint Workshop on Hands-Free Speech Communication and Microphone Arrays (HSCMA)*, Villers-les-Nancy, France, May 2014, pp. 11–15.

[155] F. Weninger, J. Le Roux, J. R. Hershey, and S. Watanabe, "Discriminative NMF and its application to single-channel source separation," in *Interspeech*, Singapore, Singapore, Sep. 2014, pp. 865–869.

[156] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, Dec. 1, 1943.

[157] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *IRE WESCON Convention*, Los Angeles, CA, Aug. 1960, pp. 96–104.

[158] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, Jan. 1, 1991.

[159] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, May 17, 2006.

[160] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, Chia Laguna Resort, Sardinia, Italy, May 2010, pp. 249–256.

[161] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *International Conference on Machine Learning*, Haifa, Israel, Jun. 2010, pp. 807–814.

[162] S. Tamura and A. Waibel, "Noise reduction using connectionist models," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, New York, NY, USA, Apr. 1988, pp. 553–556.

[163] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust ASR," in *Interspeech*, Portland, OR, USA, Sep. 2012, pp. 22–25.

[164] Y. Wang and D. Wang, "Boosting classification based speech separation using temporal dynamics," in *Interspeech*, Portland, OR, USA, Sep. 2012, pp. 1528–1531.

[165] Y. Wang and D. Wang, "Cocktail party processing via structured prediction," in *Advances in Neural Information Processing Systems (NIPS)*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., South Lake Tahoe, NV, USA, Dec. 2012, pp. 224–232.

[166] S. Suhadi, C. Last, and T. Fingscheidt, "A data-driven approach to a priori SNR estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 186–195, Jan. 2011.

[167] J. Chen, Y. Wang, S. E. Yoho, D. Wang, and E. W. Healy, "Large-scale training to increase speech intelligibility for hearing-impaired listeners in novel noises," *The Journal of the Acoustical Society of America*, vol. 139, no. 5, pp. 2604–2612, 2016.

[168] J. Chen, Y. Wang, and D. Wang, "A feature study for classification-based speech separation at low signal-to-noise ratios," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1993–2002, Dec. 2014.

[169] 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, Dec. 2014.

[170] H. Erdogan, J. R. Hershey, S. Watanabe, and J. le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, QLD, Australia, Apr. 2015, pp. 708–712.

[171] Y. Wang, K. Han, and D. Wang, "Exploring monaural features for classification-based speech segregation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 270–279, Feb. 2013.

[172] M. Delfarah and D. Wang, "Features for masking-based monaural speech separation in reverberant conditions," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1085–1094, May 2017.

[173] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[174] N. Moritz, J. Anemüller, and B. Kollmeier, "Amplitude modulation spectrogram based features for robust speech recognition in noisy and reverberant environments," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 5492–5495.

[175] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, Oct. 1994.

[176] M. R. Schädler, B. T. Meyer, and B. Kollmeier, "Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 131, no. 5, pp. 4134–4151, May 1, 2012.

[177] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *IEEE Global*

Conference on Signal and Information Processing (GlobalSIP), Atlanta, GA, USA, Dec. 2014, pp. 577–581.

[178] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 483–492, Mar. 2016.

[179] D. Wang, U. Kjems, M. S. Pedersen, J. B. Boldt, and T. Lunner, "Speech intelligibility in background noise with ideal binary time-frequency masking," *The Journal of the Acoustical Society of America*, vol. 125, no. 4, pp. 2336–2347, 2009.

[180] J. Jensen and R. Hendriks, "Spectral magnitude minimum mean-square error estimation using binary and continuous gain functions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 92–102, Jan. 2012.

[181] Z. Wang, X. Wang, X. Li, Q. Fu, and Y. Yan, "Oracle performance investigation of the ideal masks," in *IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, Xi'an, China, Sep. 2016, pp. 1–5.

[182] L. Sun, J. Du, L. R. Dai, and C. H. Lee, "Multiple-target deep learning for LSTM-RNN based speech enhancement," in *Hands-Free Speech Communications and Microphone Arrays (HSCMA)*, San Francisco, CA, USA, Mar. 2017, pp. 136–140.

[183] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Transactions on Signal Processing*, vol. 39, no. 9, pp. 2101–2104, Sep. 1991.

[184] N. Benvenuto and F. Piazza, "On the complex backpropagation algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 967–969, Apr. 1992.

[185] L. Drude, B. Raj, and R. Haeb-Umbach, "On the appropriateness of complex-valued neural networks for speech enhancement," in *Interspeech*, San Francisco, CA, USA, Sep. 2016, pp. 1745–1749.

[186] Y. S. Lee, C. Y. Wang, S. F. Wang, J. C. Wang, and C. H. Wu, "Fully complex deep neural network for phase-incorporating monaural source separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 281–285.

[187] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 1562–1566.

[188] ——, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2136–2147, Dec. 2015.

[189] Y. H. Tu, J. Du, L. R. Dai, and C. H. Lee, "Speech separation based on signal-noise-dependent deep neural networks for robust speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, QLD, Australia, Apr. 2015, pp. 61–65.

[190] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Montreal, QC, Canada, 2014, pp. 2672–2680.

[191] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech enhancement generative adversarial network," in *Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 3642–3646.

[192] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Interspeech*, Lyon, France, Aug. 2013.

[193] Y. Xu, J. Du, L. R. Dai, and C. H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, Jan. 2014.

[194] D. Liu, P. Smaragdis, and M. Kim, "Experiments on deep learning for speech denoising," in *Interspeech*, Singapore, Singapore, Sep. 2014.

[195] S. Nie, H. Zhang, X. Zhang, and W. Liu, "Deep stacking networks with time series for speech separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 6667–6671.

[196] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. L. Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, ser. Lecture Notes in Computer Science, Liberec, Czech Republic: Springer, Cham, Aug. 25, 2015, pp. 91–99. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-22482-4_11 (visited on 10/11/2017).

[197] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[198] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning (ICML)*, Atlanta, GA, USA, Jun. 2013, pp. 1310–1318.

[199] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1, 1997. [Online]. Available: https://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735 (visited on 08/03/2018).

[200] L. Hui, M. Cai, C. Guo, L. He, W. Q. Zhang, and J. Liu, "Convolutional maxout neural networks for speech separation," in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Abu Dhabi, United Arab Emirates, Dec. 2015, pp. 24–27.

[201] S.-W. Fu, Y. Tsao, and X. Lu, "SNR-aware convolutional neural network modeling for speech enhancement," in *Interspeech*, San Francisco, CA, USA, Sep. 2016, pp. 3768–3772.

[202] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," Sep. 12, 2016. arXiv: 1609.03499.

[203] D. Rethage, J. Pons, and X. Serra, "A Wavenet for speech denoising," Jun. 22, 2017. arXiv: 1706.07162.

[204] K. Qian, Y. Zhang, S. Chang, X. Yang, D. Florêncio, and M. Hasegawa-Johnson, "Speech enhancement using Bayesian wavenet," in *Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 2013–2017.

[205] E. W. Healy, S. E. Yoho, Y. Wang, and D. Wang, "An algorithm to improve speech recognition in noise for hearing-impaired listeners," *The Journal of the Acoustical Society of America*, vol. 134, no. 4, pp. 3029–3038, Oct. 1, 2013.

[206] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1381–1390, Jul. 2013.

[207] J. Chen and D. Wang, "Long short-term memory for speaker generalization in supervised speech separation," *The Journal of the Acoustical Society of America*, vol. 141, no. 6, pp. 4705–4714, Jun. 1, 2017.

[208] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 7398–7402.

[209] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "Dynamic noise aware training for speech enhancement based on deep neural networks," in *Interspeech*, Singapore, Singapore, Sep. 2014.

[210] A. Kumar and D. Florencio, "Speech enhancement in multiple-noise conditions using deep neural networks," in *Interspeech*, San Francisco, CA, USA, Sep. 2016, pp. 3738–3742.

[211] Q. Wang, J. Du, L. R. Dai, and C. H. Lee, "Joint noise and mask aware training for DNN-based speech enhancement with sub-band features," in *Hands-Free Speech Communications and Microphone Arrays (HSCMA)*, San Francisco, CA, USA, Mar. 2017, pp. 101–105.

[212] S. Mirsamadi and I. Tashev, "Causal speech enhancement combining data-driven learning and suppression rule estimation," in *Interspeech*, San Francisco, CA, USA, Sep. 2016, pp. 2870–2874. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-437.

[213] M. Sun, X. Zhang, H. V. hamme, and T. F. Zheng, "Unseen noise estimation using separable deep auto encoder for speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 93–104, Jan. 2016.

[214] M. Kim and P. Smaragdis, "Adaptive denoising autoencoders: A fine-tuning scheme to learn from test mixtures," in *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Liberec, Czech Republic, Aug. 25, 2015, pp. 100–107.

[215] R. Rehr and T. Gerkmann, "On the bias of adaptive first-order recursive smoothing," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2015.

[216] ——, "An analysis of adaptive recursive smoothing with applications to noise PSD estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 2, pp. 397–408, Feb. 2017.

[217] ——, “Bias correction methods for adaptive recursive smoothing with applications in noise PSD estimation,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 206–210.

[218] ——, “On the importance of super-Gaussian speech priors for machine-learning based speech enhancement,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 357–366, Feb. 2018.

[219] ——, “MixMax approximation as a super-Gaussian log-spectral amplitude estimator for speech enhancement,” in *Interspeech*, Stockholm, Sweden, Aug. 2017.

[220] ——, “A combination of pre-trained approaches and generic methods for an improved speech enhancement,” in *ITG Conference on Speech Communication*, Paderborn, Germany, Oct. 2016, pp. 51–55.

[221] ——, “Robust DNN-based speech enhancement with limited training data,” in *ITG Conference on Speech Communication*, Oldenburg, Germany, Oct. 2018.

[222] D. Mauler and R. Martin, “Improved reproduction of stops in noise reduction systems with adaptive windows and nonstationarity detection,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 1, p. 469 480, 2009.

[223] M. Berouti, R. Schwartz, and J. Makhoul, “Enhancement of speech corrupted by acoustic noise,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Washington, D.C., USA, Apr. 1979, pp. 208–211.

[224] P. J. Schreier and L. L. Scharf, *Statistical Signal Processing of Complex-Valued Data: The Theory of Improper and Noncircular Signals*. Cambridge University Press, 2010.

[225] (2010). Exp-gamma distribution, [Online]. Available: https://reference.wolfram.com/language/ref/ExpGammaDistribution.html (visited on 01/12/2018).

[226] Y. Ephraim and M. Rahim, “On second-order statistics and linear estimation of cepstral coefficients,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 162–176, Mar. 1999.

[227] I. S. Gradshteyn and I. W. Ryzhik, *Table of Integrals, Series, and Products*, 7th ed., D. Zwillinger and V. Moll, Eds. Academic Press, Feb. 2007.

[228] M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions : With Formulas, Graphs, and Mathematical Tables*, 9th ed., New York: Dover Publ., 1973.

[229] J. Taghia, J. Taghia, N. Mohammadiha, J. Sang, V. Bouse, and R. Martin, “An evaluation of noise power spectral density estimation algorithms in adverse acoustic environments,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 4640–4643.

[230] “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,” International Telecommunication Union, ITU-T recommendation, Jan. 2001. [Online]. Available: http://www.itu.int/rec/T-REC-P.862-200102-I/en.

[231] R. C. Hendriks, J. Jensen, and R. Heusdens, “DFT domain subspace based noise tracking for speech enhancement,” in *Interspeech*, Antwerp, Belgium, Aug. 2007, pp. 830–833.

[232] L. Breiman, *Probability*, ser. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1968.

[233] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.

[234] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. 15, no. 1, pp. 52–60, Feb. 1967.

[235] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, Mar. 1951.

[236] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[237] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. McGraw-Hill, 2002, vol. McGraw-Hill Series in Electrical Engineering: Communications and Signal Processing.

[238] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229–238, Jan. 2008.

[239] H. J. M. Steeneken and F. W. M. Geurtsen, "Description of the RSG.10 noise database," TNO Institute for perception, Technical Report IZF 1988-3, 1988.

[240] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, *TIMIT acoustic-phonetic continuous speech corpus*, 1993.

[241] U. Şimşekli, J. Le Roux, and J. R. Hershey, "Non-negative source-filter dynamical system for speech enhancement," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 6206–6210.

[242] C. Breithaupt and R. Martin, "Analysis of the decision-directed SNR estimator for speech enhancement with respect to low-SNR and transient conditions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 277–289, Feb. 2011.

[243] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, no. 1–3, pp. 133–147, 1998.

[244] L. Tóth, "Phone recognition with deep sparse rectifier neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 6985–6989.

[245] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 8609–8613.

[246] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.

[247] D. Nguyen and B. Widrow, “Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights,” in *International Joint Conference on Neural Networks (IJCNN)*, San Diego, CA, USA, Jun. 1990, 21–26 vol.3.

[248] fxprosound audio design, *Traffic Roadsounds*, Jul. 2009. [Online]. Available: https://www.freesound.org/s/75375/.

[249] “BS.1534-3: Method for the subjective assessment of intermediate quality levels of coding systems,” International Telecommunication Union, ITU-T recommendation, Oct. 2015. [Online]. Available: http://www.itu.int/rec/R-REC-BS.1534-3-201510-I/en.

[250] M. H. Radfar and R. M. Dansereau, “Single-channel speech separation using soft mask filtering,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2299–2310, Nov. 2007.

[251] M. H. Radfar, A. H. Banihashemi, R. M. Dansereau, and A. Sayadiyan, “Nonlinear minimum mean square error estimator for mixture-maximisation approximation,” *Electronics Letters*, vol. 42, no. 12, pp. 724–725, Jun. 2006.

[252] H. Yu and T. Fingscheidt, “Black box measurement of musical tones produced by noise reduction systems,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 4573–4576.

[253] P. J. Moreno, B. Raj, and R. M. Stern, “A vector Taylor series approach for environment independent speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Atlanta, GA, USA, May 1996.

[254] J. C. Segura, Á. de la Torre, C. Benítez, and A. M. Peinado, “Model-based compensation of the additive noise for continuous speech recognition. Experiments using the AURORA II database and tasks,” in *Eurospeech*, Aalborg, Denmark, Sep. 2001.

[255] L. Deng, J. Droppo, and A. Acero, “Enhancement of log Mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 133–143, Mar. 2004.

[256] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, “An overview of noise-robust automatic speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, Apr. 2014.

[257] R. F. Astudillo and T. Gerkmann, “On the relation between speech corruption models in the spectral and the cepstral domain,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 7044–7048.

[258] H. Kameoka, N. Ono, K. Kashino, and S. Sagayama, “Complex NMF: A new sparse representation for acoustic signals,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 3437–3440.

[259] B. J. King and L. Atlas, “Single-channel source separation using complex matrix factorization,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2591–2597, Nov. 2011.

[260] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247–251, 1993.

[261] S. Gonzalez and M. Brookes, "PEFAC - a pitch estimation algorithm robust to high levels of noise," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 518–530, Feb. 2014.

[262] 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, Sep. 2011.

[263] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Processing*, vol. 81, no. 11, pp. 2403–2418, 2001.

[264] G. Hu. (2005). A corpus of nonspeech sounds, [Online]. Available: http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html (visited on 01/09/2018).

[265] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[266] A. Field, *Disocvering Statistics Using SPSS*, 3rd ed. SAGE Publications Ltd., 2009.

[267] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965.

[268] J. W. Mauchly, "Significance test for sphericity of a normal n-variate distribution," *The Annals of Mathematical Statistics*, vol. 11, no. 2, pp. 204–209, Jun. 1940.

[269] S. W. Greenhouse and S. Geisser, "On methods in the analysis of profile data," *Psychometrika*, vol. 24, no. 2, pp. 95–112, Jun. 1959.

[270] S. Holm, "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

[271] B. Cauchi, I. Kodrasi, R. Rehr, S. Gerlach, A. Jukić, T. Gerkmann, S. Doclo, and S. Goetze, "Joint dereverberation and noise reduction using beamforming and a single-channel speech enhancement scheme," in *The REVERB Challenge*, Florence, Italy, May 2014.

[272] ——, "Combination of MVDR beamforming and single-channel spectral processing for enhancing noisy and reverberant speech," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 61, 2015.

[273] N. Mohammadiha and S. Doclo, "Speech dereverberation using non-negative convolutive transfer function and spectro-temporal modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 276–289, Feb. 2016.

[274] D. S. 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. 25, no. 7, pp. 1492–1501, Jul. 2017.

[275] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, "On the Lambert-W function," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

LIST OF PUBLICATIONS

The dissertation is based on the following publications.

JOURNALS

[1] R. Rehr and T. Gerkmann, “On the importance of super-Gaussian speech priors for machine-learning based speech enhancement,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 357–366, Feb. 2018.

[2] R. Rehr and T. Gerkmann, “An analysis of adaptive recursive smoothing with applications to noise PSD estimation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 2, pp. 397–408, Feb. 2017.

PEER-REVIEWED CONFERENCE PAPERS

[3] R. Rehr and T. Gerkmann, “Robust DNN-based speech enhancement with limited training data,” in *ITG Conference on Speech Communication*, Oldenburg, Germany, Oct. 2018.

[4] R. Rehr and T. Gerkmann, “MixMax approximation as a super-Gaussian log-spectral amplitude estimator for speech enhancement,” in *Interspeech*, Stockholm, Sweden, Aug. 2017.

[5] R. Rehr and T. Gerkmann, “A combination of pre-trained approaches and generic methods for an improved speech enhancement,” in *ITG Conference on Speech Communication*, Paderborn, Germany, Oct. 2016, pp. 51–55.

[6] R. Rehr and T. Gerkmann, “Bias correction methods for adaptive recursive smoothing with applications in noise PSD estimation,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 206–210.

[7] R. Rehr and T. Gerkmann, “On the bias of adaptive first-order recursive smoothing,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2015.

FURTHER PUBLICATIONS (NOT RELATED TO THIS THEISIS)

JOURNALS

[1] B. Cauchi, I. Kodrasi, R. Rehr, S. Gerlach, A. Jukić, T. Gerkmann, S. Doclo, and S. Goetze, "Combination of MVDR beamforming and single-channel spectral processing for enhancing noisy and reverberant speech," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 61, 2015.

[2] F. Xiong, B. T. Meyer, N. Moritz, R. Rehr, J. Anemüller, T. Gerkmann, S. Doclo, and S. Goetze, "Front-end technologies for robust ASR in reverberant environments – spectral enhancement-based dereverberation and auditory modulation filterbank features," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 70, 2015.

PEER-REVIEWED CONFERENCE PAPERS

[3] R. Rehr and T. Gerkmann, "Cepstral noise subtraction for robust automatic speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, QLD, Australia, Apr. 2015, pp. 375–378.

[4] B. Cauchi, I. Kodrasi, R. Rehr, S. Gerlach, A. Jukić, T. Gerkmann, S. Doclo, and S. Goetze, "Joint dereverberation and noise reduction using beamforming and a single-channel speech enhancement scheme," in *The REVERB Challenge*, Florence, Italy, May 2014.

[5] F. Xiong, N. Moritz, R. Rehr, J. Anemüller, B. Meyer, T. Gerkmann, S. Doclo, and S. Goetze, "Robust ASR in reverberant environments using temporal cepstrum smoothing for speech enhancement and an amplitude modulation filterbank for feature extraction," in *The REVERB Challenge*, Florence, Italy, May 2014.

[6] M. Krawczyk, R. Rehr, and T. Gerkmann, "Phase-sensitive real-time capable speech enhancement under voiced-unvoiced uncertainty," in *European Signal Processing Conference (EUSIPCO)*, Marrakech, Morocco, Sep. 2013.

[7] T. Gerkmann, M. Krawczyk, and R. Rehr, "Phase estimation in speech enhancement - unimportant, important, or impossible?" In *IEEE Convention of Electrical and Electronics Engineers in Israel (IEEEI)*, Eilat, Isreal, Nov. 2012, pp. 1–5.

ABSTRACTS

[8] R. Rehr, M. Krawczyk, and T. Gerkmann, "A comparison of state-of-the-art speech fundamental frequency estimators in noisy and reverberant environments," in *Deutsche Jahrestagung Für Akustik (DAGA)*, Oldenburg, Germany, Mar. 2014, pp. 487–488.

[9] R. Rehr, S. Goetze, D. Hollosi, J.-E. Appell, and J. Bitzer, “Speech / non-speech discrimination for acoustic monitoring considering privacy issues,” in *Deutsche Jahrestagung Für Akustik (DAGA)*, Düsseldorf, Germany, Mar. 2011, pp. 879–880.