Real-Time capable Noise Reduction Methods

Maximilian Luz
luzmaximilian@gmail.com

Abstract

Speech enhancement plays a fundamental role in modern voice communication and interaction. It is often used in automatic speech recognition systems or communication applications, where real-time constraints are required and it is desirable to have a technique that is robust to changes in noise and environment. In this paper, we will present multiple noise reduction techniques, with the goal of finding a method fulfilling both of these properties. To this end, we look at simple spectral subtraction and improvements thereof, such as over-subtraction and spectral flooring, combined with some basic noise estimation techniques, as well as more complex methods, based on stochastic models. The latter techniques are based on minimum mean-square error estimation, which ultimately will lead us to the optimally-modified log-spectral amplitude estimation method. We find that, when combined with the minima controlled recursive averaging noise estimator to obtain a robust and adaptive noise estimate, this technique provides a successful solution to our problem.

1 Introduction

Speech processing is omnipresent in our current everyday life: From phone calls and voice messaging, digital assistants like Apple's Siri or Google Now, to many of the videos and movies we watch, speech processing, and especially speech enhancement, is a core component empowering these technologies. An integral part of such enhancement procedures is noise reduction, or suppression, in which the goal is to recover a desired speech signal from a signal that was corrupted by additive noise. As such, we will, in the following, use these two terms, speech enhancement and noise reduction, interchangeably. Even though the quality of microphones themselves has greatly improved over the past decades, noise is still a considerable source of speech degradation, be it from man made machinery, such as cars and planes, other speakers in the background, or even nature itself, e.g., from wind and waters. To this end, many modern devices, such as mobile phones and digital assistants, include multiple microphones for techniques like dual-microphone speech enhancement (see e.g., Jeub et al. [12] and Yousefian and Loizou [26]) and beamforming (see e.g., Habets and Benesty [10]). Generally, the use of multiple microphones is preferable over single-microphone solutions, due to better quality achieved while having significantly less to no distortion in speech [2, 16].

In some cases, however, there may only be a single audio signal. Many examples for this can be found in scenarios where product vendors do not have control over the hardware, such as voice-over-IP software, where speech enhancement may be a wanted feature to set ones product apart from other competitors. Other examples can be found in settings where noise reduction is only an afterthought or a convenient feature: Even today, while most mobile

phones do use multiple microphones, headsets or microphones intended for use on a PC, and even many modern laptops and notebooks, do not.

Additionally, the choice of speech enhancement technique may further be impaired by the requirement of a real-time guarantee. First, this means that we have only access to past and current information and we cannot use future information to, for example, estimate the noise or adapt to changes in it. Secondly, this means that we only have a limited time to process the signal. Studies, e.g., by Egger, Schatz, and Scherer [5], suggest that one-way delays in communication up to 600 ms are still acceptable, whereas the official recommendation of the International Telecommunication Union [11] notes strong dissatisfaction of users at the same latency. Further raising this delay may eventually lead to an increase in double talking, i.e., speakers talking over each other, severely impacting intelligibility. Studies agree that delays up to 200 ms generally have a low to negligible impact on perceived quality [5, 11]. Note, however, that these numbers do not factor in any transport delay, reducing the time left for processing, and even slight delays may be considered annoying by some users.

In this paper, we will discuss algorithms suitable for use in single-microphone real-time scenarios. Before we begin with this, we will define the goals of this paper and the accompanying project in the next subsection. After that, we will quickly look at the noise-reduction problem and thereafter present an overview of speech enhancement techniques, providing a short introduction to the foundations of many contemporary methods. We will next highlight some common techniques used throughout the algorithms described in this paper, beginning with the short-time Fourier transform (STFT) and its inverse via the weighted overlap-add (WOLA) method. Following this, we will discuss the actual algorithms, starting with simple spectral subtraction, from which we will present a generalized view of short-time spectrum based techniques with references to the Wiener filter and the maximum likelihood estimation based method by McAulay and Malpass [18]. As these algorithms heavily rely on noise estimation, we will discuss some basic ideas for this thereafter, with an improved method in a later section. Next, we will discuss some enhanced noise reduction methods, specifically the minimum mean-square error short-time spectral amplitude estimation (MMSE) algorithm by Ephraim and Malah [7] and a modified version thereof, minimum mean-square error log-spectral amplitude estimation (log-MMSE)[6]. We will build on the log-MMSE method by incorporating speech presence uncertainty, leading to the optimally-modified log-spectral amplitude estimation (OM-LSA) algorithm by Cohen and Berdugo [4]. Thereafter, we will highlight the minima-controlled recursive averaging (MCRA) method as an improvement over the basic noise estimation methods presented previously, before we, lastly, discuss our findings and compare the behavior and performance of the algorithms.

1.1 Goals and Implementation

As stated above, the main goal of this paper and the corresponding project[1] is to explore algorithms for real-time noise reduction in settings with a single microphone. A resulting final algorithm should furthermore be capable of handling and adapting to different noise situations, e.g., changing signal-to-noise ratio or changes in frequency of signal, i.e., speech, and/or noise. These goals essentially form the requirements for an algorithm usable for real-time voice communication. In addition, we also limit ourselves to unsupervised techniques in which the processing takes place in frequency domain, substantially reducing the number algorithms to consider. This means, that we will not consider algorithms that require any kind of supervised pre-training step, such as neural network based solutions.

For the implementation itself we chose the Rust programming language [23]. Rust was originally developed by Mozilla Research with the goal of creating a safe and fast language for use in their Firefox browser. The main design goals of this language are memory safety and concurrency guarantees, that eliminate the possibility of segmentation faults and data races

[1] The source-code for the project can be found at `https://github.com/qzed/noisereduce/`.

through a strong type system, all while rivaling the speed of other compiled languages such as C or C++[17]. While those safety guarantees are a considerable argument in favor of Rust themselves, we, however, chose Rust mainly for other reasons: its modernity and flexibility. The standard Rust compiler is based on the LLVM toolchain and a significant effort has been made to bring Rust to many devices and even browsers via WebAssembly. This provides great flexibility for potential uses, e.g., in voice-over-IP (VoIP) web applications, standard desktop applications, mobile devices, or even embedded systems. Another benefit of Rust is the modern ecosystem it brings with it: In contrast to C and C++, Rust, like many other modern languages, focuses on a standardized build system with integrated package manager, Cargo. Due to this, setting-up and managing projects gets, in general, significant easier, making it possible to focus more on the application being built itself, than on managing its build system. Furthermore, Rust allows integration of pre-existing C libraries in a fairly straight-forward and simple fashion.

Additionally, we would like to highlight our use of the GNU Scientific Library (GSL), a free and open-source C library for numerical computations in mathematics and science, originally developed by physicists of the Los Alamos National Laboratory [8]. This library provides us with well-tested implementations of the more complex and tricky to implement numerical functions used in some of the techniques described in this paper, specifically Bessel functions and the exponential integral.

2 The Noise-Reduction Problem

As with all tasks, it is crucial to understand the problem before attempting to solve it. Auditory noise may come in various forms, each with different characteristics, and can be described by a set of key properties. By understanding these properties, we can build a model of the type of noise we want to reduce and via that a corresponding algorithm. In this section, we will have a brief look at the noise-reduction problem via some of the properties that can be used to describe noise.

A first characterization can be drawn in the way the noise $d(t)$ is mixed with the clean speech signal $x(t)$ to produce the noisy signal $y(t)$. Here, we can differentiate between additive, convolutive, and multiplicative noise, which are applied according to their respective name.

Additive noise assumes that the noisy signal is simply the sum of the clean signal and the noise signal, i.e.,

$$y(t) = x(t) + d(t). \tag{2.1}$$

This is by far the most common form that we encounter when dealing with audio, as it encompasses almost all occurrences of noise in our everyday lives, including car noise, wind noise, other people speaking in the background, etc. Most methods dealing with speech enhancement target this type of degradation, such as, for example, the spectral subtraction method (see Section 5.2) in which the spectral amplitude $D(\omega)$ of the noise signal is estimated and subtracted from the spectral amplitude $Y(\omega)$ of the observed noisy signal. Further examples are the short-time spectral amplitude methods discussed throughout this paper and noise-cancellation algorithms as, for example, used in the homonymous headphones.

Convolutive noise assumes that the noise signal is computed by convolving clean speech and noise signals, i.e.,

$$y(t) = x(t) * d(t). \tag{2.2}$$

Note that this is equivalent to a multiplication in the Fourier domain. Convolutive noise is often encountered as reverberation or echo. This type of noise can, for example, be targeted by approximating the convolutive transfer function of a room, e.g., via microphone arrays as done in the technique by Talmon, Cohen, and Gannot [22], or blind multi-channel identification, such as the method by Li, Gannot, and Horaud [15]. The aim of the blind multi-channel identification approach is to detect the multiple audio sources in a room by use

of a microphone array and reconstruct their original individual signals from the observation signals. In case of reverberation and echos, each reflection is interpreted as a separate source to be reconstructed, which then, combined with the identification of the original source of the clean speech signal, can be used to estimate the undegraded speech.

Multiplicative noise, on the other hand, is by far the least common type of noise when dealing with audio. It is applied by multiplying clean and noise signals, i.e.,

$$y(t) = x(t) \cdot d(t), \tag{2.3}$$

and can occur when processing analog audio signals or via interference during such processing. As this type of noise is, in general, fairly uncommon and rather a matter of the technical equipment used, it is rarely discussed and we will not consider it further.

Another major distinguishing factor between different noise types is the correlation of the noise signal with the clean speech signal. Most common forms of noise are uncorrelated, such as car noise, wind noise, etc., however, some types, such as echos and reverberation, directly correlate with the clean speech signal. When dealing with noise caused by other speakers in the background, there may also be psychological effects which can cause a more indirect correlation, e.g., by the others lowering or raising their volume during parts in which the main speaker speaks. In general, correlated signals pose a more difficult problem for speech enhancement techniques.

Lastly, there is the variation, both in frequency and volume, of the noise signal over time. This variation is, for example, inherent in background speakers, but can also be observed in noise caused by nature or cars passing by, whereas other machinery often generates fairly stationary noise. Apart from the changes in frequency and volume themselves, the duration in which these changes occur is of importance: Statistical-model-based single microphone techniques, such as the ones discussed in this paper, assume that the noise signal varies considerably slower than the clean speech signal. This assumption is required to successfully discriminate between the two signals and is thus central to those methods.

With that in mind, we can finally look at some of the more known forms of auditory noise. For this, let us first look at some of the stochastic noise prototypes, specifically white, red, and gray noise. These types are stationary, but can be modified in volume, pitch shifted, or blended together to incorporate variation over time. The key difference between these so-called colors of noise is their distribution in frequency. *White noise* is characterized by equal intensity of different frequencies, i.e., each frequency is represented in equal parts. *Grey noise* follows a similar approach, however, here equality is defined via a psychoacoustic loudness curve, i.e., the goal of gray noise is to represent equal intensity in frequencies as observed by humans. *Red noise*, often also referred to as *Brown* or *Brownian noise*, on the other hand has a logarithmic (linear in decibel) decrease of power density with respect to increasing frequencies, i.e., lower frequencies have a higher volume and higher frequencies a lower volume. Further colors, such as pink, blue, violet, and green describe additional frequency distributions. While these models are first and foremost of theoretical nature, many noise types observed in the real world can be approximated by mixtures of them, applied additively to the clean speech signal. On the subject of real world noise, common convolutive examples are, as previously mentioned, *echo* and *reverberation*. These can be described by multiple different mathematical models as well, e.g., based on convolutive transfer functions, which describe the properties of the room causing these degradations. A further important type of additive noise when considering speech enhancement techniques is noise caused by other speakers in the background, so-called *babble noise*. Babble noise is more difficult to deal with than other additive noise types, especially in single-microphone settings, and even in multi-microphone settings often considered as one of the most challenging types of speech degradations. This is due to the fact that it expresses the same or very similar characteristics as the clean signal, as both are spoken by humans. It is also more difficult to model, due to its relatively high variations in both frequency and volume. For a more

in-depth analysis and a corresponding model representing this type of noise, see for example the paper by Krishnamurthy and Hansen [14].

3 An Overview of Speech Enhancement Algorithms

Real-time speech enhancement is a long-standing topic in modern communication. As such, many techniques have been developed over the past decades. With the insights into noise given in the previous section, we can now present an overview of speech enhancement methods, with a more detailed look at statistical-model-based noise reduction algorithms later in Section 5.

In addition to the type of noise they target and the model they assume for it, which we have discussed at length in the previous Section, we can distinguish algorithms by the number of microphones they use for processing. While, we focus on single-microphone solutions only in this paper, the use of multiple microphones or microphone arrays can improve performance and significantly reduce distortion of speech [2, 16], and even be essential to techniques like blind source separation (see e.g., Li, Gannot, and Horaud [15]), beamforming (see e.g., Habets and Benesty [10]), or spatio-temporal filtering methods. In terms of actual differences in the algorithms, this means that single-microphone methods need to rely on an inherently stochastic model to differentiate between speech and noise, whereas multi-microphone solutions can also make use of spatial information. In blind multichannel separation, this spatial information is used to divide the observed signal into one channel per source contributing to said observed signal, via which the unwanted noise-sources can be discarded. In beamforming, on the other hand, the goal is to spatially focus on the direction from which the clean source signal originates, blending out noises from different directions. There are also techniques, such as the dual-microphone methods often observed on smartphones, that rely directly on a fixed orientation of the microphones. In this case, one microphone is oriented towards the source of the clean speech signal (e.g., the mouth of the speaker) and another microphone is oriented toward the area from which we assume the noise originates (e.g., it may be located on the back of the phone), to make use of spatio-temporal information and serve as a reference for the noise signal.

A further differentiation can be made in the way algorithms process the signals: On the top-level, we can discern between processing in time and frequency domain. Processing in frequency domain can make the design of speech enhancement techniques significantly easier, but involves more overhead as we need to transfer the signal from the time-domain to the frequency domain and back. This is usually done by use of the Fourier transform or one of its variations. As it is impractical to apply the Fourier transform on the complete signal, which might not be available, we usually rely on a specialized application scheme of it: the short-time Fourier transform (STFT). The STFT, separately discussed in Section 4.1, essentially splits the signal into multiple overlapping slices and then computes the Fourier transform on each slice. This provides a set of time-localized spectra, i.e., a spectrogram, and has the additional advantage of this localization. Whereas slow changes of the signal over time are directly encoded as frequencies when applying the Fourier transform on the complete signal, with the STFT they are encoded as changes over the spectrum slices. Thus, with the right length of the slices (usually 20 ms to 40 ms [7]), we can focus the spectra on the frequencies relevant to humans and human speech. Banded processing techniques can be seen as a mixture of both categories. In these techniques the signal is split into different frequency bands, e.g., by use of band-pass filters, with each signal being processed individually. Such banded techniques can, for instance, use the psychoacoustic Bark scale [28] as a reference to obtain frequency bands that better represent human perception.

Finally, we should also discuss some of the different strategies that can be chosen to address the speech enhancement problem, for which we will follow the categorization given by Loizou [16]. An, at least on the first look, very simple option are *spectral subtractive algorithms*.

These techniques are directly based on the assumption that the noise is mixed additively with the clean speech signal. Specifically, these methods try to estimate the noise signal, e.g., via secondary microphones or statistical techniques, and then subtract this estimation from the corrupted speech signal, either in time or frequency domain. The somewhat hidden complexity of these methods stems from the noise estimation problem, which is common to a wide variety of speech enhancement algorithms. This problem can be tackled in many different ways, for instance we could update our noise estimate in times when the speech signal is absent. Throughout this paper, we will discuss various different ways to estimate the noise signal using a single microphone, as this issue is also central to the techniques presented herein. Spectral subtractive techniques will be discussed more extensively in Section 5.2, as they provide a convenient introduction into the statistical-model-based-techniques presented thereafter.

Statistical-model-based algorithms are a second category of techniques. As the name implies, these methods rely on an underlying statistical model, through which they define a cost- or objective-function, which is then subsequently optimized to obtain the final algorithm. Examples for this include many neural-network based techniques, the Wiener filter (see e.g., Chen et al. [2] and Loizou [16]) — both in time and frequency domain — where the minimum mean-square error (MMSE) between clean and estimated signal is minimized, the method of McAulay and Malpass [18], in which the the likelihood of the Fourier coefficients of the clean signal is maximized, as well as most algorithms discussed in this paper, i.e., the MMSE technique by Ephraim and Malah [7], its improvement, log-MMSE [6], and derivations. Another noteworthy group of techniques that we should mention here are based on actively reconstructing the clean signal by re-synthesizing speech components, such as for example the method by Xiao and Nickel [25]. In these methods, features are extracted from the pre-processed noisy signal which are then used to reconstruct potentially missing or occluded speech components while reducing noise to create an enhanced version of this signal. Via this, these methods try to avoid or reverse the destruction of speech components that is common in conventional noise-reduction methods. Usually, the generated signal is not a complete artificial reconstruction, but rather blended with an enhanced or pre-processed input signal. While these methods can be seen as a separate group, their foundations are largely rooted within statistical models.

A somewhat less statistical approach are *subspace algorithms*. These methods are derived from linear algebra, with the fundamental idea that the space of all possible signals can be divided into a subspace that is occupied primarily by the clean speech signal and a subspace that is occupied primarily by the noise signal. We can thus enhance the signal by identifying these two subspaces and zeroing the noise subspace, so that only the subspace containing the clean speech signal remains. For this decomposition, the usual methods, such as singular-value decomposition, eigendecomposition, and other matrix factorization techniques can be used and, for example, applied to the covariance matrix of the signal [16].

A final category worth noting is formed by *binary mask algorithms*. In contrast to most of the methods discussed above, these algorithms do not attempt to estimate a real-valued correction, but rather mask out noisy parts of the signal completely, e.g., by removing channels or zeroing frequency bins in the short-time spectral amplitude. Due to this binary nature, they can be seen as transforming the speech enhancement problem into a classification problem, that can, for example, be solved via neural networks. This can be seen as a similarity to subspace algorithms, however, instead of identifying subspaces, we here identify the parts corresponding to the speech signal directly, and the bins or components to remove may change with every frame. In this sense, it is more closely related to the voice activity detection (VAD) problem. In fact, it can be seen as an extension of this problem to speech enhancement. As these algorithms do not attempt to improve the segments containing speech themselves, they do not degrade speech as much as other algorithms might do. This can, at least in theory, be better suited for speech enhancement in high signal-to-noise ratio (SNR)

conditions [16] and also be of particular interest to automatic speech recognition (ASR) systems, of which the performance may be impacted by said degradation. In the following, we will only concern ourselves with algorithms based on statistical models, but may also look at other techniques for comparison, specifically spectral subtractive methods due to their simplicity.

4 Common Processing Methods for Real-Time Noise Reduction

Before we further work towards specific speech enhancement methods, it is advisable that we discuss some of the methodology and tricks fundamental to real-time noise reduction and real-time signal processing in general. While processors have seen a significant increase in computing power and decrease in size and power-consumption over the last couple of decades, real-time programming still needs special considerations, especially when we concern ourselves with embedded or low-power devices, as, for example, in an ASR setting. But not only technical aspects have to be considered: Arguably the biggest impact on performance stems from the algorithms we use. Simple operations, such as computing a minima or an average over n time-steps require, using a naive implementation, $\mathcal{O}(n)$ in both time and space. With some fairly trivial changes, we can bring this down to $O(1)$. In this section, we will discuss such considerations and tricks that help achieve real-time performance, beginning with the short-time Fourier transform and its inverse via the (weighted) overlap-add method. Following this, we will look at the exponentially weighted moving average as a technique for tracking lower-frequent changes over time as well as windowed extrema computation to obtain extremas in a limited time-frame leading up to the current sample. Finally, we will examine some of the more technical aspects when programming for real-time audio processing.

4.1 Short-Time Fourier Transform

The short-time Fourier transform (STFT) is essential to all real-time capable speech enhancement algorithms processing data in the spectral domain, as, in contrast to the normal Fourier transform, the STFT does not require the full input signal to be known. Instead, the STFT processes overlapping slices of the signal by computing a Fourier transform independently on each slice, allowing it to handle potentially infinite signals in real-time (i.e., it is able to satisfying our real-time constraints as elaborated in Section 1). Due to this sliced processing, the STFT computes a series of time-localized spectra. By localizing the spectra we now encode low-frequent changes, such as changes in acoustic cues of the speech (e.g., vowels, nasals, stops, etc.), as changes between the spectra, rather than in the spectra themselves. Inherent with this adaption of the Fourier transform, however, is the problem of how we should choose the size of such slices. Note that, in the discrete setting, this has a direct impact on the frequency resolution of the resulting spectra. If we choose a larger slice, the spectra spans a larger interval in time leading to a worse localization, but we also get a higher resolution in frequency. This is then often referred to as narrow-band spectrum due to the narrow frequency bands. If, in contrast, we choose a smaller slice, we get a better localization in time due to the smaller interval covered, but we also get a worse resolution in frequency. This is often referred to as wide-band spectrum. The aforementioned idea of encoding these low-frequent changes of the non-stationary speech signal as changes between adjacent spectra directly presents us with an answer to this problem: By choosing the size of a slice such that this encoding holds, as well as via practical testing, we can constrain the length to 20 ms to 40 ms per slice [16].

Mathematically, the STFT is defined as

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)\, w(t-\tau)\, e^{-j\omega t}\, dt, \tag{4.1}$$

where x represents the input signal in time-domain, X the complex spectrum of x depending on time τ and angular frequency $\omega = 2\pi f$, $e^{-j\omega t}$ the complex root as used in the normal Fourier transform, and w a window function centered around τ. This windowing process essentially computes the time-localization as mentioned above, as we assume that the window function w is non-zero only on a finite interval centered around zero. The window function does not only play an important role for the slicing process, but, similar to the standard Fourier transform, is also used to avoid introducing unwanted artifacts in the frequency domain. Discretization in time is done mostly analogously to the standard Fourier transform, yielding

$$X^{[m]}(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, w[n - mR]\, e^{-j\omega n}. \tag{4.2}$$

In addition to the discretization, we also introduce a hop-size R which allows us to move R samples forward with each slice. We will see shortly that, depending on the window function, this, in fact, is required to guarantee inversibility of the discrete STFT. The relationship between hop-size R to overlap M and slice length N can be directly given via $R = N - M$. Further, note the connection to the discrete-time Fourier transform (DTFT)

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n}. \tag{4.3}$$

By introduction of the windowing term, however, we introduce an additional time dimension in the output of the STFT. Also note, that, in this discretized form, we can split the sum and compute individual DTFTs for each slice, which is essential for the final algorithm. The algorithmic computation of this process is illustrated in Figure 1.

A fundamental property with regards to window functions for the STFT is the fulfillment of the (general) constant overlap-add (COLA) constraint

$$\exists c_{\text{ola}} \in \mathbb{R} : \forall n \in \mathbb{Z} : \sum_{m=-\infty}^{\infty} w[n - mR] = c_{\text{ola}}. \tag{4.4}$$

This constraint states that, when overlapping a suitable window function w with itself using a hop-size of R, summing up the overlapping parts should yield the same constant c_{ola} for each fixed time-step n. Essentially, this constraint ensures that no new frequencies, i.e., frequencies that are not present in the original input signal, are introduced when adding up the individual spectra produced by the STFT. In fact, we can show that, if the COLA condition is fulfilled, computing this sum yields the DTFT of the whole signal x, scaled by the constant c_{ola} [21]:

$$\begin{aligned}
\sum_{m=-\infty}^{\infty} X^{[m]}(\omega) &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n]\, w[n - mR]\, e^{-j\omega n} && (4.5)\\
&= \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \cdot \underbrace{\sum_{m=-\infty}^{\infty} w[n - mR]}_{c_{\text{ola}} \text{ iff } w \text{ fulfills COLA}(R)} \\
&= c_{\text{ola}} \cdot \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} && (4.6)\\
&= c_{\text{ola}} \cdot [\text{DTFT}(x)]\,(\omega) && (4.7)\\
&= c_{\text{ola}} \cdot X(\omega) && (4.8)
\end{aligned}$$

As the DTFT is reversible, this should already give us an indication that, if the COLA constraint is fulfilled, the STFT can also be reversed. In the next subsection, we will look at the inverse method.

As window function w, we can use the common window functions, such as for example the Hamming, Hann. Blackman, or Blackman-Harris functions. Additionally, one might want to consider padding the input signal at start and end, e.g., via mirroring, so that the full signal can be reconstructed. If this is not done, the windowing process can introduce fade-in and fade-out effects, as illustrated in Figure 1 and Figure 2.

4.2 Weighted Overlap-Add Method

The weighted overlap-add (WOLA) method provides an inverse to the STFT discussed in the previous subsection. Mathematically, the inverse short-time Fourier transform (ISTFT), discretized in time only, can be written as

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{m=-\infty}^{\infty} X^{[m]}(\omega)\, e^{j\omega n}\, d\omega, \tag{4.9}$$

$$= \sum_{m=-\infty}^{\infty} \frac{1}{2\pi} \int_{-\pi}^{\pi} X^{[m]}(\omega)\, e^{j\omega n}\, d\omega, \tag{4.10}$$

$$= \sum_{m=-\infty}^{\infty} x^{[m]}[n]. \tag{4.11}$$

This set of equations is simply derived by applying the inverse DTFT (IDTFT) to Equation (4.8), under the assumption that $c_{\text{ola}} = 1$. Note that this assumption does not cause any loss of generality, as we can scale the window function w accordingly. Equation (4.11) directly represents the key aspect of the overlap-add (OLA) method, which forms the basis of the WOLA method. Given this equation, the idea of the OLA method is fairly straightforward: First, we reconstruct the individual time-domain signals $x^{[m]}$ from their corresponding complex spectrum $X^{[m]}$ using an IDTFT, then align (i.e., overlap) them by their respective original position via the hop-size R, i.e.,

$$x^{[m]} = \text{shift}_{mR}\left(\text{IDTFT}\left(X^{[m]}\right)\right),$$

and finally add the individual segments together (cf., Equation (4.11)).

The WOLA method can now be derived from this by applying an additional so-called *synthesis window* h (also referred to as *output window* or *postwindow* [21]) to the individual time-signal frames before adding them up, i.e.,

$$x[n] = \sum_{m=-\infty}^{\infty} x^{[m]}[n]\, h[n - mR]. \tag{4.12}$$

While, as we have seen in Equations (4.9) to (4.11), applying a secondary window function h here is not required to reconstruct the signal, it has a significant benefit. Applying a window function here reduces blocking effects, i.e., suppresses audible discontinuities, by fading out any spectral errors on the frame boundaries. These errors may occur due to the processing in frequency domain, especially when dealing with nonlinear techniques [21].

Introducing this new window, however, comes with new constraints for the reconstructability of the signal: Expanding equation Equation (4.12) by Equation (4.2) at some fixed time n for an arbitrary signal x and hop-size R, i.e., a round-trip from time- to frequency-domain and back, yields

$$x[n] = \sum_{m=-\infty}^{\infty} x[n]\, w[n - mR]\, h[n - mR]$$

$$= x[n] \sum_{m=-\infty}^{\infty} w[n - mR]\, h[n - mR],$$

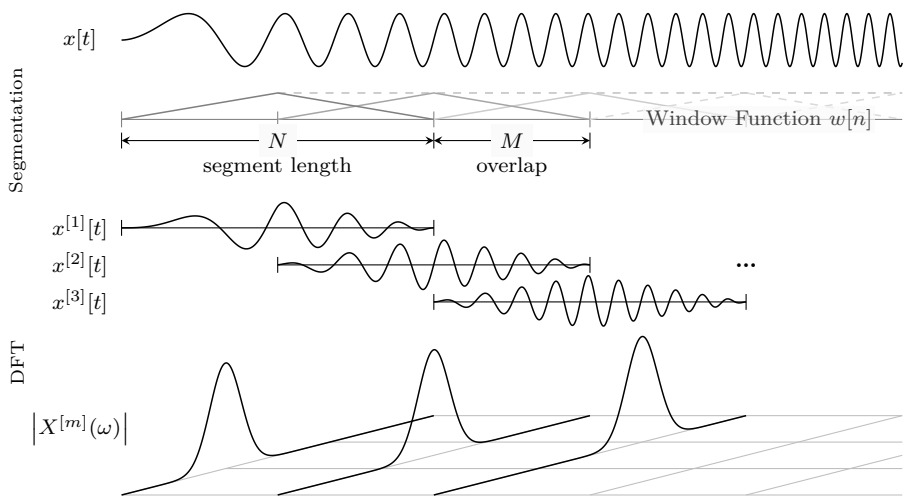

Figure 1: Short-Time Fourier Transform. The discrete input signal x in time-domain is first split into slices of N samples with an overlap of M samples. Each slice is then multiplied with a window function w to avoid introducing unwanted artifacts in the frequency domain, resulting in the individual $x^{[k]}$. Finally, a discrete Fourier transform is computed for each $x^{[k]}$, e.g., via FFT, to obtain the resulting series of complex spectra X, here illustrated by their magnitude.

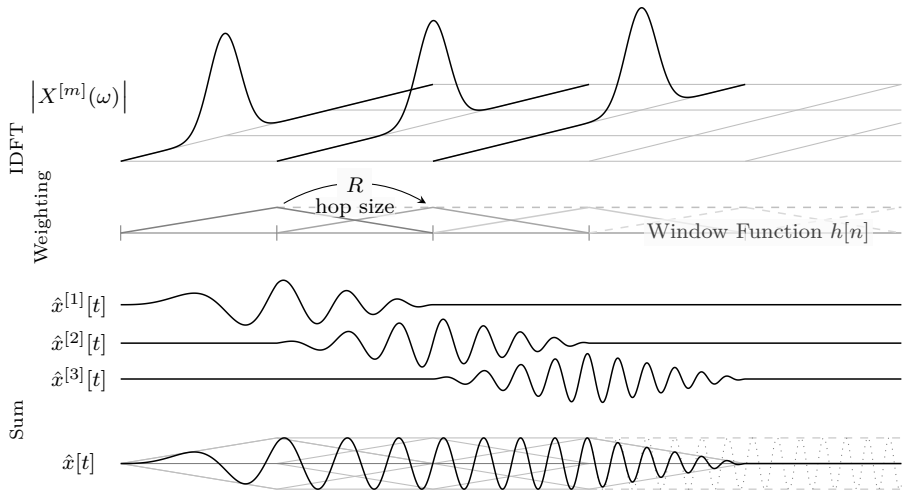

Figure 2: Weighted Overlap-Add Method. First, the series of complex input spectra X, here represented by their magnitude, is transformed into the time-domain by use of the inverse discrete Fourier transform, e.g., via IFFT. In this process, each spectrum in the series is transformed independently, resulting in one time-domain signal per spectrum. In the *weighted* overlap-add method, a synthesis window h is applied to each signal afterwards. In case of the non-weighted overlap-add method, this step is skipped. Finally, the individual signals are shifted via the hop-size R and added up to compute the resulting signal $\hat{x}$. Note that we need pad the original signal before performing the STFT if we want to fully reconstruct this signal, as, due to windowing, fade-in and fade-out effects may occur on start and end.

constraining both window functions, w and h. From this, we can generalize the WOLA method similarly to the COLA constraint in Equation (4.4) by introducing a normalization factor c_{wola}, leading to the constraint

$$\exists c_{\text{wola}} \in \mathbb{R} : \forall n \in \mathbb{Z} : \sum_{m=-\infty}^{\infty} w[n - mR]\, h[n - mR] = c_{\text{wola}}. \tag{4.13}$$

This observation leads to a common choice of window functions $w = h = \sqrt{f}$ for some arbitrary window function f that fulfills the COLA constraint (i.e., Equation (4.4)) with overlap R. Note, that setting $h = 1$ results in the OLA method and the COLA constraint required for w.

4.3 Exponentially Weighted Moving Average

The exponentially weighted moving average (EWMA or EMA), also referred to as recursive average, is an averaging method commonly used in time-series- and signal-processing. In these settings, there often arises a need for an average over some time-period leading up to the current element, for example to filter out high-frequent noise or to observe more low-frequent trends. While we could store the n latest data-points and simply average over them, doing so may not be feasible for large n, as this comes with the costs of $\mathcal{O}(n)$ in both time and space. The solution for this problem lies with the recursive formulation of the normal average. We can derive this by

$$\begin{aligned}
\tilde{x}_t &\coloneqq \frac{x_1 + x_2 + \ldots + x_t}{t} \\
&= \frac{x_1 + x_2 + \ldots + x_{t-1}}{t} + \frac{x_t}{t} \\
&= \frac{t-1}{t} \cdot \frac{x_1 + x_2 + \ldots + x_{t-1}}{t-1} + \frac{1}{t} x_t \\
&= \frac{t-1}{t} \cdot \tilde{x}_{t-1} + \frac{1}{t} x_t,
\end{aligned} \tag{4.14}$$

with a base-case of $\tilde{x}_1 = x_1$ for the recursion. If we now replace the time-varying multiplier $\frac{t-1}{t}$ in Equation (4.14) with a constant α, we obtain the EWMA, defined as

$$\bar{x}_t \coloneqq \begin{cases} x_1, & t = 1 \\ \alpha \bar{x}_{t-1} + (1 - \alpha) x_t, & t > 1. \end{cases} \tag{4.15}$$

This formulation already gives us a hint at the exponential nature of this average, which becomes explicit by recursively expanding it to

$$\bar{x}_t = (1 - \alpha) \left(x_t + \alpha x_{t-1} + \alpha^2 x_{t-2} + \ldots + \alpha^k x_{t-k} \right) + \alpha^{k+1} \bar{x}_{t-k-1}.$$

The EWMA thus averages elements in a time-series, strongly preferring the current element with an exponential decay for past elements, where α as a parameter defines a trade-off between how much of the history is kept and how big the impact of the current element is going to be.

4.4 Windowed Extrema Computation

In addition to averaging, discussed in the previous subsection, we may also need to compute extremas over the last n element of a time-series. A naive implementation of this, however, has the same issue: We will, again, need both $\mathcal{O}(n)$ time and space. While we can not solve this exact problem in $\mathcal{O}(1)$, we can approximate it. Specifically, we can start by resetting the stored extrema value every n steps by setting it to the current element. This by itself is not very useful, as we would usually need a certain minimum number of elements to

compute the extrema over. For example, if we want to track the noise floor of a signal by tracking the minimum of the spectral magnitude frames computed from this signal, resetting the minimum to the current frame would have drastic effects if the current frame contains speech, as we would now assume that the this frame, and thus the speech, represents the noise floor. However, we can use this to reset a second extrema value, by, every n elements, setting the second extrema value equal to the first extrema value and then resetting the first extrema value to the current element, as done in Algorithm 1 for minima computation. Note that adapting this algorithm for maxima computation is straightforward by modifying the initialization step to use negative infinity and replace the minimum operations with maximum operations. This gives us an algorithm that computes the extrema over at least the last n and at most the last $2n$ elements. Note that, if this range is too large, we could extend this algorithm to compute the minima of at least the last $(k-1) \cdot n$ and at most the last kn elements, by storing k extrema values, leading to costs of $\mathcal{O}(k)$ in time and space.

Algorithm 1: Constant-Time Windowed Minima Approximation

Data: Sequence $x = (x_1, x_2, \ldots)$, window size n
Result: $\min \approx \min\{x_t, x_{t-1}, x_{t-2}, \ldots x_{t-n+1}\}$

```
1 tmp ← ∞                          /* stores minima of up to last n elements */
2 min ← ∞                          /* stores minima of n to 2n last elements */
3 foreach x_t ∈ x do                          /* for each element in x, ordered */
4     tmp ← min{tmp, x_t}                          /* update minima normally */
5     min ← min{min, x_t}
6     if t ≡ 0 mod n then                                  /* every n elements */
7         min ← tmp                        /* set to minimum of last n elements */
8         tmp ← x_t                                        /* reset completely */
9     yield min          /* minima of at least n and at most 2n last elements */
```

4.5 General Technical Considerations

Even though, in this paper, we focus on the mathematical and algorithmical aspects of speech enhancement, we should, due to its relevance to the topic, also discuss some considerations that have to be made when programming for real-time applications. Real-time audio processing, such as in direct voice communication, poses hard constraints on processing time, that, when violated, have a direct impact on audio quality and can usually be heard as crackling or popping noises. To avoid such degradations, it is imperative that we meet these constraints at all times. In other words, we have, for each audio sample to be processed, a fixed time budget that we need to adhere to. This, in turn, means that we should do our utmost to avoid any operations with a non-deterministic execution time inside the processing thread or callback function, first and foremost dynamic memory allocations. However, also other functions such as semaphores and locking should be avoided, and in general everything that involves switching into the operating-system (OS) kernel context (at least if running on a non-real-time OS). Common concepts in audio processing to deal with these restrictions are fixed-size lock-free and wait-free queues as well as ring-buffers, in combination with pre-allocating any memory required for processing, if necessary in the form of memory pools with designated blocks having pre-defined sizes, depending on their usage. Further, it is common practice to process blocks of samples at a time. While this increases the latency, it is generally faster than processing samples individually: First of all, this reduces overhead, e.g., associated with obtaining or relaying samples. Secondly, this makes better use of caching, both for code and data, and third, we can use better optimizations, such as vector operations (e.g., SSE, AVX, etc.) and loop-unrolling. Together, this generally leads to sub-linear scaling in performance for block-sizes up to a certain length. Additionally, outliers in processing

time, to a limited amount, may not have as huge of an impact, as they are now averaged over the whole block size. The choice of block-size directly implies a trade-off between latency and performance, which has to be tuned individually for each application, hardware, and usage scenario.

5 Noise Reduction based on the Short-Time Spectral Amplitude

A particular class of speech enhancement methods suitable for real-time performance are based on processing the short-time spectral amplitude. The short-time spectrum is, as previously discussed in Section 4.1, obtained by use of the STFT, with the individual spectrum representing a localized time-window. Back-transform from frequency- into time-domain is generally performed by the WOLA method, discussed in Section 4.2. In the processing step, only the magnitude (i.e., amplitude) of the spectra is modified to enhance the signal, while the phase remains untouched. These methods, in general, also include multi-microphone techniques, but we will, due to the scope of this paper, content ourselves mostly with single-microphone statistical-model-based algorithms (as outlined in Section 4) throughout this section. First, however, we will highlight some basic assumptions on the signal, that we will thereafter use to introduce a fairly simple idea: spectral subtraction. Next, we will discuss the statistical estimation of the spectral components of the clean speech signal, with Wiener filtering in frequency domain as example. We will further discuss the connection between the Wiener filter and spectral subtraction and finally introduce the *a priori* and *a posteriori* signal-to-noise ratios, leading to a general formulation of all algorithms discussed in this paper.

5.1 Assumptions on the Signal

Following the fundamental assumption of all speech enhancement techniques discussed in this paper, we, first of all, assume that the corrupted speech signal $y(t)$ is based on the clean speech signal $x(t)$ and an additive noise signal $d(t)$, i.e.,

$$y(t) = x(t) + d(t). \tag{5.1}$$

Note that due to the addition theorem of the Fourier transform, this directly implies additivity in Fourier coefficients:

$$Y(\omega) = X(\omega) + D(\omega). \tag{5.2}$$

Many techniques further assume that the noise signal $d(t)$ has zero mean and is generated by a stochastic process that is uncorrelated with, i.e., independent from, the clean speech signal $x(t)$. With this second assumption applied to the expected power spectrum, we obtain

$$\begin{aligned}
\mathbb{E}\left\{|Y(\omega)|^2\right\} &= \mathbb{E}\left\{Y(\omega)\cdot Y^*(\omega)\right\} \\
&= \mathbb{E}\left\{(X(\omega)+D(\omega))\cdot(X^*(\omega)+D^*(\omega))\right\} \\
&= \mathbb{E}\left\{|X(\omega)|^2\right\} + \mathbb{E}\left\{|D(\omega)|^2\right\} + \mathbb{E}\Big\{X(\omega)\cdot D^*(\omega)\Big\} + \mathbb{E}\Big\{X^*(\omega)\cdot D(\omega)\Big\}
\end{aligned}$$

where Y^* is the complex conjugate of Y. Independence between clean speech X and noise D lets us rewrite the cross-terms as

$$\begin{aligned}
&\mathbb{E}\left\{X(\omega)\cdot D^*(\omega)\right\} = \mathbb{E}\left\{X(\omega)\right\}\cdot\mathbb{E}\left\{D^*(\omega)\right\} \quad \text{and} \\
&\mathbb{E}\left\{X^*(\omega)\cdot D(\omega)\right\} = \mathbb{E}\left\{X^*(\omega)\right\}\cdot\mathbb{E}\left\{D(\omega)\right\}.
\end{aligned}$$

Assuming zero mean for the noise signal $d(t)$ leads to $\mathbb{E}\left\{D(\omega)\right\} = \mathbb{E}\left\{D^*(\omega)\right\} = 0$ and thus the final result

$$\mathbb{E}\left\{|Y(\omega)|^2\right\} = \mathbb{E}\left\{|X(\omega)|^2\right\} + \mathbb{E}\left\{|D(\omega)|^2\right\}, \tag{5.3}$$

which serves as a useful basis for approximations, for example as in the power spectrum subtraction algorithm discussed below.

As mentioned above, we do, however, not process the complete spectrum $Y(\omega)$, but rather we handle each spectra in the series computed by the STFT individually, i.e., spectra of short overlapping time-slices, as discussed more extensively in Section 4.1. This then leads to the actual assumptions

$$Y^{[t]}(\omega) = X^{[t]}(\omega) + D^{[t]}(\omega) \tag{5.4}$$

and

$$\mathbb{E}\left\{\left|Y^{[t]}(\omega)\right|^2\right\} = \mathbb{E}\left\{\left|X^{[t]}(\omega)\right|^2\right\} + \mathbb{E}\left\{\left|D^{[t]}(\omega)\right|^2\right\}, \tag{5.5}$$

respectively. Again note the connection between discrete-time STFT and DTFT, described by Equation (4.8) (here assuming c_{ola} is one, e.g., by scaling the window function appropriately).

5.2 Spectral Subtraction

Spectral subtraction is directly based on Equations (5.2) and (5.4) and the idea that we can subtract the noise from the noisy signal in the spectral domain to obtain the clean signal, hence the name. Again, all operations are performed on the short-time spectrum, however, due to readability, we will omit indexing them in this subsection. To be able to subtract the complex spectral coefficients, we need to estimate them, due to them being unknowns. As estimating the complex noise coefficients is rather tricky, we, in typical computer science fashion, divide this problem into two sub-problems. By splitting the complex spectral coefficients $D(\omega)$ of the noise into magnitude $|D(\omega)|$ and phase $\phi_d(\omega)$ using the polar form

$$D(\omega) = |D(\omega)|\, e^{j\phi_d(\omega)}, \tag{5.6}$$

we can estimate them separately. Henceforth, we denote estimates with a hat, e.g., $\hat{D}(\omega)$. Let us assume for now that we have an oracle which can determine if speech is present given some spectral frame of the signal, i.e., some voice activity detection (VAD) algorithm. With this, we can estimate the noise magnitude $|\hat{D}(\omega)|$ by simply averaging over the last couple of frames that did not contain speech. We will look at this in more detail in the next subsection. For the phase, we follow a much simpler approach: Instead of trying to estimate it, we simply assume that the phase $\phi_x(\omega)$ of the clean signal can be approximated directly by the phase $\phi_y(\omega)$ of the noisy signal. While the phase does provide significant information impacting the quality of the speech signal, the impact on speech intelligibility is, for small frame lengths as commonly used in speech enhancements, little [19]. Estimating the phase of the clean speech is difficult and comes with a significant increase in complexity of the resulting technique [16]. Furthermore, it is possible to (geometrically) bound the difference between the noisy and the clean signal phase by the (*a priori*) signal-to-noise ratio (SNR), see, for example, Loizou [16]. This means that, as long as the SNR is large enough, the phase difference is inaudible, which leads to good results in practice. Combining these estimations finally leads to the mathematical formulation of spectral subtraction as

$$\hat{X}(\omega) = \left|\hat{X}(\omega)\right| e^{j\phi_y(\omega)} \tag{5.7}$$

where

$$\left|\hat{X}(\omega)\right| = \left|Y(\omega)\right| - \left|\hat{D}(\omega)\right|, \tag{5.8}$$

meaning we estimate the magnitude spectrum $|\hat{X}(\omega)|$ of the clean speech signal by subtracting the estimated magnitude spectrum $|\hat{D}(\omega)|$ from the noisy input signal magnitude spectrum $|Y(\omega)|$ (Equation (5.8)) and then use the noisy phase spectrum to reconstruct the clean signal with it (Equation (5.7)).

While this may, at first, seem sound in theory, in practice we are susceptible to a significant issue. Inaccuracies in the prediction of the noise magnitude spectrum can lead to negative

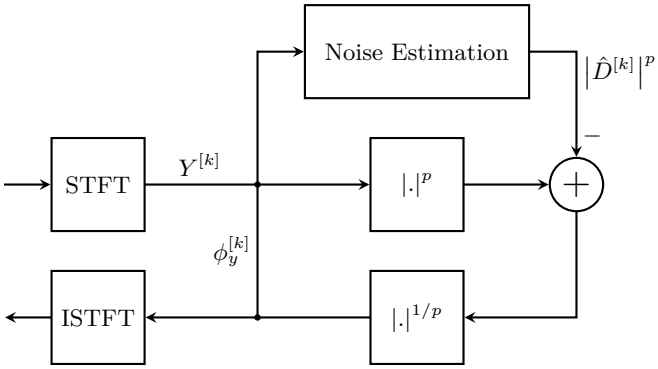

Figure 3: Generalized Spectral Subtraction Algorithm. The short-time spectrum provided by the STFT is divided into magnitude- and phase-spectrum. A noise estimate is computed based on current and past spectra of the noisy signal, which is then subtracted from the magnitude spectrum of the input signal ($p = 1$). The clean signal spectrum estimate is then obtained by combining the noisy phase spectrum with the result of this subtraction, which has been rectified to ensure its non-negativity. Optionally, the subtraction is performed on the power spectrum ($p = 2$).

values in the estimation of the clean magnitude spectrum. We thus have to ensure that, either after subtraction or during noise estimation, the final spectrum $|\hat{X}(\omega)|$ is always non-negative. The easiest solution for this is to half-wave-rectify the spectral magnitude, i.e., set the negative components of it to zero, leading to

$$\left|\hat{X}(\omega)\right| = \max\left\{\left|Y(\omega)\right| - \left|\hat{D}(\omega)\right|, 0\right\}. \tag{5.9}$$

There are many other techniques addressing this problem (see e.g., Loizou [16]), for example it may make sense to retain a spectral noise floor which may improve the perceived quality of the resulting signal, as shown by Berouti, Schwartz, and Makhoul [1].

By taking equation Equation (5.3) into account, we can extend the idea of spectral subtraction to the power-spectrum subtraction method. In some cases, it might be better to work with the power spectra rather than the magnitude spectra [16]. Both forms can then be mathematically represented by

$$\left|\hat{X}(\omega)\right|^p = \left|Y(\omega)\right|^p - \left|\hat{D}(\omega)\right|^p, \tag{5.10}$$

with $p \in \{1, 2\}$. The complete algorithm is illustrated in Figure 3. Note that, based on Equation (5.3), the power-spectrum subtraction, in addition to additivity of noise, also assumes statistical independence of speech and noise signals as well as zero mean for the noise signal.

Another very simple modification of this algorithm is over-subtraction, i.e., subtracting a small multiple of the noise estimate. As the estimate of the spectral noise magnitude is usually some sort of average, this can significantly increase the amount of noise eliminated. Combined with a spectral noise-floor rectifier, this leads to

$$\left|\hat{X}(\omega)\right|^p = \begin{cases} \left|Y(\omega)\right|^p - \alpha\left|\hat{D}(\omega)\right|^p & \text{if } \left|Y(\omega)\right|^p > (\alpha+\beta)\left|\hat{D}(\omega)\right|^p \\ \beta\left|\hat{D}(\omega)\right|^p & \text{else,} \end{cases} \tag{5.11}$$

with parameters $\alpha \geq 1$ as over-subtraction factor and β with $0 \leq \beta \ll 1$ as spectral floor parameter. The spectral noise floor can be useful to somewhat mask peaks which can improve the perceived quality of the signal. For $p = 2$ this equals the technique proposed by Berouti, Schwartz, and Makhoul [1]. A drawback of over-subtraction, however, is additional speech degradation. By removing more than our current estimate of the noise, we are inevitably

bound to remove parts of the speech signal as well, making α an important parameter for the trade-off between distortion in speech and the amount of noise removed.

Lastly, we can rewrite spectral subtraction using a so-called gain function $H(\omega)$, which may make sense when implementing these algorithms in a modular framework, via

$$\left|\hat{X}(\omega)\right|^p = H_p(\omega)\left|Y(\omega)\right|^p \tag{5.12}$$

where

$$H_p(\omega) = 1 - \frac{\left|\hat{D}(\omega)\right|^p}{\left|Y(\omega)\right|^p}, \tag{5.13}$$

again $p \in \{1, 2\}$. As we will see in the remainder of this paper, all algorithms presented here can be characterized by their respective gain function $H(\omega)$. Note that for $p = 1$, we can directly estimate the complex spectral components of the clean signal by incorporating the noisy phase information via

$$\hat{X}(\omega) = H_1(\omega)\, Y(\omega), \tag{5.14}$$

using H_1 as defined above, due to $H_p \in \mathbb{R}$. By taking square-roots on both sides, we can also express the power-spectrum based variant in this fashion, leading to

$$\hat{X}(\omega) = \sqrt{H_2(\omega)}\, Y(\omega), \tag{5.15}$$

which, as can be seen, is directly applicable to the magnitude spectrum.

5.2.1 Basic Noise Estimation

As previously mentioned, a basic idea to estimate the noise magnitude spectrum is to average over the last couple of frames that do not contain speech. In this subsection, we will concretize and extend this idea, and provide some basic algorithms founded on a simple thresholding scheme for the detection of speech presence. Let us for simplicity denote

$$Y^{[t]} = R_y^{[t]} e^{j\phi_y^{[t]}}, \quad D^{[t]} = R_d^{[t]} e^{j\phi_d^{[t]}}$$

and the power spectra

$$\lambda_y^{[t]} = \left(R_y^{[t]}\right)^2, \quad \lambda_d^{[t]} = \left(\hat{R}_d^{[t]}\right)^2$$

where $\lambda_d^{[t]} \in \mathbb{R}^n$, $R_y^{[t]} \in \mathbb{R}^n$, $\phi_y^{[t]} \in \mathbb{R}^n$, $R_d^{[t]} \in \mathbb{R}^n$, $\phi_d^{[t]} \in \mathbb{R}^n$, are all vectors with one element per spectral component. Note that for λ_d we use the power spectrum estimate, as we cannot determine it directly but rather want to estimate it.

A first attempt of this can be made via a the energy. By thresholding the energy of the spectral frame of the noisy signal we can derive a simple indicator for speech presence. Using this indicator in combination with an exponentially weighted moving average yields a technique that can adapt to slight changes in noise. The update rule for the noise estimate can then be given as

$$\lambda_d^{[t]} = \begin{cases} \alpha\lambda_d^{[t-1]} + (1-\alpha)\lambda_y^{[t]} & \text{if } \left\|R_y^{[t]}\right\|^2 < \vartheta \\ \lambda_d^{[t-1]} & \text{else} \end{cases} \tag{5.16}$$

with $\vartheta \in \mathbb{R}^+$ as decision threshold. If the energy of the spectral frame falls beneath this threshold, the frame is classified as noise and thus incorporated in the new noise estimate. If the energy is above this threshold, the noise estimate is not updated. To simplify the selection of a threshold one could consider assuming that the first m frames of the signal contain noise and thus can be used to form an initial noise estimate. With this estimate, we can then compute our threshold via a SNR-like decision parameter $\delta \in \mathbb{R}^+$, i.e.,

$$\vartheta = \delta \left\|\bar{R}_y^{[0:m]}\right\|^2, \tag{5.17}$$

meaning that if the noisy signal exceeds the initial noise estimate by a certain factor ($> \delta$), it is classified as speech for the given frame. The resulting noise estimator is somewhat adaptive, however, adaptivity is strongly limited by the initial noise estimate or the threshold value, due to which we may want to consider adapting this over time. Strategies towards this will be discussed in a later section. Furthermore, the parameter δ (or alternatively the threshold ϑ) need to be tuned accordingly.

A fairly straightforward extension of this is to handle each frequency band individually. This means that instead of looking at the energy of the whole frame, we look at the power of the individual spectral coefficients. This directly leads to the update rule

$$\lambda_d^{[t,k]} = \begin{cases} \alpha\lambda_d^{[t-1,k]} + (1-\alpha)\lambda_y^{[t,k]} & \text{if } \lambda_y^{[t,k]} < \vartheta^{[k]} \\ \lambda_d^{[t-1,k]} & \text{else,} \end{cases} \tag{5.18}$$

for which we can again represent the threshold ϑ via a (small) multiple of an initial noise estimate, e.g., from the first m frames, leading to

$$\vartheta^{[k]} = \delta\bar{\lambda}_y^{[0:m,k]}. \tag{5.19}$$

In contrast to the energy based thresholding, we here also index the spectral component via k. This, again, has the same limitations as the previous method, i.e., it is fairly limited in adaptivity and the parameters need to be tuned. As a simple solution for the first issue, we could adapt the threshold ϑ by updating it each time-step using the current noise estimate, i.e.,

$$\vartheta^{[t,k]} = \delta\lambda_d^{[t-1,k]}. \tag{5.20}$$

Note that this, however, may be prone to incorporating speech into the noise estimate if the value for the decision ratio δ is chosen too large. Specifically, δ should be chosen such that the probability of misclassifying noise as speech is near-zero, as a mistakenly classifying noise as speech generally has more severe consequences (i.e., distorting or outright removing speech) than the other way around.

5.2.2 Evaluation

With spectral subtraction and the basic noise estimation techniques presented in the previous subsection, we now have a first complete method for speech enhancement. For the analysis, we chose a dataset designed to train and evaluate speech enhancement methods, provided by the University of Edinburgh [24]. Specifically, we chose a sample clip overlapped with moderate, additive, and slightly varying street noises[2], as this best fits the adaptivity goal of this paper. With the help of this, we will first look at the differences between magnitude- and power-spectrum subtraction followed by a comparison of the noise estimation methods presented in the previous section and finally a look at over-subtraction and spectral flooring.

For all evaluations, we chose a STFT (Section 4.1) with segment length of 20 ms, overlap of half the segment length, and a square-root periodic Hann window. Reconstruction of the time-signal from the frequency spectra is performed via the weighted overlap-add method (Section 4.2) with the same window as used in the STFT as synthesis window. All initial noise estimates (e.g., as used in Equations (5.17) and (5.19)) are constructed by averaging the first nine spectral frames of the signal. To ensure the magnitude or power-spectrum is always positive, we used half-wave rectification. Unless specified otherwise, neither over-subtraction nor spectral flooring is performed.

Let us begin by comparing magnitude- with power-spectrum subtraction, as shown in Figure 4. For this, we only used the initial noise estimate as described above. There is no update performed on this estimate during the enhancement process. Due to this, the changes in

[2]The specific file is `noisy_trainset_56spk_wav/p241_087.wav`, containing the sentence "Global Scotland will be held at the Glasgow Royal Concert Hall".

noise over time in the input signal (noise gets louder towards the end) are also apparent in both results. Furthermore, both results show clear reduction in noise, however, they also highlight a major problem encountered in various speech enhancement techniques: musical noise. This type of residual noise is usually described as warbling with tonal quality [16] and can be characterized as containing more musical aspects than other noise, giving it its unique name. It can be identified by the small isolated peaks in the spectrogram, having the length of an analysis frame, and is a result multiple factors, including nonlinear processing, inaccurate estimation of the noise spectrum, and large variance in the estimates of the noisy and noise signal spectra [16]. Musical noise can have a significant impact on intelligibility, in some instances it can even be of worse quality to the listener [16]. Even though there are small differences between the two results, better observable by auditory comparison and mostly in the volume of noise removed, they are largely similar. This suggests, that the additional assumption made in the power-spectrum subtraction method (i.e., noise- and clean-speech-signal independence as well as zero-mean of the noise signal, see Section 5.1 and Equation (5.3)) are, in practice, reasonable.

As we have seen during discussion of the spectral subtraction algorithm in Section 5.2, noise estimation is an important component of this, and also other subsequently discussed techniques. To this end, we have, in the the previous subsection (Section 5.2.1), had a look at some very basic methods, which we will now compare via their application to (magnitude-based) spectral subtraction, with results presented in Figure 5. As references, we use both the original noisy input signal and spectral subtraction using a non-varying noise estimate computed by averaging the first nine spectral frames (as already seen in the previous evaluation and corresponding figure). Both adaptive noise estimation methods, energy- and power-threshold based, use a threshold based on the initial noise estimation, as given by Equations (5.17) and (5.19), with $\delta = 0.8$, meaning that anything above 0.8 times the initial noise estimate is classified as speech. The exponentially weighted moving average used in the update-rules (Equations (5.17) and (5.19)) keeps the old estimate with a factor of $\alpha = 0.8$. No over-subtraction or spectral flooring is performed. When comparing the results, we can clearly see that, overall, more noise is removed when using adaptive estimates. Furthermore, the adaptivity also becomes apparent when looking at the noise changes over time. In the stationary noise estimate, these changes are still visible after the enhancement process, however, when using adaptive estimates, these variations are reduced, but still audible. When comparing both adaptive methods with each other, we can see significant differences: Due to classifying the whole frame as speech or noise, the energy-threshold based technique (second-to-bottom) also classifies some of the weaker speech parts (e.g., *s, sj, ʒ, z, zj*, etc.) as noise, which can be seen in the shadow they leave. This is significantly improved in the power-threshold based algorithm. Although there is still a shadow visible, it is reduced both in duration and its spread across frequencies. Note that neither method, however, is capable of reducing atypical disturbances, such as the high-pitched tone (likely braking noises) at the start and end of the clip. While the energy-threshold based technique performs slightly better, due to it considering the energy of the whole frame instead of individual frequency bands and thus averaging over such frequency-localized perturbations, it is still not able to completely remove it due to its significant differences compared to the rest of the noise. As should be expected, neither of the adaptive estimation methods shows any significant difference in musical noise compared to the stationary estimate.

Finally, we can discuss enhancements of the spectral subtraction technique by means of over-subtraction and spectral flooring with the results presented in Figure 6. To this end, we base our comparison on spectral subtraction (magnitude) with power-threshold based noise estimation, with the same parameters as used in the previous evaluation (i.e., $\delta = 0.8$ and $\alpha = 0.8$ for noise estimation). This method, without over-subtraction and spectral flooring, as well as the noisy input signal serve as a base-line for the evaluation. The remaining two results both incorporate over-subtraction with $\alpha = 1.5$, with the difference that the first of them uses no spectral flooring (i.e., $\beta = 0$) and the second uses spectral flooring with a noise

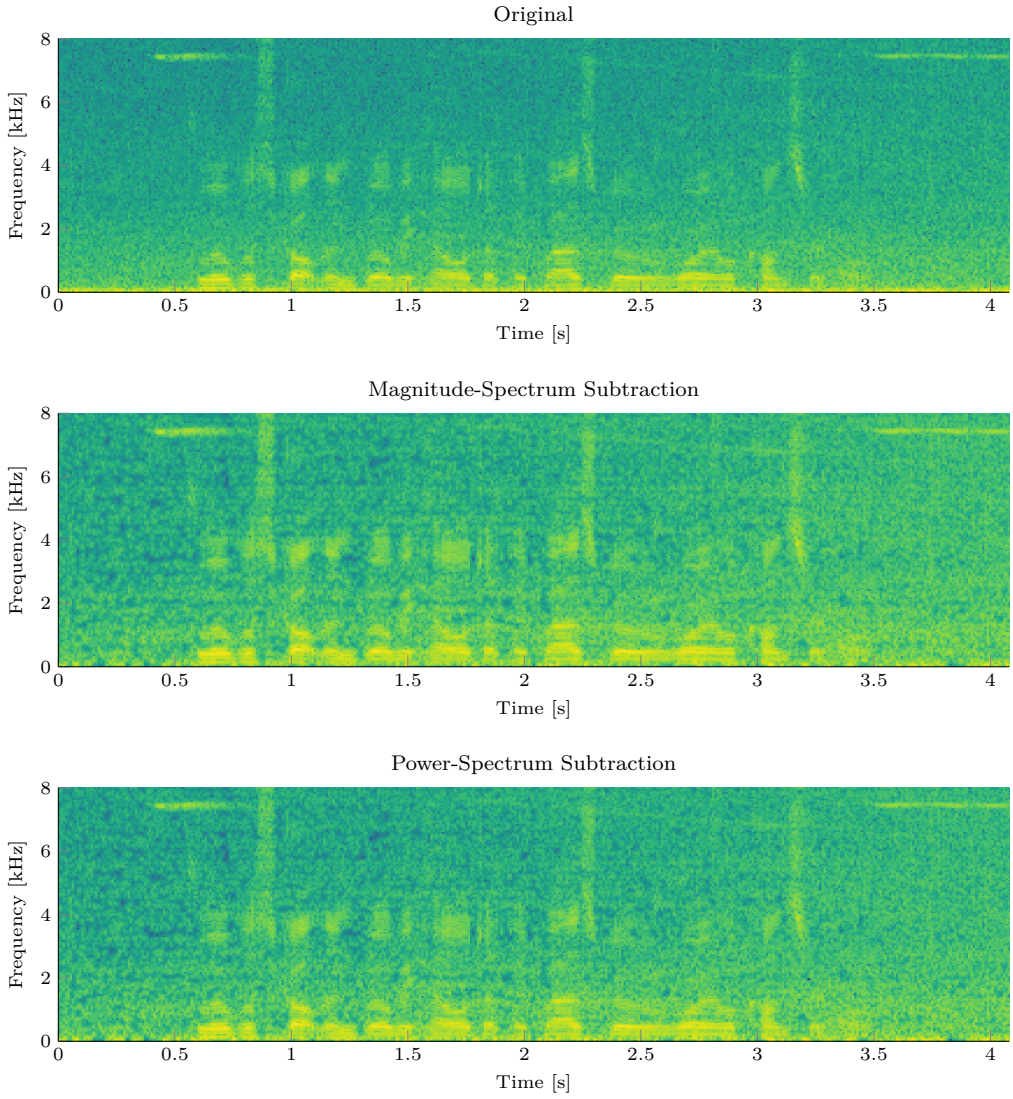

Figure 4: Spectral subtraction using the magnitude-spectrum compared to power spectrum subtraction. The noise estimate subtracted is computed by averaging the first nine frames. Notice the slight change in volume of the noise in the spectrogram of the original towards the end (top). Due to the fixed noise estimate, this change is also apparent in the processed results (middle and bottom). Both results also show signs of musical noise, i.e., small, randomly distributed peaks and valleys. In addition, both spectrograms are largely similar, however, in auditory comparison magnitude-spectrum based subtraction (middle) seems to reduce noise slightly more than power-spectrum based subtraction (bottom).

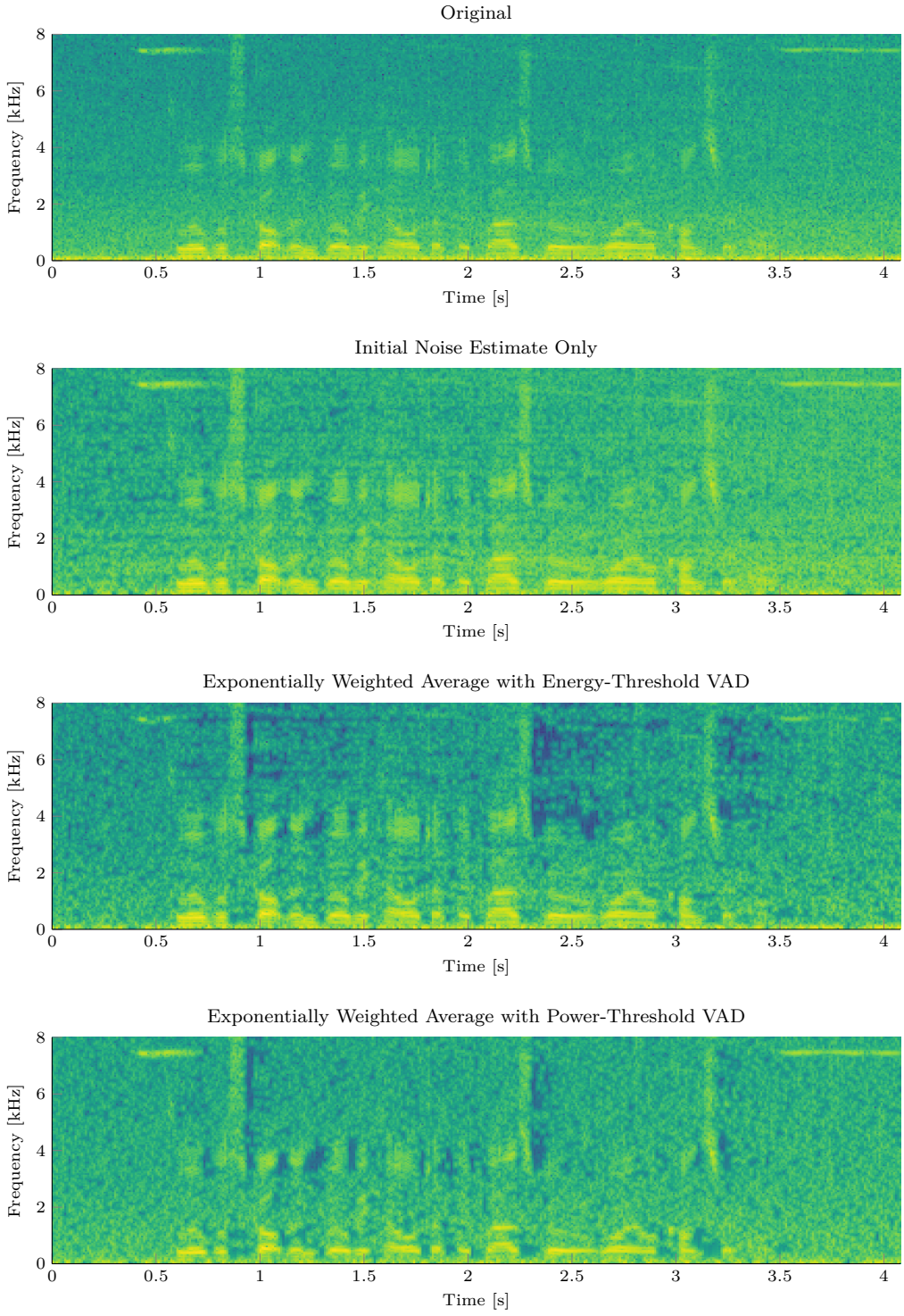

Figure 5: Basic noise estimation methods compared using spectral subtraction. Changes in noise of the original audio spectrogram (top) can be seen in the spectral subtraction method using the averaged first nine frames as noise estimate (second from top), whereas the two more advanced methods (second-to-bottom and bottom, both $\delta = 0.8$) do not show this change as much. Shadows stemming from misclassifying speech as noise can be seen in the energy-threshold based technique (second-to-bottom) and are significantly reduced in impact in the power-threshold based algorithm (bottom). Again, all results show musical noise.

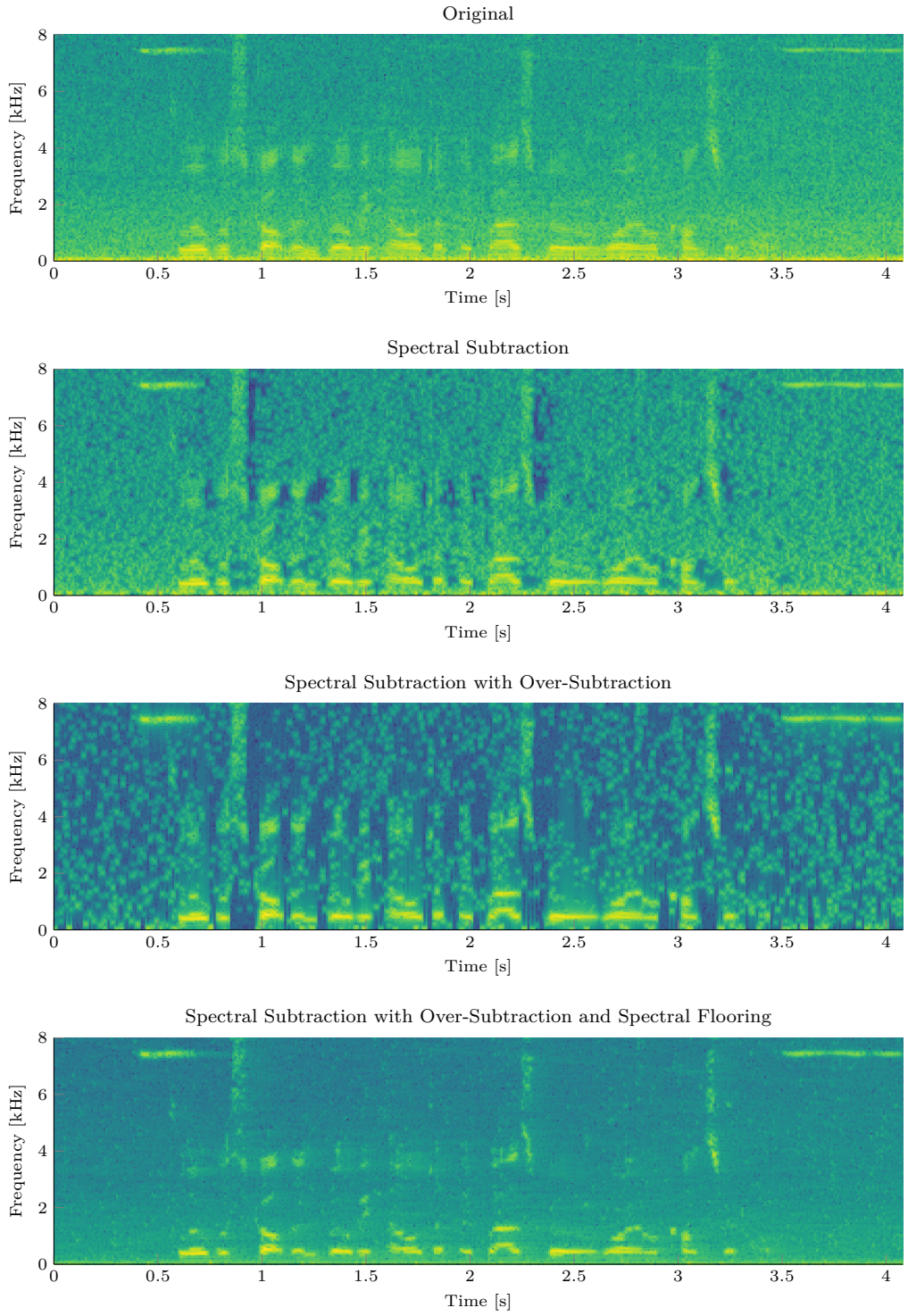

Figure 6: Over-subtraction and spectral flooring in comparison. Over-subtraction by a factor of $\alpha = 1.5$ (second-to-bottom) reduces significantly more noise as plain spectral subtraction (second from top), while keeping speech largely intact. However, this also results in strong musical noise. Adding a noise-floor of $\beta = 0.066$ (bottom) can mask this musical noise at the expense of stronger residual white(-ish) noise. All methods use power-threshold based noise estimation with the same parameters from the previous figure.

floor of $\beta = 0.066$ times the noise estimate. As can be seen in Figure 6, spectral subtraction with over-subtraction significantly increases the amount of noise removed, however, also affects the speech signal more than plain spectral subtraction, for example at the second formant around the 2.5 s mark. Note again, that the over-subtraction-factor α poses a trade-off between speech distortion and amount of noise removed. While this added distortion is noticeable, in this instance, the benefits outweigh the costs, as, in auditory analysis, the difference compared to the original speech signal is only perceivable as a sort of hall effect. A problem significantly worsened when using over-subtraction is musical noise. Due to scaling-up the amount of noise removed, the spectral peaks are significantly accentuated and not masked as much by residual noise, leading to stronger musical noise. While this noise is still not strong enough to significantly reduce intelligibility, due to its irregularity, it has a strong impact on the signal quality as perceived by human listeners. We attempt to solve this problem by introducing a spectral noise floor. Figure 6 (bottom) shows that this adaption is indeed capable of masking the spectral peaks created during the subtraction process, although at the cost of raising the noise floor. However, the resulting white-ish noise is generally considered more pleasant than musical noise, due to which this enhancement to the spectral subtraction algorithm is capable of improving the perceived quality compared to plain over-subtraction. By computing the noise-floor multiplicatively from the noise estimate, however, we also re-introduce variation in noise of the input signal back into the resulting signal.

Overall, spectral subtraction is a very flexible technique, which provides two major opportunities for a trade-off between performance and complexity: noise estimation and the post-subtraction rectification process. Most of the complexity is hidden behind these two parts, and both are integral to a good enhancement technique (see Figure 6). Noise estimation, specifically, is a complex yet important process not only for this technique, as we will see later, and can have a significant impact (Figure 5). We will improve on the noise estimation methods shown here later in Section 7. Equally as important as a good noise estimation method, is keeping musical noise in check. Here, we have seen that spectral flooring can help with that, at the cost of re-introducing a small amount of the original noise back into the signal. In addition to spectral flooring, musical noise can be reduced by several other techniques. For an overview thereof, see for example Loizou [16]. Lastly, spectral subtraction is inevitably bound to reduce the volume of the speech signal, as the noise estimate is subtracted across the complete spectrum. To compensate this, it may be desirable to apply a constant multiplicator to the signal (or spectrum) after performing the subtraction.

5.3 Statistical Estimation of Clean Speech Spectral Components

With the insights from spectral subtraction gained in the previous subsection, we can now move towards statistical-model-based methods. All speech enhancement methods discussed in the remainder of this paper fit into this category and, more specifically, are based on estimating either the full complex spectral components of the clean speech signal or their magnitude, in the latter case again with the phase taken from the noisy input signal. In this subsection, we will first explore the Wiener filter in frequency domain as one such method, which will then lead us to some important metrics for speech enhancement, the *a priori* and *a posteriori* signal-to-noise ratio, and via this we will explore connection of the Wiener filter to spectral subtraction. Finally, we will have a look at a generalized formulation of the enhancement algorithms discussed in this paper.

5.3.1 Wiener Filter in Frequency Domain

One such method is the Wiener filter, which we will look at in the frequency domain. The fundamental idea of this filter, applied to speech enhancement, is to reconstruct the clean speech signal x from the noisy input signal y. In the infinite impulse response (IIR) form,

this is done via

$$\hat{x}[t] = \sum_{k=-\infty}^{\infty} h_k \, y[t-k] \tag{5.21}$$

for $-\infty < n < \infty$ with the goal of eliminating the error defined as

$$e[t] = x[t] - \hat{x}[t], \tag{5.22}$$

as is usually done by minimizing its mean-square. To simplify Equation (5.21), we can write it as convolution, i.e.,

$$\hat{x}[t] = (y * h)[t], \tag{5.23}$$

which we can then transform into a finite impulse response (FIR) filter by simply limiting the domain of h to $h : [-m : m] \to \mathbb{R}$. Note that convolution in time-domain can be expressed by multiplication in frequency domain and thus

$$\hat{X}(\omega) = H(\omega)\, Y(\omega). \tag{5.24}$$

With the insights from the previous subsections, we can again represent this using the STFT, leading to

$$\hat{X}^{[k]}(\omega) = H^{[k]}(\omega)\, Y^{[k]}(\omega). \tag{5.25}$$

Essentially, these equations suggest that we derive a gain function H to estimate the clean speech spectral components (cf., Equation (5.12)). For readability, we will again only discuss the non-indexed variant. Due to the additivity theorem, we can similarly represent the error in the frequency domain via

$$E(\omega) = X(\omega) - \hat{X}(\omega). \tag{5.26}$$

Minimizing this via the minimum mean-square error (see for example Loizou [16]) leads to

$$H(\omega) = \frac{P_{xy}(\omega)}{P_{yy}(\omega)} \tag{5.27}$$

where

$$P_{yy}(\omega) = \mathbb{E}\left\{|Y(\omega)|^2\right\} \qquad \text{and}$$
$$P_{xy}(\omega) = \mathbb{E}\left\{Y(\omega) \cdot X^*(\omega)\right\}$$

are the power and cross-power spectra of the noisy as well as noisy and clean signal respectively. While the power spectrum P_{yy} of the noisy signal is real-valued, the cross-power spectrum P_{xy}, which is unknown and needs to be estimated, e.g., by help of a noise estimation method, is complex valued and thus H is also complex.

5.3.2 Signal-to-Noise Ratios and General Algorithm Formulation

We can now introduce two important measures for statistical-model-based speech enhancement methods: the *a priori* and *a posteriori* signal-to-noise ratio (SNR) and via those present a generalized problem formulation for the enhancement methods discussed in this paper. The *a priori* SNR ξ describes the SNR related to the clean speech signal x and is defined as

$$\xi(\omega) \coloneqq \frac{\lambda_x(\omega)}{\lambda_d(\omega)}, \tag{5.28}$$

whereas the *a posteriori* SNR γ describes the SNR with regards to the distorted signal y and is defined as

$$\gamma(\omega) \coloneqq \frac{\lambda_y(\omega)}{\lambda_d(\omega)}, \tag{5.29}$$

i.e., they describe the SNR prior and posterior to applying the distortion to the clean speech signal respectively, where the expected power spectra λ_x, λ_y, and λ_d are defined via

$$\lambda_y(\omega) \coloneqq |Y(\omega)|^2, \quad \lambda_x(\omega) \coloneqq \mathbb{E}\left\{|X(\omega)|^2\right\}, \quad \lambda_d(\omega) \coloneqq \mathbb{E}\left\{|D(\omega)|^2\right\}$$

Note that we use the expected value, given some model assumptions, in case the value cannot be determined exactly. We should further make apparent a significant connection between both SNRs. Under the assumption that $\mathbb{E}\{\lambda_y(\omega)\} = \lambda_x(\omega) + \lambda_d(\omega)$, we can write

$$\mathbb{E}\{\gamma(\omega)\} = \mathbb{E}\left\{\frac{\lambda_y(\omega)}{\lambda_d(\omega)}\right\} = \mathbb{E}\left\{\frac{\lambda_x(\omega) + \lambda_d(\omega)}{\lambda_d(\omega)}\right\} = \mathbb{E}\{\xi(\omega) + 1\}. \tag{5.30}$$

One should, however, realize that in practical application this often does not hold exactly, due to the mixture of measured values in case of the spectral components Y and the expectations as produced by our model in case of X and D. Further, this also assumes that noise and speech signal are uncorrelated and the speech signal has zero mean, i.e., that the cross-power-terms between clean speech and noise signal are zero. Nevertheless, this relation can be useful to robustify techniques in practice.

With the definition of the *a priori* SNR, we are now able to explore a relationship between spectral subtraction using the power spectrum and Wiener filtering, and via that present a more generalized view of speech enhancement methods via gain functions. From the formulation of power-spectrum based spectral subtraction given in Equations (5.13) and (5.15), we know that its gain function as applied to the magnitude is

$$H_{\text{PSS}}(\omega) = \sqrt{1 - \frac{\left|D(\omega)\right|^2}{\left|Y(\omega)\right|^2}} \tag{5.31}$$

We can now re-express this in different ways, using *a priori* and *a posteriori* SNR, i.e.,

$$H_{\text{PSS}}(\omega) = \sqrt{1 - \frac{\lambda_d(\omega)}{\lambda_y(\omega)}} = \sqrt{1 - \frac{1}{\gamma(\omega)}} = \sqrt{1 - \frac{1}{\xi(\omega) + 1}} = \sqrt{\frac{\xi(\omega)}{\xi(\omega) + 1}}$$

Similarly, we can re-write the gain function of the Wiener filter (Equation (5.27)). Again assuming independence between noise and speech signals as well as zero mean for the noise, we obtain

$$\begin{aligned} P_{xy}(\omega) &= P_{xx}(\omega) \\ P_{yy}(\omega) &= P_{xx}(\omega) + P_{dd}(\omega) \end{aligned}$$

and via this

$$H_{\text{Wiener}}(\omega) = \frac{P_{xy}(\omega)}{P_{yy}(\omega)} = \frac{P_{xx}(\omega)}{P_{xx}(\omega) + P_{dd}(\omega)} = \frac{\lambda_x(\omega)}{\lambda_x(\omega) + \lambda_d(\omega)} = \frac{\xi(\omega)}{\xi(\omega) + 1}$$

Thus $H_{\text{Wiener}} = H_{\text{PSS}}^2$, i.e., the gain function of the Wiener filter is equivalent to the squared gain function of the power-spectrum based spectral subtraction algorithm.

From these representations of the algorithms using gain functions depending on the *a priori* and *a posteriori* SNR, we can now come to a generalized form of all speech enhancement algorithms presented in this paper, illustrated in Figure 7, and many other estimation based techniques, such as for instance the method by McAulay and Malpass [18], based on a maximum likelihood estimation for the magnitude-spectrum of the clean speech signal. In short, we multiply a gain function $H(\xi, \gamma)$, usually depending only on the *a priori* and *a posteriori* SNR, to the complex noisy spectra $Y(\omega)$, obtained by a STFT, to estimate the clean speech coefficients $X(\omega)$ and then reconstruct the time-domain signal via the WOLA or OLA method. For techniques that are only concerned with estimating the magnitude, such as spectral subtraction, we usually use the phase of the noisy signal and thus $H(\xi, \gamma) \in \mathbb{R}^+$. Note that both SNRs are unknowns and, due to this, need to be estimated, for example based on a noise estimate. Methods towards this will be discussed in the next Section.

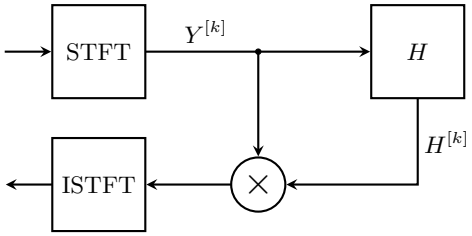

Figure 7: Generalized block-diagram of gain-based speech enhancement algorithms. The complex noisy spectrum Y obtained by a STFT is multiplied with a (potentially complex) gain function H to estimate the clean-speech coefficients X. To obtain the time-domain signal, these coefficients are back-transformed using an ISTFT method, e.g., WOLA or OLA.

6 Minimum Mean-Square Error Spectral Amplitude Estimation

Expanding on the enhancement methods provided in the previous section, we will now discuss a set of algorithms based on estimating the short-time spectral amplitude (STSA) of the clean speech signal using a minimum mean-square error (MMSE) formulation. This set of algorithms is based on the MMSE STSA estimation by Ephraim and Malah [7] and forms the core of this paper. In accordance with this fundamental technique, we will also look at approaches for estimating the *a priori* SNR, which then gives us all the required parts for a complete speech enhancement system. Additionally, we will discuss two noteworthy enhancements of the MMSE STSA algorithm: first, using a logarithmic error measure, i.e., estimating the log-spectral amplitude, as again proposed by Ephraim and Malah [6], and second, an adaption to incorporate uncertainty in the speech signal presence, proposed by Cohen and Berdugo [4]. These techniques will be combined with an enhanced noise estimation method, discussed in Section 7, and finally evaluated in Section 8.

For the MMSE STSA technique proposed by Ephraim and Malah [7], we again use additivity of the noise as basis. Additionally we limit the signal to an analysis frame of length T and assume that the interval is normalized to $[0, T]$. Combined, this leads to

$$y(t) = x(t) + d(t), \quad 0 \leq x \leq T.$$

Discretizing the signal y in time and performing a discrete Fourier transform then yields the same fundamental assumption in frequency domain, i.e.,

$$Y^{[k]} = X^{[k]} + D^{[k]},$$

where k is the index to the respective spectral component. For readability, we will express the spectral components in terms of their spectral amplitude R and phase ϕ via

$$Y^{[k]} \coloneqq R_y^{[k]} \exp\left(j\phi_y^{[k]}\right), \quad X^{[k]} \coloneqq R_x^{[k]} \exp\left(j\phi_x^{[k]}\right), \quad \text{and} \quad D^{[k]} \coloneqq R_d^{[k]} \exp\left(j\phi_d^{[k]}\right).$$

With those definitions, we can now express the underlying estimation goal of the method in continuous form as

$$\hat{R}_x(\omega) = \mathbb{E}\left\{ R_x(\omega) \;\middle|\; y(t), \quad 0 \leq t \leq T \right\}, \tag{6.1}$$

which we discretize in time- and frequency-domain, leading to

$$\hat{R}_x^{[k]} = \mathbb{E}\left\{ R_x^{[k]} \;\middle|\; Y^{[0]}, Y^{[1]}, \ldots, Y^{[N]} \right\}. \tag{6.2}$$

Due to the inversibility of the Fourier transform, this is the same estimation modulo errors introduced during discretization. To estimate the clean-speech spectral amplitude R_x, Ephraim and Malah use a Gaussian model, which we will omit here for brevity and simplicity.

A fundamental assumption in this model is the statistical independence of the individual spectral components, i.e., each spectral component is seen as an independent random variable. With it, we can simplify Equation (6.2), yielding

$$\hat{R}_x^{[k]} = \mathbb{E}\left\{ R_x^{[k]} \;\middle|\; Y^{[k]} \right\}. \tag{6.3}$$

This assumption is equivalent to the assumption that the Fourier expansion coefficients are uncorrelated, which in turn is ratified by the fact that correlation between different Fourier coefficients approaches zero with increasing frame lengths [7]. Note that, due to the limited frame length, this assumption does not strictly hold in practical applications. Finally, applying the model of Ephraim and Malah to Equation (6.3) gives us (neglecting indexing of spectral components for readability)

$$\hat{R}_x = H_{\text{MMSE}}(\xi, \gamma) \cdot R_y \tag{6.4}$$

with the gain function

$$H_{\text{MMSE}}(\xi, \gamma) = \Gamma(1.5)\frac{\sqrt{\nu}}{\gamma} \exp\left(-\frac{\nu}{2}\right)\left[(1+\nu)\, I_0\left(\frac{\nu}{2}\right) + \nu I_1\left(\frac{\nu}{2}\right)\right], \tag{6.5}$$

where $\gamma^{[k]}$ is the *a posteriori* SNR as defined in Equation (5.29), discretized in frequency for spectral component k, and

$$\nu^{[k]} := \frac{\xi^{[k]}}{1+\xi^{[k]}}\gamma^{[k]} \tag{6.6}$$

represents a corrected *a priori* SNR (cf., Equation (5.30)) using the *a priori* SNR $\xi^{[k]}$ (Equation (5.28)), again discretized in frequency. Further, $I_0(\cdot)$ and $I_1(\cdot)$ denote the modified Bessel functions of zeroth and first order, respectively, and $\Gamma(.)$ with $\Gamma(1.5) = \frac{\sqrt{\pi}}{2}$ the gamma function.

Here, we again use the phase of the noisy signal, i.e., $H_{\text{MMSE}}(\xi^{[k]}, \gamma^{[k]})$ be can directly multiplied to the k-th spectral component of the noisy signal as show in Figure 7 and discussed in Section 5.3.2, however, Ephraim and Malah also experimented with estimations for the complex exponential $e^{j\phi_x(\omega)}$ as well as the optimal phase $\phi_x(\omega)$, both relying on the same aforementioned statistical model. They concluded that trying to estimate the complex exponential leads to a trade-off in optimality between amplitude estimation and phase estimation, where improving one would adversely affect the other, and further that an estimator which does not affect the amplitude equals the complex exponential of the noisy signal [7]. The last result is also shown in their approach to estimate the phase directly, which yields that the optimal phase estimation of the clean speech signal is the phase of the noisy signal [7].

6.1 Estimating the *A Priori* Signal-to-Noise Ratio

While the *a posteriori* SNR $\gamma := {}^{\lambda_y}/_{\lambda_d}$ can directly be computed using a noise estimate $\hat{\lambda}_d$, the *a priori* SNR $\xi := {}^{\lambda_x}/_{\lambda_d}$ requires estimations for the spectral power of both noise and clean speech signals. As we have previously discussed techniques for estimating the spectral noise power λ_d, it remains to investigate on how to estimate the spectral signal power λ_x. To this end, we will look at two methods, again proposed by Ephraim and Malah [7], one based on estimating λ_x and the other based on estimating ξ more directly.

The first method is based on a maximum likelihood estimation of the spectral power $\lambda_x^{[k]}$ of the clean signal. For this, we use an estimation window of L previous observations of the spectral component, i.e., observations of $\{Y^{[t,k]}, Y^{[t-1,k]}, \ldots, Y^{[t-L+1,k]}\}$. From this, the same Gaussian model is used to estimate the spectral signal power λ_x for component k, under the assumption that these previous observations are uncorrelated. Note that this does

not hold in practice, as the analysis frames need to overlap to ensure reversibility of the STFT. Applying the model yields

$$\hat{\lambda}_x^{[t,k]} = \max\left\{\frac{1}{L}\sum_{l=0}^{L-1}\lambda_y^{[t-l,k]} - \lambda_d^{[t,k]},\ 0\right\} \tag{6.7}$$

which then suggests

$$\hat{\xi}^{[t,k]} = \max\left\{\frac{1}{L}\sum_{l=0}^{L-1}\gamma^{[t-l,k]} - 1,\ 0\right\} \tag{6.8}$$

by plugging in the definition of the *a priori* SNR. As previously mentioned in Section 4.3, a running average like this is not very well suited for real-time signal processing, thus, in practice, we replace it with a recursive average, leading to the final formulation

$$\bar{\gamma}^{[t,k]} = \alpha\bar{\gamma}^{[t-1,k]} + (1-\alpha)\frac{\gamma^{[t,k]}}{\beta}, \quad 0 \leq \alpha < 1,\ \beta \geq 1 \tag{6.9}$$

$$\hat{\xi}^{[t,k]} = \max\left\{\bar{\gamma}^{[t,k]} - 1,\ 0\right\}, \tag{6.10}$$

in which β is a correction factor for $\gamma^{[k]}$, related to the spectral subtraction gain function [7].

The second technique is obtained via a more constructive methodology and is referred to as the decision-directed estimation approach. It is directly based on the definition of the *a priori* SNR (Equation (5.28)) and the relation of it to the *a posteriori* SNR (Equation (5.30)), i.e.,

$$\xi^{[t,k]} = \frac{\lambda_x^{[t,k]}}{\lambda_d^{[t,k]}} \qquad \text{and} \tag{6.11}$$

$$\xi^{[t,k]} = \mathbb{E}\left\{\gamma^{[t,k]} - 1\right\}. \tag{6.12}$$

The fundamental idea is that we combine these equations, yielding

$$\xi^{[t,k]} = \mathbb{E}\left\{\frac{1}{2}\frac{\lambda_x^{[t,k]}}{\lambda_d^{[t,k]}} + \frac{1}{2}\left(\gamma^{[t,k]} - 1\right)\right\}, \tag{6.13}$$

which is expected to be more stable with regards to errors in the individual estimations. Note that the second part resembles the maximum likelihood estimation of the *a priori* SNR. Instead of trying to estimate the amplitude of the current frame, we use the estimate of the previous frame as an approximation. Generalizing the previous equation with a mixing ratio $0 < \alpha < 1$ and dropping expectation operators results in

$$\hat{\xi}^{[t,k]} = \alpha\frac{\hat{\lambda}_x^{[t-1,k]}}{\lambda_d^{[t-1,k]}} + (1-\alpha)\max\left\{\gamma^{[t,k]} - 1,\ 0\right\}. \tag{6.14}$$

We can now obtain the estimate $\hat{\lambda}_x^{[t-1,k]}$ from the gain function of the previous frame by use of the identities $\lambda_x^{[t,k]} = \mathbb{E}\{|\hat{X}^{[t,k]}|^2\}$ and $|\hat{X}^{[t,k]}| = H(\hat{\xi}^{[t,k]}, \gamma^{[t,k]}) \cdot |Y^{[t,k]}|$, yielding the final update rule of the decision-directed estimation approach

$$\hat{\xi}^{[t,k]} = \alpha H^2\left(\hat{\xi}^{[t-1,k]}, \gamma^{[t-1,k]}\right)\gamma^{[t-1,k]} + (1-\alpha)\max\left\{\gamma^{[t,k]} - 1,\ 0\right\}. \tag{6.15}$$

For initialization, Ephraim and Malah [7] propose

$$\hat{\xi}^{[0,k]} = \alpha + (1-\alpha)\max\left\{\gamma^{[0,k]} - 1,\ 0\right\}. \tag{6.16}$$

With these estimation methods, we now have all requirements for a complete and modular speech enhancement algorithm, an illustration of which is provided in Figure 8. Before a final evaluation in Section 8, we will in the next subsections look at further improvements to the gain function.

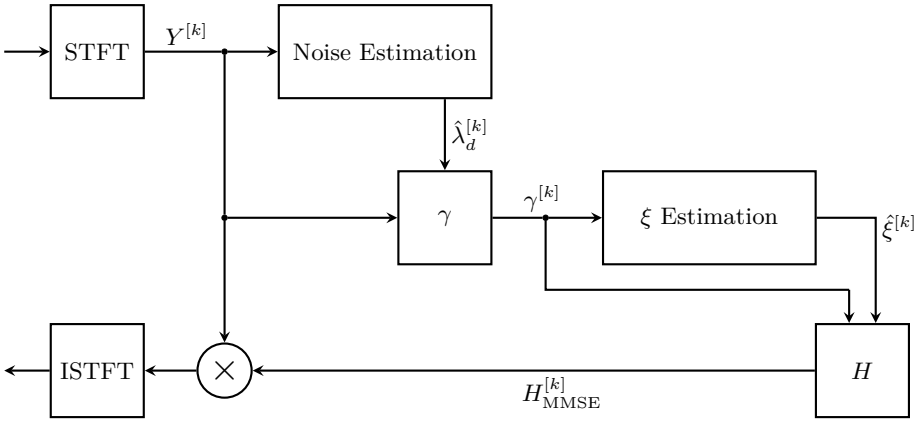

Figure 8: Full speech enhancement algorithm using MMSE STSA estimation. A noise estimate is updated for each spectral analysis frame, with which the *a posteriori* SNR is computed. This is then used to estimate the *a priori* SNR, which may additionally use the previous value of the gain function (in case of the decision-directed approach). The gain function is computed from both SNRs and is finally multiplied to the complex spectral components resulting in an estimate of the clean speech spectrum.

6.2 Estimating the Logarithmic Spectral Amplitude

In a later work, Ephraim and Malah [6] discuss an improvement to the previously presented gain function H_{MMSE} obtained via MMSE STSA estimation. Instead of estimating the spectrum directly via a minimum squared error function, they propose to estimate the logarithm of the spectrum. The rationale behind this is founded on reports, e.g., by Gray et al. [9], showing that a distortion measure based on the log-spectrum is more suitable for speech processing. This essentially only represents a change in the error measure, as we will use the same Gaussian model below, which can be expressed as

$$e^{[k]} = \mathbb{E}\left\{\left(\log R_x^{[k]} - \log \hat{R}_x^{[k]}\right)^2\right\} \tag{6.17}$$

and is to be minimized. This directly leads to the estimation

$$\hat{R}_x^{[k]} = \exp\left(\mathbb{E}\left\{\ln R_x^{[k]} \mid y(t), \quad 0 \le t \le T\right\}\right) \tag{6.18}$$

and following the same reasoning as in the MMSE STSA technique above to

$$\hat{R}_x^{[k]} = \exp\left(\mathbb{E}\left\{\ln R_x^{[k]} \mid Y_k\right\}\right). \tag{6.19}$$

Using the same Gaussian model as above then leads to the gain function

$$H_{\text{log-MMSE}}\left(\xi^{[k]}, \gamma^{[k]}\right) = \frac{\xi^{[k]}}{1+\xi^{[k]}} \exp\left(\frac{1}{2}\int_{\nu^{[k]}}^{\infty} \frac{e^{-t}}{t}\, dt\right) \tag{6.20}$$

where $\nu^{[k]}$ is defined in Equation (6.6) and the integral is known as the exponential integral. Many scientific computation libraries, such as the GSL [8] provide methods for computing said integral efficiently.

6.3 Incorporating Speech Signal Uncertainty

The idea of incorporating speech signal uncertainty was already proposed by Ephraim and Malah in their original paper [7]. It is based on the fact that speech contains a significant

amount of pauses during which only the noise signal is present. The optimally-modified log-spectral amplitude (OM-LSA) method we present here was formulated by Cohen and Berdugo [4] and is founded on two hypotheses

$$\mathcal{H}_0^{[t,k]} : Y^{[t,k]} = D^{[t,k]} \qquad \text{and} \tag{6.21}$$

$$\mathcal{H}_1^{[t,k]} : Y^{[t,k]} = X^{[t,k]} + D^{[t,k]}, \tag{6.22}$$

assuming that speech is absent ($\mathcal{H}_0$) and speech is present ($\mathcal{H}_1$), respectively. Based on this, we can define the conditional speech presence probability p via

$$p[t,k] \coloneqq P\big(\mathcal{H}_1^{[t,k]} \mid Y^{[t,k]}\big). \tag{6.23}$$

We further define the *a priori* probability for speech absence q as

$$q[t,k] \coloneqq P\big(\mathcal{H}_0^{[t,k]}\big). \tag{6.24}$$

Via the same Gaussian model as used above in the derivation of the MMSE and log-MMSE STSA methods, we can express the conditional speech presence probability p in dependence on the speech absence probability q, resulting in

$$p[t,k] = \left(1 + \frac{q[t,k]}{1-q[t,k]} \cdot \left(1+\xi^{[t,k]}\right) \cdot \exp\left(-\nu^{[t,k]}\right)\right)^{-1}. \tag{6.25}$$

Using this conditional speech presence probability, we can formulate the expected log-spectral amplitude via

$$\begin{aligned}\mathbb{E}\Big[\log R_x^{[t,k]} \;\Big|\; Y^{[t,k]}\Big] = p[t,k] \cdot \mathbb{E}\Big[\log R_x^{[t,k]} \;\Big|\; Y^{[t,k]}, \mathcal{H}_1^{[t,k]}\Big] \\ + (1-p[t,k]) \cdot \mathbb{E}\Big[\log R_x^{[t,k]} \;\Big|\; Y^{[t,k]}, \mathcal{H}_0^{[t,k]}\Big],\end{aligned} \tag{6.26}$$

leading to our estimation approach

$$\begin{aligned}\hat{R}_x^{[t,k]} = \exp\left(\mathbb{E}\Big[\log R_x^{[t,k]} \;\Big|\; Y^{[t,k]}, \mathcal{H}_1^{[t,k]}\Big]\right)^{p[t,k]} \\ \cdot \exp\left(\mathbb{E}\Big[\log R_x^{[t,k]} \;\Big|\; Y^{[t,k]}, \mathcal{H}_0^{[t,k]}\Big]\right)^{1-p[t,k]}.\end{aligned} \tag{6.27}$$

This formulation now suggests that we split this into two estimations and consider them separately, leading to two separate gain functions, i.e.,

$$\exp\left(\mathbb{E}\Big[\log R_x^{[t,k]} \;\Big|\; Y^{[t,k]}, \mathcal{H}_0^{[t,k]}\Big]\right) = H_{\min} \cdot \left|Y^{[t,k]}\right| \tag{6.28}$$

in case of speech absence and

$$\exp\left(\mathbb{E}\Big[\log R_x^{[t,k]} \;\Big|\; Y^{[t,k]}, \mathcal{H}_1^{[t,k]}\Big]\right) = H_{\mathcal{H}_1}^{[t,k]} \cdot \left|Y^{[t,k]}\right| \tag{6.29}$$

in case of speech presence. We assume $H_{\min}$ to be a constant, which is determined by a subjective criteria for noise naturalness [4] and has a similar purpose as the spectral flooring we previously discussed for spectral subtraction. As the fundamental assumptions of $\mathcal{H}_1$ and the log-MMSE method discussed above are equal, including that they both assume the speech signal is present, this estimation yields $H_{\mathcal{H}_1} = H_{\text{log-MMSE}}$. Using Equation (6.27) and the previous observations, the combined gain function can now be expressed as

$$H\big(\xi^{[t,k]}, \gamma^{[t,k]}\big) = H_{\mathcal{H}_1}^{p[t,k]}\big(\xi^{[t,k]}, \gamma^{[t,k]}\big) \cdot H_{\min}^{1-p[t,k]} \tag{6.30}$$

and by plugging in $H_{\mathcal{H}_1}$

$$H_{\text{OM-LSA}}\big(\xi^{[t,k]}, \gamma^{[t,k]}\big) = H_{\text{log-MMSE}}^{p[t,k]}\big(\xi^{[t,k]}, \gamma^{[t,k]}\big) \cdot H_{\min}^{1-p[t,k]}. \tag{6.31}$$

6.3.1 Modified *A Priori* SNR Estimation

Due to the integration of speech presence uncertainty into the gain function H, Cohen and Berdugo also propose a modification to the decision-directed estimation approach for the *a priori* SNR as given in Equation (6.15). They propose to use

$$\hat{\xi}^{[t,k]} = \alpha H_{\mathcal{H}_1}^2\left(\hat{\xi}^{[t-1,k]}, \gamma^{[t-1,k]}\right)\gamma^{[t-1,k]} + (1-\alpha)\max\left\{\gamma^{[t,k]} - 1,\ 0\right\}, \tag{6.32}$$

i.e., that instead of using the modified gain function, we should only use the gain function $H_{\mathcal{H}_1}$ for the case in which the speech is present. This should already make sense intuitively, as the decision-directed approach depends on the gain function to compute a direct estimate of the *a priori* SNR, and by adapting the gain function with a subjectively determined constant, we actively influence this estimate. Mathematically, we can show this via the gain-related part of the estimation, which we can expand using the idea behind it presented in Section 6.1, i.e.,

$$\begin{aligned}
\hat{\xi}_{\text{gain}}^{[t,k]} &= H^2\left(\hat{\xi}^{[t-1,k]}, \gamma^{[t-1,k]}\right)\gamma^{[t-1,k]} \\
&= H^2\left(\hat{\xi}^{[t-1,k]}, \gamma^{[t-1,k]}\right)\cdot\frac{\lambda_y^{[t-1,k]}}{\lambda_d^{[t-1,k]}} \\
&= \left[H_{\mathcal{H}_1}\left(\xi^{[t,k]}, \gamma^{[t,k]}\right)\cdot\frac{\lambda_y^{[t-1,k]}}{\lambda_d^{[t-1,k]}}\right]^{p[t,k]}\cdot\left[H_{\min}\cdot\frac{\lambda_y^{[t-1,k]}}{\lambda_d^{[t-1,k]}}\right]^{1-p[t,k]} \\
&= \left[\frac{\hat{\lambda}_x^{[t-1,k]}}{\lambda_d^{[t-1,k]}}\right]^{p[t,k]}\cdot\left[\frac{\lambda_{x\min}}{\lambda_d^{[t-1,k]}}\right]^{1-p[t,k]}.
\end{aligned}$$

In case of speech absence, i.e., $p[t,k] = 0$, we now get a minimum *a priori* SNR implied by the constant $H_{\min}$ and thus also $\lambda_{x\min}$ instead of the expected zero. Also remember that we introduced $H_{\min}$ to retain a natural noise floor and via this mask artifacts. As this is purely intended for the subjective quality of the result, our *a priori* SNR estimate should not contain such modifications. Further mathematical analysis is presented by Cohen and Berdugo [4].

6.3.2 *A Priori* Speech Absence Probability Estimation

To compute the modified gain function, we rely on the conditional speech presence probability p, which, in turn, depends on the *a priori* speech absence probability q. Thus, an estimator thereof is required before we can implement this as an algorithm. For this, Cohen and Berdugo [4] propose a multi-scale approach, based on a recursive average in time $\bar{\xi}$ of the *a priori* SNR ξ, i.e.,

$$\bar{\xi}^{[t,k]} = \beta\,\bar{\xi}^{[t-1,k]} + (1-\beta)\,\hat{\xi}^{[t-1,k]}. \tag{6.33}$$

This is then averaged locally over frequencies using normalized window functions h_λ of size $2w_\lambda + 1$ respectively via

$$\bar{\xi}_\lambda^{[t,k]} = \sum_{i=-w_\lambda}^{w_\lambda} h_\lambda^{[i]}\cdot\bar{\xi}^{[t,k-i]} \tag{6.34}$$

where λ denotes the scale of the window function. Specifically, two scales are used: a larger so-called global one and a smaller so-called local one, thus $\lambda \in \{\text{local, global}\}$ and, without loss of generality, $w_{\text{global}} > w_{\text{local}}$. Further, a third scale is used to incorporate the complete analysis frame via

$$\bar{\xi}_{\text{frame}}^{[t]} = \underset{k}{\text{mean}}\left\{\bar{\xi}^{[t,k]}\right\}. \tag{6.35}$$

Combined, this forms the basis of a robust multi-scale estimation method, which should be capable of handling errors and outliers in the *a priori* SNR estimate. These three estimates

are then used to compute three respective likelihoods for speech presence, which will then be used to obtain the final speech absence probability estimation. Local and global likelihoods P_{local} and P_{global} are computed on a logarithmic scale, confined to a range between zero and one, as

$$P_{\lambda}^{[t,k]} = \begin{cases} 0 & \text{if } \bar{\xi}_{\lambda}^{[t,k]} \leq \bar{\xi}_{\min}, \\ 1 & \text{if } \bar{\xi}_{\lambda}^{[t,k]} \geq \bar{\xi}_{\max}, \\ \frac{\log\left(\bar{\xi}_{\lambda}^{[t,k]}\right) - \log\left(\bar{\xi}_{\min}\right)}{\log\left(\bar{\xi}_{\max}\right) - \log\left(\bar{\xi}_{\min}\right)} & \text{otherwise.} \end{cases} \tag{6.36}$$

The empirically chosen constants $\bar{\xi}_{\min}$ and $\bar{\xi}_{\max}$ characterize this mapping and are selected to attenuate noise while retaining weak speech components [4]. The speech-presence likelihood of the full frame, P_{frame}, is designed specifically towards this objective via

$$P_{\text{frame}}^{[t]} = \begin{cases} 0 & \text{if } \bar{\xi}_{\text{frame}}^{[t]} \leq \bar{\xi}_{\min}, \\ 1 & \text{if } \bar{\xi}_{\text{frame}}^{[t]} > \bar{\xi}_{\min} \text{ and } \bar{\xi}_{\text{frame}}^{[t]} > \bar{\xi}_{\text{frame}}^{[t-1]}, \\ \mu^{[t]} & \text{otherwise.} \end{cases} \tag{6.37}$$

Again, *a priori* SNR values below a threshold are interpreted as zero likelihood for speech presence, however, we additionally set the likelihood to one as soon as we detect a rising SNR, to avoid clipping of speech startings or weak components [4]. The term μ, defined as

$$\mu^{[t]} \coloneqq \begin{cases} 0 & \text{if } \bar{\xi}_{\text{frame}}^{[t]} \leq \bar{\xi}_{\text{peak}}^{[t]} \cdot \bar{\xi}_{\min}, \\ 1 & \text{if } \bar{\xi}_{\text{frame}}^{[t]} \geq \bar{\xi}_{\text{peak}}^{[t]} \cdot \bar{\xi}_{\max}, \\ \frac{\log\left(\bar{\xi}_{\text{frame}}^{[t]} / \bar{\xi}_{\text{peak}}^{[t]}\right) - \log\left(\bar{\xi}_{\min}\right)}{\log\left(\bar{\xi}_{\max}\right) - \log\left(\bar{\xi}_{\min}\right)} & \text{otherwise,} \end{cases} \tag{6.38}$$

essentially delays the transition from $\mathcal{H}_1$, i.e., the speech presence hypothesis, to $\mathcal{H}_0$, i.e., the speech absence hypothesis, by relying on the most recent maxima $\bar{\xi}_{\text{peak}}$. This peak value is updated via

$$\bar{\xi}_{\text{peak}}^{[t]} = \begin{cases} \min\left\{ \max\left\{ \bar{\xi}_{\text{frame}}^{[t]},\ \bar{\xi}_{\text{pmin}} \right\},\ \bar{\xi}_{\text{pmax}} \right\} & \text{if } \bar{\xi}_{\text{frame}}^{[t]} > \bar{\xi}_{\min} \text{ and } \bar{\xi}_{\text{frame}}^{[t]} > \bar{\xi}_{\text{frame}}^{[t-1]} \\ \bar{\xi}_{\text{peak}}^{[t-1]} & \text{otherwise,} \end{cases} \tag{6.39}$$

i.e., each time the SNR rises, and is confined by $\bar{\xi}_{\text{pmin}}$ and $\bar{\xi}_{\text{pmax}}$, which are again empirical constants. The transition itself is again on a logarithmic scale, and the constants define the delay with which it will fade out towards $\mathcal{H}_0$. This delay is intended to reduce the misdetection of weak speech tails [4]. Finally, we can combine the three likelihoods for speech presence P_{local}, P_{global}, and P_{frame} into the *a priori* speech absence probability q by

$$\hat{q}^{[t,k]} = 1 - P_{\text{local}}^{[t,k]} \cdot P_{\text{global}}^{[t,k]} \cdot P_{\text{frame}}^{[t]}. \tag{6.40}$$

To further reduce the possibility of speech distortion, $\hat{q}$ is restricted to be smaller than a threshold $q_{\max}$, cf., Equation (6.25), resulting in

$$\hat{q}^{[t,k]} = \min\left\{ 1 - P_{\text{local}}^{[t,k]} \cdot P_{\text{global}}^{[t,k]} \cdot P_{\text{frame}}^{[t]},\ q_{\max} \right\}. \tag{6.41}$$

The parameter $q_{\max}$ can thus be seen as an uncertainty parameter for erring on the safe side, i.e., not attenuating (weak) speech components. Further note that, because this estimate only depends on the *a priori* SNR ξ, using it for the modified gain function, i.e., Equations (6.30) and (6.31), provides a drop-in replacement for any standard gain function depending only on the *a priori* and *a posteriori* SNRs.

7 Noise Estimation via Minima-Controlled Recursive Averaging

The last part for a fully robust and adaptive speech enhancement method is an estimator $\hat{\lambda}_d$ for the time-varying noise spectrum, incorporating these properties. To this end, we will improve upon the basic ideas presented in Section 5.2.1 via the minima-controlled recursive averaging (MCRA) method proposed by Cohen and Berdugo [4]. Similar to previously presented techniques, its fundament is a recursive average to capture the (assumed to be) slowly varying changes in the noise power spectrum. In this instance, however, we again introduce a (modified) speech presence probability p_d, determining when and by how much the average is updated, leading to the update-rule

$$\hat{\lambda}_d^{[t+1,k]} = p_d^{[t,k]} \hat{\lambda}_d^{[t,k]} + \left(1 - p_d^{[t,k]}\right) \left[\alpha_d \hat{\lambda}_d^{[t,k]} + (1 - \alpha_d) \left|Y^{[t,k]}\right|^2\right], \tag{7.1}$$

$$= \tilde{\alpha}_d^{[t,k]} \hat{\lambda}_d^{[t,k]} + \left(1 - \alpha_d^{[t,k]}\right) \left|Y_d^{[t,k]}\right|^2, \tag{7.2}$$

with time-varying smoothing parameter $\tilde{\alpha}_d$ defined as

$$\tilde{\alpha}_d^{[t,k]} \coloneqq \alpha_d + (1 - \alpha_d)\, p_d^{[t,k]}, \tag{7.3}$$

where $0 < \alpha_d < 1$ is the weighting parameter for the average. In contrast to the previously discussed methods, this new technique updates the estimate by weighting the newly added power spectrum instead of only making a binary decision, allowing for incorporation of uncertainty.

As hinted on by notation, there is a significant difference between the conditional speech presence probability p discussed in Section 6.3 and the adapted conditional speech presence probability p_d used here. This difference stems from the different cost of errors in making decisions between the hypotheses $\mathcal{H}_1$ (speech present), and $\mathcal{H}_0$ (speech absent). In Section 6.3 we rather want to err on the side of speech presence, meaning that we rather choose $\mathcal{H}_1$ mistakenly, than choosing $\mathcal{H}_0$ mistakenly, as this results in less distortion of speech. The objective of minimizing speech degradation stays the same here, however, the way to achieve, or rather protect, it has changed: Let us assume we would erroneously classify speech as noise, i.e., wrongly choose $\mathcal{H}_1$, then our noise estimate gets updated with a power-spectrum containing speech, and thus, subsequently, *a priori* and *a posteriori* SNRs get compromised. In the worst-case scenario, this will lead to wrong decisions in the probability estimation of the modified gain function (Section 6.3) and impact the gain function itself, causing speech to be identified as noise and thus removed, potentially causing a sort of chain-reaction. It is thus, for noise estimation, much more preferable to mistakenly classify speech as noise and not update the noise estimate at all, especially since we assume that the noise is only varying slowly over time. Note that the possibility of such a runaway process, slowly classifying more and more speech as noise, is also a strong reason why we do not want to base the estimate for p_d onto a measure such as the *a priori* SNR, which itself is estimated based on the noise estimate. Due to this, we introduce the adapted hypotheses $\mathcal{H}_0'$ and $\mathcal{H}_1'$, as well as an estimation approach suited for this task, originally proposed by Cohen and Berdugo [4].

This estimation approach, illustrated in Figure 9, is based on the spectral power of the signal and robustified against outliers by first averaging in the frequency domain using a window function h of size $2w + 1$ via

$$S_f^{[t,k]} = \sum_{i=-w}^{w} h^{[i]} \cdot \left|Y^{[t,k-i]}\right|^2 \tag{7.4}$$

and then averaging in time via the recursive average

$$S^{[t,k]} = \alpha_s S^{[t-1,k]} + (1 - \alpha_s)\, S_f^{[t,k]} \tag{7.5}$$

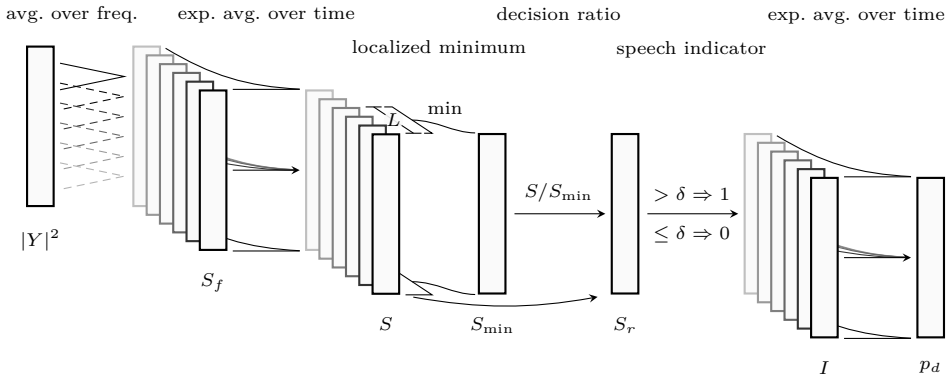

Figure 9: Minima-controlled conditional speech presence estimation approach for noise estimation. First, the power spectrum $|Y|^2$ is averaged over frequencies (S_f) using a window function and then over time (S) via a recursive average. By tracking the minima constrained to the latest L frames, a decision ratio S_r is computed, which when compared to the decision parameter δ provides a binary indicator function I for conditional speech presence. Recursively averaging this indicator function over time yields the final estimator.

with $0 < \alpha_s < 1$ as the usual control parameter. Based on this, we compute $S_{\min}$ as a windowed minimum over the last L to $2L$ frames using Algorithm 1, described in Section 4.4. From this, we can compute the ratio S_r defined as

$$S_r^{[t,k]} := \frac{S^{[t,k]}}{S_{\min}^{[t,k]}}. \tag{7.6}$$

This ratio between local energy and time-constrained minimum of the noise signal forms the fundament of the remaining estimation process. Using the *a priori* probabilities for speech absence and speech presence, $P(\mathcal{H}_0)$ and $P(\mathcal{H}_1)$, respectively, we can formulate the Bayes minimum-risk decision rule

$$\frac{p\left(S_r \mid \mathcal{H}_1\right)}{p\left(S_r \mid \mathcal{H}_0\right)} \underset{\mathcal{H}_1'}{\overset{\mathcal{H}_0'}{\gtrless}} \frac{c_{1,0} \cdot P\left(\mathcal{H}_0\right)}{c_{0,1} \cdot P\left(\mathcal{H}_1\right)} \tag{7.7}$$

where $c_{i,j}$ is the cost, i.e., risk, for mistakenly deciding $\mathcal{H}_i'$ when actually $\mathcal{H}_j'$ holds true. As the likelihood ratio of the conditional probabilities for the ratio S_r is a monotonic function [4], we can express the decision rule as

$$S_r^{[t,k]} \underset{\mathcal{H}_0'}{\overset{\mathcal{H}_1'}{\gtrless}} \delta, \tag{7.8}$$

using a fixed decision parameter $\delta > 1$. With this, we can then create an indicator function for the modified speech presence hypothesis $\mathcal{H}_1$ as

$$I^{[t,k]} := \begin{cases} 1 & \text{if } S_r^{[t,k]} > \delta \\ 0 & \text{otherwise.} \end{cases} \tag{7.9}$$

This, again, gives us a binary decision. However, as this decision is based on the ratio between current local energy and the recent minimum, it is adaptive to changes in intensity and type of noise. Further, the probability of $|Y|^2 \gg \lambda_d$ is very small when $S_r < \delta$, and thus a false decision on speech absence, i.e., $\mathcal{H}_0'$, when speech is actually present, i.e., $\mathcal{H}_1$ is fulfilled, only leads to an insignificant increase in the estimated noise [4]. Incorporating the strong correlation of speech presence in subsequent frames using a recursive average leads to the final, non-binary, estimation $\hat{p}_d$ of the conditional speech presence probability, given as

$$\hat{p}_d^{[t,k]} = \alpha_p \hat{p}_d^{[t-1,k]} + (1 - \alpha_p)\, I^{[t,k]}. \tag{7.10}$$

8 Evaluation

In this section, we will evaluate the performance of the short-time spectral amplitude estimation based methods discussed in Section 6, as well as the improved noise estimation approach provided in Section 7, and compare them to the spectral subtraction approach of Section 5.2. As both, estimation and spectral subtraction methods, are highly modular in terms of noise estimation, gain function or modifications to it, and choice of parameters, only selected combinations of what we found to perform best will be discussed. Furthermore, we will not discuss performance in terms of computational throughput, as none of the algorithms presented here poses a challenge to contemporary computing hardware, and, rather, solely focus on their auditory properties. We will first compare the maximum likelihood and decision-directed *a priori* SNR estimators, followed by MMSE, log-MMSE, and OM-LSA gain functions. Thereafter, we will look at the MCRA noise estimator, and finally compare the OM-LSA method using MCRA noise estimation with the spectral subtraction method using over-subtraction and spectral flooring.

Unless stated otherwise, we will again use the same parameters as in Section 5.2.2 for the STFT, i.e., a segment length of 20 ms, an overlap of half the segment length, and a square-root periodic Hann window, which will also be used for synthesis during the WOLA method. Again, all initial noise estimates are constructed by averaging the first nine spectral frames of the signal. Further parameters will be discussed within the individual evaluations.

We first compare the maximum likelihood and decision-directed *a priori* SNR estimators via both MMSE and log-MMSE gain functions. For the maximum likelihood approach, we chose $\alpha = 0.725$ and $\beta = 1.5$, for the decision-directed approach, we chose $\alpha = 0.98$, both similar to the suggestion by Ephraim and Malah [7]. In all cases, we use power-threshold noise estimation with a recursive average using $\alpha = 0.8$ and decision ratio $\delta = 0.8$. As we can see in Figure 10, the maximum likelihood estimator achieves significantly better noise reduction compared to the decision-directed estimator in combination with both, MMSE and log-MMSE gain functions, however, it also introduces strong musical noise. In some instances, this type of noise can even be more detrimental to the quality of the speech signal as perceived by humans than the original noise. The difficulty in removing musical noise, without introducing additional white-ish noise to mask it as done in spectral flooring, makes the decision-directed method preferable. To understand this difference, we need to look at the estimators as given in Equations (6.10) and (6.15). The maximum likelihood approach solely depends on the *a posteriori* SNR, whereas the decision-directed approach also depends on the value of the gain function for the previous frame, and can be seen as combining two separate estimators. This makes the decision-directed estimator arguably less susceptible to outliers in noise estimation. Together with the dependency on the gain function itself, which in turn incorporates the previous *a priori* SNR estimation and our best guess at the real clean signal magnitude, this improved robustness likely results in less musical noise, as it combats the typical per-analysis-frame peaks of spectral subtraction.

Next, we will look at the MMSE, log-MMSE, and OM-LSA gain functions themselves. For this, we will use both, power-threshold based noise estimation (Figure 11) and the MCRA noise estimator (Figure 12). While, for now, we focus only on the gain functions, the different noise estimation methods will be compared separately later. For power-threshold based noise estimation, we use the same parameters as above, i.e., $\alpha = 0.8$ and $\delta = 0.8$, to estimate the SNR, we use the decision-directed approach, again with $\alpha = 0.98$. To estimate the conditional speech presence probability for the OM-LSA gain, we use $\beta = 0.8$ for the recursive average, as well as two Hamming windows of size $w_{\text{local}} = 1$ and $w_{\text{global}} = 5$ for averaging over frequencies. Additionally, we use $\bar{\xi}_{\text{min}} = 1 \times 10^{-3}$ and $\bar{\xi}_{\text{max}} = 1 \times 10^{3}$ for the likelihood mapping, $\bar{\xi}_{\text{pmin}} = 1$ and $\bar{\xi}_{\text{pmax}} = 1 \times 10^{5}$ for the confined peak, as well as $q_{\text{max}} = 0.95$ to combat speech distortion. The minimal gain is chosen as $H_{\text{min}} = 0.05$. For MCRA noise estimation, we chose $\alpha_d = 0.95$ for the recursive average of the spectra. To estimate the conditional speech presence probability for MCRA, we chose $\alpha_s = 0.8$ for the recursive

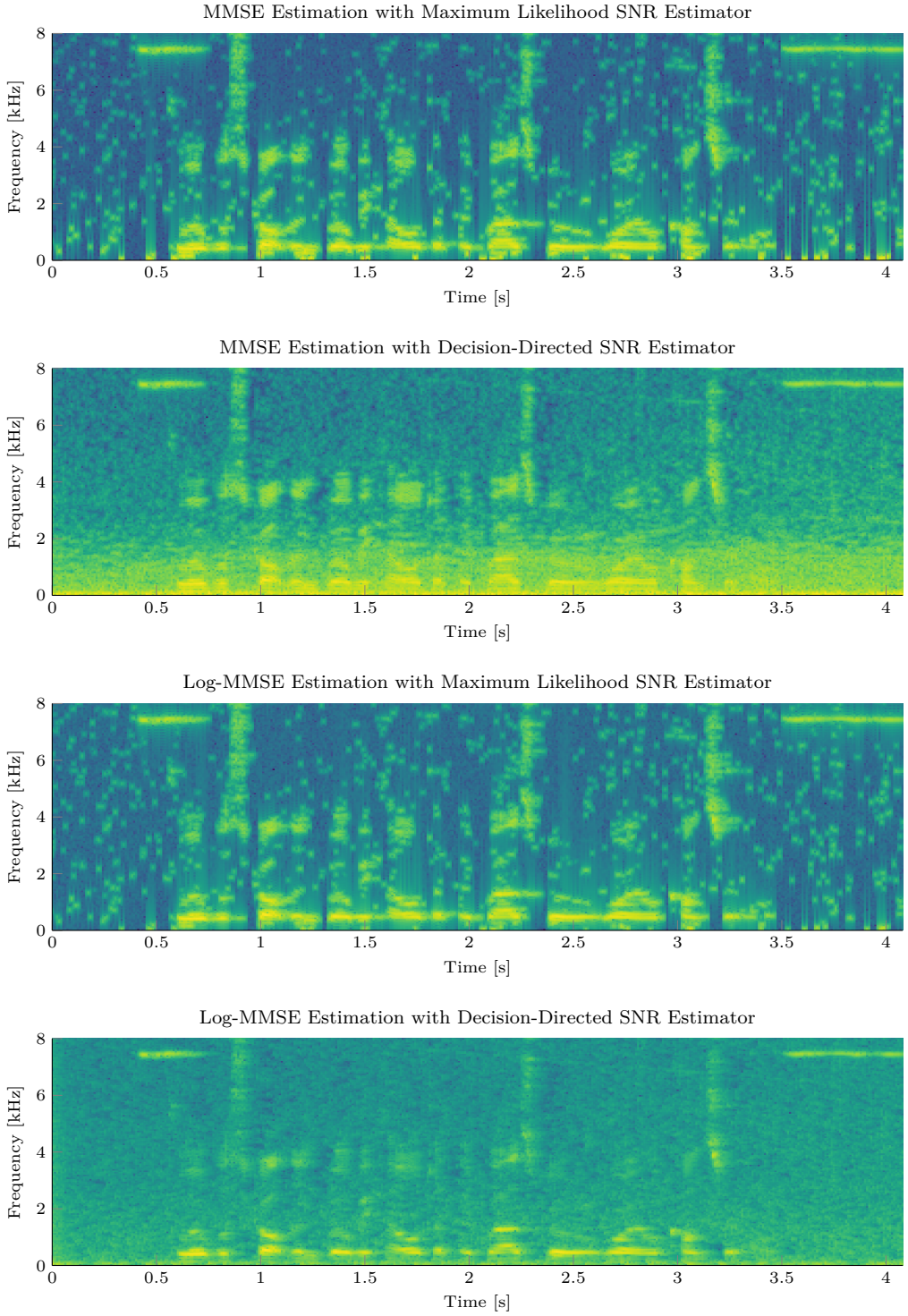

Figure 10: Maximum likelihood and decision-directed *a priori* SNR estimators in comparison. The decision-directed estimation approach shows improved robustness towards musical noise in combination with both MMSE and log-MMSE gain functions, however, has higher overall residual noise.

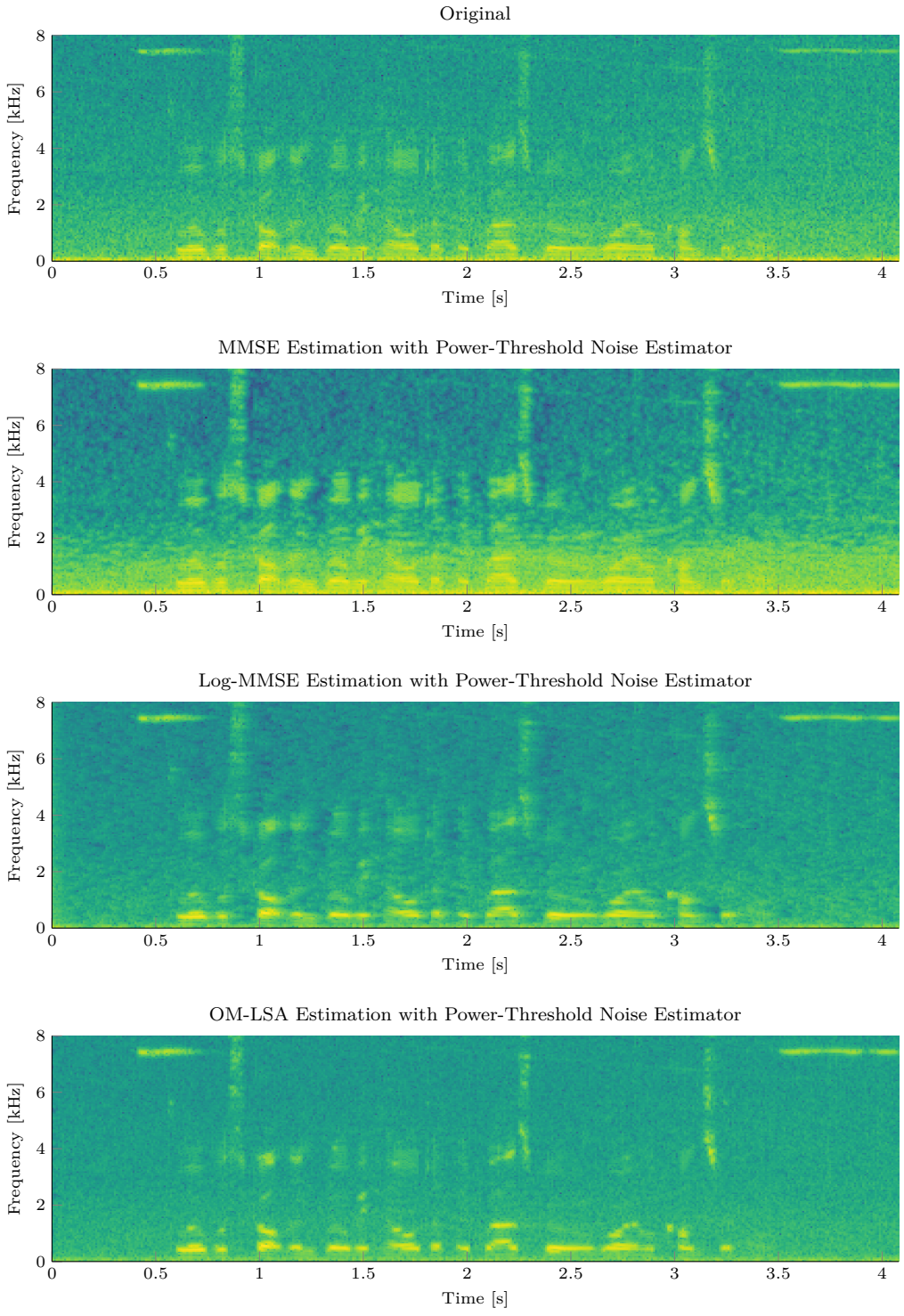

Figure 11: MMSE, log-MMSE, and OM-LSA gain functions with power-threshold based noise-estimation. The OM-LSA method shows superiority in performance and quality of the residual noise, although at the cost of some low-energy speech components removed.

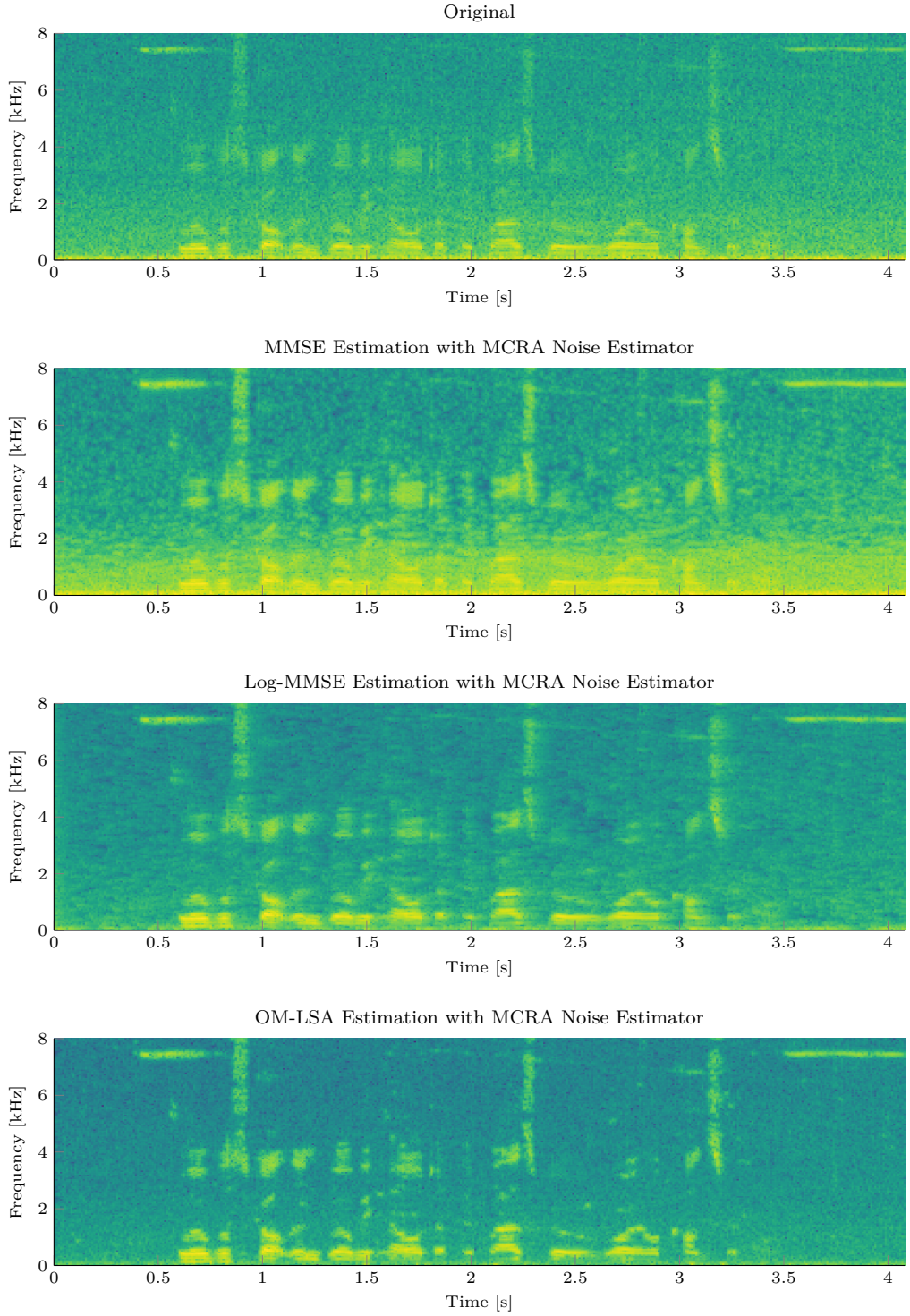

Figure 12: MMSE, log-MMSE, and OM-LSA gain functions with MCRA noise-estimation. The OM-LSA method shows superiority in performance and quality of the residual noise. In combination with MCRA noise estimation, even low-energy speech components remain intact.

average of the spectrum together with a Hamming window of size $w = 1$ for averaging over frequencies, a window length for minimum computation of $L = 45$, a decision ratio of $\delta = 5$ and $\alpha_p = 0.2$ for the recursive average to generate the probability estimate itself. In Figures 11 and 12, we can see that all methods achieve a significant reduction in noise, yet differ in characteristics of their residual noise. MMSE estimation achieves good results for mid and high frequencies, however, fails to remove noise below 2 kHz. Both, log-MMSE and OM-LSA methods improve upon this, due to their logarithmic error measure, and also provide good results for lower frequencies. Further, MMSE estimation results in a more speckled noise, reminding of musical noise with very low intensity. This is improved in log-MMSE estimation, where the noise assumes a more white-ish character, and even further in the OM-LSA method, where the remaining noise seems to be truly white. An interesting thing to note in Figure 11 is that the shadows trailing behind weaker speech segments, stemming from from noise estimation as previously noted in the evaluation of spectral subtraction methods in Section 5.2.2, are less present in the log-MMSE method and virtually invisible in the OM-LSA method. In terms of speech degradation, the MMSE algorithm influences speech the least, however, it also seems to keep parts that may not belong to the original speech signal, resulting in a somewhat blurry visual representation of speech components in the spectrogram. The log-MMSE method seems to achieve a better cut-off, with OM-LSA yielding the best result in this regard, however, both, anti-proportionally to this, also remove more speech components in higher frequencies. This is also audible when comparing audio outputs, however, other than voices sounding slightly more tinny, this does not affect the speech and its perceived quality.

Comparing Figure 11 with Figure 12 allows us to evaluate the performance of the MCRA noise estimator in relation to the power-threshold based one. When looking at the MMSE and log-MMSE gain functions, we can see that the MCRA noise estimate is more accurate as it does not cause shadows behind speech components, meaning that the noise estimate does not include (as much) parts of the speech, thus fulfilling one of its design goals. Further, MCRA seems slightly better at adapting to the rising noise intensity at the end of the audio clip being analyzed, which can be confirmed via auditory comparison. In combination with the OM-LSA gain function, it is capable of achieving a superior performance compared to other noise estimation methods, while at the same time also retaining more weak speech components and thus causing less degradation of the speech.

Finally, we can compare the results of spectral subtraction, already evaluated in Section 5.2.2, with the OM-LSA method. For this, we use spectral subtraction using over-subtraction and spectral flooring, combined with power-threshold based noise estimation as previously seen in Section 5.2.2, keeping the same choice of parameters, i.e., over-subtraction with $\alpha = 1.5$ and spectral flooring with $\beta = 0.066$. For power-threshold based noise-estimation, we again use $\alpha = 0.8$ and $\delta = 0.8$. With regards to the OM-LSA and MCRA methods, we also use the same parameters from previous evaluations, with decision-directed *a priori* SNR and conditional speech presence estimation as described above. Additionally, we combine spectral subtraction with the MCRA noise estimator for a thorough comparison. Figure 13 shows, that spectral subtraction and OM-LSA based speech enhancement methods can achieve similar reduction in noise. The major difference between the methods is the amount of speech distortion. When combining spectral subtraction with the MCRA noise estimator, distortion in speech is significantly reduced, however, very light musical noise is introduced. The OM-LSA method with MCRA noise estimator does not suffer the problem of musical noise and instead features the residual noise spectrum closest to pure white noise. Additionally, it further reduces speech distortion compared to spectral subtraction with MCRA noise estimation.

Neither of the methods and combinations is capable of removing the high pitched disturbances at start and end of the audio clip analyzed, as they are too atypical for noise with the high spectral power in comparison to previous frames and frequency bands, and thus recognized

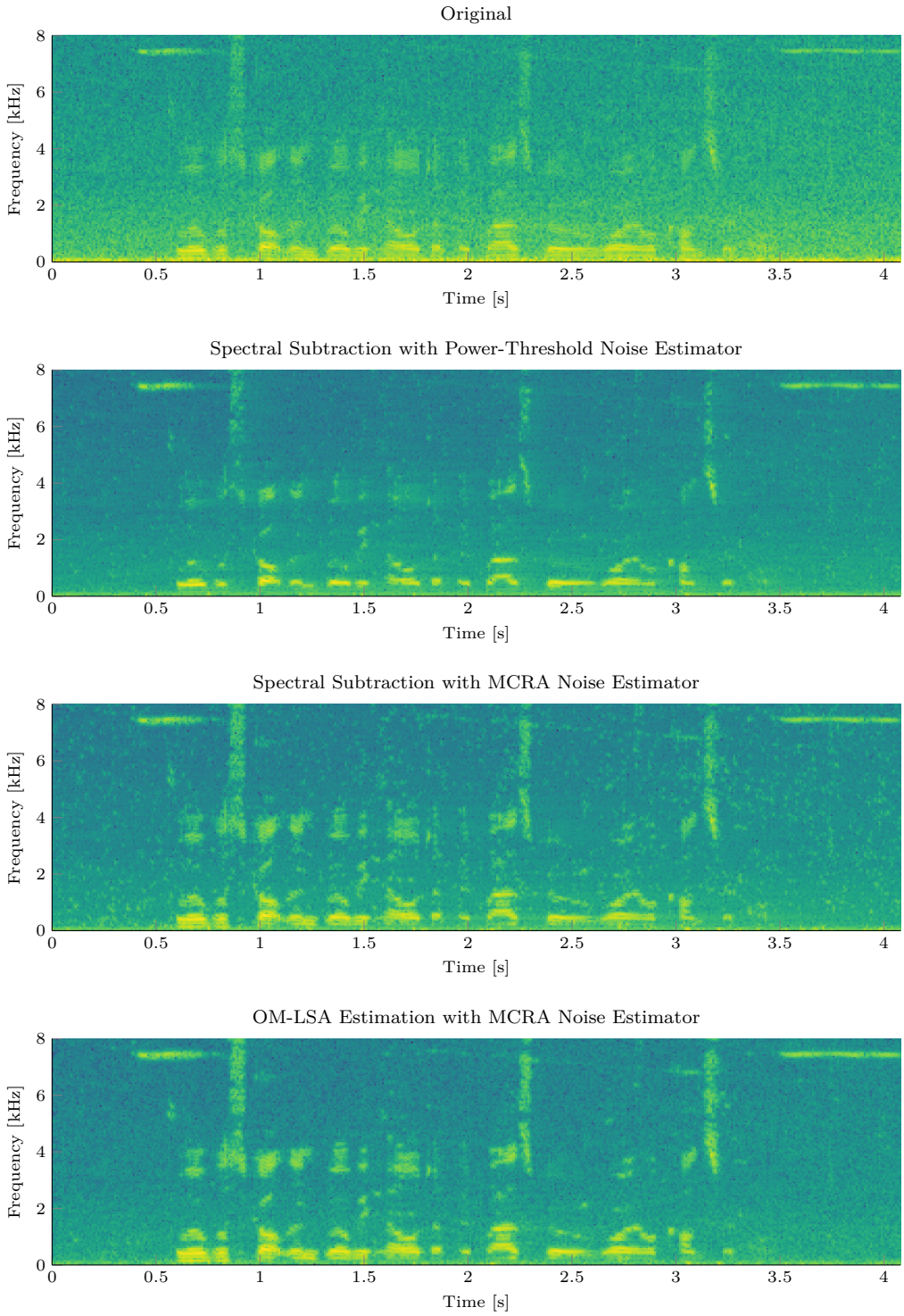

Figure 13: Comparison of spectral subtraction with OM-LSA using MCRA. The OM-LSA method is capable of achieving similar reductions in noise as spectral subtraction when combined with the MCRA noise estimator, however, shows superiority in terms of speech degradation and residual noise.

as voice in both, noise estimation and speech probability estimation processes. The best chance of removing these components in a single-microphone setting would be an MCRA approach with smaller window size for minima computation, which, however, will keep at least the beginnings of the disturbances and can introduce additional distortion of longer speech components. We found that, overall, the OM-LSA gain function in combination with the MCRA noise estimator and the parameters selected here provide the best results in terms of subjective quality of the resulting signal, auditory and according to the spectrogram, in comparison with the other methods presented here.

9 Conclusion

In this paper, we have looked at multiple methods for single-microphone, real-time capable speech enhancement, with the goal to find a method that is both adaptive to changes in noise, so that it can be used in environments with non-stationary disturbances, and performs well in terms of amounts of noise removed and speech degradation introduced. To this end, we have discussed the noise reduction problem in general (Section 2), together with different characteristics of noise, after which we presented an overview of speech enhancement techniques (Section 3). Thereafter, we looked at common methods used in real-time signal processing and speech enhancement (Section 4), including STFT, WOLA method, exponentially weighted (recursive) average, windowed extrema computation, and general technical considerations for this setting. While looking closer into noise reduction via the short-time spectral amplitude after this (Section 5), we discussed some general assumptions, basic noise estimation methods, and spectral subtraction as a simple yet effective algorithm. In this section, we also discussed the foundations of statistical estimation techniques, which we expanded on thereafter via MMSE based spectral amplitude estimation (Section 6), where we discussed various improvements to this method, specifically log-MMSE and OM-LSA, and estimation of the *a priori* SNR. Finally, we discussed MCRA as an improved noise estimation technique (Section 7) and performed an evaluation of the discussed methods (Section 8).

We found that both, spectral subtraction with over-subtraction and spectral flooring, as well as OM-LSA with MCRA noise estimator and decision-directed SNR estimator, provide good results in terms of noise reduced, whereas the OM-LSA method leads significantly in terms of retaining weaker speech components. Both methods achieve mostly colorless residual noise, leading to a subjectively good quality of the resulting signal, with no musical noise. Most major drawbacks of these methods, such as misclassifying anomalistic disturbances as speech instead of noise, can be related to the single-microphone approach, which requires statistically independent speech and noise signals. This limitation also leads to a generally bad performance on babble noise, which although not shown in the evaluation section, is, using OM-LSA gain with MCRA noise estimator and depending on the parameters, either largely preserved, or removed together with major parts of the original speech, due to the likeness of noise and clean speech signals. For many other, more regular yet potentially non-stationary, noise types, such as caused by cars, trains, machinery, wind, or waters, OM-LSA with MCRA shows reasonably good performance in low and medium SNR situations, however, some parameter adaptions may be needed.

Log-MMSE and OM-LSA methods, as well as MCRA noise estimation, form the basis of many modern speech enhancement techniques, and as such there are many improvements to them. For noise estimation, Cohen [3] proposed an improved MCRA (IMCRA) algorithm, which is based on two iterations of smoothing and minimum tracking. A similar strategy for noise estimation has been proposed by Rangachari and Loizou [20]. As improvement to the noise reduction part, Yuan and Xia [27] discussed improving OM-LSA by choosing a pre-defined parameter set based on the type of noise, determined via a classification algorithm. Jia et al. [13] on the other hand looked at using phase reconstruction in combination with a modified MMSE LSA estimation, based on the same binary hypothesis model as OM-LSA.

References

[1] M. Berouti, R. Schwartz, and J. Makhoul. "Enhancement of Speech Corrupted by Acoustic Noise." In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1979 (cit. on p. 15).

[2] J. Chen et al. "New Insights Into the Noise Reduction Wiener Filter." In: *IEEE Transactions on Audio, Speech and Language Processing* 14.4 (July 2006), pp. 1218–1234 (cit. on pp. 1, 5, 6).

[3] I. Cohen. "Noise Spectrum Estimation in Adverse Environments: Improved Minima Controlled Recursive Averaging." In: *IEEE Transactions on Speech and Audio Processing* 11.5 (Sept. 2003), pp. 466–475 (cit. on p. 40).

[4] I. Cohen and B. Berdugo. "Speech Enhancement for Non-Stationary Noise Environments." In: *Signal Processing* 81.11 (Nov. 2001), pp. 2403–2418 (cit. on pp. 2, 25, 29–33).

[5] S. Egger, R. Schatz, and S. Scherer. "It Takes Two to Tango - Assessing the Impact of Delay on Conversational Interactivity on Perceived Speech Quality." In: *Proceedings of the 11th Annual Conference of the International Speech Communication Association*. Jan. 2010, pp. 1321–1324 (cit. on p. 2).

[6] Y. Ephraim and D. Malah. "Speech Enhancement using a Minimum Mean-Square Error Log-Spectral Amplitude Estimator." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33.2 (Apr. 1985), pp. 443–445 (cit. on pp. 2, 6, 25, 28).

[7] Y. Ephraim and D. Malah. "Speech Enhancement using a Minimum-Mean Square Error Short-Time Spectral Amplitude Estimator." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32.6 (Dec. 1984), pp. 1109–1121 (cit. on pp. 2, 5, 6, 25–28, 34).

[8] M. Galassi et al. *GNU Scientific Library Reference Manual*. 2018. URL: `https://www.gnu.org/software/gsl/` (cit. on pp. 3, 28).

[9] R. Gray et al. "Distortion Measures for Speech Processing." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.4 (Aug. 1980), pp. 367–376 (cit. on p. 28).

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

[11] International Telecommunication Union. *One-Way Transmission Time. Series G: Transmission Systems and Media, Digital Systems and Networks*. Tech. rep. International Telecommunication Union, 2003 (cit. on p. 2).

[12] M. Jeub et al. "Noise Reduction for Dual-Microphone Mobile Phones Exploiting Power Level Differences." In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, Mar. 2012 (cit. on p. 1).

[13] H. Jia et al. "Speech Enhancement Using Modified MMSE-LSA and Phase Reconstruction in Voiced and Unvoiced Speech." In: *International Journal of Pattern Recognition and Artificial Intelligence* 33.02 (Oct. 2018), p. 1958002 (cit. on p. 40).

[14] N. Krishnamurthy and J.H.L. Hansen. "Babble Noise: Modeling, Analysis, and Applications." In: *IEEE Transactions on Audio, Speech, and Language Processing* 17.7 (Sept. 2009), pp. 1394–1407 (cit. on p. 5).

[15] X. Li, S. Gannot, and R. Horaud. "Blind MultiChannel Identification and Equalization for Dereverberation and Noise Reduction based on Convolutive Transfer Function." Nov. 2017. URL: `https://hal.inria.fr/hal-01568835` (cit. on pp. 3, 5).

[16] P. C. Loizou. *Speech Enhancement*. CRC Press, Feb. 2013 (cit. on pp. 1, 5–7, 14, 15, 18, 22, 23).

[17] N. D. Matsakis and Felix S. Klock I. "The Rust Language." In: *Proceedings of the 2014 ACM SIGAda Annual Conference on High Integrity Language Technology*. HILT '14. ACM, 2014, pp. 103–104 (cit. on p. 3).

[18] R. McAulay and M. Malpass. “Speech Enhancement Using a Soft-Decision Noise Suppression Filter.” In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.2 (Apr. 1980), pp. 137–145 (cit. on pp. 2, 6, 24).

[19] K. K. Paliwal and L. D. Alsteris. “On the Usefulness of STFT Phase Spectrum in Human Listening Tests.” In: *Speech Communication* 45.2 (Feb. 2005), pp. 153–170 (cit. on p. 14).

[20] S. Rangachari and P. C. Loizou. “A Noise-Estimation Algorithm for Highly Non-Stationary Environments.” In: *Speech Communication* 48.2 (Feb. 2006), pp. 220–231 (cit. on p. 40).

[21] J. O. Smith III. *Spectral Audio Signal Processing*. W3K Publishing, 2011 (cit. on pp. 8, 9).

[22] R. Talmon, I. Cohen, and S. Gannot. “Multichannel Speech Enhancement using Convolutive Transfer Function Approximation in Reverberant Environments.” In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, Apr. 2009 (cit. on p. 3).

[23] The Rust Project Developers. *The Rust Programming Language*. https://www.rust-lang.org/. Online; Accessed 01-August-2019. 2019 (cit. on p. 2).

[24] C. Valentini-Botinhao. *Noisy Speech Database for Training Speech Enhancement Algorithms and TTS Models*. https://datashare.is.ed.ac.uk/handle/10283/2791. University of Edinburgh. School of Informatics. Centre for Speech Technology Research (CSTR), 2016 (cit. on p. 17).

[25] X. Xiao and R. M. Nickel. “Speech Enhancement With Inventory Style Speech Resynthesis.” In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.6 (Aug. 2010), pp. 1243–1257 (cit. on p. 6).

[26] N. Yousefian and P. Loizou. “A Dual-Microphone Speech Enhancement Algorithm Based on the Coherence Function.” In: *IEEE Transactions on Audio, Speech, and Language Processing* (2011) (cit. on p. 1).

[27] W. Yuan and B. Xia. “A speech enhancement approach based on noise classification.” In: *Applied Acoustics* 96 (Sept. 2015), pp. 11–19 (cit. on p. 40).

[28] E. Zwicker. “Subdivision of the Audible Frequency Range into Critical Bands (Frequenzgruppen).” In: *The Journal of the Acoustical Society of America* 33.2 (Feb. 1961), pp. 248–248 (cit. on p. 5).