Lehrstuhl Kognitive Systeme
IESK
Otto-von-Guericke-Universität Magdeburg

Nuance Communications Deutschland GmbH
Zweigniederlassung Ulm

Diploma Thesis

Improved Noise Reduction for Hands-Free Communication in Automobile Environments

Ingo Schalk-Schupp

November 12, 2012

Examiners/Prüfer: Prof. Dr. rer. nat. Andreas Wendemuth
Prof. Dr.-Ing. Achim Kienle

Supervisors/Betreuer: Dr.-Ing. Mohamed Krini
Dr.-Ing. Markus Buck
Dipl.-Inf. Ronald Böck

Abstract

In this work, new single-channel extensions to existing speech enhancement methods in hands-free automobile communications are investigated.

Firstly, an algorithm for detecting formants based on power spectral density information in noisy speech signals is proposed. The formants localized in the frequency domain are described by a "boosting function" that can be arbitrarily used in subsequent processing steps. One such step introduced is the narrow-band modification of a recursive noise reduction filter. By placing hysteresis flanks in accord with the "boosting function", speech information is protected from over-attenuation in frequency-bands where formants are present. Another is the direct application of gain on the noise-reduced signal. A subjective test assesses the latter's merits.

Secondly, as part of a speech reconstruction module, an spectral envelope estimator for low frequencies is developed. The performance is evaluated with a distance measure from the clean speech signal.

Zusammenfassung

In der vorliegenden Arbeit werden einkanalige Erweiterungen zu bestehenden Verfahren zur Sprachsignalverbesserung in Automobil-Freisprecheinrichtungen untersucht.

Zum einen wird ein Algorithmus zur Detektion von Formanten vorgestellt, der auf der Leistungsdichte in rauschgestörten Sprachsignalen basiert. Die so im Frequenzspektrum lokalisierten Formanten werden mit einer „Boosting-Funktion" beschrieben, die in nachfolgenden Rechenschritten beliebig eingesetzt werden kann. Als ein solcher Schritt wird die Modifikation eines rekursiven Geräuschfilters, wie er im Stand der Technik eingesetzt wird, vorgeschlagen. Durch zielgerichtete Plazierung der Hysteresefanken anhand der „Boosting-Funktion" werden Sprachinformationen beim Vorliegen von Formanten frequenzselektiv vor zu starker Bedämpfung geschützt. Ein weiterer ist das direkte Aufprägen einer Verstärkung auf das Geräuschreduzierte Signal. Die Eigenschaften dieser Methode werden durch einen subjektiven Test beurteilt.

Zum anderen wird ein Schätzer für die Einhüllende des Amplitudenspektrums als Teil eines Sprachsignal-Rekonstruktionsmoduls entwickelt. Die Qualität wird anhand eines Distanzmaßes der Schätzung vom reinen Sprachsignal bewertet.

Statement of Authorship

I hereby declare that the work at hand has been composed by myself, and describes my own work. No further aids or resources were used unless otherwise stated.

All references and verbatim extracts quoted from other people's work have been specifically acknowledged by cross-referencing to the respective source.

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit eigenständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Alle Textstellen, die wortwörtlich oder sinngemäßanderen Werken oder sonstigen Quellen entnommen sind, habe ich unter Angabe der jeweiligen Quelle gekennzeichnet.

Aufgabenstellung

Zur Sprachkommunikation per Telefon werden häufig Freisprechsysteme eingesetzt, bei denen sich das Mikrofon in einiger Entfernung vom Mund des Sprechers befindet. Im Vergleich zu einer mundnahen Mikrofonposition wie beispielsweise bei einem Headset tritt ein deutlich schlechteres Signal-zu-Rauschverhältnis (SNR) auf. Zur Verbesserung der Signalqualität wird das gestörte Signal verarbeitet, bevor es zum Gesprächspartner weitergeleitet wird. Für einkanalige Systeme gibt es eine Vielzahl an etablierten Verfahren zur Geräuschreduktion. In der Regel schätzen diese schmalbandig das SNR und ermitteln darauf basierend zeitvariante Filtergewichte, die dann frequenzselektiv auf das Signal angewendet werden. Im Kraftfahrzeug wird das Störgeräusch meist von tiefen Frequenzen dominiert. Deshalb liegt dort oft ein sehr schlechtes SNR vor, was in der Praxis dazu führt, dass das Geräuschreduktionsfilter zu wenig öffnet. Das Ausgangssignal hört sich in solchen Fällen dann sehr dünn an, da die Tiefen im Sprachsignal fehlen.

Das Ziel dieser Arbeit ist, die Signalqualität von tieffrequent gestörten Sprachsignalen zu verbessern. Im Rahmen der Arbeit sollen zwei bis drei bekannte Standardverfahren zur Geräuschreduktion implementiert, untersucht und verglichen werden. Darauf aufbauend soll für eines dieser Verfahren eine Erweiterung entwickelt werden, die die tieffrequenten Sprachanteile vor zu starker Bedämpfung schützt. Zum Einen soll der Verlauf der Einhüllendenden des Sprachsignalspektrums in die Bestimmung der schmalbandigen Filtergewichte einbezogen werden. Zum Anderen sollen Möglichkeiten für eine Signalrekonstruktion untersucht werden, bei der die tieffrequenten Sprachsignalanteile künstlich erzeugt und mit dem originalen Ausgangssignal gemischt werden. Neben der Bewertung mit instrumentellen Bewertungsmaßen soll abschließend auch ein subjektiver Hörtest durchgeführt werden.

Der Anwendungsbereich soll sich auf einkanalige Systeme im Kraftfahrzeug beschränken. Als Programmiersprache soll MATLAB verwendet werden.

Task Assignment

Hands-free systems are commonly used in speech telecommunication systems. The microphone is usually placed at a certain distance from the speaker's mouth, resulting in a much lower signal-to-noise ratio (SNR) compared to a microphone position closer to the mouth, e.g. a headset. In order to improve the signal quality, the degraded signal is processed before being transmitted to the communication partner. For single-channel systems, there is a wide variety of well-established techniques for noise reduction. These methods generally estimate the SNR for narrow frequency bands and, based on the estimate, compute time-dependent filter weights, which are in turn applied to the signal's frequency bands. In car environments, the interfering noise signal is usually dominated by lower frequency components. This results in a very low SNR, which in practice prevents the filter from opening up sufficiently in these areas. In these cases, the resulting output signal sounds thin, because the lower frequency components are missing.

It is the goal of this work to improve the signal quality of speech signals degraded in the lower frequency domain. Within the scope of the work, two or three standard noise reduction methods shall be implemented, investigated, and compared. Based on this, an extension to one of these methods shall be developed that protects the lower-frequency speech signal components from being overly attenuated. As one part of the task, the course of the smoothed speech signal spectrum shall be taken into account for computing the narrow-band filter weights. As another part, the student shall explore speech signal reconstruction possibilities, in which the lower-frequency components are artificially generated and mixed with the output signal. Aside from an evaluation relying on instrumental rating measures, a concluding subjective listening test shall be performed.

The scope of application is restricted to single channel automobile systems. The programming language used for this work shall be MATLAB.

Contents

List of Figures

List of Tables

1. Overview

This diploma thesis is concerned with speech enhancement in an industrial environment focused on automobile embedded systems. In order to provide a better understanding of the conditions and current problems in this area, chapter 2 attends to the technical basis of speech audio processing relevant to this work.

Before the main part, chapter 3 treats the creation of a noise and speech database specifically created for the tasks at hand. Available databases are reviewed and the necessity for a new database is justified.

The first part of the main task, the use of spectral features for selective attenuation or amplification, is treated in chapter 4. The detection of speech formants is established and put to use in two separate ways. A subjective test evaluation concludes the chapter.

Chapter 5 describes the development of a speech signal envelope estimator at low frequencies. Several estimator concepts are investigated, optimized, and compared with an objective performance measure.

Finally, the results of this work are discussed and summarized in chapter 6, and perspectives on future work are given.

The appendix provides additional information for the interested reader.

2. State of the Art

In today's embedded speech signal enhancement (SSE) systems, audio data are either processed by a multipurpose microcontroller or by a specialized digital signal processor (DSP). In both cases, physical audio signals need to be converted to digital form first. This is commonly achieved through utilization of audio recording equipment integrating all the stages necessary for conversion: A microphone mechanically picks up sound waves from the medium (reception) and reactively generates an analog electric signal (transduction—Wong and Embleton). The electric signal, which can be a voltage, current or other, is then low-pass filtered to avoid aliasing in the sampling process. A sample and hold circuit performs time discretization on the signal. Finally, the signal is quantized in its amplitude domain and digitally coded for further processing (Wendemuth et al., p. 5). Depending on the hardware, the last two steps can be separate or joint.

There are many parameters that interdependently change the outcome of the conversion process. We will assume that numerical values linearly map to actual momentary pressure levels. Both the values and sampling points in time should be uniformly spaced. These assumptions are justified by the physical accuracy of modern sensors and by the actual setup in practical systems. The errors introduced by the transformation process are of minimal relevance for the challenges in this work.

This chapter is meant to give an overview of the basics underlying this work. As processing of audio signals mostly takes place in the frequency domain, the transformation is shortly defined. Common noise estimation algorithms as well as noise reduction filters are compared. Eventually, an acoustic speech production model is discussed, and a phenomenon called the Lombard reflex is introduced.

2.1. Short-Term Frequency Domain Representation of Audio Signals

For many applications in audio signal processing, operations are carried out in the frequency domain instead of the time domain, which is achieved by a short-term Fourier transform (STFT). In this process, the sequence of sampled amplitude values x in dependence of the sample index i is multiplied with a window sequence w for every Rth sample and then discretely transformed to the frequency domain. This step is called analysis, and its realization is often referred to as an analysis filter bank.

$$X(k,\mu) = \sum_{i=0}^{L-1} x(i+Rk)\, w(i)\, \mathrm{e}^{-\mathrm{j}\frac{2\pi\mu i}{N}} \tag{2.1}$$

Table 2.1.: Symbols used for integer quantities

Quantity	Count	Index
sample	I	i
frame	K	k
frequency bin	M	μ
formant	N_F	ν_F
frame shift	R	
window length	L	
DFT size	N	

The symbols used here, along with others, are summarized in table 2.1.

A selection of window sequences and their respective frequency representations are shown in figure 2.1. For better comparison between the sequences, the window length was set to $L = 64$.

After processing in the frequency domain, the resulting spectrum is transformed to the time domain again by an inverse STFT. In analogy to the previous step, this one is called synthesis, and its implementation is referred to as the synthesis filter bank.

2.2. Noise Estimation

Most modern speech enhancement algorithms apply a noise reduction filter. These filters try to optimally attenuate noise while retaining desired signal information. Usually, they rely on an estimate of the relative expected power of desired signal s to the noise signal's d power, i. e. the estimated SNR:

$$\widehat{\xi}\,(k,\mu) := \frac{\widehat{\Phi}_S(k,\mu)}{\widehat{\Phi}_{\widehat{D}}(k,\mu)}. \tag{2.2}$$

Regardless of whether such an estimator directly estimates the noise power $\widehat{\Phi}_{\widehat{D}}$ or instead the SNR, it is called a noise estimator. Figure 2.2 illustrates the signal flow in a noise reduction implementation including a noise estimator. Similar to the SNR estimate, the input-to-noise ratio (INR) estimate used is:

$$\widehat{\zeta}\,(k,\mu) := \frac{\widehat{\Phi}_X(k,\mu)}{\widehat{\Phi}_{\widehat{D}}(k,\mu)}. \tag{2.3}$$

An overview over signal symbols used in this work can be found in table 2.2.1.

2.2.1. General Approaches

This work assumes the presence of additive noise d in the noisy signal x. The clean speech signal is demarked with s:

$$x\,(i) = s\,(i) + d\,(i)\,. \tag{2.4}$$

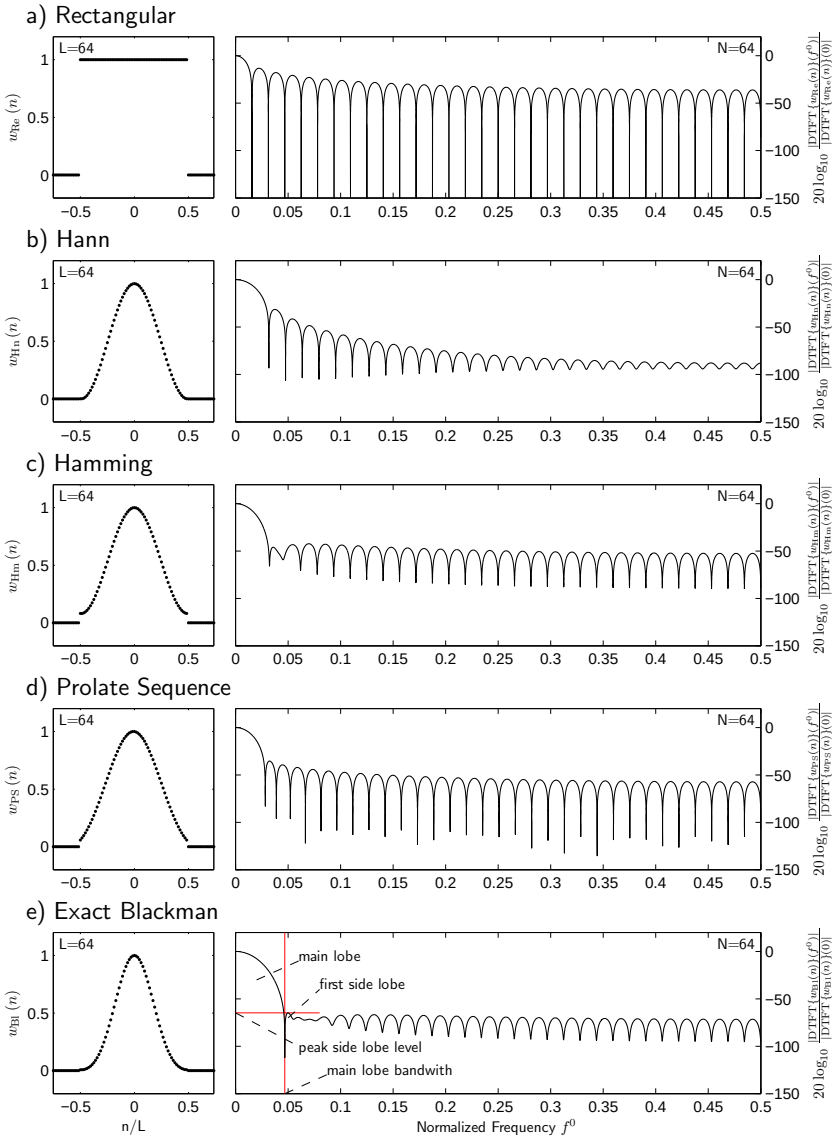

Figure 2.1.: Several windowing sequences of length $L = 64$. Left column: Window sequences in time domain, normalized to their length L, and centered around zero. Right column: DTFT frequency representation with $N = 64$. The spectra were normalized to their respective maxima. The Exact Blackman frequency representation features some factors useful in characterizing different window shapes.

Table 2.2.: Symbols used for different signals. For the function arguments used, see table 2.1.

Signal	Time Domain	Frequency Domain	True Power	Estimated Power
Clean Speech[1]	$s\,(i)$	$S\,(k,\mu)$	$\Phi_S\,(k,\mu)$	$\widehat{\Phi}_S(k,\mu) := \lvert S(k,\mu)\rvert^2$
Clean Speech Estim.	$\widehat{s}\,(i)$	$\widehat{S}\,(k,\mu)$	$\Phi_{\widehat{S}}\,(k,\mu)$	$\widehat{\Phi}_{\widehat{S}}(k,\mu) := \lvert\widehat{S}(k,\mu)\rvert^2$
Noise[1]	$d\,(i)$	$D\,(k,\mu)$	$\Phi_D\,(k,\mu)$	$\widehat{\Phi}_D(k,\mu) := \lvert D(k,\mu)\rvert^2$
Noise Estimate[2]	–	$\widehat{D}\,(k,\mu)$	–	$\widehat{\Phi}_{\widehat{D}}(k,\mu)$
Noisy Speech	$x\,(i)$	$X\,(k,\mu)$	$\Phi_X\,(k,\mu)$	$\widehat{\Phi}_X(k,\mu) := \lvert X(k,\mu)\rvert^2$

[1] The estimated power of noise and clean speech can only be calculated if the signals are actually known.

[2] The noise signal is never actually estimated, but only its power estimate. The plain symbol $\widehat{D}$ is still introduced because it is used in the power estimate's index.

According to Loizou (p. 401), noise estimation is based on three observations:

1. Silence occurs in speech pauses, in unvoiced fricatives at low frequencies, and in vowels at high frequencies.

2. In each frequency bin, the short-term power estimate drops to the noise power level.

3. Histograms counting the power level occurrences in any frequency bin often feature one or two distinctive modes.

Loizou (p. 403) relates these observations to three classes of noise estimators:

- minimal-tracking algorithms,
- time-recursive averaging algorithms, and
- histogram-based algorithms.

The algorithm used for all purposes in this work was IMCRA, which belongs to the second of the above classes, in an existing implementation.

An algorithm of the minimal-tracking class was implemented by the author as a preliminary part of this work for training purposes. In the following, the calculation steps are briefly described and the result is shown.

Smoothing over Time. As is the case in most noise estimators, the squared magnitude of the noisy speech signal serves as the basis for subsequent computations. A time-smoothed version $\widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{t}}}}$ of the noisy speech power spectrum $\widehat{\Phi}_X$ is used as an initial guess for the noise spectrum. The smoothing is implemented with a first-order

infinite impulse response (IIR) filter on the estimated microphone signal power $\widehat{\Phi}_X$ with discrete-time smoothing constant $\gamma_{\overline{\mathrm{t}}}$:

$$\widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{t}}}}(k=0,\mu) := \widehat{\Phi}_X(k=0,\mu) \tag{2.5}$$

$$\widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{t}}}}(k,\mu) := \gamma_{\overline{\mathrm{t}}}\widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{t}}}}(k-1,\mu) + (1-\gamma_{\overline{\mathrm{t}}})\,\widehat{\Phi}_X(k,\mu)\,. \tag{2.6}$$

The discrete-time smoothing constant $\gamma_{\overline{\mathrm{t}}}$ is given by:

$$\gamma_{\overline{\mathrm{t}}} = \exp\left(\frac{-R}{\tau_{\overline{\mathrm{t}}} f_{\mathrm{s}}}\right), \tag{2.7}$$

where R is the frame shift, $\tau_{\overline{\mathrm{t}}}$ is a chosen natural (continuous) decay time constant, and f_{s} is the sampling frequency.

Smoothing over Frequency. Next, the result $\widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{t}}}}$ is also smoothed in the frequency domain. Because IIR-smoothing slurs local features, it is applied once in forward direction:

$$\widehat{\Phi}'_{\widehat{D}^{\overline{\mathrm{f}}}}(k,\mu=0) := \widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{t}}}}(k,\mu=0) \tag{2.8}$$

$$\widehat{\Phi}'_{\widehat{D}^{\overline{\mathrm{f}}}}(k,\mu) := \gamma_{\overline{\mathrm{f}}}\cdot\widehat{\Phi}'_{\widehat{D}^{\overline{\mathrm{f}}}}(k,\mu-1) + \left(1-\gamma_{\overline{\mathrm{f}}}\right)\cdot\widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{t}}}}(k,\mu) \tag{2.9}$$

and once in backward direction:

$$\widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{f}}}}(k,\mu=M-1) := \widehat{\Phi}'_{\widehat{D}^{\overline{\mathrm{f}}}}(k,\mu=M-1) \tag{2.10}$$

$$\widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{f}}}}(k,\mu) := \gamma_{\overline{\mathrm{f}}}\cdot\widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{f}}}}(k,\mu+1) + \left(1-\gamma_{\overline{\mathrm{f}}}\right)\cdot\widehat{\Phi}'_{\widehat{D}^{\overline{\mathrm{f}}}}(k,\mu)\,. \tag{2.11}$$

The effective result is a second-order-IIR-filtered power spectrum $\widehat{\Phi}_{D^{\overline{\mathrm{f}}}}(k,\mu)$, which keeps local features in place.

Minimum Rising. Finally, the noise estimate should not rise too quickly after speech sets in, so it is bounded by a maximum rise constant ϵ:

$$\widehat{\Phi}_{\widehat{D}^{\mathrm{MR}}}(k,\mu) := (1+\epsilon)\cdot\min\left\{\widehat{\Phi}_{\widehat{D}^{\overline{\mathrm{f}}}}(k,\mu)\,,\widehat{\Phi}_{\widehat{D}^{\mathrm{MR}}}(k-1,\mu)\right\}. \tag{2.12}$$

2.2.2. Improved Minima-Controlled Recursive Averaging

A short history of the noise estimator used in this work will be given.

The improved minimum-controlled recursive averaging (IMCRA) method developed by Cohen is based on previous work by the same author (Cohen and Berdugo). The minimum-controlled recursive averaging (MCRA) class of noise estimators calculates probability measures for speech activity in every frame. The goal is to update the noise estimate in speech pauses, and to freeze the estimate during speech activity. The hypotheses of speech pause and activity are combined by their expected values.

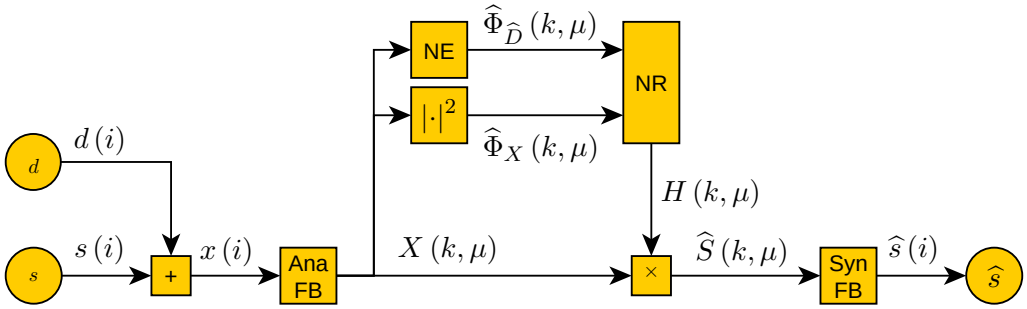

Figure 2.2.: Signal flow in a noise reduction system. "Ana FB" stands for an analysis filterbank, which performs windowing and an STFT. Analogously, "Syn FB" marks a synthesis filterbank, that performs an inverse STFT and sums the frames in the time domain. "NE" is the noise estimator and "NR" is the noise reduction algorithm.

While early versions still lagged on rising noise, this was improved in an algorithm called MCRA-2, which features a continuous mimimum search. Furthermore, the latter performs a more sophisticated speech activity probability calculation by dividing the frequency spectrum in three characteristic bands and analyzing their respective powers.

Finally, IMCRA makes use of a likelihood ratio approach in place of the former a posteriori SNR. Other than most noise estimators, it brings with it a bias compensation, releasing subsequent noise reduction algorithms from having to introduce a noise overestimation parameter.

2.3. Noise Reduction Filters

One of the most apparent tasks in speech audio signal processing is the elimination of additive noise from a degraded signal. This is usually done in the frequency domain using a setup including a noise estimator and a noise reduction filter. The general structure of a common noise reduction scheme is shown in figure 2.2.

2.3.1. Wiener Filter

Many noise-filtering applications use the Wiener filter as a basis. The Wiener filter is, in a minimum mean-squared-error sense, an optimal filter for suppressing additive noise on a signal assuming the power spectral density (PSD) of both signals is known, stationary.

Although these assumptions do not hold in general for speech signal, the application of a Wiener filter for noise reduction still provides good results.

The original formulation is based on statistical signals and their properties, and there is no requirement for the filter to be causal. In practical realizations for mobile telephony however, causality is mandatory because additional lag in the system's output is undesired. Moreover, not all of the signal properties are known. Therefore, a simplified formulation is used:

$$H_{\mathrm{W}} := \frac{\xi}{\xi + 1}, \tag{2.13}$$

where ξ is the true SNR:

$$\xi := \frac{\Phi_S}{\Phi_D} \tag{2.14}$$

The filter coefficients H_{W} are to be multiplied with the noisy signal, so that ideally the speech signal estimate $\widehat{S}'$ becomes:

$$\widehat{S}' := H_{\mathrm{W}} \cdot X. \tag{2.15}$$

The prime sign in $\widehat{S}'$ is used here to indicate the ideal Wiener filter coefficients.

Assuming that noise and speech signal be orthogonal, the sum of clean speech signal power Φ_S and noise power Φ_D constitutes the combined noisy signal power Φ_X:

$$\Phi_X = \Phi_S + \Phi_D. \tag{2.16}$$

Inserting into (2.13) the SNR definition (2.14) and (2.16), one gets:

$$H_{\mathrm{W}} \overset{(2.2)}{=} \frac{\Phi_S/\Phi_D}{\Phi_S/\Phi_D + 1} = \frac{\Phi_S}{\Phi_S + \Phi_D} \overset{(2.16)}{=} \frac{\Phi_X - \Phi_D}{\Phi_X - \Phi_D + \Phi_D} = \frac{\Phi_X - \Phi_D}{\Phi_X} = 1 - \frac{\Phi_D}{\Phi_X}. \tag{2.17}$$

Since the true noisy signal power Φ_S and the true noise signal power Φ_D are unknown, estimators are inserted, equation (2.13) transforms to a basic filter rule, which applies separately for each frame k and each frequency bin μ:

$$\widehat{H}_{\mathrm{W}}(k, \mu) := 1 - \frac{\widehat{\Phi}_{\widehat{D}}(k, \mu)}{\widehat{\Phi}_X(k, \mu)}, \tag{2.18}$$

so that the estimated speech signal $\widehat{S}$ becomes:

$$\widehat{S}(k, \mu) := \widehat{H}_{\mathrm{W}}(k, \mu) \cdot X(k, \mu). \tag{2.19}$$

Due to fluctuations in the noise estimation process, the estimated noise power $\widehat{\Phi}_{\widehat{D}}$ may occasionally be higher than the noisy signal's power $\widehat{\Phi}_X$, which would lead to a negative value for the filter coefficient. This is why it is bounded to positive values.

What is more, one problem of the Wiener filter is a phenomenon called "musical noise". It occurs in speech pauses when single frequency bins in a frame are only slightly attenuated as opposed to surrounding bins in time and frequency. These bins stand out well-audibly and irritate the listener. In order to make these bins stand

out less, a maximum allowed attenuation for all bins is introduced, which bounds the filter coefficients to become no lower than a positive value β.

Finally, a correction factor α for a biased noise estimator is introduced, called the overestimation factor.

Altogether, the practical Wiener filter with spectral floor and overestimation is defined by:

$$\widehat{H}_{\mathrm{W}}(k,\mu) := \max\left\{\beta, 1-\alpha\frac{\widehat{\Phi}_{\widehat{D}}(k,\mu)}{\widehat{\Phi}_{X}(k,\mu)}\right\}. \tag{2.20}$$

2.3.2. Recursive Wiener-Filter

Although the musical noise from the Wiener filter is audibly reduced by some amount through the introduction of the spectral floor, it is still present both audibly, and objectively. Linhard and Haulick have proposed an extension that counteracts this problem.

It is the basic idea to add to the Wiener filter a dynamic overestimation factor that depends on the filter coefficient in the previous frame. Thus, a recursive filter rule is constructed:

$$H(k,\mu) = \max\left(\beta, 1-\frac{\alpha}{H(k-1,\mu)}\cdot\frac{\widehat{\Phi}_{\widehat{D}}(k,\mu)}{\widehat{\Phi}_{X}(k,\mu)}\right). \tag{2.21}$$

A comparison of the Wiener filter's and the recursive Wiener filter's characteristics in dependence of the INR is depicted in figure 2.3.

The recursion together with the spectral floor leads to a hysteresis in the recursive filter's characteristic. This hysteresis divides the INR in three sections:

- At low INR below the hysteresis, the filter stays shut, that is, at the spectral floor.
- For high INR above the hysteresis, the filter approaches the Wiener filter.
- Between these, there is a transient interval.

In the transient interval, the characteristic depends strongly on the previous INR: Coming from higher values, a moderate attenuation is applied, while coming from lower INR values results in maximum attenuation.

This property is used effectively to eliminate musical noise, as the filter has a tendency to stay shut in speech pauses, while still opening very fast when speech sets in with a minimum INR.

Find a mathematically detailed discussion of the recursive Wiener filter in subsection 4.4.1.

2.4. Models of Speech Production

There are many models of the human speech generation process, ranging from ones aimed at preferably attaining a mathematical system accurately describing the speech

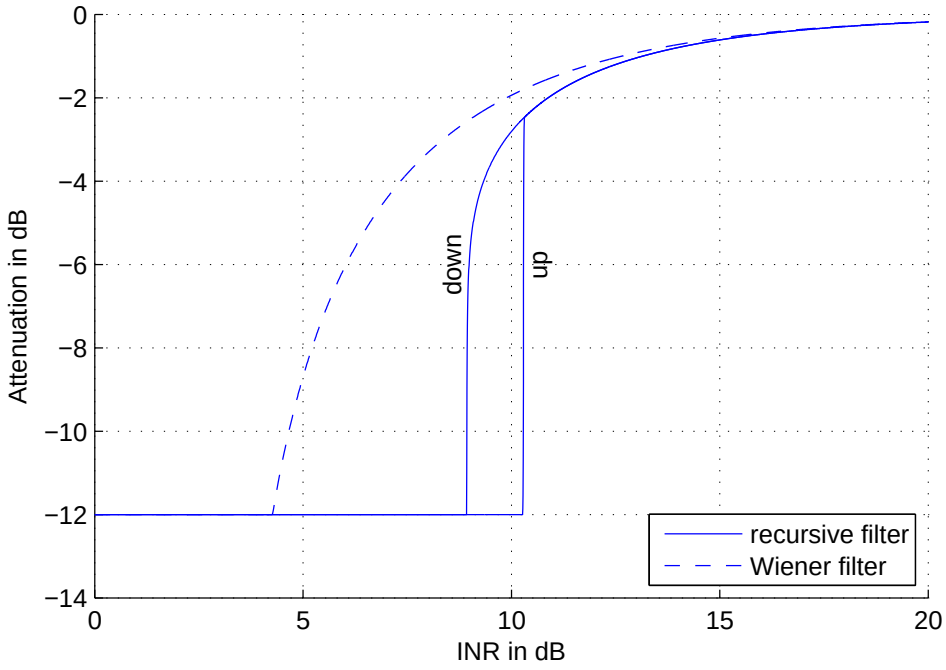

Figure 2.3.: Comparison of the Wiener filter with its recursive counterpart. The two filters' characteristics are shown as functions of the INR with overestimation factor $\alpha = 2$ and spectral floor $\beta = -12\,\mathrm{dB}$ for both filters. The hysteresis was captured by creating a phase plot of the recursive filter's output with slowly increasing and decreasing INR.

apparatus' physiognomy, to ones primarily trying to reduce the mathematical system's complexity. Examples on the complex end might simulate the physical mechanics of lungs, larynx, and the vocal tract (Quatieri) or even go as far as including fluid mechanics of turbulent air flow (Fant, p. 34), while models facilitating complexity might describe a speech signal as a sum of shifted prototype waveform impulses in the time domain.

For the purposes of this work, the so-called source-filter-model explained below will be assumed. Moreover, only voiced sounds are considered, because both the formant boosting and the reconstruction of voiced speech deal solely with these signal portions.

2.4.1. Voiced Excitation and Vocal Tract

Let us consider a periodic voiced speech signal x with zero bias, which can be represented as a Fourier series, that is a sum of weighted sine and cosine signals, such that for every integer p, the frequency of the p-th sine and cosine signal pair is a multiple of the so-called pitch frequency or fundamental frequency f_0, while a_p and b_p are the weighting coefficients:

$$x(t) := \sum_{p=1}^{\infty} \left(a_p \sin\left(2\pi p f_0 t\right) + b_p \cos\left(2\pi p f_0 t\right)\right) \tag{2.22}$$

By the identity

$$a \sin x + b \cos x = \sqrt{a^2 + b^2} \cdot \sin\left(x + \operatorname{sgn} b \cdot \arccos\left(\frac{a}{\sqrt{(a^2 + b^2)}}\right)\right), \tag{2.23}$$

(2.22) can be equivalently expressed as a sum of phase-shifted sine signals:

$$x(t) = \sum_{p=1}^{\infty} A_p \sin\left(2\pi p f_0 t + \phi_p\right) \tag{2.24}$$

with the amplitudes A_p and the phases ϕ_p given by:

$$A_p := \sqrt{a_p^2 + b_p^2} \tag{2.25}$$

$$\phi_p := \operatorname{sgn} b_p \cdot \arccos\left(\frac{a_p}{\sqrt{\left(a_p^2 + b_p^2\right)}}\right). \tag{2.26}$$

In this case, the amplitude A_p is always equal to or greater than zero. A negative sign would only result in a phase change by 180°. The form (2.24) is better-suited for considerations on amplitude and phase components in the frequency domain, as the contributing variables occur independently.

Signals that can be described by (2.22) or (2.24) are called harmonic signals. In a mathematically ideal amplitude frequency representation, they are described by

a family of singular Dirac distributions at the frequencies pf_0 and constant zero at all other frequencies. The actual digital amplitude frequency representation of a recorded harmonic signal, however, smears the theoretical peaks in the frequency domain depending on the window function used in the underlying STFT.

When modeling the voiced speech signal production process, the source-filter-model is a now-common approach proposed by Fant in 1960. In this approach, the vocal cords are regarded as a signal source, while the vocal tract acts as a filter on the signal generated by the vocal cords. The filter is thought not to have a feedback effect on the source. The source signal is also referred to as the excitation signal.

As mentioned above, an ideal excitation signal consists of discrete, uniformly spaced peaks in the amplitude spectrum. The pitch frequency is the frequency at which the lowest such peak occurs. As all other peaks' frequencies pf_0 are multiples of the pitch frequency f_0, the frequency difference between neighboring peaks coincides with the pitch frequency:

$$(p+1)\,f_0 - pf_0 = (p+1-p)\,f_0 = f_0. \tag{2.27}$$

As such, the placement of the pitch lines in the frequency domain do not bear any additional information beyond that provided by the pitch frequency itself f_0.

Excitation Signal. The source-filter-model demands that the unfiltered excitation spectrum remain static in some sense. That is to say that, aside from the overall signal power, the remaining degrees of freedom, namely the amplitude coefficients, should be attributed to the vocal tract. To achieve this, the excitation signal's coefficients $A_{p,\text{exc.}}$ are set to constant values (thus carrying no information) for any given pitch frequency f_0. They are then multiplicatively modified by the filter. Hence, the effective coefficients A_p are composed of the absolute sound pressure level A_{fix}, the excitation's contribution $A_{p,\text{exc.}}$, and the vocal tract's contribution $A_{p,\text{voc.}}$:

$$A_p := A_{\text{fix}} \cdot A_{p,\text{exc.}} \cdot A_{p,\text{voc.}}. \tag{2.28}$$

As a standard source model, a constant logarithmic slope of $q = -12\,\text{dB/octave}$ is often assumed (Fant, p. 272), although the value can vary. For example, a logarithmic slope of $q = -15\,\text{dB/octave}$ may be assumed for female speakers instead (Karlsson, p. 20). This fixes the relations between the excitation amplitude coefficients $A_{p,\text{exc.}}$ for all $p > 0$, as the coefficients come to lie on a straight line $A_{\text{cont.}}(f)$ in the logarithmic space. The vertical placement of this line is still undefined. To remedy this, define the absolute power level A_{fix} to also lie on the straight line (in the logarithmic domain) at an arbitrary given frequency f_{fix}.

$$A_{\text{cont.}}(f) := 10^{\frac{q}{20}\left(\frac{f}{f_{\text{fix}}}-1\right)+\log_{10} A_{\text{fix}}}. \tag{2.29}$$

All other coefficients $A_{p,\text{exc.}}$ and $A_{p,\text{voc.}}$ are now power values relative to A_{fix}. In this work though, power ratios are of interest in most cases, so the absolute value is canceled out. What remains for describing the characteristics of a speech utterance is the product of excitation and vocal tract contributions.

Vocal Tract. In the source-filter-model, the final change to the effective coefficients A_p is contributed by the vocal tract filter's coefficients $A_{p,\text{voc.}}$. Common vocal tract models produce one or several formants and anti-formants in their amplitude frequency spectrum. Formants are resonances of anti-resonances in the vocal tract that filter the excitation signal.

According to Wendemuth (p. 50), anti-formants are attributed to the nasal section and act as a band-stop filter. Formants on the other hand act as band-pass filters and thus accentuate a certain frequency band each and can be characterized by their respective central frequency, gain, and bandwidth. An ensemble of formants constitutes what humans perceive as vocals, which are an important part of speech information.

2.4.2. Lombard Effect

Junqua describes the so-called Lombard reflex of Lombard effect. According to his work, the effect is a reaction to different forms of stress leading to several changes in the speech signal spectrum including, among others, a higher fundamental frequency, a higher power level, and shifting of formants. The effect varies between subjects (Junqua, p. 16).

The one stress factor of particular interest for this work is the presence of noise. Driving noise in a car makes most people raise their voice according to the noise level. As speech characteristics can change drastically, this leads to a different noisy speech signal than mixing noise with a clean speech signal with no Lombard effect.

3. Signal Database

A signal database was created as part of this work for use in different processing and evaluation setups. Several databases with noise signals, speech signals, and noisy speech signals were available, including NOIZEUS (Hu and Loizou), the IEEE corpus (Rothauser et al.), SPEECON (Iskra et al.), and TIMIT (Garofolo et al.). Details on these can be seen in table 3.1.

In the following, several requirements on the signals will be defined, and the available databases will be accordingly reviewed. Eventually, the newly created database will be shortly described.

3.1. Goal

As the algorithms and methods researched in this work were to be put into operation in real-life car environments, they had to be applied to audio signals appropriate for this scenario. These signals should consist of human speech disturbed by typical in-car noise.

3.1.1. In-car Reverberation and Microphone Characteristics

In order to model operating conditions as exactly as possible, the full path of an audio signal from its production to its digitized form should be considered. This applies to spoken utterances as well as to noise signals. While the two types of signals differ considerably in their respective production process, they share a common path section as they both pass through the recording microphone and analog to digital converter.

Speech Signal Acquisition. In a typical hands-free car environment, speech utterances are subject to reverberation from the car cabin's interior faces through reflections (Kuttruff). This results in an audible sound modification. This is best described

Database Name	Clean Speech	Car Noise	Noisy Sp.	S×U[1)]	Remarks
NOIZEUS	from IEEE	N/A	8 kHz	6×5	
IEEE corpus	25 kHz	N/A	N/A	1×720	a)
SPEECON	N/A	16 kHz	16 kHz	75×35	
TIMIT	16 kHz	N/A	N/A	630×10	

Table 3.1.: Overview of available speech signal databases. [1)] S×U: number of speakers times number of utterances. Remarks: a) inconsistent recording conditions

by the played-back utterance sound as if spoken "in a box" and "far away" compared to the same utterance spoken in an anechoic environment (Naylor and Gaubitch).

Also, the recording equipment has a decisive impact on the resulting signals' quality. Typical sampling rates are 8 kHz for older car systems and 16 kHz for newer ones. In 8 kHz systems, many fricatives are lost, as their main power band lies above the corresponding Nyquist frequency. Altogether, the changes imposed on the original speech signal consist in information loss and as such, degradation.

One obvious approach to acquire a speech signal is to set up operating conditions and perform a recording. This is done by choosing a car equipped with appropriate recording hardware. This way, every utterance naturally traverses the car environment and thus is transformed the same way as in actual operation.

Given a high-quality clean speech signal database however, another approach comes up—the car environment and recording equipment can be modeled based on the measurement of an impulse response in a chosen car and transforming of the high-quality signals. This is usually done by assuming a linear transfer function for the physical signal path. A so-called artificial mouth is installed at the speaker's location in a car to produce test signals. These are recorded with the car's recording devices. Appropriate test signals can be frequency sweeps, white noise signals or maximum-length sequences (Rife and Vanderkooy). Both cover the whole frequency spectrum under considerations and are thus capable of producing accurate measurements for linear systems. While this kind of modeling provides good results for most recording systems, it fails to model non-linear systems. The reverberant car cabin features both linear and non-linear transfer properties.

Noise Signal Acquisition. It is desirable to test a speech enhancement algorithm in a variety of disturbance intensities. Results of such a test define an application range for the algorithm and provide a base for justifying its computational power cost. This is why acquiring car driving noise at different speeds is an asset.

Contrary to speech signals, noise recordings cannot easily be recorded in a high-quality way, as they do not emanate from a single localized source, but from many places in the car cabin at the same time. It would demand unreasonable effort to try and model a car's noise sources as well as their transfer paths to the microphone. Moreover, each car type differs in their noise pattern. Hence, noise measurements should be performed with the same setup that is used for acquiring speech signals by one of the above-mentioned approaches.

3.1.2. Separation of Noise and Speech Signals

Assessing an algorithm's performance usually includes computing some distance measure between the processed signal and either the clean speech signal or the pure noise signal without speech. Such measures are difficult to obtain if only the noisy speech signal is known. In this case, estimates for the separate speech and noise signals must be used, which might yield distorted results depending on the estimates' quality.

This is why noise and speech recordings were performed separately in the first place. As will be discussed in the following subsection, this approach entails additional technical difficulties as well as additional advantages.

3.1.3. Lombard Effect

Most SSE algorithms depend on the SNR rather than on any absolute signal power. This makes sense, because different audio equipment setups may lead to different digital representations due to varying transducer characteristics, digital resolution, and dynamic range.

When modeling realistic noise conditions, mixing speech with noise by merely adjusting the speech and noise signal powers to match a desired SNR neglects the Lombard effect mentioned in subsection 2.4.2, that specifically changes the power ratio of formant bands to the overall signal power. This is why another approach was chosen that provokes the Lombard reflex in the speakers.

Noise recording. In the first recording session, only car noise was recorded at different speeds via the car's built-in microphones as well as a "HEAD Acoustics NoiseBook" headset worn by the driver. The headset is equipped with microphones on the exterior sides of the ear cups and was used to provide a binaural noise image for later playback.

Speech recording. Secondly, several speaker were asked to read aloud several sentences while wearing the headset and listening to the driving noise recorded earlier. The headset's playback speakers come calibrated to provide the exact same listening experience as would be heard by the driver during the noise recording session. Thus, the Lombard effect was provoked in the speakers while retaining a quiet recording environment. Recording was performed with the same microphones as in the noise recording session.

3.2. Database Features

A signal database featuring in-car noise and appropriate Lombard speech at different speeds was recorded. The signals are provided in mono RIFF WAVE format. File names follow a concise naming scheme as defined in A.1 and A.2.

All recordings were performed in an Audi A6 car proprietary to Nuance. The database can be easily expanded with additional speakers, as long as the car hardware stays intact.

Data are suitable for evaluating speech enhancement algorithms and modules for realistic conditions. The mixed noisy speech signals resemble noisy speech signals much closer than would signals mixed from recorded clean speech with studio equipment, because transfer from the speakers' mouths through car cabin and recording equipment need not be simulated, which would introduce some amount of error. As

clean speech signals and noise signals are separately available, algorithms' performance may be tested assuming perfect noise estimation, thus decoupling the effects of serialized process units. Distance measures between estimated and known signals can be calculated.

The following is a summary of the database features at a glance:

- Noise signals:
  - 20 seconds of continuous stationary noise
  - in 5 different noise conditions
    (50, 80, 100, 130, and 160 km/h in-car noise)
  - recorded in-car for realistic transfer properties of cabin and recording equipment
  - available for 6 in-car microphones
  - and HEAD NoiseBook recordings for binaural playback to speakers.
- Speech signals:
  - 80 elicited German proverb utterances
  - spoken at the driver's seat,
  - recorded in-car for realistic transfer properties of cabin and recording equipment
  - with preceding and entailing speech pause of 1 second
  - and provoked Lombard effect, evenly distributed
  - over 2 male and 2 female speakers
  - and 5 different noise conditions as above,
  - resulting in 4 utterances per speaker and noise condition.
- Noisy speech signals:
  - All of the utterances were mixed with corresponding noise.

Use in this work. The database was successfully put to use in this work's investigations. A framework was built that automatically performs an operation with arbitrarily defined parameter combinations on a configurable set of data. Output filenames can be tagged according to actual parameter values. An example configuration can be found in appendix B.

Before creating the database, clean speech signals were mixed with noise, and the levels were adjusted to produce different SNRs. The resulting noisy speech signals sounded unnaturally disturbed. In comparison, speech signals with provoked Lombard effect sound much more credible when mixed with according noise.

Signal reconstruction performance was evaluated as described in section 5.4 based on a logarithmic distance measure. Clean signal information was used for parameter optimization.

4. Use of Formants for Speech Signal Enhancement

In this chapter, a novel approach on speech signal enhancement is introduced termed "formant boosting". The detection of formants in speech signals degraded by noise will be presented first, as it is the base for the subsequent processing stages. The next section describes the generation of a boosting function, which is a normalized representation of the presence of formants in the spectrum. The boosting function is used in two separate ways, the first of which purposefully modifies a recursive noise reduction filter in formants, while the second adds additional gain to formant bands in a noise reduced signal. The latter method is finally evaluated in a subjective listening test.

4.1. Statement of Problem

Common noise reduction algorithms make assumptions to the type of noise present in a noisy signal. The Wiener filter for example introduces the mean squared error (MSE) cost function as an objective distance measure to optimally minimize the distance between the desired and the filtered signal (Loizou, p. 143). The MSE however does not account for human perception of signal quality. Also, filtering algorithms are usually applied to each of the frequency bins independently. Thus, all types of signals are treated equally. This allows for good noise reduction performance under many different circumstances.

However, mobile communication situations in an automobile environment are special in that they contain speech as their desired signal. The noise present while driving is mainly characterized by increasing noise levels with lower frequency.

With knowledge about the characteristics of speech signals, more specialized assumptions can be made to provide for strategies better fitted to the noise reduction problem.

The novel filter extension exploits additional information from the noisy speech signal in order to improve noise filtering compared to conventional methods.

This chapter describes several techniques for the detection of formants and how the information is passed on to following processes in the form of a normalized "boosting function". Two methods applying the boosting function to modify noise reduction filter characteristics are then investigated the first of which aims to selectively modify the hysteresis characteristics of a recursive noise reduction filter in formants, while the second one directs additional gain on the noise reduced signal. The performance of the latter is evaluated in a subjective test.

4.2. Detection of Formants

Prerequisite to all formant based speech signal improvement is the correct detection of formants present in the speech signal. As mentioned in subsection 2.4.1, vocals in speech are formed by a combination of formants.

As stated in subsection 2.4.1, formants are frequency portions of a signal in which the excitation signal was amplified by a resonance filter. This results in a higher PSD compared to the excitation's PSD around any formant's central frequency and also compared to neighboring frequency bands, unless another formant is present there. Assuming that besides the vocal tract formants, no other significant formants are present (e.g. strong environment resonances), we can hope to identify formants by finding locally high PSD bands.

4.2.1. Constraints

Not all locally high PSD bands are to be identified with formants. Especially fricatives, that is speech with an unvoiced excitation, should not be taken into account. Neither should any boosting take place in frames without voice activity.

Frequency Band. In order to avoid boosting fricatives, a frequency band restriction for the detection of formants was introduced at $f_{\mathrm{F,max}} = 3500\,\mathrm{Hz}$. By reducing the number of relevant frequency bins, this restriction further reduces the computational complexity of the detection process.

Voiced Excitation Detection. In place of a standard voice activity detection, a voiced-excitation detection algorithm is applied before formant detection takes place in a frame. This is done by deciding whether the mean logarithmic INR $\widehat{\zeta}$ over a number $M_{\mathrm{F}} = \mu_{\mathrm{F,high}} - \mu_{\mathrm{F,low}} + 1$ of frequency bins,

$$P_{\mathrm{VUD}}(k) = \frac{1}{M_{\mathrm{F}}} \sum_{\mu=\mu_{\mathrm{F,low}}}^{\mu_{\mathrm{F,high}}} 10 \log_{10} \widehat{\zeta}(k,\mu)\,, \tag{4.1}$$

exceeds a certain threshold P^*_{VUD}:

$$\mathrm{VUD}(k) = \begin{cases} \text{true} & \text{for } P_{\mathrm{VUD}}(k) > P^*_{\mathrm{VUD}} \\ \text{false} & \text{otherwise.} \end{cases} \tag{4.2}$$

In the experiments, a value of $P^*_{\mathrm{VUD}} = 1\,\mathrm{dB}$ was found to be suitable for discrimination of voiced and unvoiced or no speech activity in a frequency band between $f_{\mu_{\mathrm{F,low}}} \approx 300\,\mathrm{Hz}$ and $f_{\mu_{\mathrm{F,high}}} \approx 2000\,\mathrm{Hz}$.

As this algorithm computes logarithms for M_{F} bins, it is quite costly and may be replaced by a cheaper algorithm at the expense of robustness: Exchanging the sum and the logarithm decreases the computation power needed, but no satisfactory detection performance was attained.

In Nuance's proprietary SSE however, logarithmic values for the INR are calculated anyway, so there is no additional cost involved when implementing the above algorithm on this system.

Clearance Between Formants. A minimum clearance between formants is enforced to avoid excessive overlapping. Still, some overlapping is tolerated to account for very wide formants to be detected as two separate ones that overlap. A minimum distance between the formants' central frequencies of 600 Hz was deemed useful and was used throughout this work.

4.2.2. Linear Predictor

Using a linear predictor to model the vocal tract's transfer function is a common approach in speech recognition and has been successfully applied since the 1970's (Wendemuth, p. 49). In this work, a linear predictor as used in estimator of order 15 was used as a reference method for the detection of formants. Such predictors are common use in linear predictive coding (LPC) and will in this work often be abbreviated as "LPC".

Since the minimization of the prediction error takes place in the time domain and results in a linear filter realization, it is necessary to compute its frequency representation, which involves a costly discrete Fourier transform (DFT) transform. This method becomes infeasible especially in embedded systems, so other methods were investigated as a replacement.

4.2.3. IIR Smoothed Amplitude Spectrum

The linear predictor mentioned above leads to a fair estimation of the formant's location and gain values. While the cost of determining the linear model's coefficients grows linearly with their desired number (Wendemuth, p. 50), finding the local maxima in the model's resulting amplitude spectrum requires transformation to the frequency domain. This transformation is relatively costly. Hence, computationally more efficient methods were sought while preserving the reference LPC model's estimation quality.

Note the following observations:

- Typical car noise has an approximately linear power decay of $-14\,\mathrm{dB/oct}$. See section 3.2.
- Speech formant's bandwidths are much higher than those of the excitation's harmonics. See subsection 2.4.1.
- The central frequencies of higher formants lie above the fundamental (pitch) frequency. This does not necessarily hold for the first formant in high-pitched voices like those of children or female speakers.

With this knowledge, it becomes obvious that a plain search for the noisy speech spectrum's local maxima is not suitable for formant detection. If one tried this approach, one would detect most harmonics and some disturbances, whether or not they were the result of a formant.

To eliminate the problem of the harmonic's maxima masking the superposed formants' ones, a lower frequency resolution is desired. Since the transformation parameters in embedded systems are fixed, the high resolution amplitude spectrum must be processed.

Smoothing the amplitude spectrum by applying a first-oder IIR filter is a very cost-efficient approach in terms of computational complexity. However, one has to find a trade-off smoothing constant that on the one hand provides an adequate attenuation of the harmonics' effects and on the other hand does not cancel out any formants' maxima.

Define the number of relevant frequency bins M_F as the number of bins in the formant band up to the last bin $\mu_{\mathrm{F,max}}$:

$$M_\mathrm{F} := \mu_{\mathrm{F,max}} + 1. \tag{4.3}$$

Similar to (2.8), the proposed filter with smoothing constant γ_F is applied once in forward direction

$$\overline{S}'_\mathrm{F}(k, \mu = 0) := |X(k, \mu = 0)| \tag{4.4}$$

$$\overline{S}'_\mathrm{F}(k, \mu) := \gamma_\mathrm{F} \cdot \overline{S}'_\mathrm{F}(k, \mu - 1) + (1 - \gamma_\mathrm{F}) \cdot |X(k, \mu)| \tag{4.5}$$

and once in backward direction:

$$\overline{S}_\mathrm{F}(k, \mu = M_\mathrm{F} - 1) := \overline{S}'_\mathrm{F}(k, \mu = M_\mathrm{F} - 1) \tag{4.6}$$

$$\overline{S}_\mathrm{F}(k, \mu) := \gamma_\mathrm{F} \cdot \overline{S}_\mathrm{F}(k, \mu + 1) + (1 - \gamma_\mathrm{F}) \cdot \overline{S}'_\mathrm{F}(k, \mu)\,. \tag{4.7}$$

so as to keep local features in place. Here, instead of the power values, the amplitudes of the complex microphone signal $|X|$ are smoothed.

With the given transformation parameters (sampling frequency $f_\mathrm{s} = 16000\,\mathrm{Hz}$ and window width $N = 512$) a good compromise numerical smoothing constant was found to be $\gamma_\mathrm{F} = 0.92$. This corresponds to a natural decay constant of

$$\gamma'_\mathrm{F} = \frac{N}{f_\mathrm{s}} \ln \gamma_\mathrm{F} \approx -2.668 \cdot 10^{-3}\,\mathrm{s} \mathrel{\widehat{=}} -11.588\,\frac{\mathrm{dB}}{\mathrm{kHz}} \tag{4.8}$$

for arbitrary STFT parameters. The STFT-dependent smoothing constant can then be calculated using:

$$\gamma_\mathrm{F}(N, f_\mathrm{s}) = \mathrm{e}^{\frac{f_\mathrm{s}}{N}\gamma'_\mathrm{F}}\,. \tag{4.9}$$

After smoothing the PSD, the local maxima are determined by finding the zeros of its discrete derivative. Streaks of zeros are consolidated, and an analysis of the second derivative is used to classify minima, maxima, and saddle points.

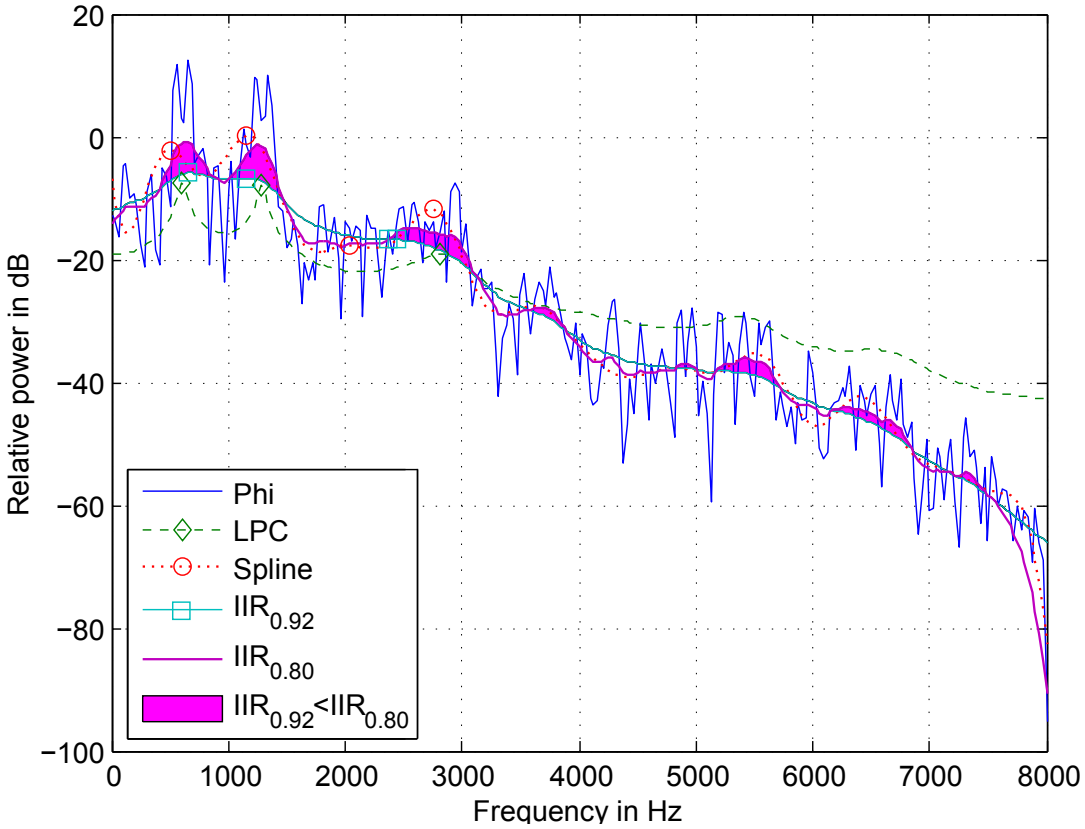

Figure 4.1.: Comparison of spectral smoothing methods for formant detection. Markers indicate formant centers. Formants are only accepted below $f_{\mathrm{F,max}} = 3500\,\mathrm{Hz}$.

As can be seen in figure 4.1, the smoothing provides the desired information. However, in frequency bands where the noisy speech PSD is flat, remnants of the harmonics still result in the detection of undesired local maxima. This can be attributed to the IIR filter's inherent information preserving property—arising from its infinite impulse response. The harmonics are attenuated, but not enough for their maxima to disappear in the smoothed spectrum's slope.

4.2.4. Spline

As another method for smoothing, the fitting of a piecewise cubic spline curve was investigated. A MATLAB script called SPLINEFIT (Lundgren) was used to fit spline curves to the noisy speech signal's logarithmic power spectrum $10\log_{10}\widehat{\Phi}_X$ with $N_{\mathrm{spline}} = 20$ pieces:

$$\overline{S}_{\mathrm{spline}}(k) := \texttt{splinefit}\left(1\ldots M, 10\log_{10}\widehat{\Phi}_X(k), N_{\mathrm{spline}}\right). \tag{4.10}$$

The resulting maxima are very similar to that of the LPC smoothing, but it was found that an order of 20 is necessary to achieve results comparable to those of an LPC with order 15. In this case, the computing time needed is about 75 % higher according to the MATLAB profiler. Therefore, this approach was rejected.

4.2.5. Fast and Slow IIR Smoothing

The final method for formant detection is done by applying two IIR filters (see (4.4)) with different smoothing constants on the microphone spectrum and comparing their magnitudes. In figure 4.1, the filled areas mark frequencies where the fast filter's output exceeds the slow filter's one. The smoothing constants are $\gamma_{\text{fast}} = 0.80$ and $\gamma_{\text{slow}} = 0.92$ as noted in the legend.

While retaining a low computational complexity, this features one specific advantage over the others in that it provides information about the formants' widths.

Additional requirements can be defined to any presumed formant's properties. A minimum area threshold ensures that the formant is not sporadic. Too narrow a formant is likely the result of a singular excitation peak in the spectrum; it can be ruled out by a minimum formant width.

This method performed well in first impression and has been implemented in the proprietary Nuance SSE. Additional optimization is recommended, as this method was not used in the subjective tests.

4.3. Generation of a Boosting Function

In the following, let $N_{\text{F}}(k)$ be the number of formants detected in frame k, and $\nu_{\text{F}} \in \{1, \ldots, N_{\text{F}}(k)\}$ the according index variable. After detection of any formants, information is available about their central frequency $f_{\text{F}}(k, \nu_{\text{F}})$ and—in the case of fast and slow smoothing, see subsection 4.2.5—about their respective widths $\Delta f_{\text{F}}(k, \nu_{\text{F}})$.

It is the tentative goal to create a boosting function $B(k, f)$ with codomain $[0, 1]$, where a value of 0 should represent the absence of any formants at the respective frequency, while a value of 1 should mark a formant's center. These values will later be used to modify the noise reduction filter's behavior. Since the actual effect is not yet determined at this stage in the algorithm, one can think of 0 as "no effect", while 1 means "full boosting".

4.3.1. Boosting Window

We introduce the prototype boosting window function $b_{\text{prot}}(x) : \mathbb{R} \to [0, 1]$ with

$$b_{\text{prot}}(x) := \begin{cases} \widetilde{b}_{\text{prot}}(x) & \forall x \in \left[-\frac{1}{2}, \frac{1}{2}\right], \\ 0 & \text{otherwise}, \end{cases} \tag{4.11}$$

where $\widetilde{b}_{\text{prot}}(x) : \left[-\frac{1}{2}, \frac{1}{2}\right] \to [0, 1]$ defines the actual prototype window shape. Example window functions are depicted in figure 4.2.

In this work, the gaussian window function was used for all boosting applications.

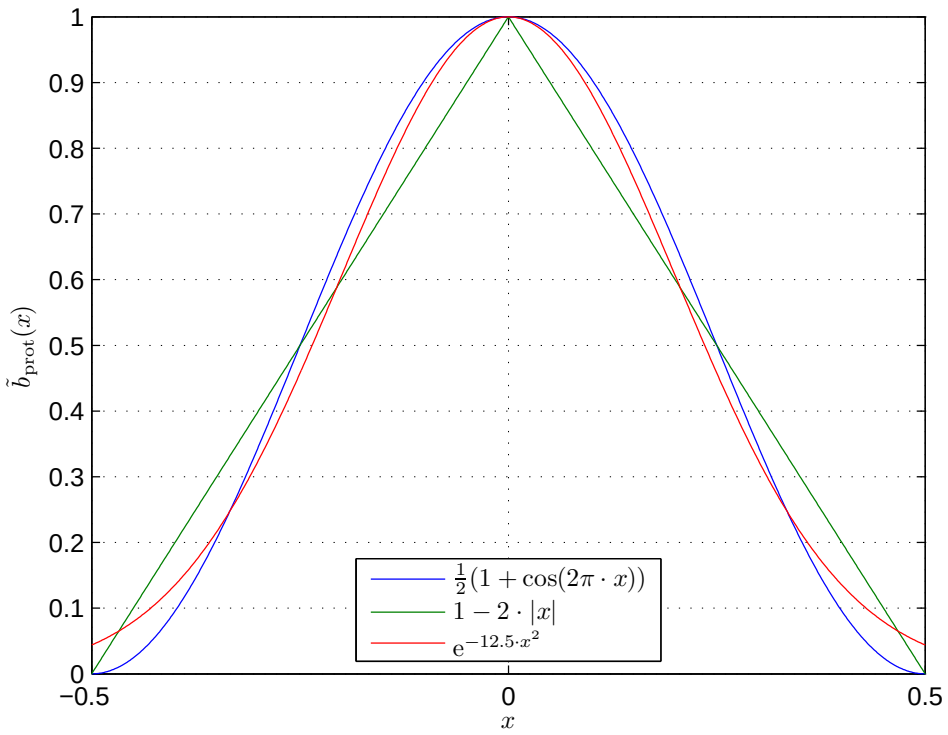

Figure 4.2.: Example prototype boosting window functions centered around zero with unity width. A cosine, a triangle, and a Gaussian window are shown.

Shape. Within any formant, the highest SNR can be expected at its center, so the risk of introducing noise by boosting the signal increases towards formants' borders. This is why a typical boosting around a formant's center should fall of gently. In this work, a Gaussian function has been used as a prototype boosting window function to implement this requirement. Other prototype window functions could be used as well, e.g. cosine or triangular functions. As can be seen from (4.11), they are assumed to be centered around $x = 0$ and have unity width.

Width. For each formant detected, the prototype window function is stretched by a factor $w(\nu_\mathrm{F}, k)$ to match the formant's width, if it is known—as is the case for the approach with fast and slow smoothing. Otherwise, it should be stretched to a decent constant frequency width w_fix of about 600 Hz:

$$\forall \nu_\mathrm{F} : \; w(\nu_\mathrm{F}, k) := \begin{cases} \Delta f_\mathrm{F}(\nu_\mathrm{F}, k), & \text{if known,} \\ w_\mathrm{fix} & \text{otherwise.} \end{cases} \tag{4.12}$$

Location. Finally, the boosting window is shifted by the formant's central frequency.

4.3.2. Boosting Function

The boosting function is defined to be the sum of the stretched and shifted boosting window functions:

$$B(f, k) := \sum_{\nu_\mathrm{F}=1}^{N_\mathrm{F}(k)} b_\mathrm{prot}\left(\frac{f - f_\mathrm{F}(\nu_\mathrm{F}, k)}{w(\nu_\mathrm{F}, k)}\right). \tag{4.13}$$

This sum could potentially result in values for $B(f, k) > 1$. However, this is avoided by adjusting the minimum clearance between formants mentioned in subsection 4.2.1, so that the sum of neighboring formants cannot exceed 1.

Since practical application takes place in the STFT space, only frequencies corresponding to frequency bins' centers are to be considered:

$$B(k, \mu) = \sum_{\nu_\mathrm{F}=1}^{N_\mathrm{F}(k)} b_\mathrm{prot}\left(\frac{f\mu - f_\mathrm{F}(\nu_\mathrm{F}, k)}{w(\nu_\mathrm{F}, k)}\right) \quad \forall \mu \in \left\{0, \ldots, \tfrac{N}{2}\right\}. \tag{4.14}$$

4.4. Application of the Boosting Function

In each frame, the boosting function $B(k, \mu)$ with its codomain of $[0, 1]$ can be arbitrarily used to modify any filter parameter by applying an appropriate transformation. This allows for the filter to be responsive to the presence of formants and thus potentially improve signal quality compared to conventional methods.

In this work, the boosting function was applied in two different ways. The first approach modifies the recursive Wiener filter's input to provoke admission of formants. The second approach changes the same filter's output, allowing the resulting filter coefficients to grow beyond the former upper boundary of 0 dB.

4.4.1. Modified Recursive Wiener Filter

As mentioned in subsection 2.3.2, Linhard and Haulick developed a recursive gain function that results in a multistable filter system with a hysteretic characteristic. This extension effectively counteracts the problem of musical noise.

It is one of the tasks stated in this work's task assignment to protect the noisy speech signal from being overly attenuated. This idea inherently bears the risk of introducing noise artifacts into the filtered signal compared to a conventionally filtered signal. However, taking this risk is justified if speech intelligibility can be sufficiently improved.

It will be shown that by further analyzing the recursive Wiener filter's underlying structure, one can use the boosting function to freely choose the position of the filter's hysteresis flanks.

Detailed Analysis of the Recursive Wiener Filter. The recursive Wiener filter's system function as given by Linhard and Haulick is

$$H(k,\mu) = \max\left(\beta, 1 - \frac{\alpha}{H(k-1,\mu)} \cdot \frac{\widehat{\Phi}_{\widehat{D}}(k,\mu)}{\widehat{\Phi}_X(k,\mu)}\right), \tag{4.15}$$

where α is the overestimation factor, and β is the spectral floor. Here, the spectral floor acts as both a feedback limit, and the classical maximum attenuation that masks musical noise. Replace the estimated powers with the estimated INR as per (2.3) to get

$$H(k,\mu) = \max\left(\beta, 1 - \frac{\alpha}{H(k-1,\mu) \cdot \widehat{\zeta}(k,\mu)}\right). \tag{4.16}$$

Consider first a similar system H' without spectral floor. Its system equation is:

$$H'(k,\mu) = 1 - \frac{\alpha}{H'(k-1,\mu) \cdot \widehat{\zeta}(k,\mu)}. \tag{4.17}$$

To find the equlibrium map in its input-state space, set

$$H'(k,\mu) \stackrel{!}{=} H'(k-1,\mu) =: H'_{\mathrm{eq}} \tag{4.18}$$

and

$$\widehat{\zeta}(k,\mu) =: \widehat{\zeta}'_{\mathrm{eq}}. \tag{4.19}$$

This leads to

$$H'_{\mathrm{eq}} = 1 - \frac{\alpha}{\widehat{\zeta}'_{\mathrm{eq}} \cdot H'_{\mathrm{eq}}}. \tag{4.20}$$

This is an implicit representation of the reduced system's equilibrium map. It can be transformed to give the $\widehat{\zeta}'_{\mathrm{eq}}$ as a function of the system's output H'_{eq}:

$$\widehat{\zeta}'_{\mathrm{eq}}\left(\alpha, H'_{\mathrm{eq}}\right) = \frac{\alpha}{H'_{\mathrm{eq}} \cdot \left(1 - H'_{\mathrm{eq}}\right)}, \tag{4.21}$$

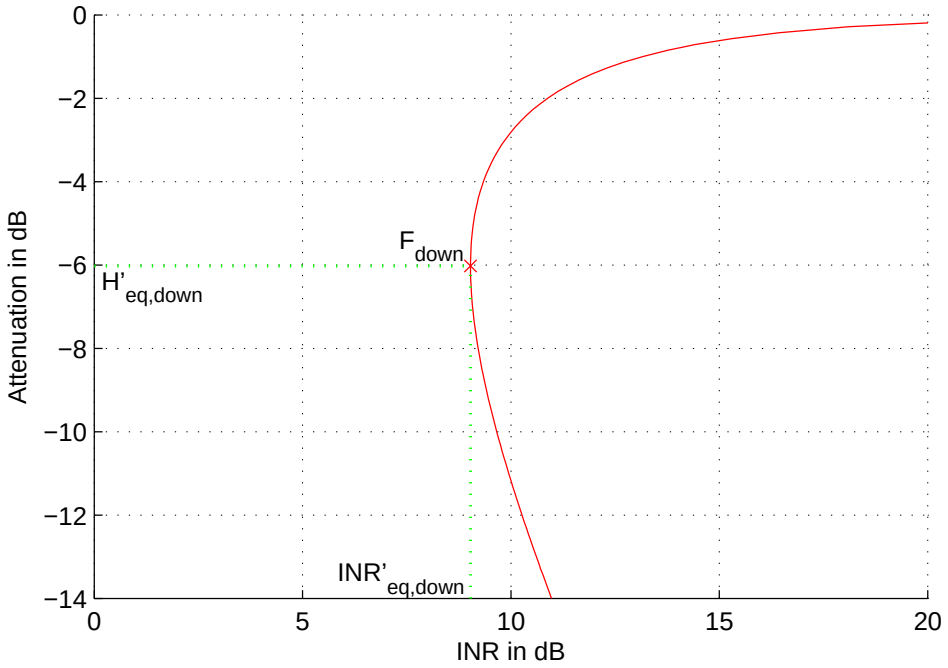

Figure 4.3.: Reduced recursive Wiener filter's equilibrium map with overestimation factor $\alpha = 2$ and no spectral floor. The bifurcation point F_{down} is shown at the inverse parabola's vertex.

or to give a quasi-function of H'_{eq} with two branches in the $\widehat{\zeta}'_{\mathrm{eq}}$ domain:

$$H'_{\mathrm{eq}}\left(\alpha, \widehat{\zeta}'_{\mathrm{eq}}\right) = \tfrac{1}{2} \pm \sqrt{\tfrac{1}{4} - \frac{\alpha}{\widehat{\zeta}'_{\mathrm{eq}}}}. \tag{4.22}$$

Downwards flank. The function described by (4.21) has the form of an inverse parabola. In figure 4.3, the inverse parabola was plotted in the logarithmic domain. It can be shown that at the left of this curve, the filter's output decreases with each step, while at the right of the curve, it increases. This means that there is a bifurcation point F_{down} at the inverse parabola's vertex $\left(\widehat{\zeta}'_{\mathrm{eq,down}}, H'_{\mathrm{eq,down}}\right)$.

Right of the bifurcation point, that is to say with an input $\widehat{\zeta}\,(k,\mu) > \widehat{\zeta}'_{\mathrm{eq,down}}$, the system has two distinct equilibria. The upper branch is a stable equilibrium, as on both of its sides, the system approaches it. The lower branch is an instable equilibrium, as on both of its sides, the system moves farther away from it.

Left of the bifurcation point, the filter's output constantly decreases towards zero ($-\infty$ in logarithmic space), so the filter is closed almost completely as soon as a low input INR is reached. Even if subsequent INR input raises again, it would have to get very high for the filter system state to cross the equilibrium curve and the output to increase again, as the inverse parabola approaches infinity near filter outputs close to zero:

$$\lim_{H'_{\mathrm{eq}} \to 0} \widehat{\zeta}'_{\mathrm{eq}} = \infty. \tag{4.23}$$

The critical INR at the downwards flank, $\widehat{\zeta}'_{\mathrm{eq,down}}$, can be determined by finding the minimum of (4.21) with respect to $\mathrm{H}'_{\mathrm{eq}}$. The only maximum of the fraction's denominator clearly is at

$$H'_{\mathrm{eq,down}} = \tfrac{1}{2}, \tag{4.24}$$

so the corresponding $\widehat{\zeta}'_{\mathrm{eq,down}}$ can be expressed as a function of the overestimation factor α:

$$\widehat{\zeta}'_{\mathrm{eq,down}}(\alpha) = \frac{\alpha}{H'_{\mathrm{eq,down}} \cdot \left(1 - H'_{\mathrm{eq,down}}\right)} = \frac{\alpha}{\frac{1}{2} \cdot \left(1 - \frac{1}{2}\right)} = 4\alpha. \tag{4.25}$$

Of course, the original filter's downwards flank $\widehat{\zeta}_{\mathrm{eq,down}}$ is at the same spot:

$$\widehat{\zeta}_{\mathrm{eq,down}}(\alpha) = 4\alpha. \tag{4.26}$$

Upwards flank. If a state near zero was reached, the reduced recursive Wiener filter would stay shut forever. This is why the spectral floor was installed as an artificial equilibrium. This way, a third function branch is created, which keeps the filter from locking on zero output. The resulting system's phase plot (with input added as a state) is illustrated in figure 4.4. In the configuration of Linhard and Haulick, the spectral floor is used as a limit to the feedback term in (4.15).

As can be seen from figure 4.4, the filter's output is bounded by the spectral floor, where it locks until the INR exceeds the equilibrium INR at the intersection of the equilibrium map and the spectral floor line. This roughly corresponds to the actual hysteresis' upward flank, as the filter's output still needs some time to increase.

Note that the spectral floor β itself limits the feedback used in the filter's recursion, hence $\beta = H_{\mathrm{eq,up}}$. The INR at the upwards flank can be determined by inserting $H_{\mathrm{eq,up}}$ into (4.21):

$$\widehat{\zeta}_{\mathrm{eq,up}}(\alpha, \beta) = \frac{\alpha}{H_{\mathrm{eq,up}} \cdot (1 - H_{\mathrm{eq,up}})} = \frac{\alpha}{\beta \cdot (1 - \beta)}. \tag{4.27}$$

Further notes:

- The theoretical flank INRs only roughly correspond to the observed flanks, as the filter still needs some time steps to reach stationary output values.
- As the equilibrium map loses its bifurcation property when the spectral floor is chosen too high ($\beta > 0.5$), no hysteresis will occur in these cases.

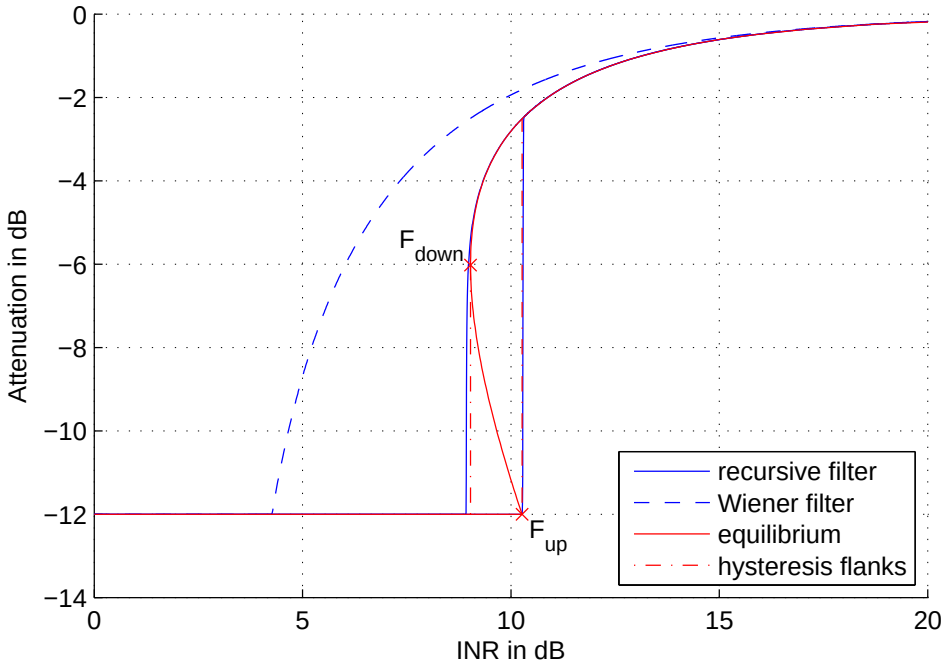

Figure 4.4.: Comparison of the Wiener filter with its recursive counterpart including equilibrium map, bifurcation points, and theoretical flanks. The two filters' characteristics are shown as functions of the INR with overestimation factor $\alpha = 2$ and spectral floor $\beta = -12\,\mathrm{dB}$ for both filters. The hysteresis was captured by creating a phase plot (with input added as a state) of the recursive filter's output with slowly increasing and decreasing INR. Depicted within the hysteresis area is the equilibrium map's instable branch of the recursive filter and the resulting theoretical flanks, that result from the bifurcation points marked F_{down} and F_{up}.

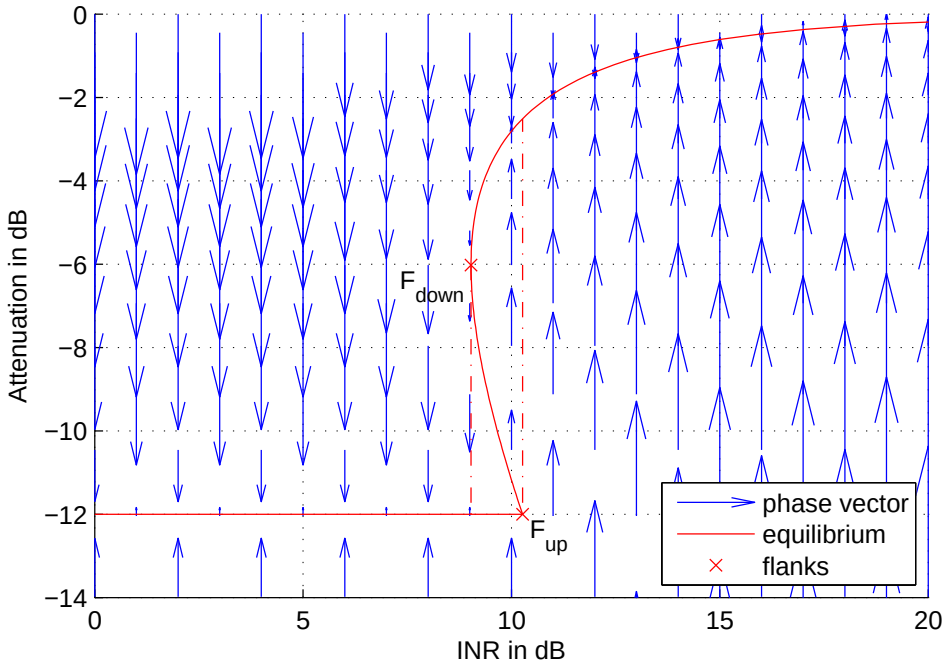

Figure 4.5.: Recursive Wiener filter's phase plot, equilibrium map, and hysteresis flanks. The phase plot was obtained by initializing the filter with a system state and having it calculate one time step at the same input INR. Arrows (not to scale) conceptually indicate the system's trend in its phase space. The filters' equilibrium map is shown as an implicit function of the filter's output at equilibrium H'_{eq} and the corresponding $\widehat{\zeta}'_{\mathrm{eq}}$ with overestimation factor $\alpha = 2$, feedback limit $\beta = -12\,\mathrm{dB}$, and maximum attenuation $\gamma = -12\,\mathrm{dB}$. The resulting theoretical flanks result from the bifurcation points marked F_{down} and F_{up}.

- As the INR increases, the recursive Wiener filter approaches the normal Wiener filter's behavior.
- The recursion concept can also be applied to similar noise reduction filters with similar results (Hänsler and Schmidt, p. 366).

With these considerations, the hysteretic behavior of the recursive Wiener filter is fully explained.

Extension of the recursive Wiener Filter. While the recursive Wiener filter effectively reduces musical noise, it also attenuates speech at low INRs (Hänsler and Schmidt, p. 367). Obviously, the placement of the recursion flanks determines at which INR signals are attenuated down to the spectral floor. Proper placement of the flanks will lead to a good trade-off between musical noise suppression and speech signal fidelity.

It is desirable to modify the flanks' positions according to circumstance. In areas with only noise—the term area is used here to describe time spans as well as frequency bands—the musical noise suppression should remain prevalent while in areas with speech signal components (e.g. in formants), preserving the speech signal gets more important. By detecting important speech component in the form of formants, one gets a good weighting function between the two.

Reconsider the flanks' theoretical positions resulting from the chosen parameters:

$$\widehat{\zeta}_{\mathrm{eq,down}}\left(\alpha\right) = 4\alpha, \tag{4.26}$$

and

$$\widehat{\zeta}_{\mathrm{eq,up}}\left(\alpha,\beta\right) = \frac{\alpha}{\beta\cdot\left(1-\beta\right)}. \tag{4.27}$$

This system can be rearranged to describe the parameters α and β as functions of the flanks' desired INR:

$$\alpha\left(\widehat{\zeta}_{\mathrm{eq,down}}\right) = \frac{\widehat{\zeta}_{\mathrm{eq,down}}}{4} \tag{4.28}$$

$$\beta\left(\widehat{\zeta}_{\mathrm{eq,up}},\widehat{\zeta}_{\mathrm{eq,down}}\right) = \frac{1-\sqrt{1-\dfrac{\widehat{\zeta}_{\mathrm{eq,down}}}{\widehat{\zeta}_{\mathrm{eq,up}}}}}{2}. \tag{4.29}$$

The flanks can be independently placed by choosing adequate overestimation α and spectral floor β. However, as stated at the beginning of subsection 4.4.1, it serves dual purpose. If one chose β arbitrarily small, for example, to move the upwards flank towards a higher INR, this would also result in a very low maximum attenuation, which might be undesirable.

This problem was eliminated by introducing a separate parameter γ that does not contribute to the feedback, but limits the output attenuation anyway. For the

purposes of this work, it is called maximum attenuation. The proposed system $\widetilde{H}$ is described by

$$H(k,\mu) = \max\left(\beta, 1 - \frac{\alpha}{H(k-1,\mu)\cdot\widehat{\zeta}(k,\mu)}\right) \tag{4.30}$$

and

$$\widetilde{H}(k,\mu) = \max\left(\gamma, H(k,\mu)\right), \tag{4.31}$$

where the maximum attenuation γ cannot have any effect when it is smaller than the spectral floor β.

The proposed filter is a generalization of the recursive Wiener filter as proposed by Linhard and Haulick. It can be tailored to different conditions better than could the conventional filter. An example application can be seen in figure 4.6.

The boosting function defined in (4.14) can be put to use in this setup by defining the default flank positions ($\widehat{\zeta}^0_{\text{up}}$, $\widehat{\zeta}^0_{\text{down}}$) and their desired maximum deviations ($\Delta\widehat{\zeta}_{\text{up}}$, $\Delta\widehat{\zeta}_{\text{down}}$) in the center of formants. Then, the filter parameters are updated in every frame and for every bin according to the presence of formants as indicated by the boosting function B:

$$\alpha(k,\mu) = \frac{\widehat{\zeta}^0_{\text{down}} + B(k,\mu)\cdot\Delta\widehat{\zeta}_{\text{down}}}{4} \tag{4.32}$$

and

$$\beta(k,\mu) = \frac{1 - \sqrt{1 - \dfrac{\widehat{\zeta}^0_{\text{down}} + B(k,\mu)\cdot\Delta\widehat{\zeta}_{\text{down}}}{\widehat{\zeta}^0_{\text{up}} + B(k,\mu)\cdot\Delta\widehat{\zeta}_{\text{up}}}}}{2}. \tag{4.33}$$

Now, in the absence of formants, the default values will be used, while with gradual presence of formant, as given by the boosting function, the deviation from the default is used, which can be set to arbitrary positive or negative values. The filter thus reacts differently to formants than it would normally.

As this feature does not permit the filter to output coefficients beyond 0 dB, the audible impact on the estimated speech signal remains low compared to the gain method described in the next chapter. It is recommended to put further work in the proposed recursive filter modification, as optimization and listening tests were concentrated on the following approach.

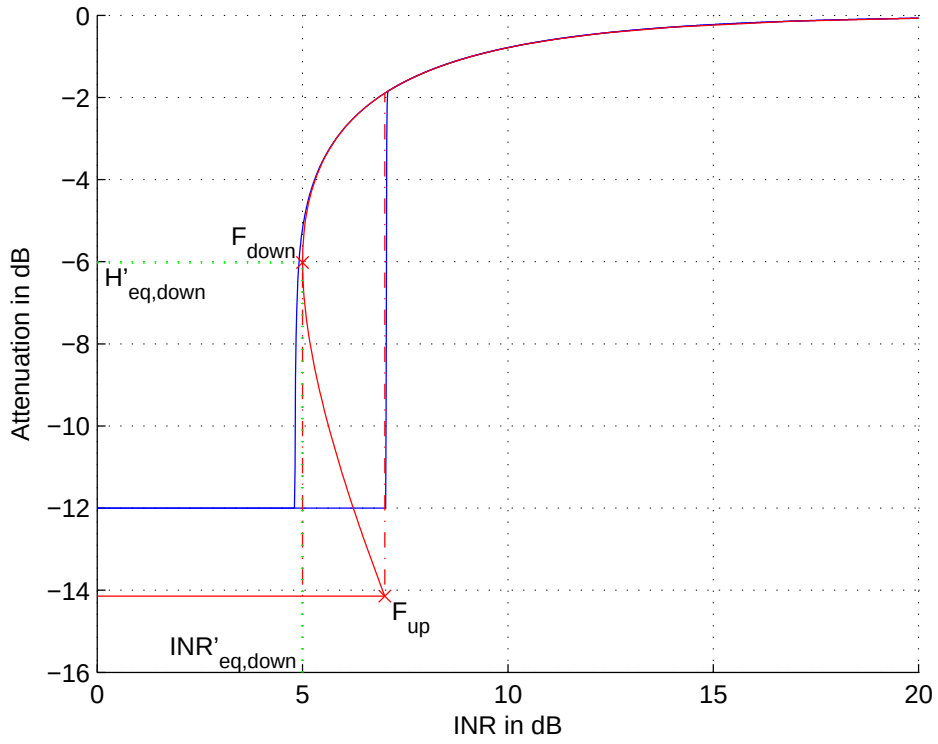

Figure 4.6.: Modified recursive Wiener filter. The phase plot was obtained by initializing the filter with a system state and having it calculate one time step at the same input INR. The conceptual difference between the feedback limit $\beta \approx -14\,\mathrm{dB}$ and the maximum attenuation $\gamma = -12\,\mathrm{dB}$, both formerly the same thing termed spectral floor, is reflected. Flanks positions were set to exactly 5 dB and 7 dB, leading to the aforementioned feedback limit and the overestimation $\alpha = 0.791$. Bifurcation points are indicated by F_{up} and F_{down}.

4.4.2. Gain after Arbitrary Filter

Another application of the boosting function defined in (4.14) is to amplify the noise filtered signal depending where formants are detected. As stationary driving noise in automobiles usually falls off monotonically in frequency, most speech formants are bound to feature a higher local SNR than neighboring frequencies. Hence, by identifying and amplifying frequency regions with better-than-average SNR, a better broadband SNR ca be achieved.

Any noise reduction filter's output $H(k,\mu)$—called the filter coefficients—consists of values between 0 and 1 for each frequency bin μ in a frame k. In the conventional setup, the noisy speech signal's amplitude spectrum is then multiplied element-wise with the filter coefficients, see (2.15).

In order to perform the formant-dependent amplification mentioned, a multiple of the boosting function $B(k,\mu)$ is used as a power gain factor. This is done by converting it to linear amplitude values and multiplying the result with the filter coefficients:

$$H_{\mathrm{FB}}(k,\mu) := H(k,\mu) \cdot 10^{B_{\max} \cdot B(k,\mu)/20} \tag{4.34}$$

In the absence of formants, the filter output remains unchanged, as the boosting function is zero there. At the center of formants, the maximum gain $B_{\max}$ (in decibels) is applied. The amount of gain applied to intermediate frequency bins follows the prototype window function used.

Note that the boosting function B is interpreted here as a series of power gain values.

Power Adjustment. As the algorithm only adds power, the formant boosted signal is always louder compared to the corresponding conventionally noise reduced signal. This can lead to clipping if the system's dynamic range is exceeded. What is more, the speech signal's overall power in the formant band grows in relation to its power in the fricative band. This mismatch is disturbing to the listener.

The power contrast between formants' centers and frequency bands without formants is determined by the maximum power gain in decibels $B_{\max}$. The power contrast is responsible for the intelligibility increase and should not be reduced. Instead, after selective amplification, the frequency band that potentially contained formants (up to $f_{\mathrm{F,max}} = 3500\,\mathrm{Hz}$) can be attenuated as a whole.

The amount of necessary attenuation is determined heuristically: The motivation here is that the boosting function should not inherently introduce a broadband power gain, while it should retain the power contrast between formants and other speech signal components.

Recall from (4.3) that M_{F} is the number of frequency bins in the formant band. Then the mean power gain introduced in a frame by formant boosting is:

$$B_{\mathrm{mean}}(k) := \frac{B_{\max}}{M_{\mathrm{F}}} \sum_{\mu=0}^{M_{\mathrm{F}}-1} B(k,\mu)\,. \tag{4.35}$$

This mean value can be subtracted from the boosted spectrum in each frame, so that for the relevant bins, the boosting rule from (4.34) becomes:

$$H_{\mathrm{FB,neut}}(k,\mu) := H(k,\mu) \cdot 10^{(B_{\max} \cdot B(k,\mu) - B_{\mathrm{mean}}(k))/20} \quad \forall \mu \in \{0,\ldots,M_{\mathrm{F}}-1\}. \tag{4.36}$$

This provides for the boosting function to be power neutral, which can be useful for listening comparison with other filters, if equal mean volume is desired, as is the case for the subjective test performed for the evaluation of the formant boosting process. See section 4.5 below.

Time Smoothing. With the power adjustment explained above, the maximum temporal power step resulting from onset of the formant amplification is lower than it were without the adjustment, as now it needs to step up by $B_{\max} - B_{\mathrm{mean}}$ decibels at most, instead of the whole $B_{\max}$. The same holds vice versa for transitions from formant boosting to frames without boosting. Still, power steps at some formants' centers remain audible. Therefore, temporal smoothing on the power gain is applied by restricting its maximum rate of change in each frequency bin. This so-called slew rate can be adjusted separately for increasing $\dot{B}_{\mathrm{up}}$ and decreasing $\dot{B}_{\mathrm{down}}$ values:

$$B_{\mathrm{slew}}(k,\mu) := \min\Big\{\max\Big\{B(k,\mu)\cdot B_{\max} - B_{\mathrm{mean}}, B_{\mathrm{slew}}(k-1,\mu) - \dot{B}_{\mathrm{down}}\Big\}, B_{\mathrm{slew}}(k-1,\mu) + \dot{B}_{\mathrm{up}}\Big\} \tag{4.37}$$

The slew rate constants must be specified in decibels per frame. The resulting and final boosting rule then is:

$$H_{\mathrm{FB,slew}}(k,\mu) := H(k,\mu) \cdot 10^{B_{\mathrm{slew}}(k,\mu)/20}. \tag{4.38}$$

4.5. Subjective Test

As the formant boosting process inherently constitutes a distortion of the speech signal, most objective measures are expected to indicate a worse performance of formant boosting compared to mere noise reduction. It is however hoped that even objective distortions might improve speech signals for human listeners. Therefore, to obtain a performance measure anyway, a subjective test was conducted.

4.5.1. Design

In their P-series (Telephone transmission quality, telephone installations, local line networks), the ITU Telecommunication Standardization Sector (ITU-T) of the International Telecommunication Union (ITU) proposes several test designs. In ITU-T, the comparison category rating (CCR) method defines a procedure test design for comparing processed against unprocessed signals with reference to a common preprocessing. The resulting variable is called comparative mean opinion score (CMOS) and features integer values between -3 and $+3$ as subjective ratings, see table 4.1.

Table 4.1.: CMOS scores and their meanings. Two speech samples are presented. This scale defines the quality of the second sample compared to the first. In the listening test, the order is randomized, but is taken into account on evaluation. Positive values support formant boosting, while negative values support noise reduction. Adapted from ITU-T.

CMOS	Description
3	Much Better
2	Better
1	Slightly Better
0	About the Same
-1	Slightly Worse
-2	Worse
-3	Much Worse

This method was used for comparing signals processed by the formant boosting algorithm against noise reduced signals without formant boosting applied. The speech signals and noise signals were taken from the database created as a part of this work and described in chapter 3.

The test was performed in three distinct passes that will be commented below. The results and insights from the first and second passes were used for the design of the following pass.

A graphical user interface (GUI) was implemented for presentation of the listening test featuring, for each pair of samples, a play button for each, and one or two CMOS scales, depending on the particular definition of the respective pass. In deviation from the CCR specification, the subjects were asked to repeat the listening when they hear no difference at first, as the audible differences are expected to be small. No time limit was imposed on the decision either. The GUI main window of the listening test is depicted in figure 4.7.

In a preparation step for each pass, a selection of speech signals with one second of leading and trailing speech pause was processed once with a recursive Wiener filter and once with the same filter and additional formant boosting. The sound samples' file names were then obfuscated with random alphanumeric character sequences while retaining a local table relating the original and obfuscated file names. A template listening test GUI was integrated with the sound files and packaged for deployment to the subjects.

When starting the listening test for the first time, the order of presentation was randomly generated individually and locally saved for each subject. The listening test was programmed to create a results file upon completion of the test by each subject.

In the development phase, two different versions of formant boosting were reviewed by the author. The first (described in subsection 4.4.1), did not feature much difference to the corresponding noise reduced signals, which is easily explained by its structural limitation to a maximum 0 dB gain. This version was not used for any test.

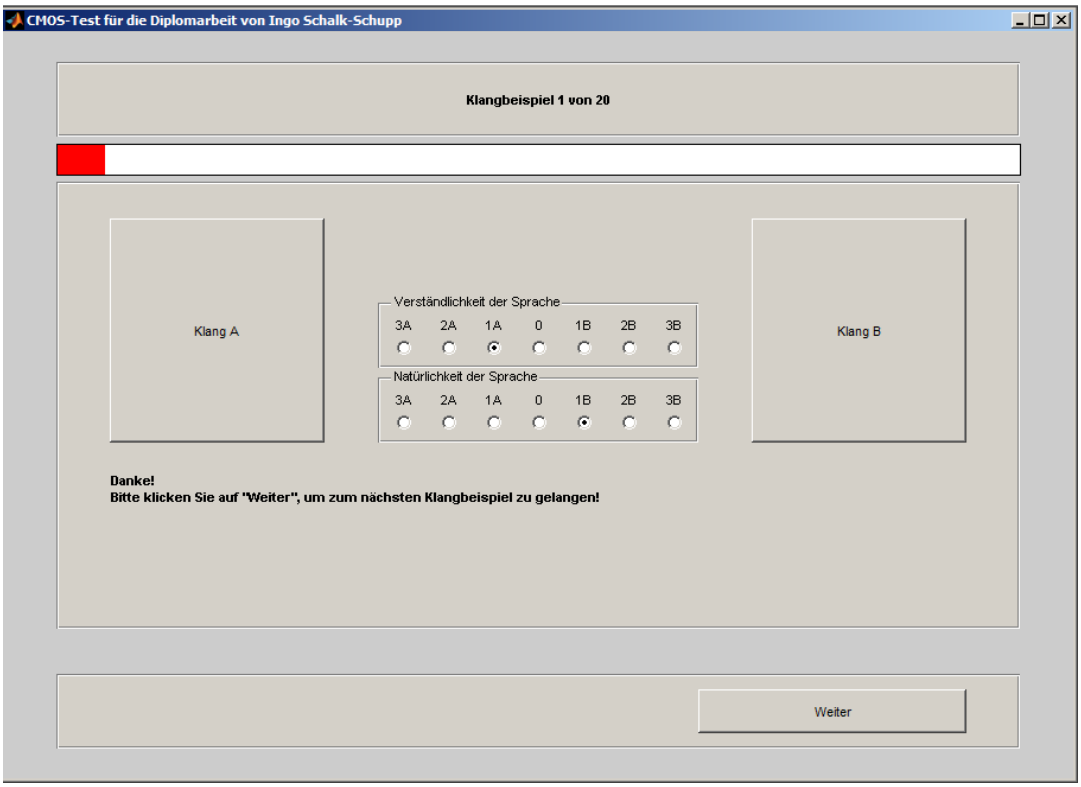

Figure 4.7.: Graphical user interface of the subjective test. The sound samples are played back by clicking on the respective buttons. After each of the samples have been played at least once each, the rating scales appear. When both ratings are done, the next pair of sound samples becomes available.

The second (decribed in subsection 4.4.2) however, seemed more promising and was hence used for all of the following test definitions.

4.6. Evaluation

After all of the files containing the results were gathered, the results were collated to the conditions and consolidated into a single data file for each pass. Evaluation is based on these data files.

Apart from a descriptive analysis of the results in the form of histograms, a mean statistic with 95%-confidence intervals was calculated. The statistics are presented for each speed separately and consolidated in an overall statistic for each CMOS scale and pass separately.

4.6.1. First Pass with Experienced Subjects

The first subjective listening test was designed to get evidence on whether the formant boosting algorithm was an improvement over noise reduction. It was performed in Nuance's speech team internally, so the subjects were experienced listeners. At the time of the first subjective listening test pass, the formant boosting algorithm lacked the time smoothing post-processing stage described in subsection 4.4.2.

Presentation. The subjects were presented 40 pairs of speech samples evenly distributed over the 5 noise conditions at 50 km/h, 80 km/h, 100 km/h, 130 km/h, and 160 km/h. For every noise condition, 2 utterances by 4 (two male, two female) speakers were presented. All of the utterances were the same for all subjects, but the order was randomized individually. Only one CMOS scale—overall quality—was presented in favor of a shorter test duration.

Results. The first test pass was performed by 11 subjects. The overall quality rating relative frequencies are depicted in figure 4.8. Positive values support formant boosting, while negative values support noise reduction. A slight tendency toward formant boosting is evident from the histograms in each of the noise conditions.

From the relative frequencies, a mean score was obtained separately for each speed and once for the complete set of ratings. A 95%-confidence interval was also computed. Figure 4.9 displays these results. The values can be seen in table 4.2.

Feedback and Discussion. Figure 4.9 supports the impression from figure 4.8: A slight preference of the formant boosting is evident with the confidence intervals consistently lying above 0. The overall mean score is 0.530 ± 0.131. The overlapping confidence intervals suggest that there is no considerable difference between the noise conditions.

Several subjects provided additional feedback, indicating that they observed separate positive and negative effects that they would have liked to differentiate. First, the

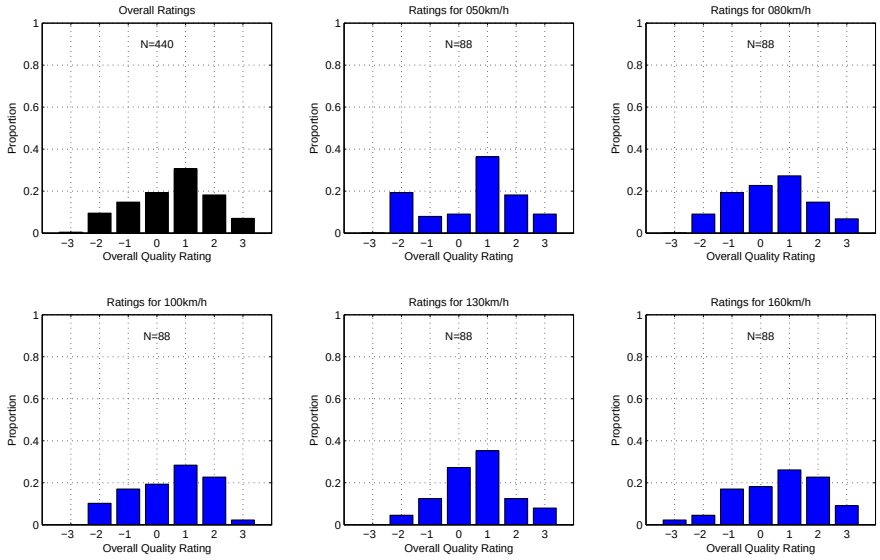

Figure 4.8.: Subjective overall quality ratings in the first pass. Relative frequencies are depicted for the overall ratings as well as for the noise conditions.

Table 4.2.: Comparative mean opinion score with 95%-confidence intervals for the scale "overall quality" in the first test pass.

Condition	Mean±Conf.
50 km/h	0.534 ± 0.331
80 km/h	0.398 ± 0.288
100 km/h	0.432 ± 0.281
130 km/h	0.625 ± 0.256
160 km/h	0.659 ± 0.305
Overall	0.530 ± 0.131

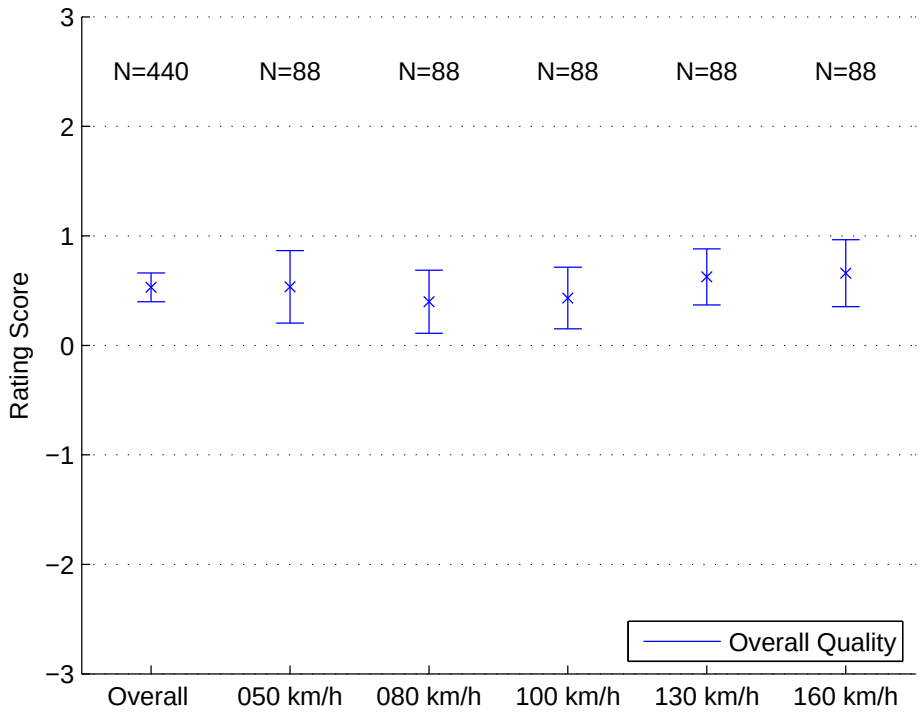

Figure 4.9.: Overall quality CMOS in the first pass. Error bars indicate the 95%-confidence intervals.

formant boosting had introduced artifacts by abruptly setting in or stopping abruptly, which was detrimental to the overall quality. On the other hand, the boosting itself was perceived as accentuating vocals, sometimes improving the signal, but sometimes distorting it too much.

While the formant boosting algorithm was slightly preferred throughout the noise conditions, the CMOS of roughly 0.5 did not meet the author's expectations.

4.6.2. Second Pass with Inexperienced Subjects

In the first pass, subjects were forced to combine their impressions to one value on an overall quality scale. Feedback given by some led to the idea to separate the rating scales in order to get a more differentiated result. Decision on two new scales, namely "intelligibility" and "naturalness", was made based on the main effects observed in the speech samples: an increase in clarity against an increase in distortion. It was hoped that the overall preference of formant boosting in the first pass could be explained by a greater preference of the formant boosting's intelligibility impaired by its decrease in naturalness.

Additionally, the problem of sudden boosting onset was tackled by the introduction of time smoothing on the amount of gain applied in boosting.

This time, the test was sent to external subjects with no experience in the area of audio signal processing.

Presentation. The subjects were presented 40 pairs of speech samples evenly distributed over the 5 noise conditions at 50 km/h, 80 km/h, 100 km/h, 130 km/h, and 160 km/h. For every noise condition, 2 utterances by 4 (two male, two female) speakers were presented. All of the utterances were the same for all subjects, but the order was randomized individually. Two CMOS scales—intelligibility and naturalness—were presented.

Results. The second test pass was performed by 9 subjects. The intelligibility and naturalness rating relative frequencies are depicted in figure 4.10 and figure 4.11, respectively. Positive values support formant boosting, while negative values support standard noise reduction. At first glance, no specific tendency is evident for the intelligibility. Compared to the first pass however, a score of 0 was chosen noticeably more often, indicating that this scale feature neither improvement nor degeneration of this merit. A much clearer picture emerges on the naturalness scale: Here too, a score of 0 occurs most frequently, but the remaining ratings clearly support the naturalness of noise reduced signals.

From the relative frequencies in the two categories, a mean score was obtained separately for each speed and once for the complete set of ratings. As before, a 95%-confidence interval was also computed. Figure 4.12 displays these results. The values are listed in table 4.3 and table 4.4.

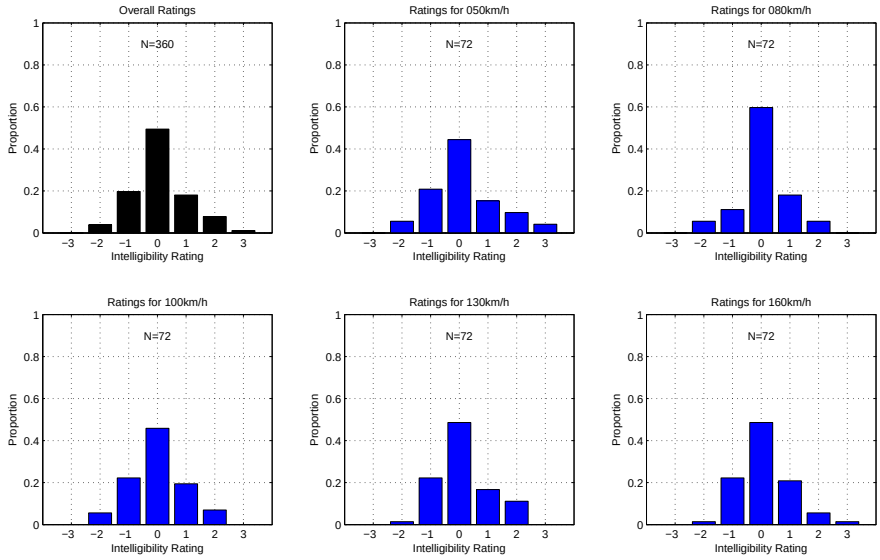

Figure 4.10.: Subjective intelligibility ratings in the second pass. Relative frequencies are depicted for the overall ratings as well as for the noise conditions.

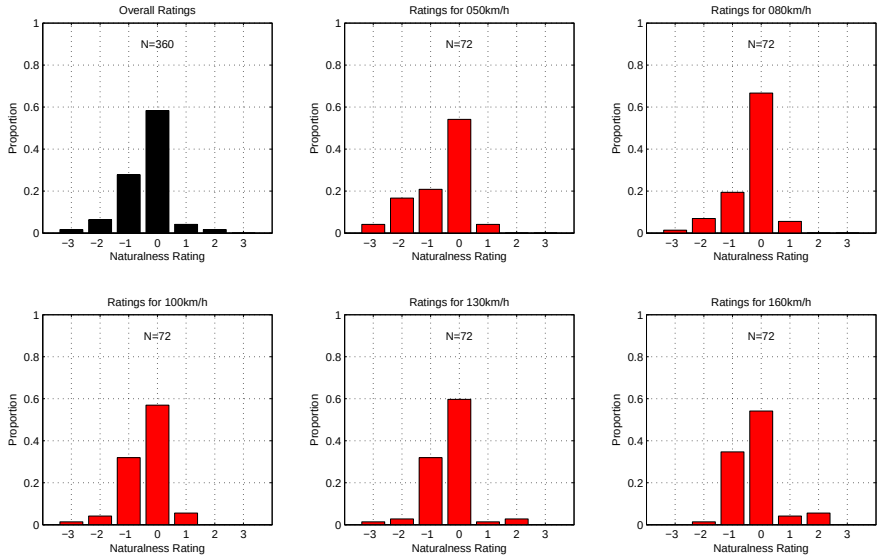

Figure 4.11.: Subjective naturalness ratings in the second pass. Relative frequencies are depicted for the overall ratings as well as for the noise conditions.

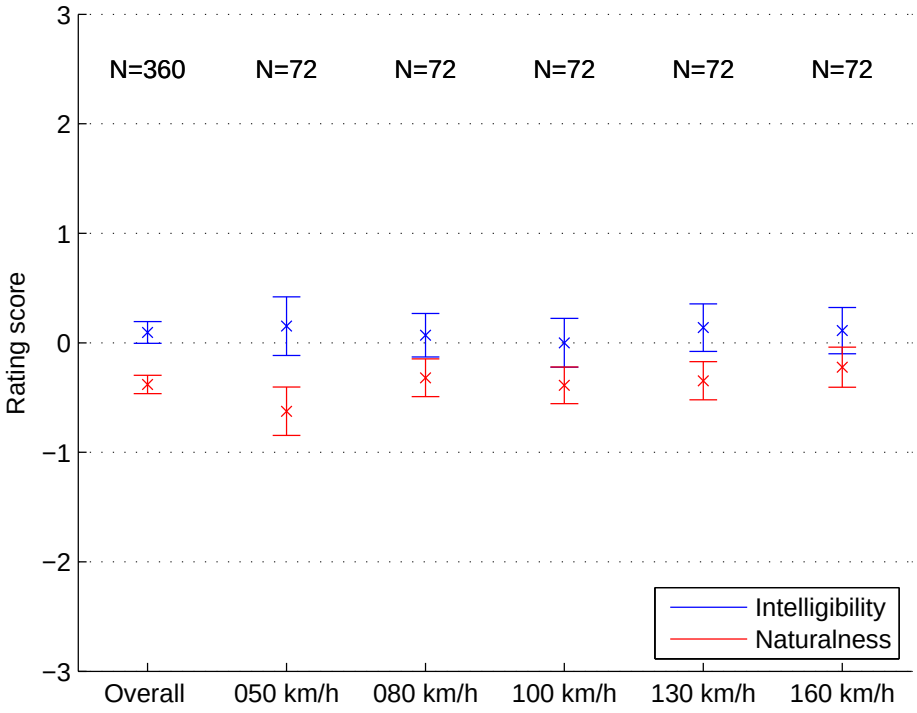

Figure 4.12.: Intelligibilty and naturalness CMOS in the second pass. Error bars indicate the 95%-confidence intervals.

Table 4.3.: Comparative mean opinion score with 95%-confidence intervals for the scale “intelligibility” in the second test pass.

Condition	Mean±Conf.
50 km/h	0.153 ± 0.268
80 km/h	0.069 ± 0.199
100 km/h	0.000 ± 0.223
130 km/h	0.139 ± 0.217
160 km/h	0.111 ± 0.211
Overall	0.094 ± 0.100

Table 4.4.: Comparative mean opinion score with 95%-confidence intervals for the scale "naturalness" in the second test pass.

Condition	Mean±Conf.
50 km/h	-0.625 ± 0.221
80 km/h	-0.319 ± 0.173
100 km/h	-0.389 ± 0.167
130 km/h	-0.347 ± 0.174
160 km/h	-0.222 ± 0.183
Overall	-0.381 ± 0.083

Feedback and Discussion. Figure 4.12 supports the impression from figure 4.8: Neither of the algorithms presented is any more intelligible than the other, as the mean values' confidence intervals all cover zero score. A slight preference of the natural sound of pure noise reduction is evident from the CMOS of -0.381 ± 0.083.

Many subjects stated in their answer email that they could barely make out any difference between the paired signals at all, which is consistent with the high frequency of zeros in both categories.

These results can for the most part be attributed to exaggerated smoothing of the boosting gain. It was introduced in order to reduce artifacts that were perceived as disturbing in the first test pass. However, the difference in overall power between pairs of signals had remarkably declined because of the high amount of smoothing applied. Another reason might be the fact that this test pass was performed only by inexperienced users who are likely to not perceiving difference that experiences listeners might be able to discern.

The second pass did however reveal that there is a relevant difference in the two proposed categories.

4.6.3. Third Pass with Experienced Subjects

In the second pass, exaggerated smoothing led to an undesirable result. Hence, a trade-off between the reduction of artifacts introduced by sudden boosting and audible improvement of vocals in the speech signal was sought. A third internal test pass was proposed, because it was discovered that even gentle smoothing greatly reduced the amount of artifacts generated. As the categories from the second pass revealed a noticeable difference in their respective scores, they were not changed in the third pass. Time restrictions imposed a limit on the number of trials in this pass, so only half the number of pairs was presented. In order to cover the whole available database and not force subjects to listen to an small a priori chosen set of speech samples, the actual speech samples were randomly chosen for each subject. For better comparability, a testing stand was setup, thus implementing a defined listening environment and equipment.

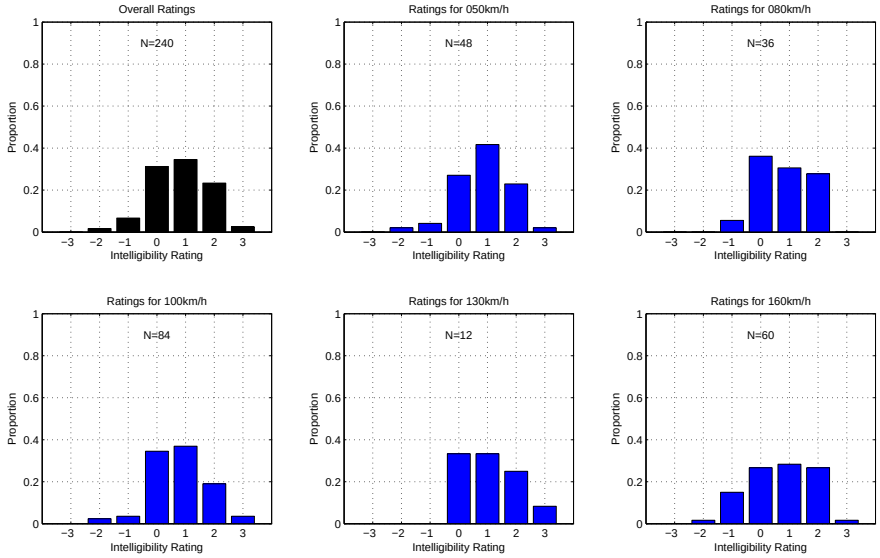

Figure 4.13.: Subjective intelligibility ratings in the third pass. Relative frequencies are depicted for the overall ratings as well as for the noise conditions.

Presentation. The subjects were presented 20 pairs of speech samples randomly chosen from the 5 noise conditions at 50 km/h, 80 km/h, 100 km/h, 130 km/h, and 160 km/h. No effort was made to attain an evenly distributed combination of speech sample pairs. Two CMOS scales—intelligibility and naturalness—were presented.

Results. The third test pass was performed by 12 subjects. The intelligibility and naturalness rating relative frequencies are depicted in figure 4.13 and figure 4.14, respectively. Positive values support formant boosting, while negative values support noise reduction. A considerable tendency in intelligibility toward formant boosting is observed. Compared to the second pass, much more even distributions can be seen. Excepting the 130 km/h noise condition, which has only 12 trials, the histograms look quite similar. As for the naturalness, a tendency towards noise reduction is visible. Again, the 130 km/h noise condition stands out while featuring the least number of trials.

From the relative frequencies in the two categories, a mean score was obtained separately for each speed and once for the complete set of ratings. Again, a 95%-confidence interval was also computed. Figure 4.15 displays these results. The values are listed in table 4.5 and table 4.6.

Feedback and Discussion. Figure 4.15 consolidates what can be seen in figure 4.8: Formant boosting outperforms noise reduction in terms of intelligibility with an overall CMOS of 0.788 ± 0.128. No dependence on the noise conditions is evident, as all of

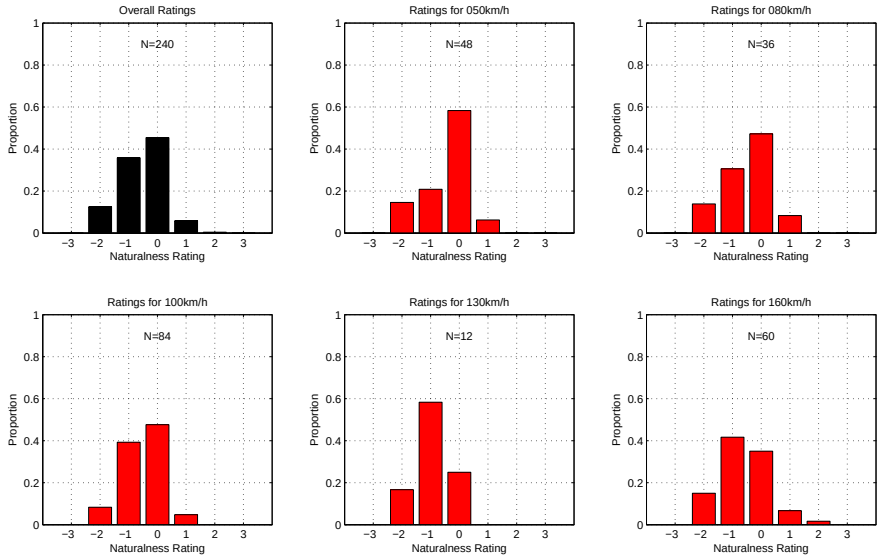

Figure 4.14.: Subjective naturalness ratings in the third pass. Relative frequencies are depicted for the overall ratings as well as for the noise conditions.

Table 4.5.: Comparative mean opinion score with 95%-confidence intervals for the scale "intelligibility" in the third test pass.

Condition	Mean±Conf.
50 km/h	0.854 ± 0.274
80 km/h	0.806 ± 0.301
100 km/h	0.774 ± 0.213
130 km/h	1.083 ± 0.564
160 km/h	0.683 ± 0.285
Overall	0.788 ± 0.128

Table 4.6.: Comparative mean opinion score with 95%-confidence intervals for the scale "naturalness" in the third test pass.

Condition	Mean±Conf.
50 km/h	-0.438 ± 0.233
80 km/h	-0.500 ± 0.276
100 km/h	-0.512 ± 0.154
130 km/h	-0.917 ± 0.378
160 km/h	-0.617 ± 0.224
Overall	-0.542 ± 0.101

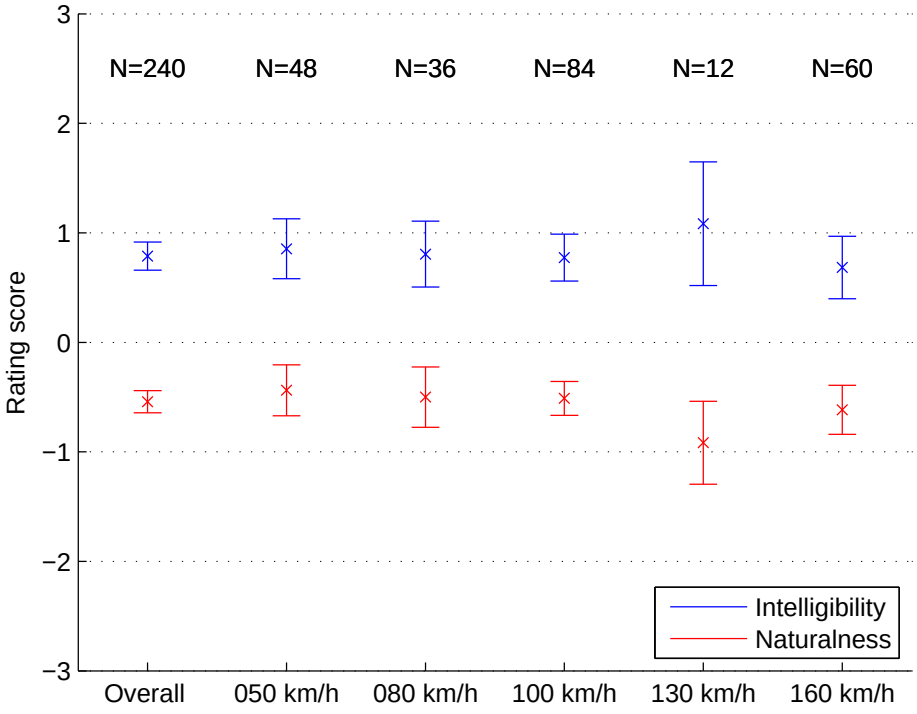

Figure 4.15.: Intelligibilty and naturalness CMOS in the third pass. Error bars indicate the 95%-confidence intervals.

the confidence intervals overlap. At 130 km/h, a suspected outlier of the mean is mitigated by the wider confidence interval resulting from the small number of trials in this condition. The preference of the naturalness of the noise reduction algorithms is just as consistent, while the effect size is slightly smaller at -0.542 ± 0.101.

The results indicate that formant boosted signals, when compared to noise reduced signals, are more intelligible but sound less natural.

Considering the second test pass, the new set of parameters used for the third pass is most likely the reason for the improvement, while the choice of subjects cannot be neglected.

4.6.4. Weighting of the Rating Scales

The question arises how the combined rating results of the first pass relate to the separate rating scales in the third pass, taking into account the fact that the 9 subjects in the third pass also participated in the first pass, out of 11. Assuming momentarily that the testing conditions had been exactly the same with only the available CMOS scales differing, one must come to the conclusion that the overall quality score $\mathrm{CMOS}_{\mathrm{ovr}}$ must be the weighted sum of the two new scores, intelligibility $\mathrm{CMOS}_{\mathrm{int}}$ and naturalness $\mathrm{CMOS}_{\mathrm{nat}}$:

$$\begin{aligned} \mathrm{CMOS}_{\mathrm{ovr}} &= a \cdot \mathrm{CMOS}_{\mathrm{int}} + (1-a) \cdot \mathrm{CMOS}_{\mathrm{nat}} \\ \Leftrightarrow 0.530 &= a \cdot -0.542 + (1-a) \cdot 0.788 \\ \Leftrightarrow a &= 0.806 \end{aligned} \quad . \tag{4.39}$$

This would mean that, for the given subjects, intelligibility was about four times as important as naturalness for overall quality. However, since the assumption is not exactly true, other effects may contribute to the scores. Firstly, it can be stated—in terms of objective distance measures—, that the algorithm used in the third test pass has improved on naturalness, pushing the favor further towards intelligibility. Secondly, subjects might weight the categories differently, but the weighting considered above relates to mean values only. Finally, there might be other merits reflected in neither of the new categories, but still contributing to an overall impression. However, none of the verbal feedback has produced any sign for this.

5. Speech Signal Reconstruction

Beyond speech enhancement algorithms that analyze and modify a signal on a subband base, more sophisticated approaches take into account additional features specific to human speech signals. One example, formant boosting, was introduced in the previous chapter. Formant boosting makes use of a signal's estimated vocal tract information that is represented by a smoothed amplitude spectrum. In this chapter, another method will be presented that aims to recover information concealed by noise in some frequency bins on the base of information contained in the remaining frequency bins. Underlying is the well-known fact that audible human speech contains redundant information.

While Warren et al. and Jax and Vary (p. 4) observe that even with missing fundamental or lower pitch frequencies, utterances can still be recovered by a human listener, Jax and Vary (p. 5) also state that the listening effort decreases for less-degraded signals. Hence, it is desirable to attain a good-quality reconstructed signal to provide for a less strenuous communication experience, which is especially important when a communication partner has to concentrate on driving a car.

In this chapter, a structure for the speech reconstruction process used at Nuance will be introduced, followed by a description of several methods to estimate the speech signal's amplitude envelope, which constitutes the specific task assigned to the author. Finally, the estimators' performances will be evaluated using a logarithmic distance measure.

5.1. Statement of Problem

In all of the above-mentioned methods, each frequency bin amplitude value was multiplied with a scalar in order to provide a modified output signal of better quality. The methods differ in how each scalar is calculated and which information the calculation is drawn upon. This means that the output spectrum consists of a frequency-selective scaling of the input spectrum, that in turn is a sum of the clean speech and the noise spectrum. Neither the frequency-selective ratio of speech power to noise power can be changed, nor can the phase.

These restrictions can be bypassed by permitting the introduction of artificially generated signal components into the output spectrum, opening the potential to recover even frequency bands in which noise is predominant in the input signal.

In hands-free automobile telecommunication environments, low-frequency noise typically masks the speech signal's fundamental harmonic and sometimes, depending on the speaker's pitch and on the noise level, higher harmonics, too. Noise reduction

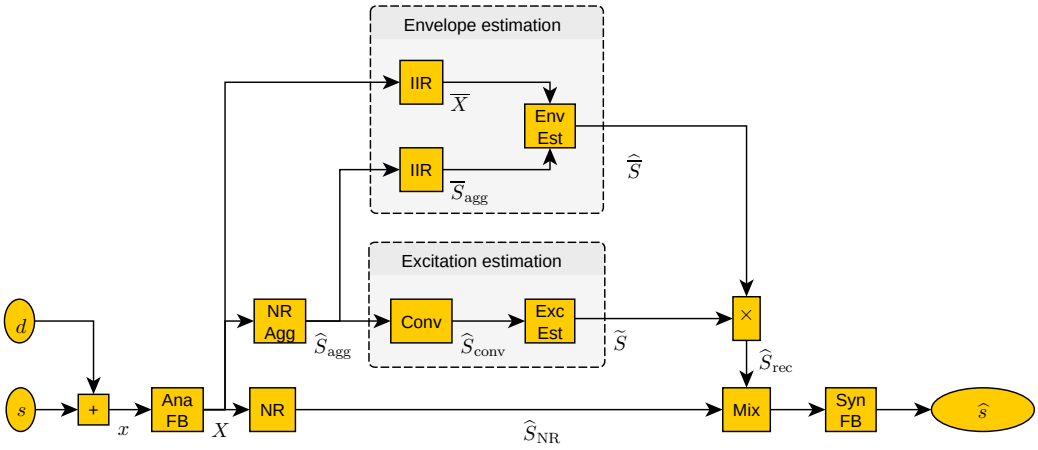

Figure 5.1.: Signal flow in the speech reconstruction process. Ana FB stands for an analysis filterbank, which performs windowing and an STFT. Analogously, Syn FB marks a synthesis filterbank, that performs an inverse STFT and sums the frames in the time domain. NR is the noise reduction algorithm. NR Agg is a similar noise reduction algorithm with a very low spectral floor. Conv and Exc Est compute the convolution of $\widehat{\overline{S}}_{\text{agg}}$ and provide an estimate for the excitation component of the speech signal. The IIR units provide a smoothed amplitude spectrum of their respective inputs. Env Est is the estimation unit for the envelope of the speech signal, meaning the vocal tract component. Excitation and envelope information are combined with the noise reduced signal in the Mix module.

filters generally attenuate these frequency bands down to the spectral floor. This results in a low noise level but also renders any remaining speech components in this area inaudible. The resulting signal sounds too “thin”.

5.2. Structure

It is the idea of speech signal reconstruction to utilize redundant information from the input signal to compensate for gaps in the frequency spectrum caused by noise.

The reconstruction process consists of three main sections: a conventional noise reduction, speech reconstruction, and a mixing stage. The reconstruction process’ overall structure can be seen in figure 5.1 and is described in the following.

Noise reduction. In the block labeled “NR Agg”, a conventional noise reduction algorithm with a high maximum attenuation is applied to the noisy speech signal.

This provides a speech signal with mainly noise-free components for the following stage.

Separation of Excitation and Vocal Tract Information. Given the physical speech production model given in subsection 2.4.1, in particular the Dirac-comb-like nature of the excitation signal, on can assume that the excitation amplitude changes fast in the frequency domain as opposed to the vocal tract information that changes only slowly. The proposed structure separates a given amplitude spectrum in a slowly-changing portion and a fast-changing portion at several points in the process. This is achieved by first applying a first-order IIR filter to the spectrum's amplitude values to extract the coarse amplitude structure and then dividing each frequency bin of the original complex spectrum by the smoothed values to get the fine amplitude structure while also retaining the phase information. As an example, see (5.1) ff. below.

The division of excitation information and vocal tract contribution relies of the physical speech production model mentioned in subsection 2.4.1, which states that the speech amplitudes are the product of the excitation and the vocal tract amplitude contributions. This assumption has proven useful for the purposes of speech reconstruction, as no information is lost in the process and envelope estimation can thus constitute an independent module.

Speech reconstruction. Whenever voiced excitation occurs, a natural spectrum with evenly spaced harmonics emerges in the frequency domain. This signal is processed by a nonlinear operator, which results in the creation of sub- and super-harmonics in the spectrum. The actual operator chosen is a circular auto-convolution of an adaptively selected part of the spectrum. A voiced-excitation detector is used here to detect and select 1-kHz-wide frequency bands with enough power.

The reconstructed harmonics feature different amplitudes that are generally not suitable for direct output, as their envelope shape depends on many factors like the place and width of the missing components, the envelope of the remaining speech signal, and the frequency band chosen for reconstruction. Therefore, the envelope of the convoluted signal is to be replaced with an estimated envelope.

As a preparation step, the convoluted complex pitch structure $\widehat{S}_{\text{conv}}$ is divided by its own amplitude envelope $\overline{S}_{\text{conv}}$, so the estimated envelope $\widehat{\overline{S}}$ can later be multiplied with the reconstructed pitch structure $\widetilde{S}$.

$$\overline{S}_{\text{conv}}(k,\mu) := \text{IIR}\left\{\left|\widehat{S}_{\text{conv}}(k,\mu)\right|\right\} \tag{5.1}$$

$$\widetilde{S}(k,\mu) := \frac{\widehat{S}_{\text{conv}}(k,\mu)}{\overline{S}_{\text{conv}}(k,\mu)}, \tag{5.2}$$

where IIR denotes smoothing along the frequency dimension and is defined as in (4.4) to (4.7) with a smoothing constant of $\gamma_{\text{env}} = 0.8$.

Underlying the process described here is the notion that the coarse structure of the amplitude spectrum resembles the vocal tract information, while the fine structure

corresponds to the excitation component. There will always be cases in which the chosen smoothing constant leads to erroneous separation, but the method is very fast and provides adequate separation in most cases.

The details of the envelope estimation are discussed in section 5.3 below.

Mixer. Finally, a mixer stage computes a weighted sum of the reconstructed spectrum's and the noise spectrum's amplitudes while keeping the microphone signal's phase. The weighting is based on the source-filter model usually applied in speech coding (Krini, Hannon, and Schalk-Schupp).

5.3. Estimation of the Low-Resolution Amplitude Spectrum

The task assigned in this work was narrowed down to one specific module embedded in a surrounding speech reconstruction structure. It is the module's goal to estimate the smoothed spectrum of the clean speech signal as good as possible. As can be seen in figure 5.1, estimation of the pitch structure is done in a separate unit, and their product is mixed with the noise-reduced spectrum in a downstream stage. The mixing steps are not within the scope of this work.

5.3.1. Separation of Training and Test Signals

Some of the methods described hereafter make use of information taken from actual speech signals. In order to produce valid results, it is necessary to separate the signals used for information retrieval from the signals used for performance evaluation. Otherwise, the results might yield performance measures that are only valid for the special set of signals used.

The speech signals recorded for this work, as described in chapter 5, were divided in a training data set and a testing data set for the purposes of this chapter. The same partitioning was used for all of the methods in question to provide for comparable results.

As four utterances were recorded for each condition (speaker, speed), half of each condition's utterances were used for training, and the other half for testing. Given the total amount of utterances available from the database, this partitioning is the most viable, as some variance in both is still present and neither is specialized on a specific sentence.

5.3.2. Estimation Methods

The performance of an estimation algorithm can be objectively evaluated. An adequate cost function for errors has to be specified to get an absolute performance measure. However, to be meaningful, the results should be compared to other algorithms with the same goal. A choice of methods was implemented and compared.

For all of the following methods, a decision has been made on how to separate the voiced excitation's and the vocal tract's contributions to the clean speech spectrum

so as to supply comparable results. As only the amplitude spectra are of interest for the estimation here, the powers $\widehat{\Phi}_X$ and $\widehat{\Phi}_{\widehat{S}}$ of the microphone signal X and the noise-reduced signal $\widehat{S}$ were smoothed using an IIR filter:

$$\overline{X}(k,\mu) := \mathrm{IIR}\{|X(k,\mu)|\} \tag{5.3}$$

$$\widehat{\overline{S}}_{\mathrm{NR}}(k,\mu) := \mathrm{IIR}\left\{\left|\widehat{S}(k,\mu)\right|\right\}. \tag{5.4}$$

Again, the smoothing is performed in frequency direction as in (4.4) to (4.7) with $\gamma_{\mathrm{env}} = 0.8$.

5.3.3. Estimation Performance Statistics

For determining the known signal's S amplitude envelope, smoothing is applied exactly as in (5.3) above:

$$\overline{S}(k,\mu) := \mathrm{IIR}\{|S(k,\mu)|\}. \tag{5.5}$$

The error E used for examining the estimators' performance is defined as the logarithmic ratio between the estimated amplitude envelope's power $\widehat{\overline{S}}$ and the clean speech signal's amplitude envelope power $\overline{S}$ in each frame k and frequency bin μ:

$$E(k,\mu) := 20\log_{10}\left(\widehat{\overline{S}}(k,\mu)\right) - 20\log_{10}\left(\overline{S}(k,\mu)\right) \tag{5.6}$$

$$= 20\log_{10}\left(\frac{\widehat{\overline{S}}(k,\mu)}{\overline{S}(k,\mu)}\right). \tag{5.7}$$

In the testing data set, $K = 2110$ frames were accepted for containing voiced speech data and were subsequently processed. The performance is later assessed by computing two measures: the average of absolute error from (5.7):

$$P_{\mathrm{avg}}(\mu) = \frac{1}{K}\sum_{k=1}^{K}|E(k,\mu)|, \tag{5.8}$$

and the maximum absolute error

$$P_{\max}(\mu) = \max_k\{|E(k,\mu)|\} \tag{5.9}$$

for every bin μ separately. Moreover, the different noise conditions (speeds) are evaluated independently.

For each relevant frame, the estimation task consists in the minimization of a cost function, which is defined here as the sum of absolute estimation error defined in (5.7):

$$\sum_{\mu=1}^{M}|E(k,\mu)| \to 0 \quad \forall k \tag{5.10}$$

Codebook-Based Estimation

On modern computers, unsupervised vector quantization methods like the codebook approach are mainly used to reduce the dimension of a feature space in order to save memory space and computation power (Wendemuth, p. 126). However, it is still common use in embedded systems, where these resources are limited (Krini, p. 26).

In the training step, codebooks store information on prototype feature vectors extracted from a database. In the testing step, incomplete feature vectors can be matched against all codebook entries by use of a cost or distance function. The codebook entry that best matches the incomplete vector is used to fill in missing features. In the task at hand, missing features are frequency bins where the corresponding filter coefficients lie below a certain threshold, thus indicating a low SNR. In these bins, the estimate amplitude value is dominated by the noise power, and the speech power is bound to be much lower.

A codebook was implemented using the well-known Linde-Buzo-Gray (LBG) algorithm for training on clean speech signal envelopes. The number of feature vectors was set to $N_{\mathrm{cb}} = 256$ and each vector consisted of M amplitude values.

Non-Parametric Estimation

In addition to the codebook approach, two methods based on bin-wise computation of available spectral information were investigated. In frequency bins where the noise estimation is accurate, the noise reduction filter will output coefficients near 1 (corresponding to 0 dB on a logarithmic scale). Here, the microphone spectrum provides a fairly good estimate for the speech spectrum.

Microphone. The noisy (also called microphone) signal's amplitude spectrum X is smoothed and used as a first approximation as defined in (5.3):

$$\widehat{\overline{S}}_{\mathrm{Mic}}(k,\mu) := \overline{X}(k,\mu). \tag{5.11}$$

As is to be expected, this approximation only works well where no or very little noise is present. It can however be useful as a source of information for other estimators.

Average. While similar to the microphone estimator, this method indirectly takes into account the filter coefficients $H(k,\mu)$ available from a preceding noise reduction. These range from the spectral floor up to a value of 1 and contain information on how much noise is present in each bin μ and frame k. Higher values indicate a better SNR.

The actual implementation relies on the smoothed noise-reduced signal's amplitude spectrum $\widehat{\overline{S}}_{\mathrm{NR}}$ as defined in (5.4):

$$\widehat{\overline{S}}_{\mathrm{Avg}}(k,\mu) := \frac{1}{2}\left(\overline{X}(k,\mu) + \widehat{\overline{S}}_{\mathrm{NR}}(k,\mu)\right). \tag{5.12}$$

Parametric-Model-Based Estimation

In a separate processing stage, the frequency-domain envelope of the original speech signal is estimated. The estimated envelope is then imprinted on the generated excitation signal in order to reconstruct the original speech spectrum as accurately as possible.

The envelope estimation is implemented in a novel way that can be applied on-line at a very low computational power cost. First, he noise suppression filter's coefficients are analyzed to distinguish bins with good and bad speech signal power estimation. The well-estimated bins are then used as supporting points for the estimation of the badly estimated ones.

The estimation itself consists in the use of a parametric model for the envelope shape. For extrapolating towards low frequencies, a logarithmic parabola is used. The shape is derived from observed envelope shapes and is modified by several parameters, such as the parabola's curvature.

Several features can easily be extracted from the signal, such as the position of the lowest well estimated frequency bin. These features are used to determine a good estimate for the curves' parameters.

Linear Extrapolation. As stated in the task assignment, disturbances in the low frequency spectrum are prevalent in hands-free automobile telecommunication systems. It is a typical situation that a number of consecutive low-frequency bins do not offer enough speech signal information for estimating the envelope at that frequency bin. In a first approach, it is tried to approximate missing envelope information by constructing a straight line in the logarithmic amplitude spectrum.

The lowest well-estimated bin's index in a frame k is called $\mu^{\circ}(k)$, and the logarithmic amplitude value in that place is $\widehat{\overline{S}}_{\mathrm{Avg}}(k, \mu^{\circ}(k))$. The line is fixed to this point, retaining one degree of freedom, namely its slope $m_{\mathrm{Lin}}(k)$. This parameter will be estimated from the signal's features as described in subsection 5.3.4.

The estimated amplitude values are computed iteratively:

$$\widehat{\overline{S}}_{\mathrm{Lin}}(k, \mu) := \frac{\widehat{\overline{S}}_{\mathrm{Lin}}(k, \mu + 1)}{m_{\mathrm{Lin}}(k)}. \tag{5.13}$$

This gives a straight line in logarithmic space depending on the slope $m_{\mathrm{Lin}}(k)$, which must be set to a value leading to a good approximation of the actual spectral amplitudes.

Parabolic Extrapolation. It is not known directly whether the low-frequency band contains an obscured speech formant or not. As an extension to the aforementioned linear extrapolation, the new method proposed features a parabolic estimation model parametrized by the slope of the reference estimation to guess the optimal course of the envelope. It is more flexible than the former method, including it as a special

case. Hence, it bears a equal-or-better potential performance but introduces higher worst-case errors.

From the reference estimation $\widehat{\overline{S}}_{\mathrm{Avg}}$, a slope m_{R} is computed:

$$m_{\mathrm{R}} := \sqrt{\frac{\widehat{\overline{S}}_{\mathrm{Avg}}\left(k, \mu^{\circ}\left(k\right)+1\right)}{\widehat{\overline{S}}_{\mathrm{Avg}}\left(k, \mu^{\circ}\left(k\right)-1\right)}}. \tag{5.14}$$

This reference slope is then used to determine the parabola's initial slope $m^{\circ}\left(k\right) = m\left(k, \mu^{\circ}\right)$ at frequency μ°. The curve is fixed by setting the remaining degree of freedom, called here the curvature $J\left(k\right)$. For an illustration, see figure 5.2.

Additionally, a restriction on the curve's maximum slope (in positive frequency direction) $m_{\max}\left(k\right)$ is introduced, and the minimum slope (also in positive frequency direction) $m_{\min}\left(k, \mu\right)$ is restricted to that of the estimated noise spectrum $\widehat{D}$:

$$m_{\min}\left(k, \mu\right) := m_{\widehat{D}}\left(k, \mu\right) := \frac{\left|\widehat{D}\left(k, \mu+1\right)\right|}{\left|\widehat{D}\left(k, \mu\right)\right|}. \tag{5.15}$$

Given the above-mentioned parameters, the parabola $\widehat{\overline{S}}_{\mathrm{Par}}\left(k, \mu\right)$ is calculated iteratively by counting the bin index μ down from μ° with the following algorithm:

$$\widehat{\overline{S}}_{\mathrm{Par}}\left(k, \mu\right) := \frac{J\left(k\right) \cdot \widehat{\overline{S}}_{\mathrm{Par}}\left(k, \mu+1\right)}{\min\left\{m_{\max}\left(k\right), \max\left\{m_{\widehat{D}}\left(k, \mu\right), m\left(k, \mu\right)\right\}\right\}}, \tag{5.16}$$

where the slope $m\left(k, \mu\right)$ is:

$$m\left(k, \mu\right) := \begin{cases} m^{\circ}\left(k\right) & \text{for } \mu = \mu^{\circ}, \\ \dfrac{\widehat{\overline{S}}_{\mathrm{Par}}\left(k, \mu+2\right)}{\widehat{\overline{S}}_{\mathrm{Par}}\left(k, \mu+1\right)} & \text{otherwise} \end{cases} \tag{5.17}$$

This results in a parabolic shape that is determined by the parameters initial slope $m^{\circ}\left(k\right)$, curvature $J\left(k\right)$, and minimum slope $m_{\min}\left(k\right)$, as well as on the course of the noise spectrum via $m_{\widehat{D}}$.

5.3.4. Parameter Optimization

Above, the construction of a parabolic curve depending on parameters was described. It was not yet defined how these parameters are derived from the features extracted from the available information. This will be described in the following.

Functional dependencies between the features and the optimal parameters are established off-line from a training data set to obtain a simple constant or linear dependency from a feature to a parameter. This way, the computational power cost under operating conditions stays minimal.

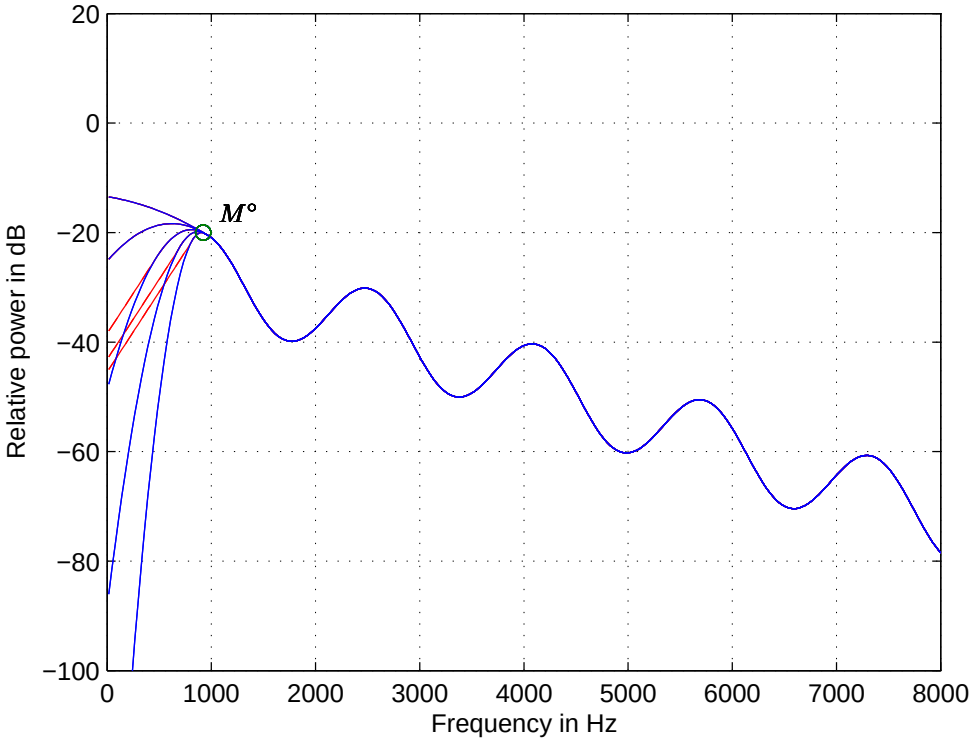

Figure 5.2.: Illustration of parabolic extrapolation. From the hypothetical envelope spectrum to the right of the point M°, the envelope from M° on to the left can be extrapolated as a parabola. The curve is fixed in M°, and its slope is also set to match the envelope's one. One degree of freedom—the parabola's curvature—remains, several examples of which are presented here. Additionally, the parabola's slope can be restricted, which is indicated here by the straight red lines.

Detecting functional dependencies is performed in two steps. In the first one, a minimum search for the cost function is performed on the available parameters. This way, the optimum approximation that a model is capable of is established. The parameters defining this optimal approximation are saved together with the feature extracted from information that would be available also in testing conditions. In the second step, the tuples of features and corresponding optimal parameters are analyzed to detect approximate functional dependencies between them. If there seems to be a dependency, a zero- to second-order polynomial is fit on the data to represent it. This can be evaluated in the testing step.

Optimum Parameter Detection and Feature Extraction. Firstly, the testing data set is searched for frames in which reconstruction can likely be successfully performed. Criterion for exclusion is insufficient voiced excitation power as in (4.1), but with modified parameters. The band used goes from $f_{\text{low}} = 800\,\text{Hz}$ to $f_{\text{high}} = 2400\,\text{Hz}$, and the threshold value is $P_{\text{VUD}} = 2\,\text{dB}$.

In the remaining frames, the noisy signals were processed and the envelope estimation parameters for each gap in the smoothed spectrum were optimized to approximate the known clean speech smoothed spectrum. The cost function is defined in (5.10) above.

Apart from the optimized parameters for each frame, several features are extracted from the noisy speech signal. These feature are likely to carry valuable information on the corresponding optimal parameters. Note that no features can be taken from the known clean speech signals, but from the noisy speech signals only. Hence, the same information will be available under operating conditions.

The optimal parameters together with the corresponding features are treated as a tuple for each relevant frame. All of the tuples are sequentially written to a single file that will be used in the next step. About 350 tuples were extracted from the testing data set.

Estimation of Optimal Parameters from Features. With the help of a tool developed in the course of this work, it is possible to visualize dependencies between features and optimal parameters stored during the previous step. This is achieved by scatter plotting an arbitrary combination of up to three optimal parameter variables or feature variables. Each variable is represented by one axis in the plot, while each of the stored frame information tuples is represented by a point in the plot area.

Beyond the visualization, it is also possible to select and modify data points directly in the plot. Similarly to the well-known MATLAB feature, data points can be brushed in a chosen color, or they can be deleted from the data set. The difference to the native MATLAB feature is the fact that modifications are persistent even when choosing a different set of variables for display.

Modifications display in real time and can be undone by means of a revert option. This allows the user to intuitively look at the data from different perspectives and track data points through more than the three visual dimensions.

All in all, the developed tool is suitable for heuristically ruling out outliers and detection errors at first glance, as well as visually characterizing dependencies between

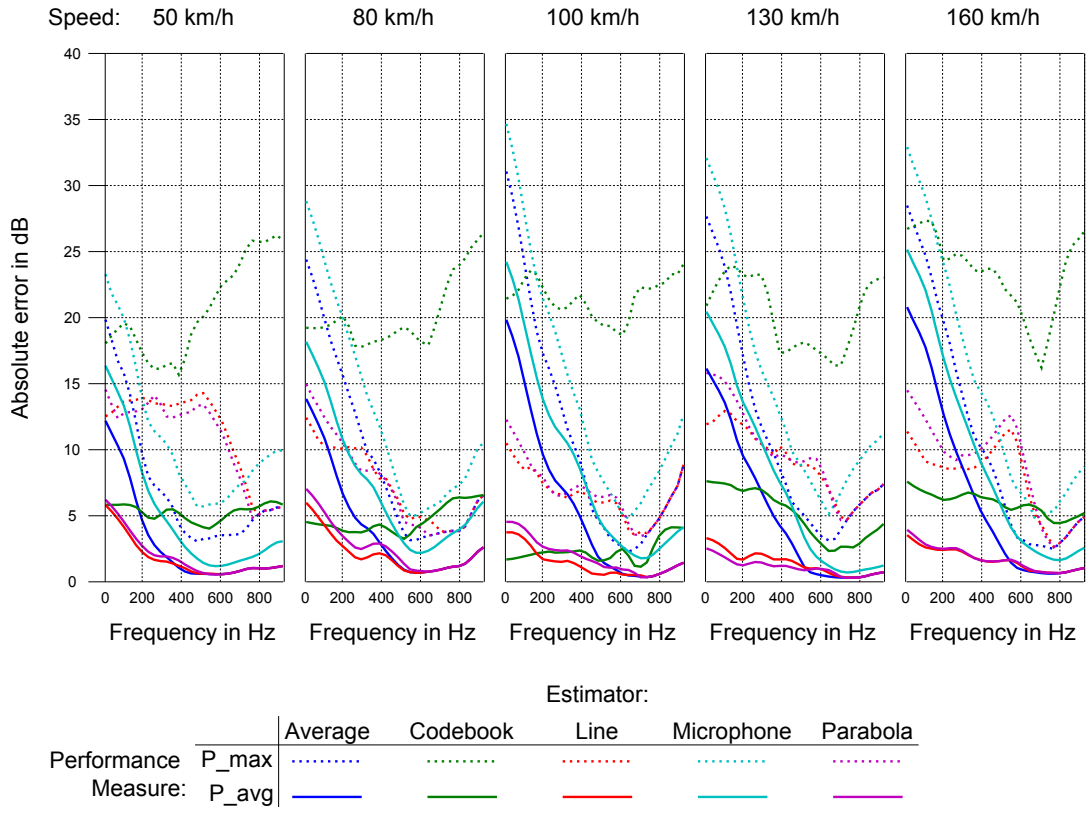

Figure 5.3.: Performance measure of different estimators without bias correction. The estimator names refer to the following variables defined in the text: Average: $\widehat{\widehat{S}}_{\mathrm{Avg}}$; Codebook: $\widehat{\widehat{S}}_{\mathrm{CB}}$; Line: $\widehat{\widehat{S}}_{\mathrm{Lin}}$; Microphone: $\widehat{\widehat{S}}_{\mathrm{Mic}}$; Parabola: $\widehat{\widehat{S}}_{\mathrm{Par}}$; P_max represents the worst-case estimation, while P_avg refers to the mean absolute estimation error.

the variables. The dependencies can then be appropriately approximated. If adequate features have been chosen in the first step, better performance can be can expected by using a model dependent on the feature compared to choosing just an optimized constant for each of the parameters.

Un-Biasing of the Estimators. In a preliminary evaluation on the training data set, an increasingly poor performance in lower frequencies was observed in all estimators except the codebook approach. The non-parametric approaches, which are based on the noisy signal's amplitude spectrum, show especially poor performance compared to the other estimators.

The evaluation is depicted in figure 5.3. The decrease in performance for lower frequencies is explained by the typical car-noise amplitude spectrum, which increases

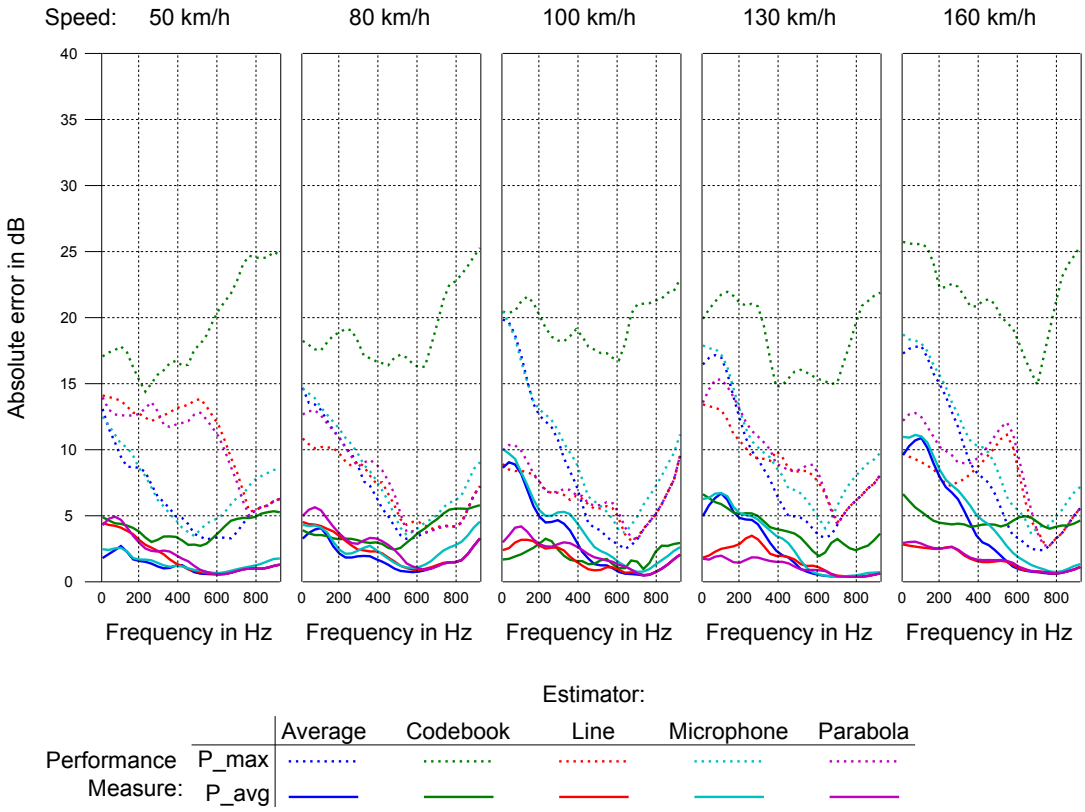

Figure 5.4.: Performance measure of different estimators that subtract a constant bias. The estimator names refer to the following variables defined in the text: Average: $\widehat{\widehat{S}}_{\mathrm{Avg}}$; Codebook: $\widehat{\widehat{S}}_{\mathrm{CB}}$; Line: $\widehat{\widehat{S}}_{\mathrm{Lin}}$; Microphone: $\widehat{\widehat{S}}_{\mathrm{Mic}}$; Parabola: $\widehat{\widehat{S}}_{\mathrm{Par}}$; P_max represents the worst-case estimation, while P_avg refers to the mean absolute estimation error.

towards low frequencies and thus masks the speech signal. Estimation is more prevalent in these bands, which is consistent with the original problem.

As can further be seen from figure 5.3, the Average and Microphone estimators' mean and maximum errors feature a near-parallel course toward low frequencies. This gives rise to the suspicion that a systematic error is involved. The test data set was therefore used to calculate the mean error for each frequency bin, estimator, and speed over all relevant frames. The resulting bias spectra were subtracted after estimation on the testing data set. The suspicion proved true for almost all cases, as the resulting estimation errors generally improved as depicted in figure 5.4.

5.4. Evaluation

Applying the resulting parametric estimation process on a testing data set yields better estimation results than the codebook approach, which is also much more expensive

in terms of memory and computation power requirements. Even the limited amount of available training data sufficed to improve the results.

While the "average" estimator gives a good reference even for some of the badly estimated frequency bins, it fails to provide a good estimation under low-SNR conditions. After un-biasing, it is still the best choice in situations with high SNR due to its very low complexity.

The codebook approach spreads the mean error evenly over the spectrum and thus performs better in very low frequencies where other estimators have difficulties, especially the non-parametric ones. It does not profit very much from un-biasing and has a singular advantage at 100 km/h in very low frequencies. It is not obvious why this is the case. Its demanding complexity in memory and computation power is not justified by its performance.

Under high-noise conditions, the "parabola" and "line" estimators yield the best results of the tested estimators, the latter slightly outperforming the former. However, the "parabola" estimation profits more from un-biasing than the "line" method. This makes it a viable choice, since the absolute computation power needed is only slightly higher than that of the "line" estimator.

6. Conclusion

In addition to the task assignment for this work, some useful spin-off achievements were attained. This chapter will first list what has been accomplished, including parts beyond academic concern. Unless otherwise stated, the following describes the author's work. After a discussion on the results of this work, a summary recapitulates and concludes the thesis.

6.1. Results

The following is a concise list of singular items adding to the outcome.

Implementation of a DSP Framework. As a basic training structure, a framework implementing a STFT and inverse STFT was created in MATLAB. The core function module is soft-coded to a high degree and can easily be adapted for other tasks, as arbitrary processing steps can be plugged in.

Automated Processing Bank. Moreover, an automated test bank system has been developed that makes use of the core function's flexibility. This system allows the textual definition of an arbitrary combination of files to process, and according processing parameters. Parameter names and one or more desired values can be listed. For each setting thus defined, the cross product of the parameter values and files is processed, and the output files named according to definition by a format string per parameter, filled in with the respective values used. Several Parameters can be combined so as not to build the cross-product for associated parameter values. A sequence of settings can be defined if desired, for example to perform noise reduction and formant boosting, which rely on totally parameter sets. This way, a large number of systematically-named output files can be produced with minimum effort, which is most useful for tasks like parameter tuning. For an example configuration see appendix B.

Signal Database. A signal database featuring in-car noise and appropriate Lombard speech at five different speeds was recorded. The files are systematically named and can be used for other applications. Details can be found in chapter 3.

Formant Detection Algorithm. Based on power spectral densities, a low-cost but reasonably robust formant detection algorithm is proposed. Subsequent processing modules can react to the presence of formants in each frequency bin by making use

of the normalized boosting function provided by the detection algorithm as described in section 4.2.

Analysis and Modification of the Recursive Wiener Filter. In the course of modifying the noise estimate serving as an input to a recursive Wiener filter, a mathematical analysis of the same was conducted (see subsection 4.4.1) yielding insight into its hysteresis characteristic and enabling arbitrary placement of the critical hysteresis flanks.

Gain Formant Boosting. Applying additional gain on frequency bins within formant was shown to improve intelligibility at the cost of naturalness, thus likely improving overall speech signal quality, compare subsection 4.6.4.

Patent Pending for Formant Boosting. A patent was filed by Nuance for the formant boosting process as a whole as well as for the single steps comprising formant detection, modification of the recursive Wiener noise reduction filter, and application of gain on formants in the speech signal. Enrolled inventors are Mohamed Krini, Ingo Schalk-Schupp, and Markus Buck. The international application number is PCT/US12/53666.

Patent Pending for Speech Signal Reconstruction. Another patent was filed by Nuance for the reconstruction process developed by Nuance as a whole as well as for the single steps including the author's envelope estimator. Enrolled inventors are Mohamed Krini, Ingo Schalk-Schupp, and Patrick Hannon. The international application number is PCT/US12/62549.

Automated Subjective Test. The whole cycle of preparing, compiling, and deploying a subjective test, up to gathering and evaluating the results has been automated to a high degree, allowing fast creation and adaption to new situations and specifications. An overview over the workflow covered is given in appendix C.

Envelope Estimation. A method for estimating the envelope of a speech signal strongly degraded by noise in lower frequencies was proposed. It was tested against other common methods and found to be very well suitable in place of other methods, especially in high-noise conditions. The module is used in a speech signal reconstruction process.

Interactive Tool for Relation Discovery. While searching for possible functional relations of signal features and estimation parameters that would help in developing effective estimators, a graphical user interface was developed that speeds up the process of discovering such relations by offering simple and intuitive access to 2D and 3D scatter plots based on optimized-parameter data provided by the aforementioned automated processing bank.

6.2. Discussion

Based on the results reported in the respective chapters, the two main parts of this thesis—formant boosting and envelope estimation—will be discussed in the following. As the work on the thesis was enabled by and done at Nuance, the main focus lay on product relevance. Computational efficiency was a main concern, as the product palette consists mainly of embedded systems, where resources are limited. The author's MATLAB implementations of both proposed methods were adapted by Nuance in their SSE product.

6.2.1. Formant Boosting

This method addresses the task to protect missing fundamental frequencies in degraded single-channel speech signals from being overly attenuated. It is based of established techniques and additionally takes into account the course of the smoothed speech signal amplitude spectrum.

Relevance Noise reduction is an area that has been well researched in the past. In today's telecommunication systems, noise reduction filters are commonly employed. Automobile manufacturers and others constantly request improved noise reduction solutions at medium and high noise conditions.

Effectiveness In subsection 4.6.4, a hypothesis on the weighting of intelligibility and naturalness of a speech signal contributing to an overall rating was discussed based on the assumption that two test passes had been identical in the material presented. These test passes can be compared to each other, albeit in a limited way. Still, intelligibility can be said to be more important to most subjects than naturalness of a speech signal. Evidence from the third pass suggests that formant boosting features a higher intelligibility at the cost of a lower naturalness. Combining the two theses implies that formant boosting generally improves the overall quality of a speech signal degraded by noise.

Efficiency The main impact on computational complexity in formant boosting is introduced by the formant detection process. However, the smoothing of the amplitude spectrum with two separate constants developed is a very efficient method that alleviates the demands on both memory and computation power compared to the use of a linear predictor—the result of which needs to be transformed to the frequency domain—or a polynomial fit that requires a high amount of calculation power. Moreover, the proposed method provides additional information on the formants' respective widths.

Outlook With the available data as they were, adequate results were achieved. As the proposed formant detection algorithms work on a per-frame basis, further research could investigate relations between the formants in time. For example, a likelihood

measure for the presence of formants based on previous frames could help to make the detection less conservative and more robust. Another approach might modify the noise power estimation based on the detection of formants. Here, feedback effects would have to be taken into account. While the application of gain in formants has been thoroughly investigated in this work, the modification of the recursive Wiener filter could be adapted to more settings aside from formant boosting. It remains to be tested whether formant boosting has any effect on automatic speech recognition.

6.2.2. Low-Frequency Envelope Estimation

The estimator solves a part of the speech signal reconstruction problem assuming the presence of a harmonic structure and providing an estimated coarse amplitude spectrum in the lower frequency band.

Relevance Although missing fundamental frequencies can be amended by human aural perception, speech quality improves with more natural sound. The reconstruction of these speech signal components leads to a better listening experience. In order to develop a functional reconstruction process, Nuance needed an envelope estimation with an objectively good performance.

Effectiveness Other estimators were compared to the proposed one and found to be unfeasible in the case of a codebook, or too badly performing. The proposed estimator together with the harmonic reconstruction results in better-sounding speech signals and was also adapted by Nuance from the author's MATLAB implementation in their SSE.

Efficiency While the harmonic reconstruction uses redundant harmonic information from the degraded signal, estimation of the envelope relies on singular features, the extraction of which requires hardly any calculation power or memory. Given the features, envelope estimation is also economic, as it iteratively constructs a polynomial curve based on a small set of parameter values.

Outlook The reference amplitude spectrum used in this work—compare (5.12)—is based on a fixed spectral floor in the noise reduction filter. This could be made more flexible by introducing a weighting factor between the noisy and the noise reduced amplitude spectra. Future work could delve into additional feature extraction concepts and derivation of parameters for the parametric models using the new tool mentioned above. A bigger database could also help in the process. Finally, speech reconstruction could be combined with the formant boosting method for additional improvement. This could include a better envelope estimation using information from the formant detection process.

6.3. Summary

In this thesis, several methods for speech signal enhancement are investigated. Using a newly-created signal database, enhancement is achieved by two methods. On the one hand, formants are used to selectively prefer important speech information in the noise reduction process. On the other hand, the amplitudes of missing fundamental frequency components are reconstructed from available signal information.

After defining the type of integral transform used in this work and introducing state-of-the-art noise estimation, basic noise reduction algorithms including the Wiener filter and an existing recursive modification thereof are described. Also, a common speech production model that separates excitation from vocal tract information in the shape of formants is given, along with a portray of the Lombard effect that influences formant characteristics.

Next, a signal database is presented, that is tailored to the requirements placed by subsequent investigations. Proper acoustic properties are ensured by in-car recording. Clean speech signals and noise signals are recorded separately in order to compute performance measures. Moreover, the Lombard effect is provoked in speakers so as to approach more realistic conditions.

Based on the presence of formants, a novel approach on noise reduction in degraded speech signals is proposed. Detection of the formants is performed when voiced excitation occurs. Several detection methods are compared, including a linear predictive coding, IIR smoothing, polynomial fitting, and IIR smoothing with two different smoothing constants. From the detected formants, a boosting function is constructed that features a pre-defined shape for each formant and controls subsequent boosting steps. Two such steps are proposed, the first of which consists in a novel extension of the recursive Wiener filter that uses the boosting function to become sensitive to formant presence, thus potentially allowing more speech information to pass into the noise reduced signal than existing noise reduction filters. The second one directly imposes gain on the detected formants to attain a better broadband signal-to-noise ratio. A subjective listening test is designed to provide a comparative mean opinion score for the second method, showing that formant boosting improves speech intelligibility at the cost of a natural acoustic impression. Heuristic arguing indicates that the overall subjective quality rating become better through the application of formant gain.

The reconstruction of low-frequency speech signal components is shortly presented. It consists of an excitation estimator, an envelope estimator, and a mixer stage, where the envelope estimator is the author's work, which is of concern in this thesis. Several envelope estimators are described and compared, opposing parametric and non-parametric groups. Two parametric models for low-frequency envelopes are proposed. Respecting separation of training and test data, parameter optimization is performed based on specific features extracted from the degraded speech signal. Finally, all of the estimators under investigation are evaluated using a logarithmic distance measure. The two newly-developed parametric models yield the best performance while also demanding very low computational effort.

A. Measurement Reports

A.1. Noise Measurement

Overview

Objective of the measurement / short description

Noise measurements for later mixing with clean speech. NoiseBook recordings can be used to provoke Lombard effect in speakers.

Date of the measurement

April 19, 2012

Persons involved

Speakers: Mohamed Krini, Timo Matheja, Ingo Schalk-Schupp
Setup: Meik Pfeffinger

Vehicle

Audi A6 (former SVOX vehicle)

Microphone positions

Eight microphones in the roof near the A-pillars, whereof four on the driver side, and the other four on the co-driver side. Not all of these were actually used, see channel mapping in table A.1 below.

Two microphones in the roof control unit, one on each side. Two microphones in the rear-seat roof, one on each side. Additionally, NoiseBook measurements were performed in different positions.

Details

Channel Mapping

The recording channels are mapped to different filenames as shown in table A.1

Laptop / Recording Software

"Demonstrator box" in trunk for in-car recordings using Audition 1.5. Laptop for NoiseBook recordings using native software.

Table A.1.: Channel mapping in the noise measurement recording session

Number	Microphone position	Filename extension
Channel 1	Control unit driver side	raw: DBE Driver cut: DbeDriver
Channel 2	Control unit co-driver side	raw: DBE CoDriver cut: DbeCodriver
Channel 3	Rear seat driver side	raw: PassengerBehindDriver cut: BackseatDriverSide
Channel 4	Rear seat co-driver side	raw: PassengeBehindCoDriverr cut: BackseatCodriverSide
Channel 5	Leftmost roof microphone on driver side	raw: DriverLeft cut: DriverLeftmost
Channel 6	Rightmost roof microphone on co-driver side	raw: CoDriverRight cut: CodriverRightmost
Channel 7	Non-functional	-
Channel 8	Non-functional	-
NoiseBook left	[Condition]	raw: [Session] cut: NoiseBookLeft
NoiseBook right	[Condition]	raw: [Session] cut: NoiseBookRight

Data Post-Processing

Raw recordings can be found in the subfolder:

```
raw recordings\Audition\[Session]
```

and:

```
raw recordings\NoiseBook
```

Cut sections in subfolder:

```
cut
```

Naming pattern is:

```
A6_[condition]_[NoiseBook position]_[warnings]_[↩
   channel].wav
```

`condition` contains speed and NoiseBook position; `warnings` may be empty.

Detailed Recording Protocol

Conditions

Measurements were performed for the conditions shown in table A.1.

Table A.2.: Noise conditions recorded

		Speed (km/h)				
Condition	NoiseBook position	50	80	100	130	160
All windows closed	Driver	√	√	√[1]	√	√
All windows closed	Co-driver	√	√	√	√[2]	√
All windows closed	Backseat co-driver side			√	√	
Driver window open	Driver		√[3]	√	√	
Driver window open	Co-driver		√	√	√	
Driver window open	Backseat co-driver side			√	√	

[1] Supplementary record acquired on May 30, 2012 with different microphone.
[2] Cut section contains artifacts, but smaller areas might still be usable for some applications.
[3] Cut section contains mobile phone artifacts in the NoiseBook recording only.

Remarks

Noise events created by overtaking cars are contained in the recordings. Also, short noise events, like clicking and popping, may appear. This should be considered when near-stationary noise is assumed.

Sampling rate for all files is 44100 Hz.

Recording Details

Four sessions were recorded in total, each including several conditions. The first three sessions were cut to receive the noise measurements only. The fourth session contains speech and is for reference only.

The conditions are distributed over the recording sessions as shown in table A.1.

Cutting Protocol

Cutting was performed by first inserting cue markers into the NoiseBook wave files only in Audition 3.0. The range markers denoting the start and the end of a desired section were named according to the respective condition.

The NoiseBook recordings were then aligned to match the accompanying demonstrator recording. Exactly one cue point was defined indicating the start (in samples) of the latter relative to the NoiseBook recording. This cue point was used for synchronizing sample offsets between the independent NoiseBook recordings and demonstrator recordings. NoiseBook channels were separated into two mono wave files for cutting tool compliance.

Since Audition does not support exporting cue markers, they were then extracted and exported as CSV files using a tool named "CueListTool", which is freely available at `http://www.tonbandstimmen.de/cuelisttool/`.

An appropriate preset was created, the definition of which can be found below. It is suitable for importing the created files in MS Excel. Absolute sample positions are

Table A.3.: Conditions contained in recording sessions

First_test	100kmh_openWinDriver_NbDriver
	100kmh_openWinDriver_NbCodriver
	100kmh_openWinDriver_NbBackseatCodriver
	100kmh_NbDriver
	130kmh_NbCodriver_INDICATORARTIFACTS_PAVINGARTIFACTS
	130kmh_openWinDriver_NbCodriver
	130kmh_openWinDriver_NbDriver
	130kmh_openWinDriver_NbBackseatCodriver
	100kmh_NbBackseatCodriver_MOBILEARTIFACTS
	130kmh_NbBackseatCodriver
Second_test	100kmh_NbCodriver
	80kmh_NbCodriver
	80kmh_NbDriver
	80kmh_openWindow_NbDriver_MOBILEARTIFACTS
	80kmh_openWindow_NbCodriver
Third_test	160kmh_NbCodriver
	160kmh_NbDriver
	50kmh_NbCodriver
	50kmh_NbDriver_TUNNELARTIFACTS

automatically created, assuming that each NoiseBook recording session starts before demonstrator recording sets in.

Using the sample offsets and condition names obtained this way, actual cutting was performed by inserting them into the Nuance cutting tool separately for NoiseBook and demonstrator files.

Preset Definition

Cue ranges :

```
%begin%tab%end%tab"%label"%tab=A%cue-$A$1%tab=B%↩
    cue-$A$1
```

Cue points :

```
%begin%tab%tab"%label"
```

%sep :

```
-
```

Footer :

```
start%tabend%tablabel%tabstart%tabend%nlFile: %↩
   file
```

Time format :

```
%p
```

A.2. Lombard Speech Measurement

Overview

Objective of the measurement / short description

Lombard speech measurements for mixing with previously recorded noise. NoiseBook recordings were played back to speakers to provoke Lombard effect.

Date of the measurement

May 3, 2012

Persons involved

Speakers: Ilona Holtz, Frank Dangel, Lars Tebelmann
Setup: Ingo Schalk-Schupp

Vehicle

Audi A6 (former SVOX vehicle)

Microphone positions

Eight microphones in the roof near the A-pillars, whereof four on the driver side, and the other four on the co-driver side.

Two microphones in the roof control unit, one on each side. Two microphones in the rear-seat roof, one on each side.

Not all of these were actually used, see channel mapping in table A.4 below.

Details

Channel Mapping

The recording channels are mapped to different filenames as shown in table A.4

Table A.4.: Channel mapping in the Lombard recording session

Number	Microphone position	Filename extension
Channel 1	Control unit driver side	DbeDriver
Channel 2	Control unit co-driver side	-
Channel 3	Rear seat driver side	-
Channel 4	Rear seat co-driver side	-
Channel 5	Leftmost roof microphone on driver side	DriverLeft
Channel 6	Rightmost roof microphone on co-driver side	-
Channel 7	Non-functional	-
Channel 8	Non-functional	-
NoiseBook left	[Condition]	NoiseBookLeft
NoiseBook right	[Condition]	NoiseBookRight

Laptop / Recording Software

"Demonstrator box" in trunk for in-car recordings using Audition 1.5. Laptop for NoiseBook recordings using native software.

Data Post-Processing

Naming pattern is:

```
A6_Lombard_[speed]_[gender][speaker]_[sheet][sentence]↩
   _[channel].wav
```

`speed` is a zero-padded three-characters number indicating speed condition. `gender` can be `m` for male or `f` for female speakers. `speaker` is a zero-padded two-characters number identifying the actual speaker together with `gender`. `sheet` is a capital alphabetic character. Each speaker read aloud a different sheet. `sentence` is the zero-padded two-characters number of the sentence on the respective sheet. Together with the latter, it identifies the actual sentence spoken. `channel` marks the microphone used.

Detailed Recording Protocol

Conditions

For every speaker, a sheet with 20 German idioms in random order was read aloud, four sentences for each of the speeds: 50 km/h, 80 km/h, 100 km/h, 130 km/h, and 160 km/h.

Remarks

Sampling rate for all files is 44100 Hz.

Recording Details

The idioms used were:

- Aller guten Dinge sind drei.
- Der frühe Vogel fängt den Wurm.
- Müßiggang ist aller Laster Anfang.
- Ein blindes Huhn findet auch mal ein Korn.
- Der Klügere gibt nach.
- Reden ist Silber, Schweigen ist Gold.
- Pech im Spiel – Glück in der Liebe.
- Die Ratten verlassen das sinkende Schiff.
- Man soll den Tag nicht vor dem Abend loben.
- Ehre, wem Ehre gebührt!
- Ehrlich währt am längsten.
- Kindermund tut Wahrheit kund.
- Irren ist menschlich.
- Wer zuletzt lacht, lacht am besten.
- Liebe geht durch den Magen.
- Der Apfel fällt nicht weit vom Stamm.
- Morgenstund hat Gold im Mund.
- Wer anderen eine Grube gräbt, fällt selbst hinein.
- Schmiede das Eisen, solange es heiß ist!
- Geld allein macht nicht glücklich.

Cutting Protocol

Cutting was performed the same way as the noise measurements above. Before and after each utterance, one second of silence was allowed as exactly as possible.

B. Example Configuration for Automated Processing Bank

An example configuration actually used for `FbTestBank.m` is given. It builds the cross product of any parameters given and calls `formantBoosting.m` for each instance and for each file. For example, it would process the file

`A6_Noisy_050_m00_C01_DbeDriverside.wav`

to produce one output file for each combination of the five smoothing time parameter sets with the two power correction values true and false. All remaining parameters contain only one instance, so the cross product's magnitude does not grow with them. Given 20 input files per gender, a total of $2 \cdot 20 \cdot 10 = 400$ output files are automatically created in one go. These can be investigated or evaluated in a separate step.

```
parm = {...
    {...
        {{''}, {'%s'}, {... % input filenames
            'A6_Noisy_050_m00_C01_DbeDriverside'
            'A6_Noisy_050_m00_C02_DbeDriverside'
            'A6_Noisy_050_m01_D01_DbeDriverside'
            'A6_Noisy_050_m01_D02_DbeDriverside'

            'A6_Noisy_080_m00_C05_DbeDriverside'
            'A6_Noisy_080_m00_C06_DbeDriverside'
            'A6_Noisy_080_m01_D05_DbeDriverside'
            'A6_Noisy_080_m01_D06_DbeDriverside'

            'A6_Noisy_100_m00_C201_DbeDriverside'
            'A6_Noisy_100_m00_C202_DbeDriverside'
            'A6_Noisy_100_m01_D201_DbeDriverside'
            'A6_Noisy_100_m01_D202_DbeDriverside'

            'A6_Noisy_130_m00_C09_DbeDriverside'
            'A6_Noisy_130_m00_C10_DbeDriverside'
            'A6_Noisy_130_m01_D09_DbeDriverside'
            'A6_Noisy_130_m01_D10_DbeDriverside'

            'A6_Noisy_160_m00_C13_DbeDriverside'
            'A6_Noisy_160_m00_C14_DbeDriverside'
```

```
    'A6_Noisy_160_m01_D13_DbeDriverside'
    'A6_Noisy_160_m01_D14_DbeDriverside'
    }
}

% some static parameters
{{'estimatorName'      }, {[]}, {'IMCRA'}}
{{'displayLiveSpectra' }, {[]}, {false}  }
{{'displaySpectrograms'}, {[]}, {false}  }
{{'playSynth'          }, {[]}, {false}  }

% formant boosting specific parameters
{{'filterName'}, ...
    {[]}, ...
    {'WienerRecursiveModified'}
}

{{''}, {'_FB'}, {''}} % formant boosting tag

% voiced/unvoiced-detection thresh.
{{'fbVudThreshold'}, {[]}, {...
        1
    }
}

% formant boosting filter characteristic
{{'filterParameters'} {'_Post-06dB'} {...
       % InrMinusDbDefault, InrMinusDbBoosted,
       % InrPlusDbDefault,  InrPlusDbBoosted
       % postGainDbDefault, postGainDbBoosted
       % spectralFloorDb
       {{9.0309,  9.0309, ...
        10.2666, 10.2666, ...
         0.0000,  6.0000, ...
        -12            }}
    }
}

% smooth boosting function in time
{
    {
        'fbSmoothTime', ...
        'fbSmoothTimeUp', ...
        'fbSmoothTimeDown'}, ...
```

```
            {'_T%g' 'U%.2g' 'D%g'}, ...
            {
            0 0.0 0.0
            2 0.8 1.2
            2 1.0 1.5
            2 1.2 1.8
            2 1.5 2.0
          }
        }

        % remove theoretical power gain
        {{'bPowerCorrection'}, {'_Eq%g'}, {
                false
                true
            }
        }

        {{'fbSmoothing' 'fbSmoothingParm'}, {[] []}, {
                'LPC'    15
            }
        }

        {{'fbNFormant'}, {'_N%g'}, { % # of formants
                4
            }
        }

        {{'fbWinShape'}, {'_%s'}, {...
                'Gauss'
            }
        }

        {{'fbWinWidth'}, {'W%g'}, {...
                 800
            }
        }

        {{'fbSubtractNoise'}, {[]}, {
                false
            }
        }
    }
    {...
        {{''}, {'%s'}, {... % input filenames
```

```
    'A6_Noisy_050_w00_A01_DbeDriverside'
    'A6_Noisy_050_w00_A02_DbeDriverside'
    'A6_Noisy_050_w01_B201_DbeDriverside'
    'A6_Noisy_050_w01_B202_DbeDriverside'

    'A6_Noisy_080_w00_A05_DbeDriverside'
    'A6_Noisy_080_w00_A06_DbeDriverside'
    'A6_Noisy_080_w01_B205_DbeDriverside'
    'A6_Noisy_080_w01_B206_DbeDriverside'

    'A6_Noisy_100_w00_A201_DbeDriverside'
    'A6_Noisy_100_w00_A202_DbeDriverside'
    'A6_Noisy_100_w01_B209_DbeDriverside'
    'A6_Noisy_100_w01_B210_DbeDriverside'

    'A6_Noisy_130_w00_A09_DbeDriverside'
    'A6_Noisy_130_w00_A10_DbeDriverside'
    'A6_Noisy_130_w01_B213_DbeDriverside'
    'A6_Noisy_130_w01_B214_DbeDriverside'

    'A6_Noisy_160_w00_A13_DbeDriverside'
    'A6_Noisy_160_w00_A14_DbeDriverside'
    'A6_Noisy_160_w01_B217_DbeDriverside'
    'A6_Noisy_160_w01_B218_DbeDriverside'
    }
}

% some static parameters
{{'estimatorName'      }, {[]}, {'IMCRA'}}
{{'displayLiveSpectra' }, {[]}, {false}  }
{{'displaySpectrograms'}, {[]}, {false}  }
{{'playSynth'          }, {[]}, {false}  }

% formant boosting specific parameters
{{'filterName'}, ...
    {[]}, ...
    {'WienerRecursiveModified'}
}

{{''}, {'_FB'}, {''}} % formant boosting tag

% voiced/unvoiced-detection thresh.
{{'fbVudThreshold'}, {[]}, {...
        1
```

```
    }
}

% formant boosting filter characteristic
{{'filterParameters'} {'_Post-07dB'} {...
        % InrMinusDbDefault, InrMinusDbBoosted,
        % InrPlusDbDefault,  InrPlusDbBoosted
        % postGainDbDefault, postGainDbBoosted
        % spectralFloorDb
        {{9.0309,  9.0309, ...
         10.2666, 10.2666, ...
          0.0000,  7.0000, ...
         -12            }}
    }
}

% smooth boosting function in time
{
    {
        'fbSmoothTime', ...
        'fbSmoothTimeUp', ...
        'fbSmoothTimeDown'}, ...
    {'_T%g' 'U%.2g' 'D%g'}, ...
    {
        0 0.0 0.0
        2 0.8 1.2
        2 1.0 1.5
        2 1.2 1.8
        2 1.5 2.0
    }
}

% remove theoretical power gain
{{'bPowerCorrection'}, {'_Eq%g'}, {
        false
        true
    }
}

{{'fbSmoothing' 'fbSmoothingParm'}, {[] []}, {
        'LPC'    15
    }
}
```

```
        {{'fbNFormant'}, {'_N%g'}, { % # of formants
                4
            }
        }

        {{'fbWinShape'}, {'_%s'}, {...
                'Gauss'
            }
        }

        {{'fbWinWidth'}, {'W%g'}, {...
                800
            }
        }

        {{'fbSubtractNoise'}, {[]}, {
                false
            }
        }
    }
};
```

C. Workflow in Automated Subjective Testing

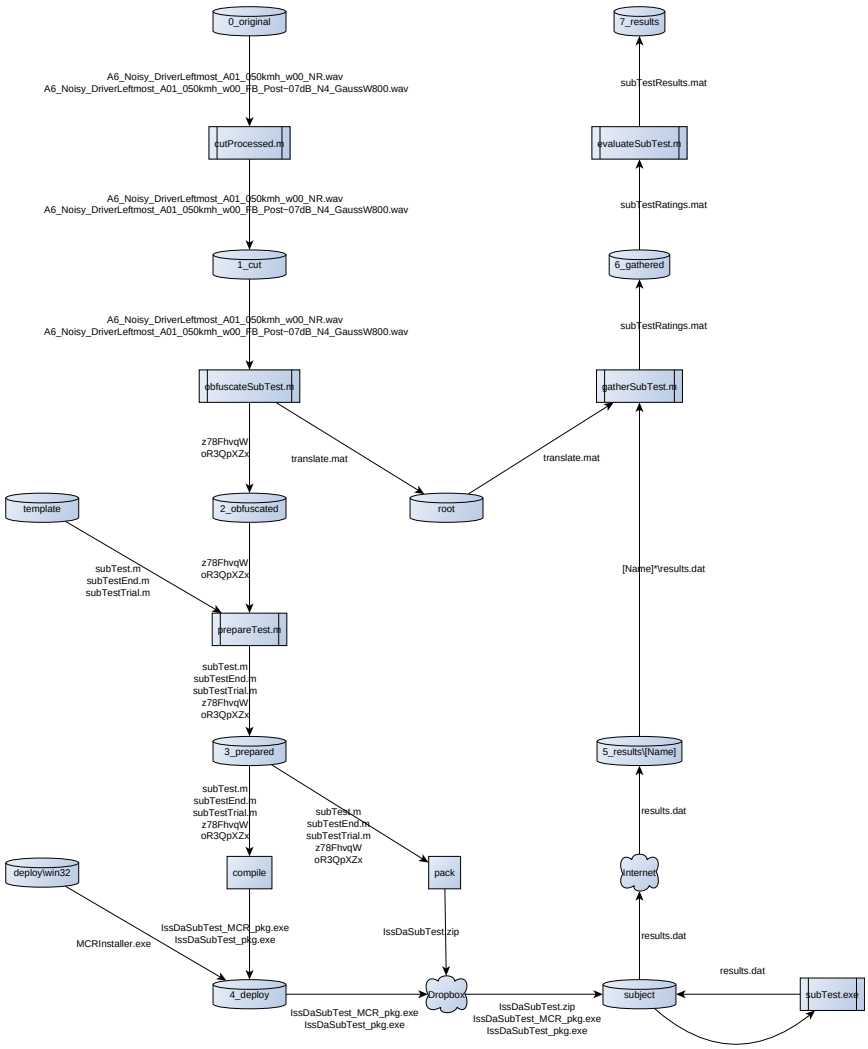

Figure C.1.: Workflow in automated subjective testing.

Acronyms

Notation	Description	Page List
CCR	comparison category rating	35, 36
CMOS	comparative mean opinion score	x, xi, 35, 36, 38–41, 43–48, 66
DFT	discrete Fourier transform	20
DSP	digital signal processor	2, 62
DTFT	discrete-time Fourier transform	4
GUI	graphical user interface	36, 37, 63
IIR	infinite impulse response	6, 21, 22, 51, 53, 66
IMCRA	improved minimum-controlled recursive averaging	5–7
INR	input-to-noise ratio	3, 9, 10, 19, 20, 26, 28, 29, 31
ITU	International Telecommunication Union	35
ITU-T	ITU Telecommunication Standardization Sector	35
LBG	Linde-Buzo-Gray	54
LPC	linear predictive coding	20, 23
MCRA	minimum-controlled recursive averaging	6
MSE	mean squared error	18
PSD	power spectral density	7, 19, 21, 22
SNR	signal-to-noise ratio	vii, 3, 7, 8, 16, 17, 25, 34, 54, 61, 66
SSE	speech signal enhancement	2, 16, 20, 23, 64, 65
STFT	short-term Fourier transform	2, 3, 7, 12, 21, 25, 50, 62

Symbols

Notation	Description	Page List
α	overestimation factor	9, 10, 26–33
β	spectral floor	9, 10, 26, 28–33
$\widehat{D}$	estimated complex noisy speech signal in frequency domain	3, 5–9, 26, 50, 56
D	complex noise signal in frequency domain	5, 6, 8
d	noise signal in time domain	3, 5, 7, 50
f_s	sampling frequency	6, 21
γ	maximum attenuation	30–33
I	sample count	3
i	sample index	2, 3, 5, 7, 50
K	frame count	3, 53
k	frame index	2, 3, 5–9, 19, 21–23, 25–27, 32, 34, 35, 50, 51, 53–56
L	window length	2–4
M	frequency bin count	3, 6, 19, 21, 22, 34, 35, 53, 54
μ	frequency bin index	2, 3, 5–9, 19, 21, 25–27, 32, 34, 35, 50, 51, 53–56
N_F	formant count	3, 23, 25
N	DFT size	2–4, 21, 25
ν_F	formant index	3, 23, 25
Φ	power of a signal	5, 8

Notation	Description	Page List
$\widehat{\Phi}$	estimated power of a signal	3, 5–9, 22, 26, 50, 53
R	frame shift	2, 3, 6
S	complex clean speech signal in frequency domain	3, 5, 7, 8, 21, 22, 50, 51, 53–56, 59, 60
$\widehat{S}$	estimated complex clean speech signal in frequency domain	5, 7, 8, 50, 51, 53
s	clean speech signal in time domain	3, 5, 7, 50
$\widehat{s}$	estimated clean speech signal in time domain	5, 7, 50
$\operatorname{sgn} x$	sign of the real scalar x	11
X	complex noisy speech signal in frequency domain	3, 5–9, 21, 22, 26, 50, 53, 54
x	noisy speech signal in time domain	2, 3, 5, 7, 50
ξ	signal-to-noise ratio	8
$\widehat{\xi}$	estimated signal-to-noise ratio	3
$\widehat{\zeta}$	estimated input-to-noise ratio	3, 19, 26–28, 30–32

Bibliography

Cohen, Israel. "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. ISSN: 1063-6676. DOI: 10.1109/TSA.2003.811544.

Cohen, Israel and Baruch Berdugo. "Spectral enhancement by tracking speech presence probability in subbands." In: *International Workshop on Hands-Free Speech Communication*. 2001, pp. 95–98.

Fant, Gunnar. *Acoustic Theory of Speech Production*. 2nd ed. D A C S R Series. Mouton De Gruyter, 1970. ISBN: 9789027916006.

Garofolo, John S. et al. "TIMIT acoustic-phonetic continuous speech corpus." In: *Linguistic Data Consortium* 10.5 (1993).

Hänsler, Eberhard and Gerhard Schmidt. *Acoustic Echo and Noise Control: A Practical Approach*. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. Wiley, 2005. ISBN: 9780471678397.

Hu, Yi and Philipos C. Loizou. "Subjective comparison of speech enhancement algorithms." In: *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. Vol. 1. IEEE. 2006.

Iskra, Dorota et al. "Speecon-speech databases for consumer devices: Database specification and validation." In: *Proc. LREC*. Vol. 2002. 2002.

ITU-T. *Methods for subjective determination of transmission quality (ITU-T Recommendation P.800e)*. Tech. rep. P.800e. Geneva: International Telecommunications Union, Aug. 1996.

Jax, Peter J. and Peter Vary. *Enhancement of Bandlimited Speech Signals. Algorithms and Theoretical Bounds*. Aachener Beiträge zu Digitalen Nachrichtensystemen. Wissenschaftsverlag Mainz, 2002.

Junqua, Jean-Claude. "The influence of acoustics on speech production: A noise-induced stress phenomenon known as the Lombard reflex." In: *Speech Communication* 20.1Ű2 (1996), pp. 13–22. ISSN: 0167-6393. DOI: 10.1016/S0167-6393(96)00041-6.

Karlsson, I. "Evaluations of acoustic differences between male and females voices: a pi." In: *STL-QPSR* 33.1 (1992), pp. 19–31.

Krini, Mohamed. *Modellbasierte Verfahren zur Sprachsignalverbesserung*. Vol. 10. Fortschrittberichte VDI 803. VDI-Verlag, 2009. ISBN: 9783183803101.

Krini, Mohamed, Patrick Hannon, and Ingo Schalk-Schupp. "Speech Enhancement by Partial Speech Signal Reconstruction." Internal invention disclosure. 2012.

Kuttruff, Heinrich. *Room acoustics*. Taylor & Francis, 2000.

Linhard, Klaus and Tim Haulick. "Spectral noise subtraction with recursive gain curves." In: *ICSLP*. ISCA, 1998, pp. 1479–1482.

Loizou, Philipos C. *Speech Enhancement. Theory and Practice*. Boca Raton: Taylor & Francis, 2007.

Lundgren, Jonas. *SPLINEFIT*. 2007. URL: `http://www.mathworks.com/matlabcentral/fileexchange/13812-fit-a-spline-to-noisy-data` (visited on 09/06/2012).

Naylor, Patrick A. and Nikolay D. Gaubitch, eds. *Speech Dereverberation*. Springer Verlag, 2010.

Quatieri, Thomas F. *Discrete-Time Speech Signal Processing: Principles and Practice*. 10th ed. Prentice-Hall, 2008. ISBN: 9780132441230.

Rife, Douglas D. and John Vanderkooy. "Transfer-Function Measurementwith Maximum-Length Sequences." In: *Journal of the Audio Engineering Society* 37.6 (1989), p. 419.

Rothauser, E. H. et al. "IEEE recommended practice for speech quality measurements." In: *IEEE Transactions on Audio Electroacoustics* 17 (1969), pp. 227–246.

Warren, Richard et al. "Spectral redundancy: Intelligibility of sentences heard through narrow spectral slits." In: *Attention, Perception, & Psychophysics* 57 (2 1995), pp. 175–182. ISSN: 1943-3921.

Wendemuth, Andreas. *Grundlagen der stochastischen Sprachverarbeitung*. München: Oldenbourg, 2004. ISBN: 9783486576108.

Wendemuth, Andreas et al. *Grundlagen der digitalen Signalverarbeitung: Ein mathematischer Zugang*. Springer-Lehrbuch. Springer, 2004. ISBN: 9783540218852.

Wong, George S. K. and Tony F. W. Embleton, eds. *AIP handbook of condenser microphones: theory, calibration, and measurements*. American Institute of Physics, 1995.