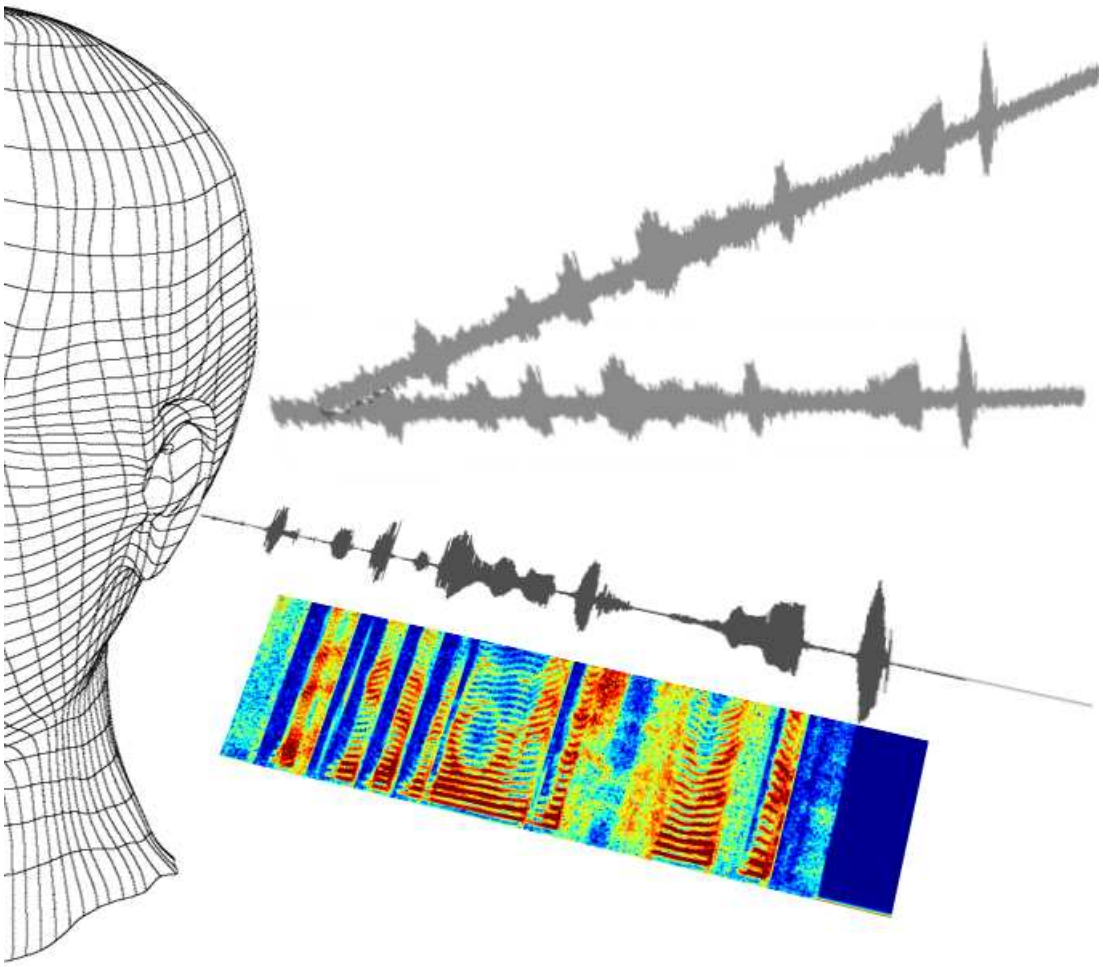
TWO-CHANNEL SPEECH ENHANCEMENT AND IMPLEMENTATION CONSIDERATIONS

– NOISE REDUCTION AND SPEECH QUALITY –



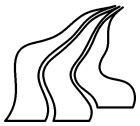
Aalborg University
Institute of Electronic Systems
Faculty of Engineering, Science, and Medicine

Project Group 943/1043, 2007

Aalborg University

Faculty of Engineering, Science, and Medicine

Institute of Electronic Systems



Frederik Bajers Vej 7 ▪ 9220 Aalborg Øst ▪ Denmark Phone +45 96 35 87 00

Title:	Two-Channel Speech Enhancement and Implementation Considerations
Subtitle:	Noise Reduction and Speech Quality
Theme:	Applied Signal Processing and Implementation
Project period:	September 5th 2006 to June 7th 2007

Project group:

943/1043

Group members:

Thomas Lynge Kjeldsen

Jacob Barsøe Kjærgaard

Martin Hammer

Supervisor:

Kjeld Hermansen

Publications:	5 (hard copies)
Pages:	166
Finished:	June 7th 2007

Abstract

Hearing impairment is a prevalent, world-wide ubiquitous problem. An increasing number of people suffer from perceptual hearing loss which cannot be remedied by mere amplification, but rather calls for advanced speech enhancement functionality offering increased speech intelligibility. This project focuses on speech enhancement for hearing-aid applications with one or two microphones available. In specific noise reduction and speech quality improving functionality is attended.

In order to measure speech quality, a survey of speech-assessment techniques lay the basis for the choice of signal-to-noise ratio (SNR) and weighted spectral slope measure (WSSM) as preferred measures of noise reduction and speech distortion, respectively.

For completeness, two single-channel noise reduction techniques are investigated. Spectral subtraction, which is a well-documented, classical method for noise reduction, and the signal subspace approach. Despite its computational complexity, the signal subspace approach is found uncompetitive. However, by extending the method to multiple channels, the GSVD-based multi-channel Wiener filter is formulated. This method relies on long-term stochastic estimates and show robust to different noise types compared to single-channel methods, fixed and adaptive beamforming. However, the multi-channel Wiener filter relies on a good voice activity detector (VAD).

Motivated by the observation, that the noise reduction problem often is shifted towards a noise estimation problem, two methods for noise estimation are investigated. For spectral subtraction, a method of minimum-statistics tracking and, for all methods, a log-energy-based VAD. Both methods prove suitable for additive white and pink noise, but fails in babble noise setups.

Simulations using reverberated environments reveal that little noise reduction and reduced speech distortion is attained. A combined dereverberation and noise reduction method is investigated. It show no significant improvement.

The complexity of the multi-channel Wiener filter is examined and, with acceptable performance degradation, a recursive GSVD-based implementation in 32 bit floating-point precision is created, running in real time on a Pentium-based workstation.

Preface

This master thesis is a documentation of the project work carried out in the time frame September 6th till June 7th by project group 943/1043 in order to fulfil the study regulations set by the Study Board for Electronics and Information Technology for obtaining the degree of "Civilingeniør i signalbehandling med speciale i Anvendt Signal Processering og Implementation", i.e. (in English) the M.Sc in Engineering degree in Applied Signal Processing and Implementation, at the faculty of Engineering, Science, and Medicine, Aalborg University.

Nomenclature, mathematical notation, and abbreviations are listed in the glossary. A MATLAB code portfolio is described in the appendix and contains a subset of the functions in the toolbox develop during this project. The functions described in this appendix is available online at http://kom.aau.dk/group/06gr943/thesis/. The two test signals used throughout the report are also available online at ./data-files/.

The equations are assigned sequentially numbers referring to the chapter, and the reference for an equation is shown in parentheses. References are found in the Bibliography after the report, before the appendices, and are sorted alphabetically using the style [1].

Thomas Lynge Kjeldsen Jacob Barsøe Kjærgaard Martin Hammer

Aalborg, June 7th 2007

Glossary

Nomenclature

x	Scalar x
$\mathbf{x}$	Vector $\mathbf{x}$
$\mathbf{X}$	Matrix $\mathbf{X}$
$x*$	Complex conjugate of scalar x
$\mathbf{X}^T, \mathbf{X}^H$	Transpose and hermitian tanspose of matrix $\mathbf{X}$
$\mathbf{X}^{-1}$	Inverse of matrix $\mathbf{X}$
$\mathbf{X}_{ij}$	The (i,j)th pivot element of matrix $\mathbf{X}$ (2×2 matrix)
x_i, X_{ij}	The ith element or (i,j)th element of vector $\mathbf{x}$ and matrix $\mathbf{X}$, respectively (scalar)
$\mathbf{x}_i$	The ith column vector of matrix $\mathbf{X}$
$\mathrm{diag}(\mathbf{x})$	Square matrix with vector $\mathbf{x}$ along the diagonal
$\mathrm{tr}(\mathbf{X})$	Trace of matrix $\mathbf{X}$ (sum of diagonal elements)
$\lvert \cdot \rvert$	Absolute value
$\lVert \cdot \rVert_2$	$\mathcal{L}_2$-norm
$\mathrm{sgn}(\cdot)$	Sign
$r_{xx}(l)$	Auto-correlation sequence of vector $\mathbf{x}$
$r_{xy}(l)$	Cross-correlation sequence of vector $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{R}_{xx} = E\{\mathbf{x}\mathbf{x}^T\}$	Auto-correlation matrix of vector $\mathbf{x}$
$\mathbf{R}_{xy} = E\{\mathbf{x}\mathbf{y}^T\}$	Auto-correlation matrix of vectors $\mathbf{x}$ and $\mathbf{y}$
$S_{xx}(\omega)$	Power spectral density of $\mathbf{x}[k]$
$S_{xy}(\omega)$	Power spectral density of $\mathbf{x}[k]$ and $\mathbf{y}[k]$
$E\{\cdot\}$	Expectation operator
$\mathcal{F}\{\cdot\}$	Discrete-time Fourier transform
$\mathcal{F}^{-1}\{\cdot\}$	Discrete-time Inverse Fourier transform
$X[r,n]$	nth frequency bin of DFT of rth frame of vector $\mathbf{x}[k]$
$\hat{x}, \hat{\mathbf{x}}, \hat{\mathbf{X}}$	Estimate of scalar x, vector $\mathbf{x}$, and matrix $\mathbf{X}$
$*$	Convolution
k	Discrete-time index
n	Discrete-frequency index
r	Block-time index, e.g. used for discrete-time STFT

ω	Normalised angular frequency in radians, $0 < \omega \leq 2\pi$ or $-\pi < \omega \leq \pi$.
t	Continuous time
μ_x^2	Mean of probability density function $f(x)$
σ_x^2	Variance of probability density function $f(x)$
$\lambda_{x,i}$	ith eigenvalue from the EVD of matrix $\mathbf{X}$
$\sigma_{x,i}$	ith singular value from the SVD of matrix $\mathbf{X}$
T_{60}	Reverberation time (time required for a sound in a room to decay by $60\,\mathrm{dB}$)

Abbreviations

ANC	Adaptive noise canceller
ASR	Automatic speech recognition
BSD	Bark spectral distortion
BTE	Behind-the ear (hearing-aid device)
CD	Cepstral distortion
cf.	*confer*, meaning compare or consult
CLS	Constrained Least Squares
DTFT	Discrete-time Fourier transform
DFT	Discrete Fourier transform
DS	Delay-and-sum (beamformer)
DS-GSC	Delay-and-sum Generalised Sidelobe Canceller (beamformer)
e.g.	*exempli gratia*, for instance or for example
et al.	*et alii*, and others
EVD	Eigenvalue decomposition
FFT	Fast Fourier transform
FIR	Finite impulse response (filter)
FLOP	Floating point operation (singular)
FLOPs	Floating point operations (plural)
FLOPS	Floating point operations per second
GEVD	Generalised singular value decomposition
GJBF	Griffiths-Jim beamformer
GSC	Generalised sidelobe canceller
GSVD	Generalised singular value decomposition
HDL	Hardware-description language
IDM	Itakura distortion measure
i.e.	*id est*, that is
ISDM	Itakura-Saito distortion measure
ITU	International Telecommunication Union
LCMV	Linear constrained minimum variance (beamformer)
LS	Least squares
LSD	Log-spectral distortion
LMMSE	Linear minimum-mean square error, see MV
LMS	Least-mean square (adaptive filter)

MCWF	Multi-channel Wiener filter
MOS	Mean-opinion scale
MV	Minimum variance, see LMMSE
MVDR	Minimum variance distortionless response
NLMS	Normalised least-mean-square (adaptive filter)
PESQ	Perceptual Evaluation of Speech Quality
PSD	Power spectral density or power spectrum
QRD	QR decomposition
SD	Spectral distortion
SNR	Signal-to-noise ratio (SNR)
SNRSEG	Segmental signal-to-noise ratio ($\mathrm{SNR}_{\mathrm{SEG}}$)
SQNR	Signal-to-quantisation noise ratio
SVD	Singular value decomposition
SDC	Spectral domain constrained (estimator)
SPL	Sound pressure level
TDC	Time-domain constrained (estimator)
TSVD	Truncated singular value decomposition
VAD	Voice activity detector
viz.	*videlicet*, namely
w.r.t.	with respect to
WSSM	Weighted spectral slope measure (WSSM)

Contents

CHAPTER 1

Introduction

Hearing impairment represent an important biomedical application. World-wide millions of people suffer from hearing loss. This is not at least due to constant exposure of loud noises (noise at work, loud music in a disco, etc.). Most people with a hearing impairment suffer from a perceptual, frequency-dependent, hearing loss, which not only causes loss of loudness perception, but also difficulty in distinguishing different sounds. People suffering from perceptual hearing impairment often looses their ability to decompose incoming acoustical signals into *desired* and *unwanted* signals. This ability is what enables a normal hearing to focus or concentrate on one speaker in a multi-speaker situation. The intelligibility will therefore not be increased by mere amplification.

In order to help people suffering from perceptual hearing loss, behind-the-ear (BTE) hearing-aid devices of today contain not only two or three microphones, but also a small digital signal processor (DSP) to do some useful speech processing of the multiple signals in order to increase the intelligibility. The amount of processing have so far been limited by the small amount of processing power available in a DSP processor small enough to built into modern hearing-aid devices, and power efficient enough to prevent the users from having to replace the batteries, or recharging several times a day. Modern technology bids more DSP power, increased battery energy, and even remote processing in a pocket DSP connected to the hearing aid by a bluetooth connection [36].

This project is motivated by the need for useful, efficient - in terms of quality-to-processing-speed ratio - and intelligibility improving, functionality. Over the last decade Moore's law has successfully predicted a large increase in computational power, thus we have to anticipate future hearing-aid devices with more computational power, with useful, yet efficient, digital signal processing algorithms.

However, whereas the processing power can be expected to increase, the physical problem of signal acquisition still poses a large problem. Because of the large distance (could be several meters) between the microphone(s) in the hearing aid and the speaker, the physical problem is speech signal acquisition in a possible adverse acoustic environment. Possible noise sources could be traffic and other types of background noise, competing speakers, and acoustic room effects, which all impede intelligibility. The speech enhancement problem posed is different than e.g. echo cancellation, in that, no reference signal is available. This poses, besides the speech enhancement problem, an additional estimation problem of the interferer, that being an acoustic room, competing speakers, or background noise. Further, the inter-distance between the microphones is quite small for a hearing-aid device ($1-2\,\mathrm{cm}$), which renders robustness of the proposed algorithms an important parameter - which is less important, e.g. in a hands-free car installation.

1.1 Initial Problem (Formulation)

A formal description of the problem can be done using a scenario depicted in Figure 1.1. The hearing aid user, "Listener", is located in an enclosure. By one or more microphones the speech signal from the "speaker" is recorded. The signal is corrupted, however, by a background "noise" source and a number of competing voice, "cocktail-party noise"[1], sources which add up to what we will categorise as background noise. Besides the background noise, the acoustic room effects, known as reverberation, causes the desired speech signal, as well as the background noise to be propagated by multiple paths and added up at the acquisition point at the microphone.

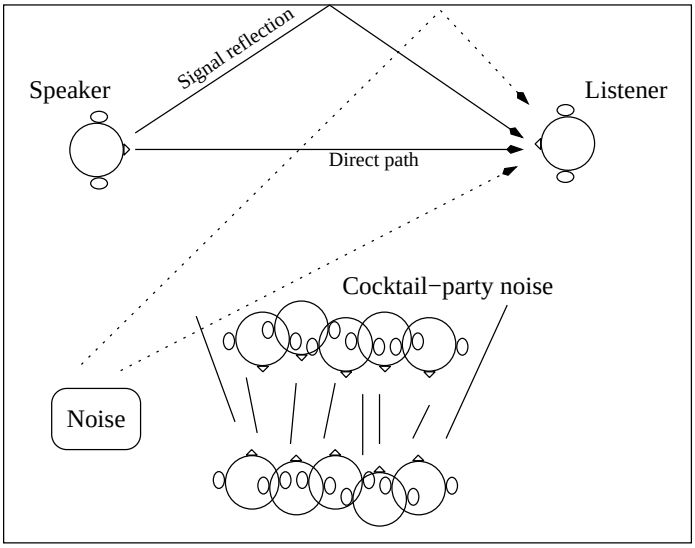


Figure 1.1: Depiction of signal acquisition scenario for a listener in a possibly noisy environment consisting of different types of background noise, competing speakers, and reverberation effects.

The description of Figure 1.1 leads us to a fundamental assumption in this project. We assume, that can the speech signal quality and intelligibility be improved for a normal hearing, we will also have improved the signal quality for a hearing-impaired. This assumption reduces the concern of this project to speech enhancement functionality, rather than physical hearing aid design. The assumption also rules out the field of audiology and modelling of human hearing. Rather we will base the project on an observation model, which forms the basis for our further work.

Remember the problem for perceptual hearing impaired was to distinguish signals. We consider the problem of de-reverberating the observed signal a secondary problem subordinate to reducing background noise, as we know that the effect of reverberation would also be present had we not used a hearing aid.

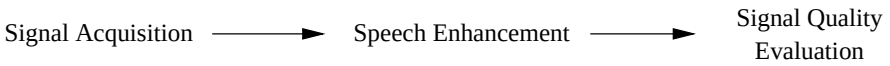


Figure 1.2: The work flow of a speech enhancement development cycle. The signal acquisition could consist of controllable, simulated signal generation. The speech enhancement method is the device under test, and evaluation is a toolbox of tools to evaluate the quality of improvement by applying the speech enhancement method.

To summarise, people who suffer from perceptual hearing impairment are in need of hearing aids which provides better speech quality and intelligibility than the apparatus available today.

[1] By *cocktail-party noise* we refer to the effect of multiple speakers spatially separated from the desired speaker. This is to be distinguished from the *cocktail-party effect*, which is concerned with the ability of normal-hearing persons to distinguish the voice of the speaker of interest from one or more competing voices. The interest in cocktail-party noise is related to its spectral concentration in the same frequency band as the desired speakers.

This project is concerned with speech enhancement functionality which can satisfy this, primarily by noise reduction, secondly by de-reverberation. Based on an observation model accommodated to signal acquisition using the built-in microphones of a hearing-aid device, the speech enhancement functionality is defined. The underlying observation model is presented in Section 1.2 together with characteristics of speech and noise signals. This is the first step in Figure 1.2, which depicts the work flow in speech enhancement evaluation. A model of the observation allows controllable (laboratory) signal generation of realistic as well as toy signals. The last step in Figure 1.2 concerns evaluation and analysis of the enhanced speech signal. Evaluation can be done by subjective and/or objective assessment techniques - where the latter is a faster, more quantitative, but less qualitative and realistic alternative to evaluation by exposing human subjects to listening tests. This is covered in Section 1.3. Finally, in Section 1.4, we introduce the preferred speech signals and noise types, together with visual inspection tools such as traditional time-plane plotting and modified spectrograms. We finish the chapter by a formal problem formulation and problem delimitation followed by an outline of the report.

1.2 Characterisation of the Acoustic Environment, Signals, and Observation Models

Signals can generally be characterised by spectral, stochastic, and spatial properties. The room acoustics can be characterised by a set of physical phenomena related to propagation of sound waves or particle movement in a medium.

It is the purpose of this section to list some important characteristics to gain insight into the physical problem, and from these obtain an approximate observation model and a signal generation model. These models should enable us to simulate, develop, and evaluate a number of signal enhancement techniques in a desirable controlled, close-to-reality setup.

1.2.1 Acoustic Environment

When a sound signal propagates through an acoustic environment it undergoes several complex transformations, which arises from physical laws. Traditionally, an acoustic environment is modelled by a linear filter, typically an FIR model of hundreds or thousand of filter taps. To characterise a room, the *reverberation time*, T_{60}, is a frequently used measure. It is the finite duration it takes an impulse to decay to inaudibility, defined as the time it takes an impulse to decay $60\,\mathrm{dB}$.

Reverberation is the effect of adding up the direct path signal and the multiple delayed and attenuated versions caused by reflections from large surfaces. One way of interpreting an acoustic impulse response is to first recognise the direct-path signal shortly followed by the reflections of large nearby surfaces. This is called early echoes. The early echoes are followed by dense, smaller echoes. This is called diffuse reverberation. In Figure 1.3 a typical impulse response of an FIR model of an acoustic impulse response can be seen. The impulse response is sampled at $16\,\mathrm{kHz}$ and obtained using an adaptive approach.

According to Everest in [25] effort has been made in order to determine an optimum reverberation time. For churches the reverberation time can be between $1\,\mathrm{s}$ for a small church to $3-4\,\mathrm{s}$ for large churches. Generally larger reverberation time is needed for music than speech. A symphony orchestra tends to desire more reverberation than do opera and chamber music. An auditoria for speech should have reverberation times lower than $1\,\mathrm{s}$ to maintain high intelligibility, whereas reverberation times for an average living room is between $400\,\mathrm{ms}$ and $1\,\mathrm{s}$. Not surprisingly reverberation time is frequency dependent.

In some approaches early echoes and diffuse reverberation are realised separately in the reverberation model. An FIR filter and a number of all-pass feedback loops [26]. More recent work additionally includes absorptive losses. Besides reverberation (reflections), also dissipation in the air, diffraction, diffusion, non-linear absorption, and temperature-dependent effects play a role

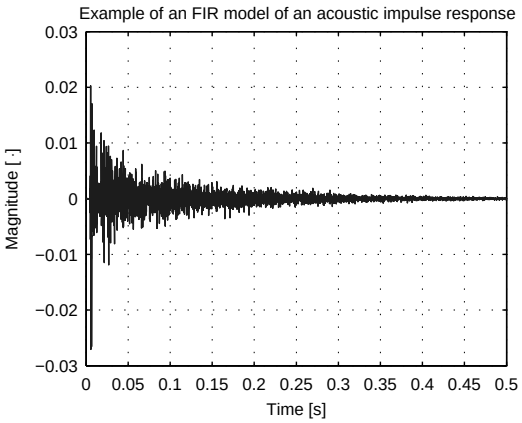


Figure 1.3: Impulse response of a room with a reverberation time of $940\,\mathrm{ms}$. The direct path and the early echoes are followed by dense, smaller echoes, called diffuse reverberation. By playing back white noise on a speaker and recording the signal at a microphone positioned elsewhere in the room the impulse response has been approximated by adaptive filtering.

in determining an acoustical characterisation. It can be readily identified, that a room model is dependent on a number of parameters: frequency (wavelength), angle of propagation, temperature, and physical position of speaker and microphone. It is important to understand the FIR model of the acoustical characteristics is an approximation, but with sufficient order, it provides a good basis for examining speech enhancement techniques, both for noise reduction as well as dereverberation.

It is in some cases reasonable to assume stationary room acoustics, but if objects in the room moves around, windows or doors are opened, or if the microphone (listener) is moving, the room acoustics will be a non-stationary concept.

1.2.2 Speech Signals

Speech signals are in nature broadband signals. In fact, some sounds are not band limited. However, normal ears will understand more than 90% of what has been said when constraining the frequency spectrum from $300\,\mathrm{Hz}$ to $3.4\,\mathrm{kHz}$ [72].

Speech signals can be categorised in several ways, normally by a sub-word characterisation. From a signal processing point-of-view, the signal can be characterised as either periodic (vowels), coloured noise (fricatives), or impulsive (plosives). This characterisation elucidates the non-stationarity of speech signals. Normally, however, speech signals are attributed quasi-stationary, viz. stationary within time segments of $20 - 40\ \mathrm{ms}$ [75].

1.2.3 Noise Signals

Whereas some knowledge is known for the signal of interest, less is normally known about the noise signal. Could one obtain a reference of the noise signal, the signal enhancement problem would be significantly easier. However, in order to do noise estimation, assumptions about the noise signal often need to be taken. A frequent assumption is the white-, stationary-noise assumption. This assumption make a clear distinction between the spectral and stochastic characteristics of speech signals and those of the noise signal. However, it is only true in a few cases - or at least only true for some component of the noise signal. Diffuse noise, such as fan noise or distant traffic noise can be considered a stationary, white-noise source. Other types of noise must be considered either coloured noise or non-stationary, or both. The most difficult case, when considering speech enhancement, is when the noise is also speech, popular called *cocktail-party noise*. In this case the noise pertains the same spectral and stochastic characteristics as the signal of primary interest. However, the cocktail-party noise has a more constant sound pressure in contrast to speech, thus the dynamic range is usually different.

1.2.4 Observation Model

Referring back to Figure 1.1 on page 2, we observe a desired signal corrupted by noise. A basic observation model would be

$$y_m(t) = s(t) + b_m(t) \qquad m = 1, \ldots, M \tag{1.1}$$

where $b_m(t)$ is the additive noise on the mth sensor and $s(t)$ the desired signal source(s) and $y_m(t)$ is the recorded signal on the mth microphone. This additive noise model is commonly used when considering *noise reduction* techniques. The additive noise is referred to as the *background*, whereas the signal of interest is referred to as the *object*. When considering single-microphone techniques $M = 1$.

One could wonder whether it was possible to have more than one source, (1.1) defines only one source, but it seems a reasonable choice to model only one source (at the time). Imagine several persons in more than one conversation. The source will then be one of the persons speaking, it might change if you wish to join another conversation, but it remains one source. Since (1.1) models only one source as in Figure 1.1, all other speakers are modelled as noise and included in the $b_m(t)$ term. Of course this make the "noise" non-stationary and assures that the noise and signal of interest covers mostly the same spectral area, since the noise is essentially also speech.

Another observation model that pertains the effects of the acoustical environment on the source speech signal, can mathematically be formulated as

$$y_m(t) = g_m(t) * s(t) \qquad m = 1, \ldots, M \tag{1.2}$$

where $g_m(t)$, the time-varying transfer function from the desired speech source to the mth sensor, is linearly convolved with the object, $s(t)$. This is called convolutional noise, and the convolutional effects due to the acoustical impulse response are referred to as *reverberation*. Equation (1.2) is used when considering the speech enhancement technique considering the problem of *dereverberation*.

Of course it is also possible to combine (1.1) and (1.2) into a combined background and convolutional noise observation mode

$$y_m(t) = g_m(t) * s(t) + b_m(t) \qquad m = 1, \ldots, M \tag{1.3}$$

Naturally the acoustical environment will also affect the noise source, thus $b_m(t)$ will now be $b_m(t) = h_m(t) * c(t)$, where $c(t)$ is the clean-noise signal. In the combined observations model both background noise and room acoustics play a role. Clearly this model is more realistic, but most techniques address only one of the two problems, and a combined noise reduction and dereverberation algorithm requires the combination of two techniques.

To illustrate the problem, Figure 1.4 will be used. The additive and convolutional noise described in (1.3) will be exemplified together with non-linear effects mentioned in Section 1.2.1 on the acoustic environment. Starting in the lower-right corner of Figure 1.4, the non-linear affected reverberated signal embedded in additive noise is shown. If we remove the noise (move to the left), we observe the reverberated signal without noise. Compared to the upper-right figure, where only the reverberated signal is shown, it is seen that the non-linear effect has affected some amplitudes different that others[2]. The effect of reverberation can be observed comparing the upper figures, however, for a single sinusoid, little effect can be observed. The attenuation is clear, but the delay of a sinusoid is constant.

Signal Generation and Observation Model

In order to do experiments with speech enhancement algorithms we introduce a signal generation / signal observation model. Starting point is a noise signal and a "pure" speech signal.

[2]This is only an example of a non-linear function. In acoustics, impingement angle, amplitude, and frequency are likely arguments to a non-linear effect/function.

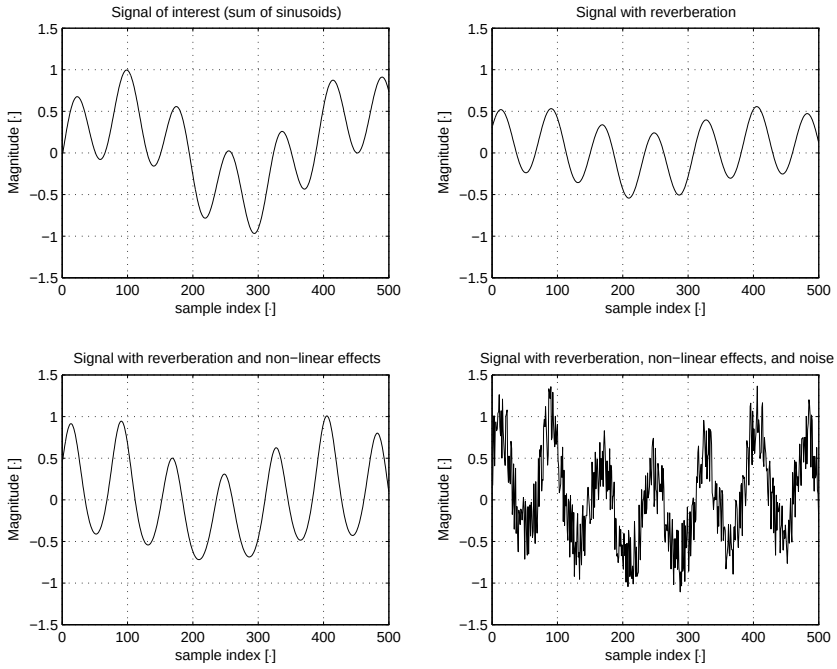


Figure 1.4: The four figures shows an increasing degradation of a "speech signal" (sinusoid) starting in the upper-left corner. Effects of reverberation, non-linear effect, and additive noise is shown.

The speech signal is reverberated by an acoustical impulse response and the signal-to-noise-ratio (SNR) between the noise and reverberated speech signal is adjusted to a desired level. This SNR corresponds to the SNR at the microphone. In the case of multi-microphone setups, one need to choose a reference microphone. These first steps corresponds to the left half of Figure 1.5.

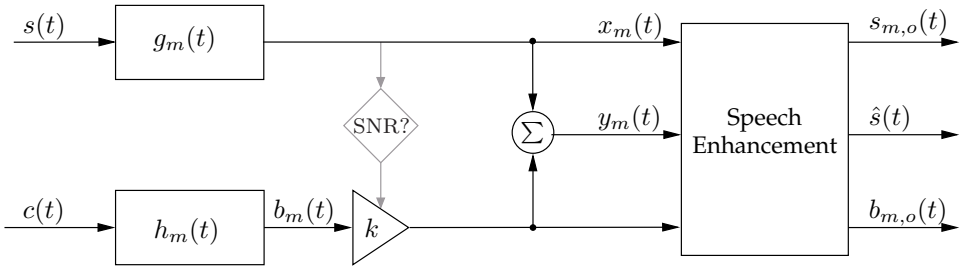


Figure 1.5: Multi-microphone observation model integrated in a signal generation and speech enhancement algorithm scenario. Based on an acoustical impulse response, and a noise and speech signal, the microphone input SNR is adjusted and the output of the algorithm computed.

By adding the noise and speech component and feed it to the speech enhancement algorithm, one would obtain an estimate of the speech signal component

$$\hat{s}(t) = \mathcal{H}[y_m(t)] = \mathcal{H}[x_m(t) + b_m(t)] \tag{1.4}$$

By assuming superposition of the speech enhancement algorithm, $\mathcal{H}$, the noise and speech components can be split

$$\mathcal{H}[x_m(t) + b_m(t)] = \mathcal{H}[x_m(t)] + \mathcal{H}[b_m(t)] = s_{m,o}(t) + b_{m,o}(t) \tag{1.5}$$

which allows one to observe the separate noise and speech components at the output, $b_{m,o}(t)$ and $s_{m,o}(t)$. While the apparent usability of this approach might seem unclear, it will turn out very valuable in Section 1.3.2 on objective measures for speech enhancement assessment. The estimated speech signal can be obtained by re-combining the noise and speech components

$$\hat{s}(t) = s_{m,o}(t) + b_{m,o}(t) \tag{1.6}$$

When we introduced the convolutional noise term in (1.3), we did not mention how it relates to real acoustical impulse responses, or if that covers all the noise effects of room acoustics. Throughout this text we will limit ourselves to approximate the acoustical impulse response by an FIR filter and disregard non-linear effects of room acoustics. Although the impulse response is approximated, one can also split the speech enhancement problem when considering (1.3) in two. One approach would consist of inverse filtering to remove all reverberation effects, while another, simpler approach, would try to remove the spectral *colouring* caused by the reverberation.

From the former equations it would seem the sole purpose of a speech enhancement algorithm to estimate the speech signal, $s(t)$. Where this might be the case in some cases, it is not always true. In some application speech enhancement is not only concerned with improved intelligibility, but also preserving naturalness and comfort of the sound. This is the topic of the next section.

1.3 Objective and Subjective Metrics for Evaluating Speech Enhancement

In order to evaluate if the processed speech signal is improved, well defined assessment techniques are needed, which can be both subjective and objective methods. Furthermore one must define what is *improvement* for a speech signal. This topic will be of interest in this section.

1.3.1 What is Good Speech Quality?

In order to assess processed speech, from a speech enhancement algorithm, one needs to consider the end user. Subjective measures based on well-defined listening tests are preferred - but impractical - hardly very quantitative or repeatable. Objective assessment metrics are in contrast characterised by repeatability and quantitativeness. Furthermore, they exhibit accurate means of comparing e.g. speech enhancement, usually by partitioning speech signals into frames of $10 - 30\,\mathrm{ms}$ and compute a mathematically well defined distance or distortion measure for each frame. The quality of objective assessment metrics are inspected by proving their high correlation with subjective methods.

To emphasise the close relationship between *quality* and *intelligibility* one could consider the plausibility of having good quality but low intelligibility. A very unlikely situation. The converse, however, seems more likely. Juxtaposing or subordinating intelligibility and quality depends on the application. We have chosen to introduce the term *comfort* in contrast to intelligibility and weight both intelligibility and comfort as a subconcept of speech quality, see Figure 1.6.

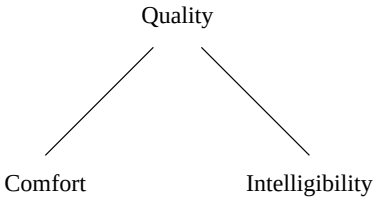


Figure 1.6: The relationship between quality, comfort, and intelligibility.

Generally one can say that intelligibility is concerned with *what* is being said, while comfort is concerned with *how* it has been said. Comfort usually has attributes as naturalness or speaker recognisability. Two examples are to illustrate the point. First consider an opera. The most important speech quality features would be to enjoy the sound of the singer and the beautiful voice. What is less important is to comprehend what has actually been said. Then consider an aeroplane pilot communicating with the control tower. The pilot is less interested in recognising the voice of the person, nor enjoying the naturalness of the speech, but rather pure intelligibility. The two following sections will describe different assessment techniques in speech enhancement. Firstly,

objective measures that represent quantitative techniques and then, secondly, subjective qualitative techniques are presented.

1.3.2 Objective Assessment Techniques for Speech Enhancement

Objective assessment metrics are characterised by quantitative, repeatable, and accurate means of comparing, e.g. speech enhancement. Objective assessment metrics can be used in simulation where both the noise-free speech signal, $s[k]$, and the estimated speech signal, $\hat{s}[k]$[3], are available [15].

The major goal of objective measures is to obtain a figure, that, with a high correlation with subjective measures, indicates the speech quality or improvement of the signal in order to evaluate performance of speech processing algorithms. Thus, having a good objective score gives an indication of whether or not the perception and/or quality has been improved. Over the last decades several methods have been proposed. Two different approaches are commonly used, namely, distortion of the speech signal and noise reduction metrics.

Measures of Noise Reduction

Noise reduction metrics are useful in determining e.g. the SNR-improvement before and after speech enhancement. In [15] the SNR-metric is defined as

$$\mathrm{SNR} = 10\log_{10}\left[\frac{\sum\limits_{k} s^2[k]}{\sum\limits_{k}(s[k]-\hat{s}[k])^2}\right] \tag{1.7}$$

where $s^2[k]$ is the energy in the clean speech signal, and $(s[k]-\hat{s}[k])^2$ represents the noise energy. However using our observation model this can be formulated as

$$\mathrm{SNR} = 10\log_{10}\left[\frac{\sum\limits_{k} s^2[k]}{\sum\limits_{k}\left(s[k]-\mathcal{H}(y[k])\right)^2}\right] \tag{1.8a}$$

$$= 10\log_{10}\left[\frac{\sum\limits_{k} s^2[k]}{\sum\limits_{k}\left(s[k]-\mathcal{H}(s[k]+b_m[k])\right)^2}\right] \tag{1.8b}$$

$$= 10\log_{10}\left[\frac{\sum\limits_{k} s^2[k]}{\sum\limits_{k}\left(s[k]-s_{m,o}[k]-b_{m,o}[k]\right))^2}\right] \tag{1.8c}$$

this can be interpreted as a ratio between the energy in the original signal and not just the noise, $b_{m,o}[k]$, but also the distortion, $s[k]-s_{m,o}[k]$, introduced by the algorithm. Assuming that no distortion is introduced by the algorithm, $s[k]-s_{m,o}[k]=0$, which is desired, but unrealistic, this method will then measure the signal to noise ratio. A noise reduction measure can then be obtained by comparing the SNR before and after applying the speech enhancement algorithm. This measure is referred as the *broadband* SNR.

A modified version of the broadband SNR has been developed as well, since the broadband SNR yields wrong results, because high energy utterances contribute to a deceptively high SNR. By framing the signal into segments this problem can be circumvented. Segmenting the signal gives the segmental signal-to-noise ratio, $\mathrm{SNR}_{\mathrm{SEG}}$, also denoted *average short-time segmental* SNR.

[3] As noted in the glossary the time index k is used to describe discrete-time processing and sampling.

In the case of M frames, it can be described as

$$\mathrm{SNR}_{\mathrm{SEG}} = \frac{1}{M}\sum_{j=0}^{M-1} 10\log_{10}\left[\frac{\sum_{k=jN+1}^{(j+1)N} s^2[k]}{\sum_{k=jN+1}^{(j+1)N}(s[k]-\hat{s}[k])^2}\right] \tag{1.9}$$

where the frame size, N, typically is in the order of $20-40$ ms. To prevent noise-only frames to contribute with a large negative SNR, the use of $0\,\mathrm{dB}$-thresholding is recommended. Furthermore, frames of $\mathrm{SNR} > 35\,\mathrm{dB}$ are not perceived by listeners as significantly higher. An upper thresholding of $35\,\mathrm{dB}$ is therefore often employed [15]. By framing the signal we introduce time resolution into the SNR expression, and by proper thresholding, we achieve results comparable to those of computing the SNR, using Eq. (1.7), over speech-dominated samples only. The final $\mathrm{SNR}_{\mathrm{SEG}}$ is the average over all segments.

The last SNR method is based on frequency weighting such that the measure better resembles the human auditory system. Frequency-weighted segment-based SNR , $\mathrm{SNR}_{\mathrm{fw-seg}}$, is an extension of the segmental SNR. Each frequency band, normally spaced equally to the ear's critical bands, is applied a perceptual weight to produce a measure more closely related to the listener's perceived notion of quality [15]. Let $E_s(j,n)$ denote the short-term signal energy in one of the M frames (indexed by j), and $E_{s-\hat{s}}(j,n)$ denote the short-term noise energy, then the metric can be described as

$$\mathrm{SNR}_{\mathrm{fw-seg}} = \frac{1}{M}\sum_{j=0}^{M-1}\left[\frac{\sum_{n=1}^{N} w_n 10\log_{10}\left[\frac{E_s(j,n)}{E_{s-\hat{s}}(j,n)}\right]}{\sum_{n=1}^{N} w_n}\right] \tag{1.10}$$

where the weight, w_n, is applied to each of the N frequency bands indexed by n.

Measures of Speech Distortion

Methods for speech enhancement are often designed in order to remove noise from a degraded speech signal, which is why noise reduction is an important performance parameter. However, it is not the only. Recently, Chen et al. [9] have brought focus on speech distortion. Single-channel noise reduction algorithms have been designed with emphasis on noise reduction, but they achieve this at the cost of speech distortion. The signal subspace approach is in fact explicitly defined on the trade-off between noise reduction and speech distortion. In introducing speech distortion we might be reducing the intelligibility, though we in fact attained our goals of reducing the noise. Therefore methods to assess speech distortion introduced by the speech enhancement - in specific, noise reduction - methods are needed.

In the following section we introduce a number of different speech distortion techniques. Generally speaking they all partition the signal into frames of $20-40$ ms, in which the measures/-metric is computed.

The Itakura-Saito distortion measure (ISDM) is described by [33]

$$d_{IS}(S,\hat{S}) = \int_{-\pi}^{\pi}(S/\hat{S})\frac{d\theta}{2\pi} - \ln(\sigma^2/\hat{\sigma}^2) - 1 \tag{1.11}$$

This equation however is not directly applicable to discrete time speech enhancement, thus a discrete version is described as

$$d_{IS}(S,\hat{S}) = \frac{\sum_{n=0}^{N-1}\left[(S_n/\hat{S}_n) - \ln(S_n/\hat{S}_n)\right]}{N} - 1 \tag{1.12}$$

using $\ln(\sigma^2/\hat{\sigma}^2) = \int_{-\pi}^{\pi}\ln(S/\hat{S})\frac{d\theta}{2\pi} = \frac{1}{N}\sum_{n=0}^{N-1}\ln(S_n/\hat{S}_n)$. S_n is the power spectral density (PSD) of the original signal in the nth of N frequency bins and $\hat{S}$ is the estimated spectrum. The ISDM

can be interpreted as the L_1-norm of the area between the two spectra. The "-1" term in Eq. (1.12) is to assure a result of 0 if the two spectra are identical.

A modified version of the IDSM is the the Itakura Distortion Measure (IDM), which tries to minimise the gain-mismatch in the ISDM presented above. The IDM does not weight a gain offset between the two spectra which is a linear frequency-independent difference. The measure can be described as

$$d_I(S, \hat{S}) = \min_{\lambda \geq 0} \left(d_{IS}(S, \lambda \hat{S}) \right)$$

This optimisation problem, however, is a global search for a minimum of a non-linear function. For both Itakura measures, superposition (additive property) and homogeneity (scaling property) holds, however, further constraints may be imposed to ensure symmetry [15], i.e. same results whether signal *A* is compared to signal *B*, or vice versa.

Another method, which for small amounts of distortion is related to the ISDM, is the log spectral deviation (LSD). The relation is

$$\text{ISDM} \approx 0.5(\text{LSD})^2 \tag{1.13}$$

and the LSD is computed as [33]:

$$d_{\ln p}(S, \hat{S}) = \left\| \log \left(\frac{S}{\hat{S}} \right) \right\|_p \tag{1.14}$$

where $\|\mathbf{x}\|_p$ is the p-norm, see Appendix A. S and $\hat{S}$ are the spectrum and the estimated spectrum, respectively.

The log spectral deviation method however has been shown to have a very high correlation with the cepstral distortion method (CD), $d^2_{\ln,2} = d^2_{\text{CD}}$

$$d^2_{\text{CD}} = \sum_{n=-\infty}^{\infty} (c_n - c'_n)^2 \tag{1.15}$$

The main difference is the computational complexity, because the sum of the cepstral coefficients can be truncated [77].

The CD technique measures the euclidean distance between the cepstral coefficients. The cepstral coefficients can be recursively computed from the linear predictive coefficients (LPC) α_l, $l = 1 \ldots p$ as shown in [3]

$$c_n = \sum_{l=1}^{n-1} (1 - l/n) \alpha_l c_{n-l} + \alpha_n \tag{1.16}$$

where p is the number of LPC coefficients, and $a_l = 0$ for $l > p$. The truncated version from Rabiner et al. [77] is

$$d^2_{\text{CD}}(Q) = \sum_{n=1}^{Q} (c_n - c'_n)^2$$

where $Q \approx \frac{3}{2}p$.

Another method to measure speech distortion is the bark scale distortion (BSD). This method is preferred over CD in [91], which focuses in objective measures of speech coders, because the correlation between estimator and MOS value (mean opinion scale, see Section 1.3.3) is better than of the CD method over a broad range of bit rates. We, however, do not operate with bit rates, thus we should be aware of this difference between quality assessment in speech coding and in speech enhancement. In speech coding the signal is distorted due to introduced quantisation noise in lossy encoding. In speech enhancement it is a matter of improving the quality. The BSD is calculated using the Bark scale to adapt the distortion in K approximated human auditory

frequency bands (critical band filters). The overall BSD is computed as [91]

$$BSD = \frac{\frac{1}{M}\sum_{j=1}^{M}\sum_{n=1}^{N}\left[L_x(j,n) - L_{\hat{x}}(j,n)\right]^2}{\frac{1}{M}\sum_{j=1}^{M}\sum_{n=1}^{N}\left[L_x(j,n)\right]^2}$$

where M is the number of frames, $L_x(j)$ is the original Bark spectrum of the jth frame, and $L_{\hat{x}}(j)$ is the Bark spectrum of the estimated speech in the jth frame and n is the index of the Bark bands.

One method, which does not measure the difference between the two spectra, but instead measures the euclidean distance between the slopes of the two spectra is the *weighted spectral slope measure* (WSSM), sometimes called *Klatt-measure* due to its inventor [50]. The method is known to include perceptual features of the human auditory system [74, 53].

The method works on short time segments, e.g. $20\,\mathrm{ms}$, in which critical band analysis is performed. In the original paper, Klatt divides the two spectra into 36 critical bands and computes the slope between each band. A weighting function is computed to emphasise on maximum subband (global maximum) and the nearest spectral peak (local maximum), which nicely models the auditory system, which is more sensitive to spectral peaks than spectral valleys. Klatt emphasised, that the actual computations of this weighting function was not of primary interest as long as it weighted spectral peaks more than valleys and largest peak more than lesser peaks.

Let S' denote the slope of the magnitude spectrum of S, expressed in dBs. Then the *slope* can be defined as $S'[n] = S[n+1] - S[n]$, and the WSSM can be computed as

$$\mathrm{WSSM} = K_{\mathrm{SPL}}(dB_{\mathrm{SPL}} - \widehat{dB}_{\mathrm{SPL}}) + \sum_{n=0}^{N} W[n]\left(S'[n] - \hat{S}'[n]\right)^2 \tag{1.17}$$

where the intensities dB_{SPL} and $\hat{dB}_{\mathrm{SPL}}$ are the two *sound pressure levels* (SPLs) of the original and estimated signal, respectively. To weight the difference in intensity, the empirical constant K_{SPL} can be used. By setting the difference to zero, we obtain

$$\mathrm{WSSM} = \sum_{n=0}^{N} W[n]\left(S'[n] - \hat{S}'[n]\right)^2 \tag{1.18}$$

where N is the number of critical bands. The weighting function, $W[n]$ is computed as a mean between two independent weighting functions $W_1[n]$ and $\hat{W}_2[n]$, where $\hat{W}$ denotes the weighting function to $\hat{S}$. The weighting functions are computed as

$$W_j[n] = W_{j,\,\max}[n] \cdot W_{j,\,\max,\,\mathrm{local}} \tag{1.19a}$$

$$W_{j,\,\max}[n] = \frac{K_{j,\,\max}}{K_{j,\,\max} + dB_{j,\,\max} - S_j[n]} \tag{1.19b}$$

$$W_{j,\,\max,\,\mathrm{local}}[n] = \frac{K_{j,\,\max,\,\mathrm{local}}}{K_{j,\,\max,\,\mathrm{local}} + dB_{j,\,\max,\,\mathrm{local}}[n] - S_j[n]} \tag{1.19c}$$

where j denotes either the true or the estimated signal, the constants $K_{\max}$ and $K_{\max,\,\mathrm{local}}$ are empirical values that define the gain to the global and local maxima. $dB_{\max}$ is the global maximum of S and $dB_{\max,\,\mathrm{local}}[n]$ is the local maximum to band n. The local maximum search is performed like this. Search in the higher bands (to the right on the frequency axis) if the slope is positive and search in lower bands if the slope is negative. For each of the two signals, a weighting function (1.19a) is computed, and the mean, $W[i]$, enters the final expression (1.18). The final WSSM measure is computed as the average over all segments.

Although the computations involved in the WSSM measure are given, it is unclear whether the computations should be based on $s[k]$ and $\hat{s}[k]$ (the clean-speech and estimated speech signal), or on $x[k] = g * s[k]$ and $\hat{s}[k]$ (the clean-speech signal convolved with the acoustic room impulse

response and the speech estimate). The latter corresponds better with the observation model used for noise reduction. It turns out, unsurprisingly, that it is crucial to choose $s[k]$ as reference and not $x[k]$. The WSSM method is discussed further in the not essential, but yet very interesting Appendix C.

A new method has been standardised by the International Telecommunication Union (ITU) which claim to have high correlation with subjective listening tests. However, the method, perceptual evaluation of speech quality (PESQ), ITU-R P.862, is developed for the purpose of assessing quality of speech coding algorithms. The method adapts to the human auditory system in loudness, perception, and time-alignment [71], which have been stressed as important for a distortion measure [53]. The method has shown high correlation with subjective listening tests, but the implementation is rather complex.

Yet another alternative to the traditional metrics is to use an automatic speech recognition (ASR) method in order to estimate speech intelligibility. Employing ASR as a measure of intelligibility has proven useful, and with a high correlation to subjective listening tests, for a broad number of applications [53]. It seem just as useful as the PESQ, however, more sensitive to low SNR (below $5\,\mathrm{dB}$, [53]).

Having introduced a number of candidates for objective assessment of either noise reduction or speech distortion - or both - we will shift focus to subjective methods, which are more qualitative, but also more time consuming. Next to the review of common subjective assessment techniques, we will introduce the methods which we have chosen to apply in the context of this project.

1.3.3 A Selection of Subjective Methods for Assessing Speech Enhancement

In subjective measures the quality of an utterance is evaluated by the opinion of a listener. This evaluation can be divided into two subgroups, those methods that measure the intelligibility and those who measure the overall quality. Some different subjective assessment methods are listed in Table 1.1.

Test		**Type of Test**
Modified Rhyme Test	MRT	Subjective intelligibility
Diagnostic Rhyme Test	DRT	Subjective intelligibility
Mean Opinion Scaling	MOS	Subjective quality
Diagnostic Acceptability Measure	DAM	Subjective quality

Table 1.1: Table of subjective quality measures.

A common method for testing intelligibility is the used *rhyme test*, where a listener responses to a set of rhyming words. The MRT is based on single-syllable words of the form consonant-vowel-consonant such as “cat”, “hat”, “fat” etc. The listener is given a sheet containing six identical words where the leading consonant is missing. The listener is then required to choose the leading consonant.

The DRT is also based on rhyme test, however each rhyme test is restricted to a pairwise comparison, so that the difference in each leading consonant pair differ in just one distinct feature. The DRT score represents the correct percentages of responses.

The intelligibility tests only measure one parameter of quality, however speech quality is not necessarily good, because the intelligibility is high, as earlier discussed. Therefore subjective methods for assessment of the quality are needed. These are usually only applied when the intelligibility is high.

One of the most often used tests is the MOS, where the listener rate the speech quality on a five-point scale. This scale is listed in Table 1.2. In order to achieve reliable results, the listeners are often trained using some test phrases, where the scores are known beforehand. Thereby it is possible to normalise listener bias for those who always judge the tested speech to be low or high

quality.

Rating	Speech Quality	Level of Distortion
5	Excellent	Imperceptible
4	Good	Just perceptible but not annoying
3	Fair	Perceptible and annoying
2	Poor	Annoying but not objectionable
1	Unsatisfactory	Vary annoying and objectionable

Table 1.2: The five-point scale in the Mean Opinion Score

Another subjective method for assessing speech quality is the DAM, which is used for evaluating medium- to high-quality speech. It evaluates the speech signal on 16 separate scales, divided into three categories: signal quality, background quality and total quality.

Four different methods for subjective assessment of speech intelligibility and quality have been described in short. However this only covers some of the techniques generally used. Table 1.3 lists some additional methods for subjective assessing speech quality.

Test		Type of Test
Isometric absolute judgment	IAJ	Subjective quality
Paired acceptability rating	PAR	Subjective quality
Parametric absolute judgment	PAJ	Subjective quality
Quality acceptance rating test	QUART	Subjective quality

Table 1.3: Other subjective quality measures

Subjective methods are by nature more qualitative than quantitative, and listening tests are a time-consuming task. These types of tests find they place in long-term projects, where time has been set of to perform several research/development iterations in order to pin-point characteristics e.g. of a speech enhancement method. One could also imagine using such extensive testing in a final approval cases. Choices for which methods to apply could stem from the position of the application in the comfort-intelligibility triangle in Figure 1.6 and indications from one or more objective assessments techniques. Also the objective of a project, say an automatic speech recognition method, might call for different choices of techniques for subjective and objective assessment, than a project on noise reduction for hearing aid devices.

1.4 Choice of Assessment Techniques and Test Scenarios

In order to assess and compare simulation results using different methods for speech enhancement, a number of objective assessment metrics have to be used. This enables fast iteration through algorithm investigation, research, and development, as opposed to using subjective metrics, which are, of course, more time consuming. We have settled for *broadband* SNR in order to measure noise reduction. This choice is based on its popularity, and on the belief that this metric can be evaluated by people not attached to this project, and the results compared to results from related projects. Broadband SNR attaches even importance to noise-only as to speech-dominated time instances. In order to alleviate this, one can choose to either compute SNR over speech-dominated segments only, e.g. by use of a voice activity detector (VAD), or to use the segmental SNR measure. We have chosen the *segmental* SNR measure ($\mathrm{SNR}_{\mathrm{SEG}}$) with a segment length of $20\,\mathrm{ms}$. In order to further emphasise speech-dominated segments, upper thresholding of $35\,\mathrm{dB}$ and lower thresholding of $0\,\mathrm{dB}$ (or $-10\,\mathrm{dB}$ where more appropriate) has been applied. Throughout the report these noise reduction metrics will be employed, with an emphasis on broadband SNR.

The SNR measures are sample-by-sample measures. It is therefore important to do timely alignment, e.g. by a correlation method, in order to obtain correct results. This stems from the fact,

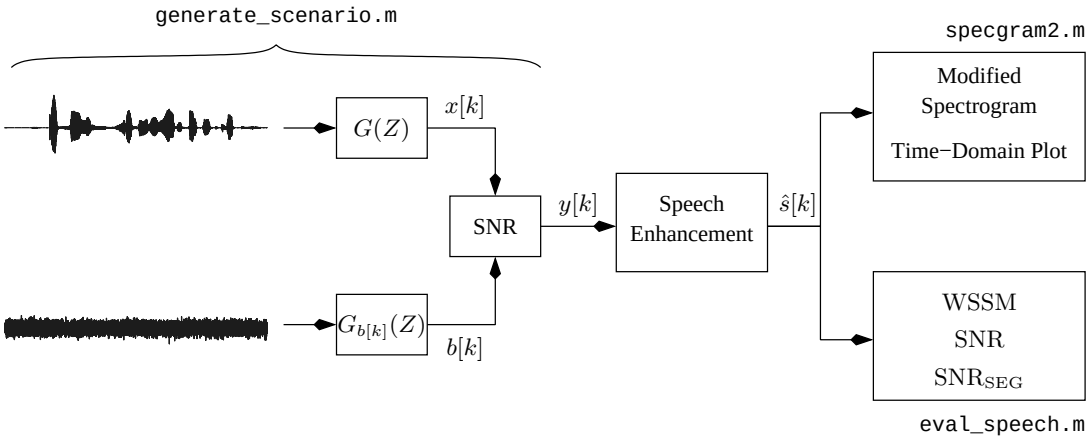


Figure 1.7: Referring to Figure 1.2 on page 2, the speech enhancement work cycle is depicted here. A test scenario with signal plus noise, possibly with reverberation (acoustic room effects), and given SNR, is created. Speech enhancement is applied. The resulting signal is evaluated by comparing it with the input using a number of objective measures and visual inspection by time-domain plotting and modified spectrogram.

that some speech enhancement methods rely on timely re-alignment of the input signal(s). The output signal might not be sample aligned to the input signal and correction is needed. In order to make investigation and testing of speech enhancement methods, we have created a testbench or speech enhancement framework. In Appendix D a selection of the files in our MATLAB portfolio are presented. The function `eval_speech.m` is used for evaluation of SNR and $\mathrm{SNR}_{\mathrm{SEG}}$. It also does time alignment by using the time lag that maximises the signal cross-correlation.

Referring to Figure 1.2 in the introduction, we iterate through signal acquisition (or signal generation), speech enhancement, and signal quality evaluation of improvement. Methods applied in this project are depicted in Figure 1.7, which is an extension of Figure 1.2 on page 2. The `eval_speech.m` function is seen in the lower right corner of Figure 1.7.

To measure speech distortion, we have chosen the *weighted spectral slope measure* (WSSM), because of its emphasis on perceptual features by its critical band analysis and weighting spectral peaks higher than spectral valleys. The method have in recent studies shown SNR-robust and with results comparable to those of PESQ when applied to estimate intelligibility for a large number of speech coding methods [53]. We have implemented the WSSM with 22 bands ($50\,\mathrm{Hz} - 9.5\,\mathrm{kHz}$) approximating the Bark scale by bandpass filters on segments of $20\,\mathrm{ms}$.[4] The WSSM measure is also included in our evaluation function (`wslope.m`).

Objective measures are useful indicators of speech enhancement, but in order to further visualise the output signal, we will make use of the spectrogram. For speech signals it is reasonable to assume limited amplitude dynamics, since the human auditory system also perceives with a limited dynamic range. We here assume $60\,\mathrm{dB}$ dynamic range. In order to reduce the effect of a small number of large peaks, we scale each signal to a standard deviation of 0.1 and saturate any amplitude above $4\,\mathrm{dB}$, and floor (saturate) values lower than $-56\,\mathrm{dB}$, i.e. we define the $60\,\mathrm{dB}$ dynamic range between $-56\,\mathrm{dB}$ and $4\,\mathrm{dB}$ for a standard-deviation scaled signal. Using a Hann (Hanning) window of 240 samples and 230 samples of overlap, we apply a STFT of 1024 discrete frequency points. The implementation, `specgram2.m`, is used extensively throughout the report. The result, when applied to a clean speech signal (see Figure 1.9), is seen in Figure 1.8. This leads us to the introduction of the signals and test scenarios applied in this project.

The signals used throughout the project are one sentence uttered by a female and one sentence uttered by a male voice. The signals are from the TIMIT database [27] sentence 160 and 305. Being gender representative, the signals are by no means statistical representative. The delimitation to only two clean-speech test signals are to simplify the exposition, obtain clear indications of the performance without addressing robustness issues in the broader sense.

[4] Referring to the description of WSSM on page 1.3.2, we have used $K_{\max} = 20$, $K_{\max,\,\mathrm{local}} = 1$, and $dB_{\mathrm{SPL}}\mathrm{s} = 0$.

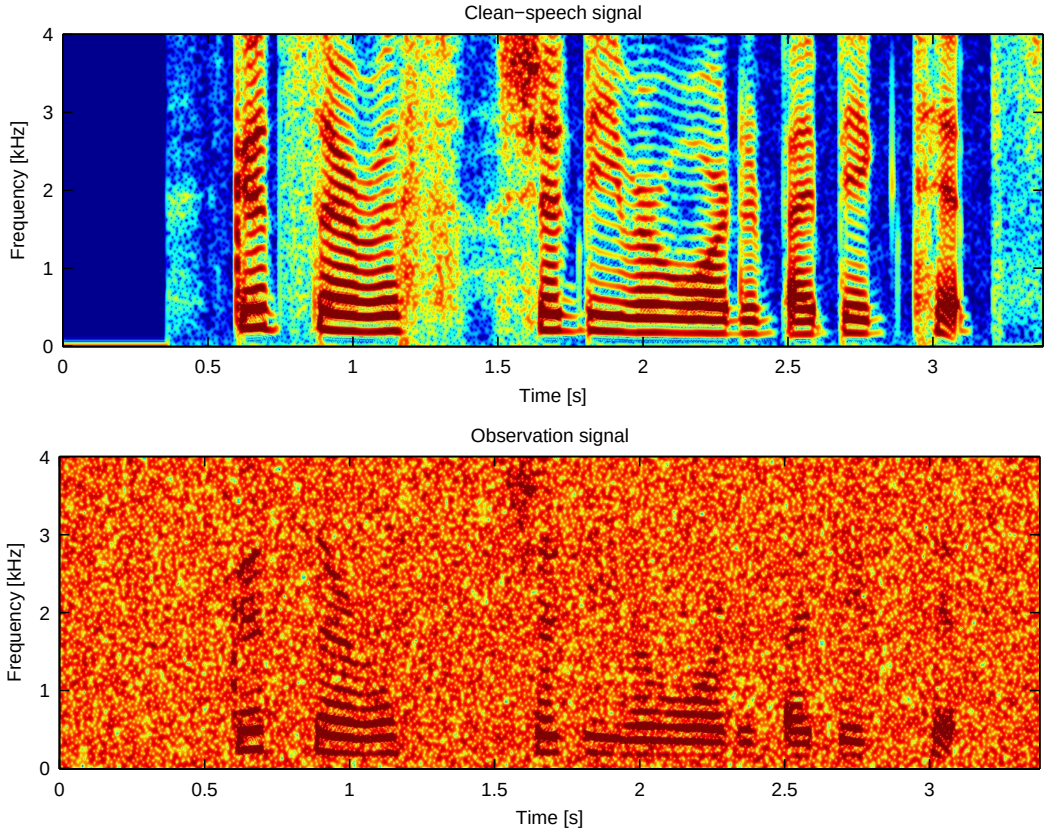


Figure 1.8: Modified spectrogram with limited dynamic range (60 dB), 240 samples Hann windowing with 230 samples overlap, and 1024 discrete frequency points applied to a signal at 8 kHz. The upper figure depicts a clean-speech signal, while the lower depicts same signal embedded in noise (SNR = 5 dB).

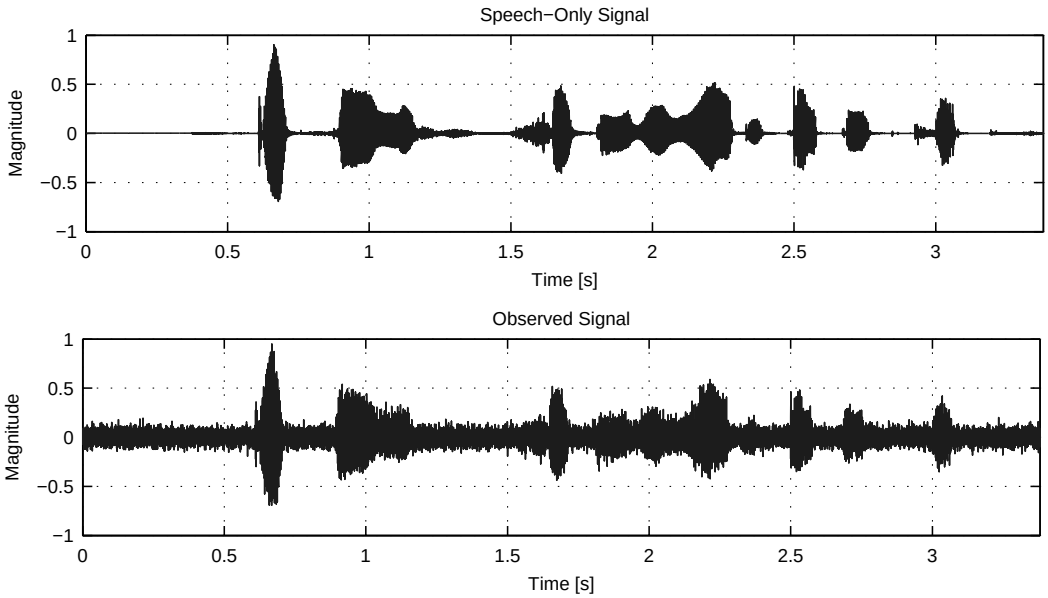


Figure 1.9: Clean speech and speech embedded in noise (SNR = 5 dB) for the sentence "*Good service should be rewarded with big tips*". The lower figure shows the observation at the microphone. No reverberation has been applied to the signal, thus the noise is of the additive type.

The clean-speech test signals, one female and one male, are

"*Good service should be rewarded with big tips*" female, TIMIT-160, (standard signal)
"*The boost is helpful, but inadequate*" male, TIMIT-305

where the female signal has been chosen as the standard signal. i.e. the signal we most often apply. This clean-speech signal is depicted in Figure 1.9. All signals are recorded at $8\,\mathrm{kHz}$. One could argue that modern DSP systems would sample at $16\,\mathrm{kHz}$ instead of $8\,\mathrm{kHz}$, but we have settled for $8\,\mathrm{kHz}$ for reduced computations (especially for the computations of the GSVD-based multi-channel Wiener filter, see Chapter 3) and because of wider availability of good test signals. Furthermore, the performance of a speech enhancement algorithm can safely be assumed independent of the sample rate.

As seen from our observation model

$$y_m(t) = g_m(t) * s(t) + b_m(t) \qquad m = 1, \ldots, M$$

introduced on page 5 in Eq. (1.3), the clean-speech signal is corrupted by acoustic room effects, reverberation, and additive noise. The additive noise has been chosen to be white noise or pink noise. The reverberation is added by convolving the speech signal and the noise signal with their respective room impulse responses created using a simplified mirror-image method [60]. To keep the exposition of the speech enhancement methods as simple as possible, we introduce each method covered in this project by the standard signal. The standard signal consists of the female voice added white noise at an SNR of $5\,\mathrm{dB}$. No reverberation - except for multi-channel methods where we assume a direct path. Secondary channels are added a delay, but no convolutional noise. The standard signal embedded in noise is depicted in Figure 1.9 lower half.

However the white additive noise is a good, well-behaved toy signal, which allow us to investigate the performance, the introduction of each speech enhancement method is backed up with a pink noise test at four SNRs: $-5, 0, 5, 10\,\mathrm{dB}$. These are the four standard SNRs to be employed.

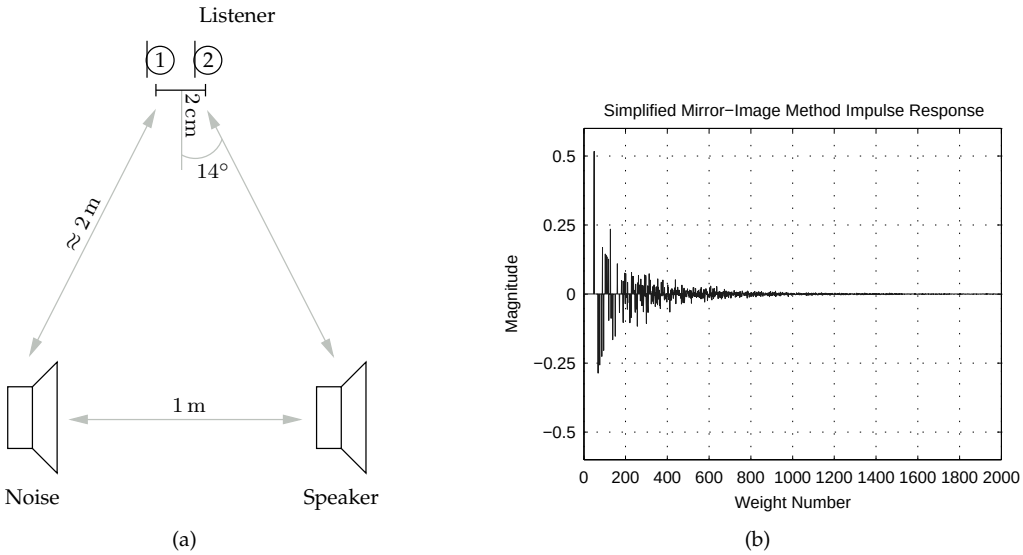


Figure 1.10: The setup for the modified room impulse response method is depicted in (a). Distance between the microphones is $2\,\mathrm{cm}$ and distance from the noise and speech source is approx. $2\,\mathrm{m}$. In Figure (b), the room impulse response with reverberation time $T_{60} = 300\,\mathrm{ms}$ is depicted.

Noise reduction is often casted using an observation model disregarding convolutional noise. This might have a profound effect on the performance when reverberation is added. Where interesting, we have test results from using an observation signal comprising additive as well as convolutional noise. The convolutional noise is added by reverberating the speech and noise signals, respectively, using a simulated room impulse response using a simplified mirror-image

method [60]. The setup used consists of a room with following dimensions $8 \times 8 \times 2.5\,\mathrm{m}$. The 2-channel observation has a microphone inter-distance of $2\,\mathrm{cm}$, and approx. $2\,\mathrm{m}$ between the noise and speaker (speech source) to the microphones, respectively. The simulated setup is depicted in Figure 1.10(a).

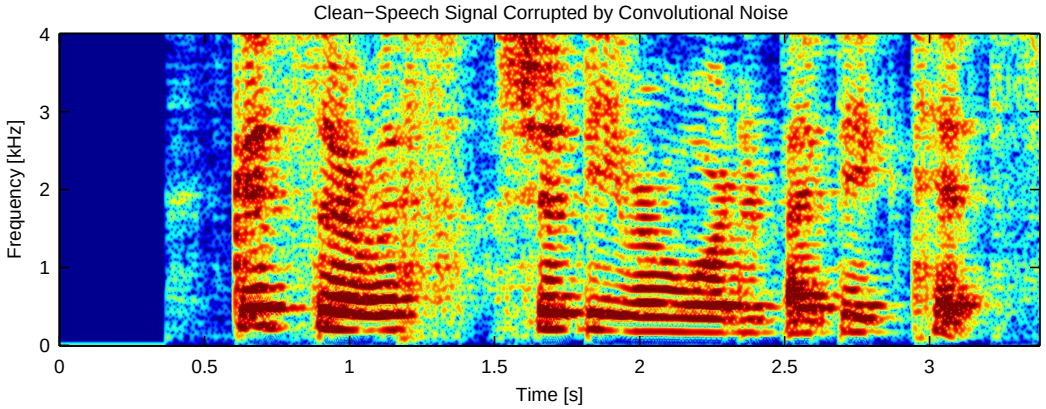


Figure 1.11: Spectrogram of the clean-speech signal of Figure 1.8 (top) convolved with the room impulse response with $T_{60} = 300\,\mathrm{ms}$ from Figure 1.10(b). The signal is seen to be spread due to the convolution with the impulse response.

The room depicted in Figure 1.10(b) has been applied to the clean-speech signal (another impulse response to the noise, of course), and the modified spectrogram of the resulting observation signal is depicted Figure 1.11. The effects are clear compared to lower half of Figure 1.8. The speech signal is seen to be timely spread or smeared. The consequence is reduced speech intelligibility. From the speech enhancement perspective it also inhibits observation of noise-dominated segments for noise estimation.

In the preceding text, the speech signal, the acoustic room impulse response and a simulation method, have been introduced together with the assessment techniques we have chosen to utilise. We are aware that the WSSM is a rather unknown measure compared to, e.g. broadband SNR. Therefore we will touch upon this problem. Regarding the signals used in the WSSM computations, it is of utmost importance to use $s[k]$ (clean-speech signal) and $\hat{s}[k]$ (estimated speech signal) as a basis for the WSSM computations - although the observation signal (might be) is degraded by convolutional noise. This important observation is elaborated in Appendix C. In Figure 1.12a figure from this appendix is reprinted. Using a $3\,\mathrm{s}$ sentence, the observation signal is degraded with different types of noise at various SNRs. The figure is to establish a relationship between input SNR and noise type with the WSSM metric - and hopefully quantify the measure and contribute towards an intuitive feeling of the speech distortion measure (similar to the widespread acceptance of SNR to measure noise).

The WSSM measure is best for low values, thus an increase in WSSM corresponds to an increase in speech distortion. The WSSM method has been divided into 23 Bark bands, which explains that low-SNR as gives rise to the most speech distortion, and that pink noise has a more profound effect on WSSM than white noise due to more energy contribution from pink noise into the low-frequency Bark bands. This corresponds very well with our expectations, as the pink noise, at a given input SNR, adds considerably more noise into the lower frequency spectrum than white noise. This affects the formants, which finally results in reduced intelligibility. When the observation is degraded with white noise and furthermore convolved with an acoustic room impulse response, the WSSM values are further increased. As expected this is most notable for high-SNR observations. For observation with low SNR, the effect of the additive noise drown out the intelligibility-degrading effect of the convolutional noise. We can conclude that the WSSM measure reflects our expected perception of intelligibility; that the measure seems to take perceptual features into account; and, that the measure is sensitive to additive and convolutional noise. Figure 1.12 furthermore provides insight into the range and scale of the WSSM measure.

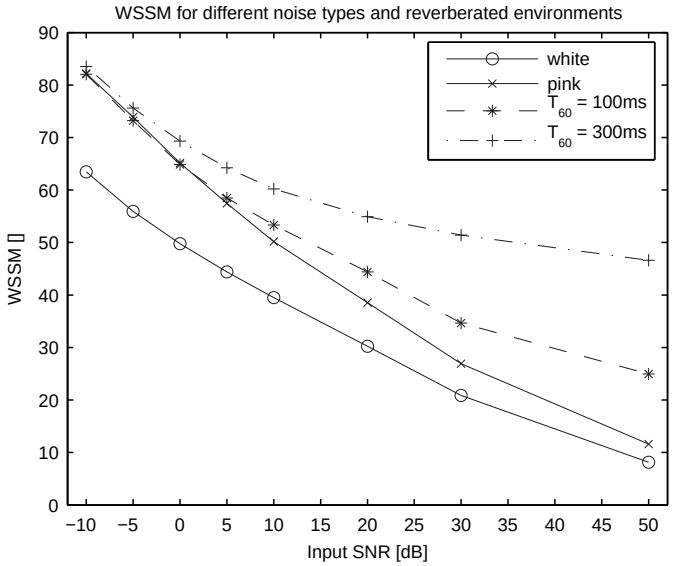


Figure 1.12: WSSM metric as a function of input SNR. Both white and pink additive noise is shown. Reverberated observations added white noise are also shown. As expected, the WSSM measure is lower for higher SNRs. Notice that pink noise has a more profound effect than white noise. As expected reverberation has a profound effect on high-SNR observations.

Having presented and argued current choices of clean-speech signals, noise signals, and reverberation effects, we have describe the functionality of `generate_scenario.m` as seen in Figure 1.7. It should by now be clear how this figure represent the work cycle for each speech enhancement method. The choices of objective assessment metrics and the modified spectrogram as a tool for visual inspecting a signal concludes the introduction chapter. Next we recast the initial problem into a problem formulation, which has grown during the project period in line width our exploration of the speech enhancement methods and experiences implementing and evaluating each method. The next section will be composed of equal parts problem formulation and project delimitation.

1.5 Problem Formulation and Project Delimitation

Motivated by the ubiquitous, world-wide problem of mitigating perceptual hearing loss among millions of hearing-impaired persons, we laid a dual-microphone hearing-aid device as the foundation of this project. Analysis of the problem field lead to characterisation of the signal(s) and room acoustics. Assessment techniques for quantitative, rapid, and repeatable speech signal evaluation lead to the choice of broadband SNR, segmental SNR ($\mathrm{SNR}_{\mathrm{SEG}}$), and the weighted spectral slope measure (WSSM), as objective metrics. Subjective assessment techniques was not included based on two assumptions. The first was that these types of tests are very time consuming and secondly, that these types of tests fits well in larger projects, with an iterative process of development, but not in a waterfall-oriented development process as enforced in this project.

Based on the fundamental assumption, that can the speech signal quality be improved for a normal hearing, we will also have improved the signal quality for a hearing-impaired, we narrow the project to algorithms meant to enhance a speech signal. This assumption implicitly opts out study of the hearing loss and audiological perspectives on regaining the hearing for hearing-impaired persons.

The goal of the project is to investigate methods to improve speech quality by digital signal processing. In specific reduce noise subject to retain, or even, improve the speech intelligibility. We define our goal as the clean-speech signal $1\,\mathrm{m}$ from the speaker source in an anechoic chamber.

As this is a master thesis, we have also taken some choices in favour of the learning process. One such choice is to gain experience in speech processing and the use and interpretation of our

chosen objective assessment methods. To that end, working with spectral subtraction enables us to work with a well-documented method. We can compare results, quantitatively by the metrics from the assessment techniques, and qualitatively by listening to the resulting speech signals, with work by other researchers. The modified spectrogram is in this respect a helpful aid, in that it reveals the structure of the signal in a manner infeasible to the objective metrics. It conveys more detailed information on a speech signal, than the metric from one of our chosen objective metrics, and comes closer to actually listening to the signal.

The dual-microphone application of course entails dual observations. A fundamental question used as a driving force of this project is whether two-channel processing can improve the noise reduction and, in specific, the speech intelligibility, compared to single-channel speech enhancement techniques. This question is further strengthened by the question whether the added cost (money) of using a dual-microphone hearing-aid is justified by an increase in performance. Motivated by these questions, we choose to include the well-known fixed beamformer and the adaptive beamformer (generalised sidelobe canceller, GSC) in the project to use a well-documented multi-channel speech enhancement reference.

We conclude this section by noting that we make a clear distinction between investigating methods and considering an implementation hereof. We will treat the functionality of a speech enhancement technique independent of whether the method in its current proof-of-concept implementation is running real time or not. This is an application of reductionism and separation-of-concerns. We believe that in order to develop high-quality speech enhancement techniques, we need to separate the concern of speech quality performance and efficient computations of an algorithm of interest.

In this section we have enumerated a number of choices and delimitations that we have found necessary and fundamental for this project. We conclude this chapter by outlining the remainder of the report and introduce the A3 methodology.

1.6 Report Outline and Reading Instructions

We have chosen to present one of many methodologies, the A3 model, in order to describe the work flow from which this report is a spin-off. The model is a common design methodology at the specialisation of applied signal processing and implementation at Aalborg University. It is one way to classify a problem or a task at hand. It defines three domains *application*, *algorithm*, and *architecture*, which in a simple manner founds the basic principles and terminology behind the discussions which have formed this project. In order to enable the reader to follow the definitions of the three domains the A3 paradigm is presented in Figure 1.13.

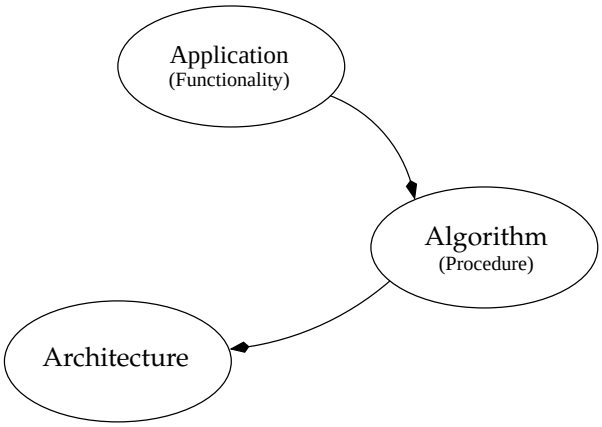


Figure 1.13: The A3 paradigm. A design methodology which uses three design domains to implicitly describe the level of abstraction of a task or project. In this report we document work on functionality of speech enhancement and implementation of one specific method, the multi-channel Wiener filter.

Basically a design can be expressed as moving from the concept or idea through the three domains, top to bottom. It is implicitly understood that a random number of iterations can be made on inter-domain and intra-domain level. Application is concerned with *what* is being done, while the algorithm is *how* it is done. For example, an adaptive filter is functionality, the NLMS and RLS are algorithms which implements this functionality. We say that an algorithm implements or realises a certain functionality. Each algorithm can be mapped to an architecture, and an architecture is concerned with *how* an algorithm is mapped to an architecture. How the algorithm is physically realised. As can be understood, the word *implementation* has different meanings in each domain. Implementing a functionality could be done by writing a specification, using use cases or MATLAB . An algorithm, on the other hand, is traditionally expressed in a computer language or mathematics using, e.g. C, C++, or MATLAB . Architecture is concerned with activities of defining and simulating yet closer to the actual hardware, e.g. using an HDL or a hardware simulation tool.

In this chapter a fundamental introduction to the topic of interest, *speech enhancement* has been given. An observation model on how to describe the problem at hand has been presented, a framework on how the techniques are measured with respect to noise reduction and speech quality was presented[5], and a problem formulation has been stated. The speech enhancement techniques that are to be investigated are yet to presented, but using the A3 notation we will start examining different methods with respect to the *functionality* of the speech enhancement techniques.

Existing techniques for speech enhancement have traditionally been split in two groups - both in literature, but also among researchers. Those techniques based on observation of a single signal, *single-microphone techniques*, and those based on multiple observations - usually two or three - called *multi-microphone techniques*. Although this can readily be justified by the structure of the algorithms, it disregards the fact that both approaches can co-exist.

In Chapter 2 we will examine single-microphone techniques, which can be seen as signal-adaptive frequency filtering of the noisy speech signal, as they exploit temporal and spectral information. A very popular method, due to its relatively low computational complexity, is the spectral subtraction technique, which we will introduce in Section 2.1 in order to have a well-defined reference method. Spectral subtraction, however, quickly reveals that the main problem in speech enhancement narrows down to obtaining an estimate of sufficient quality of the underlying stochastic process of the observation and the background noise. Evidently, this is a very complicated task impeded by the dynamics of speech and noise signals. The functionality of spectral subtraction relies mainly on having such a process estimate, thus two methods are presented; the well-established state-of-the-art method of minimum statistics proposed by Martin [59], and a voice activity detector (VAD) based on that the log-energy distribution of speech and noise have different characteristics.

In order to circumvent the need for a noise estimate, the signal subspace method is presented. Where spectral subtraction rests on the Fourier transform, the signal subspace methods uses signal-dependent transforms known from linear algebra. Specifically, the eigenvalue decomposition and the singular value decomposition are used to do principal component analysis. Having the signal transformed to a signal-dependent domain allow us to use different signal models, most notably the low-rank model of speech which facilitates integrated noise estimation.

In Chapter 3 we introduce an additional microphone to the application, such that two observations are used instead of one. Spriet et al. [83] observed that using several microphones in behind-the-ear (BTE) hearing-aid devices is a clear benefit for the resulting speech intelligibility using speech enhancement techniques. If the noise and signal sources are physically located at different positions, besides temporal information, spatial information can be exploited.

We investigate conventional beamformers to introduce different spatial filtering techniques.

[5]It is a considered a prerequisite for reading any chapter, that the reader is familiar with our choices regarding assessment techniques and test signals, which is covered in Section 1.4. To that end, Appendix C on the weighted spectral slope measure (WSSM) is considered important and enlightening, but not essential to the use and understanding of the distortion measure.

The delay-and-sum is used to illustrate the basics in beamforming techniques. We then introduce narrow- and broadband beamformers, and lastly an adaptive beamformer, the generalised side-lobe canceller (GSC). These techniques unfortunately requires some a priori information of the spatial position of the speech and noise signals in order to work. Albeit, in a hearing-aid context, the user can be assumed to look in the direction of the speaker, the small microphone inter-distance renders a priori (non-adaptive) information a robustness issue in multi-microphone speech enhancement applications [83].

Having presented known fixed and adaptive beamforming techniques, the multi-channel Wiener filter (MCWF) is presented. The method can be seen as a single-channel Wiener filter re-casted to a multi-channel scenario. Whereas the single-channel Wiener filter is based on short-time spectral information, beamformers exploits the spatial information, e.g. using the inter-channel cross-correlation. The multi-channel Wiener filter tries to combine these methods by an array of adaptive filters which are summed up in order to minimise the output error w.r.t the speech component in one of the reference channels. This turns out to be advantageous, as the method is less sensitive to short-time stationarity assumptions, and thus can alleviate some of the musical noise and other distortion artifacts known in other speech enhancement techniques.

The MCWF method is directly applicable to a dereverberation method proposed by Affes et al. [1], such that both additive and convolutional noise can be reduced. The method, however, relies on a priori information of a frequency-dependent ambiguity factor related to the acoustic room impulse response. We present this method in combination with a method we refer to as spectral addition, which is a novel approach, to our knowledge, firstly seen in this report. The method of spectral addition is used to determine this ambiguity factor.

In the Chapters 2 and 3 the application of speech enhancement has now been described using different functionalities, if we use the terminology in the A3 paradigm. The MCWF is shown to obtain promising results with respect to noise reduction, but also has a rather high computational complexity. In Chapter 4 we move on to the algorithm level in the A3 model and introduce an algorithm, the recursive GSVD, that is able to lower this high complexity such that the MCWF can be computed in real-time. A fixed-point analysis in then conducted for this real-time method. According to the A3 model an architectural analysis is not performed, the scope of this report ends with an implementable algorithm, which has been examined with respect to word lengths and complexity.

CHAPTER 2

Noise Reduction Techniques Based on a Single-Channel Observation

The objective of speech enhancement is to estimate the speech signal, $s(t)$, from a set of observations, $y_m(t)$, each degraded with background noise, $b_m(t)$, and convolved with a room response, $g_m(t)$. In the previous chapter we defined the corresponding continuous-time observation model

$$y_m(t) = g_m(t) * s(t) + b_m(t) \qquad m = 1, \ldots, M \tag{2.1}$$

where M denotes the number of different observations available, which corresponds to the number of microphones. Equation (2.1) can be divided into two major parts, removing additive background noise, $b_m(t)$, and deconvolving the effect from the acoustical environment, remove $g_m(t)$. In this chapter we will focus on the removal of additive background noise with only one observation available. The observation model for this chapter can therefore be rewritten as

$$y(t) = s(t) + b(t) \tag{2.2}$$

One of the most common single microphone speech enhancement techniques for additive noise reduction is spectral subtraction, due to its simplicity and low complexity. Pioneering work by Berouti et al. [6] and Boll [7], using power spectral subtraction and magnitude spectral subtraction, respectively, led to a generalised spectral subtraction where empirical parameters can be adjusted in order to fine-tune the technique. Spectral subtraction have been reported capable of removing stationary background noise, however, at the expense of speech distortion [13]. Spectral subtraction introduces *musical noise*, which arises from continuous on-off switching of low-level pure tones. This distortion is caused when the estimate of the noise spectrum is subtracted from the instantaneous noise spectrum yielding narrow spectral-varying peaks [6]. Many extensions to spectral subtraction have focused on minimising the effect of musical noise [30, 22, 23].

Another common method for single-channel speech enhancement is Wiener filtering, which can provide noise reduction without considerable distortion in the speech estimate [15], given the speech and background noise are uncorrelated and the background noise is stationary. Wiener filtering can, with few exceptions, be interpreted as a time domain implementation of some form of spectral subtraction [13].

A third method for noise reduction is the signal subspace technique. Where the spectral subtraction is based on a signal-independent transformation, the Fourier transform, the signal subspace is based on a signal-dependent transformation. The method has its roots in low-rank modelling of speech and is to a large extend based on linear algebra. The basic assumption is, that the noisy speech signal, the observation, can be decomposed into a signal-plus-noise subspace and a noise subspace. The clean signal can then be estimated by eliminating the noise subspace. This decomposition can theoretically be made using the Karhunen-Loève transform, but is traditionally approximated using an eigenvalue decomposition or a singular value decomposition,

proposed by Ephraim and Van Trees [24] and Jensen et al. [49], respectively. Although the method can be interpreted, as revealing interesting perceptual features, the method dependent on a noise estimate with non-white noise present and suffers from musical noise, as spectral subtraction.

In Section 2.1 spectral subtraction is presented, the method however needs a noise estimate. In Section 2.2 and 2.3 two different approaches on how to obtain this estimate are presented. The last section, 2.4, presents the signal-subspace approach, which only for non-white noise needs an external noise estimate. Adding reverberation to the observation is expected to decrease performance for both methods. This will be tested in the discussion and conclusion at the end of this chapter.

2.1 Spectral Subtraction

Spectral subtraction stems from a seemingly simple idea. Transform the observation model (2.2) to the frequency domain and rearrange the formula to recover the speech signal. However, as the transform reveals phase and magnitude in the complex coefficients, we have a problem. The idea is to discard the phase information and define spectral subtraction on the power spectrum alone [52, 7]. The idea can be formulated as

$$\left|Y(\omega)\right|^2 = \left|S(\omega)\right|^2 + \left|B(\omega)\right|^2 \tag{2.3}$$

and the clean speech signal estimated as

$$\left|\hat{S}(\omega)\right|^2 = \left|Y(\omega)\right|^2 - \left|B(\omega)\right|^2 \tag{2.4a}$$

$$\hat{S}(\omega) = \left|\hat{S}(w)\right| \angle Y(\omega) \tag{2.4b}$$

$$\hat{s}[k] = \mathcal{F}^{-1}\left\{\hat{S}(\omega)\right\}] \tag{2.4c}$$

The estimate of the clean speech spectral magnitude is recombined with the noisy phase of the observed signal, $\angle Y(\omega)$, yielding an estimate of the speech signal. This will naturally lead to a distortion caused by the noisy phase, however if the SNR in a local frame is above $6\,\mathrm{dB}$ the phase distortion has been found to be inaudible [51]. Since speech signals can only be assumed stationary for short time frames, the recorded signal is divided into frames, in which the estimated noise spectrum is subtracted from the observation giving an estimated of the speech spectrum.

The problem of obtaining an estimate of the noise, $\left|B(\omega)\right|^2$, is further described in Appendix B. One problem is the characteristics of the noise estimator, bias and variance, mainly, another consideration is on how to observed the noise signal. In a speech enhancement application only the noisy speech signal is observable. One of the early methods assumed the first second to be noise only and estimate the noise from this signal [6]. This approach had the disadvantage, that was the signal in fact speech, the method would erroneously remove speech components. Secondly, was the first second in fact noise, the assumption meant that the noise had to be stationary for the remainder period of operation. Clearly a very unrealistic assumption in most applications. As an alternative, it has often been proposed to use a voice activity detector (VAD) in order to detect speech-dominated and noise-dominated time frames. The noise-dominated time frames can be used to estimate the (possibly non-stationary) background noise, however, the reader would likely have realised, that this merely shifts the problem of noise estimation to developing reliable VADs which successfully detects speech-dominated time frames. This has lead to extensive research in good VADs. For noise estimation, alternative more exotic noise estimation procedures have recently reported promising results in non-stationary environments. Most notable the method by Martin [57, 58, 59] have been accepted as a state-of-the-art method, which works without the use of an explicit VAD. Also Cohen have shown interesting work [11, 12, 10] in the field, however, both methods common fail when applied in an environment with non-stationary, speech-like noise.

The spectral subtraction can be described using the *generalised spectral subtraction* given as

$$D(\omega) = \big|Y(\omega)\big|^{\gamma} - \alpha\big|B(\omega)\big|^{\gamma} \tag{2.5a}$$

$$\big|\hat{S}(\omega)\big| = \begin{cases} D(\omega)^{1/\gamma} & \text{if} \quad D(\omega)^{1/\gamma} > \beta\big|B(\omega)\big| \\ \beta\big|B(\omega)\big| & \text{otherwise} \end{cases} \tag{2.5b}$$

$$\text{with} \quad \alpha \geq 1 \quad \text{and} \quad 0 < \beta \ll 1$$

where α is a subtraction factor and β is a spectral floor parameter. The speech signal can be recovered using (2.4c). The exponent $\gamma = 2$ corresponds to power spectral subtraction [6] and $\gamma = 1$ corresponds to magnitude spectral subtraction [7].

In the following sections noise reduction performance of the spectral subtraction approach will be investigated. Firstly the effect of the empirical parameters are qualified by a number of tests using spectrograms, secondly the noise reduction performance is investigated for the case of white and pink coloured noise.

2.1.1 Experimental Results

We have chosen to implement the generalised spectral subtraction, in order to create a reference method to measure other methods against. This stems from the fact, that spectral subtraction, without exaggeration, must be the most widely used reference method. Secondly, we aim at gaining experience using the objective assessment techniques described in Section 1.4 and Section 1.3.2. In order to test the implementation, a test signal with a female voice uttering: "*Good service should be rewarded with big tips*", have been added white or pink noise at a given SNR. The signal and the assessment metrics are further described in Section 1.4 on our choices for test signals and assessment metrics.

For the evaluation of the spectral subtraction method, we have used two different noise power spectra. The first noise spectrum is obtained directly from the additive noise signal by Welch averaging, see Appendix B Section B.2.2, over the noise signal, $b[k]$, which is therefore not an estimate based on an observable signal, but rather the *true noise power spectrum* in a stochastic sense. The second noise spectrum is estimated by averaging noise-only frames. These are detected by an ideal VAD. The construction and use of a VAD is discussed in Section 2.3.

The noise-dominated frames are averaged using a single-pole recursive smoothing with forgetting factor $\lambda = 0.96$. The recursive noise estimate can be expressed as

$$P(r, n) = \lambda P(r-1, n) + (1-\lambda)|Y(r, n)|^2 \tag{2.6}$$

by defining the instantaneous power spectral density (PSD), $P(r, n) = |Y(r, n)|^2$, which is used as the discrete-time version of $|B(t, \omega)|$ in (2.5b), as

$$|Y(r, n)|^2 = \frac{\frac{1}{N}\left|\sum_{k=0}^{N-1} w[k]y[k + rR]\exp(-2\pi jkn/N)\right|^2}{\frac{1}{N}\sum_{k=0}^{N-1}\big|w[k]\big|^2} \tag{2.7}$$

where N is the number of frequency bins and R is the amount of increment, i.e. $N - R = L$, where L is the amount of overlap. The window function, $w[k]$, is used to tapper the time-domain signal for reducing spectral leakage.

It is well established, that whereas the mean of the periodogram estimate, (2.7), approaches zero for large sequences, the variance is independent of the signal length, and, in fact, dependent on the squared PSD magnitude [46]. Time-averaging over multiple periodograms is one way to reduce the variance of the periodogram estimate, e.g. using the recursive method of (2.6) (see Appendix B, Section B.2.2).

The settings used for the test using generalised spectral subtraction are 256-samples Hann tappered data with 50% overlap, i.e. 128 samples. Referring to (2.5b), we have found $\alpha = 5$, $\beta = 0.01$, and $\gamma = 2$ good empirical parameters for the signal under consideration. These settings have been adjusted as to optimise the spectral subtraction performance to the objective assessment metrics. Thereby, we have obtained $\alpha = 5$, which corresponds to an overestimation of the noise, as found advantageously in [6], and the spectral floor parameter, $\beta = 0.01$, corresponding $20\,\mathrm{dB}$ attenuation of the estimated (noise) signal. The spectral subtraction procedure is to be found in our implementation (`berouti79_mod.m`, see Appendix D). Using these settings, we obtain the results listed in Table 2.1.

Noise estimate	SNR[dB]	ΔSNR[dB]	$\mathrm{SNR_{SEG}}$[dB]	WSSM[·]
Observation	5.0	–	-0.4	44.4
Known noise	12.8	7.8	2.3	52.5
Ideal VAD	12.5	7.5	2.2	53.7

Table 2.1: The results obtained using the generalised spectral subtraction with empirical parameters $\alpha = 5$, $\beta = 0.01$, and $\gamma = 2$. Furthermore a 256-samples Hann window is used with 50% overlap. First line is the observation, while the following describes the true noise spectrum (averaging over the noise-only signal), and the averaged noise-dominated speech frames.

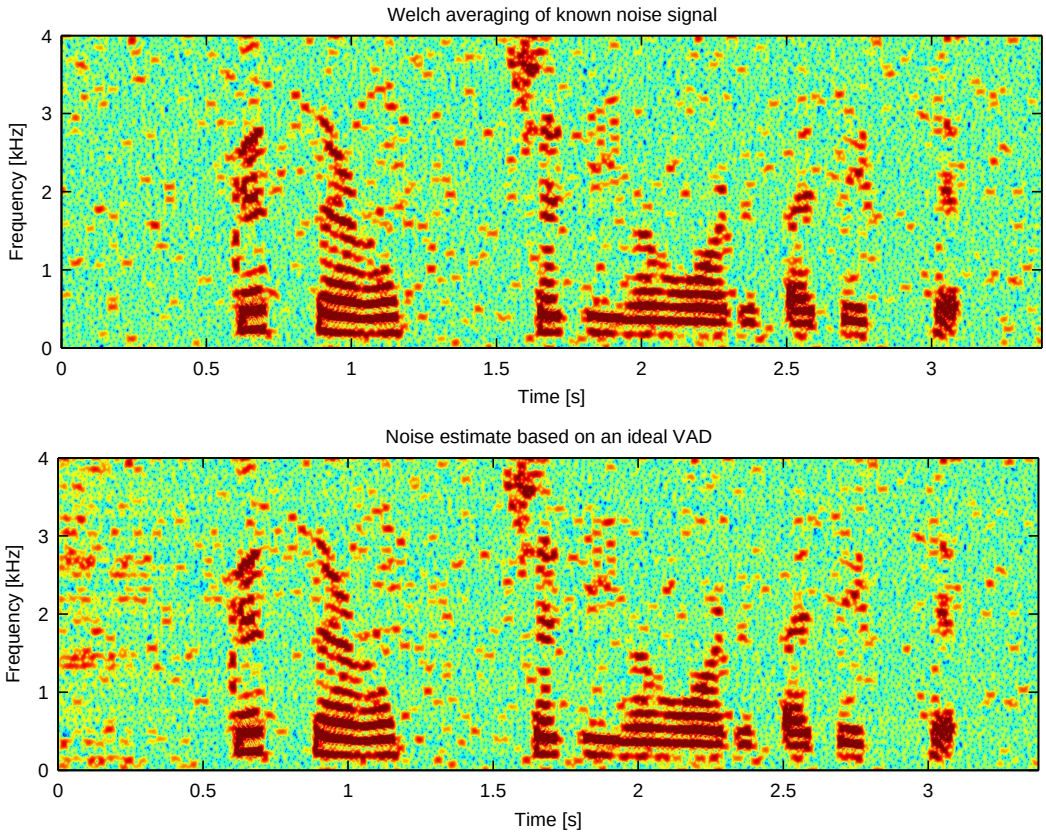


Figure 2.1: Modified spectrogram of the estimated speech signal using the generalised spectral subtraction. In the top figure, the noise is known and the spectral density estimated by Welch averaging. In the lower figure the noise is estimated using averaging of noise only frames identified using an ideal VAD. Besides adaptation/edge effects of the recursive noise estimate, the lower figure, only subtle differences are noticed between the spectrograms.

The results are depicted in Figure 2.1, where spectrograms for the two speech estimates are plotted. The green/cyan background colour is the noise floor which is adjusted by the β-value. There are some distinct dark-red areas, which are easily detected as speech energy, if we com-

pare these two figures with the clean speech spectrogram in Figure 1.8 on page 15. There are, however, also some small random red dots, which are not speech energy, these are continuous on-off switching of low-level pure tones, which causes the *musical noise*. This distortion results in an increased WSSM. This would be an convenient moment to add a discussion on the sample-by-sample measure SNR. The definition of broadband SNR was in (1.7), substituting $\hat{s}[k] = s_{m,o}[k] + b_{m,p}[k]$, defined as

$$\mathrm{SNR} = 10\log_{10}\left[\frac{\sum\limits_{n} s^2[k]}{\sum\limits_{n}(s[k] - (s_{m,o}[k] + b_{m,o}[k]))^2}\right] \tag{2.8}$$

tacitly assuming that $s_{m,o}[k]$ equals $s[k]$ in order to estimate $b_{m,o}[k]$. If this assumption does not hold, we assume, that we have not estimated the speech signal correctly. As a side-effect, the SNR measure is affected, as both residual noise, $b_{m,o}[k]$, and residual speech, $r_s = s_{m,o}[k] - s[k]$, are assumed noise sources. Although this is not wrong from a perceptive viewpoint, after all, we expect more noise if the speech residual has increased, it does render the SNR measure with the definition of (2.8) partly a measure of noise reduction, partly a measure of speech distortion. This fact is important to keep in mind when comparing and evaluation using broadband SNR.

By comparing Figure 2.1 to Figure 1.8, we observe differences which might influence the SNR measure along the lines of the discussion in the previous paragraph. The estimated speech signal is distorted because parts of the speech signals in some spectral bands are attenuated and somewhere even completely removed. To that end, remember, that the spectrograms are limited in dynamic range to 60 dB. The effect of musical noise and residual noise will also contribute to the noise in the denominator of (2.8). To finish this short discussion of the SNR measure, we add a note on WSSM. The spectral peak weighting functionality of WSSM leads to an increased score from signals containing musical noise, as these are heavily weighted in the final computation of the metric.

Returning to the spectral subtraction technique, we wish to illustrate the effect of spectral floor parameter. We have repeated the simulations from Table 2.1 using $\beta = 0$. The spectral floor is now infinity, which by (2.5b) means for all negative $D(\omega)$, the resulting speech estimate is set to zero. The simulations results using $\beta = 0$ are listed in Table 2.1.

Noise estimate	SNR[dB]	ΔSNR[dB]	$\mathrm{SNR}_{\mathrm{SEG}}$[dB]	WSSM[·]
Observation	5.0	–	-0.4	44.4
Known noise	12.7	7.7	2.4	66.3
Ideal VAD	12.4	7.4	2.4	66.9

Table 2.2: Simulations results for spectral subtraction using empirical parameters $\alpha = 5$, $\beta = 0$, and $\gamma = 2$. As for Table 2.1, 256-samples Hann window with 50% overlap is employed. The noise reduction performance is seen to be unaffected of the change in the β-parameter, however the speech distortion is substantially increased.

We can see that the broadband SNR improvement is not significantly altered by lowering the spectral floor. The WSSM, on the other hand, is considerable increased, since the musical noise now is more distinct. This can also be seen in Figure 2.2 where a modified spectrogram is shown of the speech estimate using known noise and spectral subtraction with $\beta = 0$. The red random dots that can be classified as musical noise are now more emphasised. Actually, the musical noise is not emphasised, rather the background noise is reduced. This has little effect on the noise reduction, as the noise was already substantially reduced, but increases the spectral slopes, to which the WSSM measure responds. What we observe in Table 2.2 and Figure 2.2 also fits very well to informal listening tests among the group members. The smearing of the random, fluctuating peaks in Figure 2.2 results from the periodogram as a method, not from the signal itself. This is a common problem when visualising in the joint time-frequency plane.

As the results using $\beta = 0$ revealed, the β value controls the level of background noise, therefore the name, noise floor. However, much effort has been put into devising better ways of con-

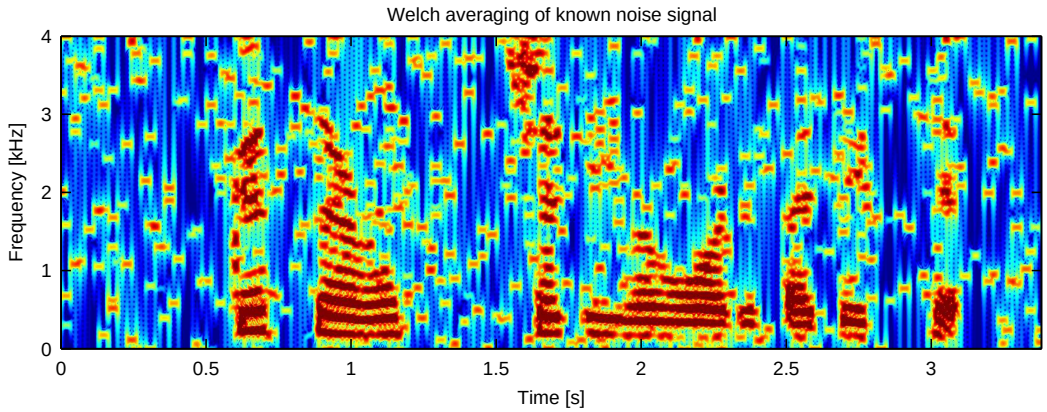


Figure 2.2: Modified spectrogram of the estimated speech signal using the generalised spectral subtraction and known noise. Parameters $\alpha = 5$, $\beta = 0$, and $\gamma = 2$ have been used. The infinity noise-floor parameter, β, has emphasised the musical noise effect.

trolling the estimator of (2.5b) in order to reduce the effects of musical noise [22, 23, 30]. If we use $\beta = 0.1$, we obtain good noise reduction ($\mathrm{SNR} = 11.5\,\mathrm{dB}$, for known-noise case), but less than using $\beta = 0.01$. As expected the increase in background noise has drowned some of the musical noise and reduced the WSSM to 41.7, which is more than 10 points lower than using $\beta = 0.01$, see Table 2.1.

2.1.2 Experimental Results For White and Pink Noise Testing

In order to test spectral subtraction against different noise types and at different input SNR levels, we have used the test setup described in Section 1.4. The setup consists of two sentences, a male uttering: "*The boost is helpful, but inadequate*"; and a female uttering: "*Good service should be rewarded with big tips*". White and pink noise have been used at SNRs, -5, 0, 5, and $10\,\mathrm{dB}$. To measure noise reduction and speech distortion we have employed SNR improvement and WSSM improvement, respectively. Because WSSM is a less-known method and seeing that it is an absolute value representing the speech quality, it might seem unjustified to measure WSSM improvement. Even so, we have chosen to use the improvement as the alternative, plotting the reference and output WSSM for each test case, is likely to cause more confusion, than clearness. Positive WSSM improvements corresponds to increase in speech quality.

We have used power spectral subtraction (generalised spectral subtraction of (2.5b) with $\gamma = 2$) with empirical parameters $\beta = 0.01$ and $\alpha = 5$. The noise estimate is obtained by averaging noise-only frames, obtained by using an ideal VAD, with a forgetting factor $\lambda = 0.96$ referring to (2.6). The results from these simulations are depicted in Figure 2.3.

In Figure 2.3 it is observed, that spectral subtraction can remove substantial noise both for the white- and pink-noise case. As expected, the noise reduction is considerably higher for low input SNR than for high input SNR, (approx. $5\,\mathrm{dB}$ max-to-min noise reduction).

From the figure, a tendency towards ΔWSSM decrease is observed when the input SNR increases. The resulting speech estimate has been distorted as a consequence of applying the noise reduction. It seems, that speech distortion is less for pink than white noise. This is due to the noise-energy concentration in the lower frequency spectrum for pink noise. When removing noise, the musical noise will be predominant in the lower frequency spectrum. The musical noise residing in the upper frequency spectrum will contribute to less distortion (as referred in Appendix C WSSM measures spectral slopes in several Bark bands, which weight low-frequency bands more than high frequency bands), than for the case of white noise.

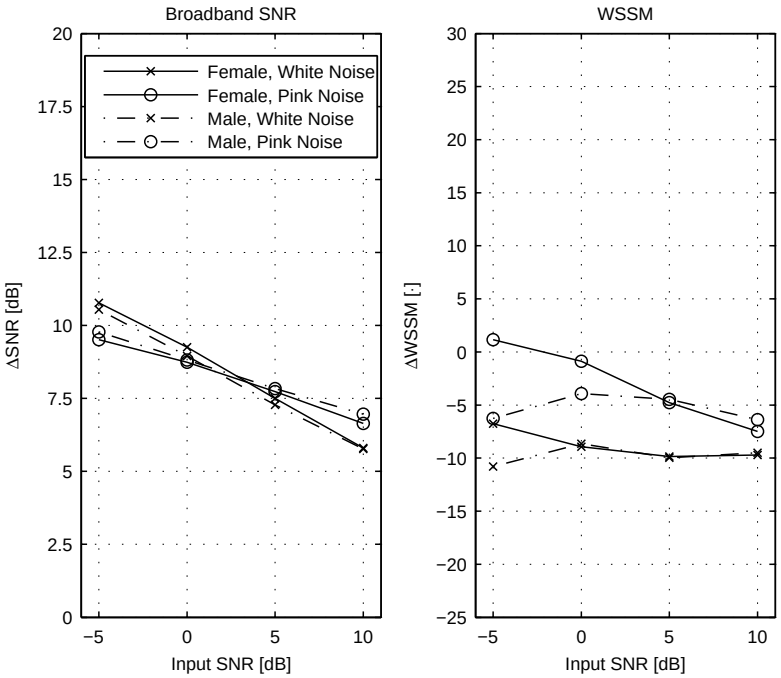


Figure 2.3: Noise reduction (SNR) and speech distortion (WSSM) for power spectral subtraction empirical parameters $\beta = 0.01$, $\alpha = 5$, and $\gamma = 2$. The noise is estimated using recursive averaging over noise-only frames based on the decision of an ideal VAD.

2.1.3 Conclusion

Spectral subtraction is a relative simple single-channel noise reduction method. It rests on the Fourier transform and simple decision-making based on a number of empirical parameters. The method is shown to be able to reduce white and pink stationary, additive noise, by an increase in broadband ΔSNR. However, the method is only able to obtain the speech estimate, which is seen as dark red traces in the spectrogram for the observation, presented in Figure 1.9 in Section 1.4 and not the speech masked by noise. Thus, the improvement comes at the cost of speech distortion. Most notable the musical noise effect, which is an annoying effect for the speech quality. Efforts have been made in many years of research in order to reduce or drown the musical noise, but as a stand-alone method, spectral subtraction fails to deliver enhanced speech signals with reduced noise without reducing the quality.

The results on which this conclusion rests are based on removing stationary, additive noise. To that end, the noise estimation is a crucial provision for the functionality of spectral subtraction. Reducing noise with non-stationary characteristics, e.g. music or traffic noise, will impede the performance of the spectral subtraction, because noise estimation is more difficult. The noise estimation in this section is based on recursive smoothing on noise-dominated time segments by employing an ideal VAD.

In the following sections we will investigate more realistic methods. First, in next section, we introduce a state-of-the-art method, which rests on frequency-dependent optimal smoothing, as opposed to the simple approach of Eq. (2.6). In Section 2.3 we introduce a VAD technique based on estimation of the noise log-energy distribution.

2.2 Minimum Statistics Based Noise Estimation and Spectral Subtraction

In noise reduction of speech signals the objective is to estimate the speech component in an observation signal of degraded speech. The dual problem of estimating the noise component is

often inherently part of the noise reduction formulation. Some sort of noise estimation is required in order to apply the method under consideration. For spectral subtraction, it is assumed that a frequency-dependent noise estimate is known, but spectral subtraction formulates no specific method on how to obtain this estimate. The problem is, so to say, just shifted from speech estimation to noise estimation.

In this section a state-of-the-art noise estimation procedure, particularly suited for spectral subtraction, is presented. As opposed to the recursive smoothing with a constant forgetting factor, λ, as presented in previous section, the current method explicitly considers non-stationarity of the noise signal. The ideal VAD employed in previous section is replaced by an SNR estimate to establish the best suitable smoothing parameter.

The noise spectral density estimation based on minimum statistics, proposed by Martin [57, 58, 59], rests on two basic assumptions: (1) the speech and interfering noise are statistically independent; (2) the power of the noisy speech signal often decays to the power level of the interfering noise. The latter is true for speech signals, as they are highly non-stationary and take on varying stochastic characteristics over time. Speech signals are known not to be continuously present and often decay to a noise floor. This is what we would like to exploit in this method.

The estimation is carried out entirely in the frequency domain without the use of a voice activity detector (VAD). The observed signal, $y[k]$, is windowed and transformed by a short-time Fourier transform, and the instantaneous power spectral density of the observation is computed (see Eq. (B.15) in Appendix B)

$$|Y(r,n)|^2 = \frac{\frac{1}{N}\left|\sum_{k=0}^{N-1} w[k]y[k+rR]\exp(-2\pi jkn/N)\right|^2}{\frac{1}{N}\sum_{k=0}^{N-1}\left|w[k]\right|^2} \tag{2.9}$$

where N is the number of frequency bins and R is the amount of increment, i.e. $N-R=L$, where L is the amount of overlap. The window function, $w[k]$, is used to tapper the time-domain signal for reducing spectral leakage. By smoothing the power spectral density and tracking the minimum over time, it has been shown by Martin, that it is possible to obtain an estimate of the noise power spectral density, even for non-stationary noise. It has been noted, though, that successful tracking depends on the *assumption of slowly-varying PSD of the interference*, and that the procedure fails for interfering sources with rapidly time-varying PSDs, such as speech or music [41].

The two elements in the algorithm are: (1) smoothing of the periodogram, and; (2) tracking the minimum over a given time window. The smoothing should be great enough (non-reactive) in order to obtain a good power spectral density estimate and suppress spurious peaks, but also low enough (reactive) to follow overall changes in the instantaneous power spectral density. The proposed recursive smoothing of the power spectral density by a moving average can be defined as

$$P(r,n) = \alpha(r,n)P(r-1,n) + (1-\alpha(r,n))|Y(r,n)|^2 \tag{2.10}$$

Notice the difference with the formulation in Section 2.1, where a constant smoothing parameter (forgetting factor) is used. In addition, the smoothing parameter, $\alpha(r,n)$, is frequency and time dependent.

It is well established, that whereas the mean of the periodogram estimate approaches zero for large sequences, the variance is independent of the signal length, and, in fact, dependent on the squared PSD magnitude [46]. Time-averaging over multiple periodograms is one way to reduce the variance of the periodogram estimate, e.g. using the recursive method of (2.10) (see Appendix B, Section B.2.2). To that end, it is desirable to keep the smoothing parameter, α, as close to one as possible. This, however, has the undesirable effect of rendering the estimator invariant to most variations of the instantaneous power spectral density, $|Y(r,n)|^2$. As a remedy Martin [59] proposes to use an optimal, time-varying, and frequency-dependent smoothing parameter

which is based on a smoothed *a posteriori* signal-to-noise ratio, $P(r-1,n)/\sigma_N^2(r,n)$,

$$\alpha_{\text{opt}}(r,n) = \frac{1}{1+(|Y(r-1,n)|^2/\hat{\sigma}_N^2(r,n)-1)^2} \tag{2.11}$$

By monitoring the error (distance) between the smoothed power spectral density and the instantaneous power spectral density, the reactiveness of the smoothing is adjusted, i.e., the α-parameter is adjusted to lower values. To avoid dead-locking α is confined to an interval, α_{min} to α_{max}. The algorithm from [59] is reprinted in Algorithm 1 for completeness. As such, it is the same as in the original paper, but a few adjustments have been made to improve readability and reflect our implementation (`martin2001.m`, see Appendix D).

The noise power spectral density is estimated by determining the minimum of the smoothed power spectral density, $P_{min\,u}$, of the observed signal within a finite-length window of length D. Since the minimum value of a set of stochastic variables is smaller than their mean, we need to compensate for the minimum estimate bias [59]. Extensive derivations, which include an approximation of the mean of the minimum based on simulations (interpolation from Table III in [59]), arrives in a number of expressions (see the original paper).

The length-D window is sub-divided into a number of sub-windows in order to reduce computational complexity (number of comparisons) while maintaining the tracking latency below a certain level. D should be long enough to bridge speech-gaps, around $500\,\text{ms} - 1\,\text{s}$. D is equal to UV, where U is the number of sub-windows and V is the number of "samples" (possible minima) in each sub-window. In the following, some values are computed for the overall window and some for the sub-window (denoted by subscript $_{sub}$). Algorithm 1 can be summarised as follows: For each time and frequency index, compute the smoothing parameter, α, the smoothed periodogram, $P(r,n)$, and the minimum statistics bias correction, B_{min} and $B_{min\,sub}$. Keep track of the local minimum for each sub-window in a circular buffer, $actmin$, indexed by $1 \leq U_c \leq U$. The samples in each sub-window is indexed by $1 \leq subwc \leq V$.

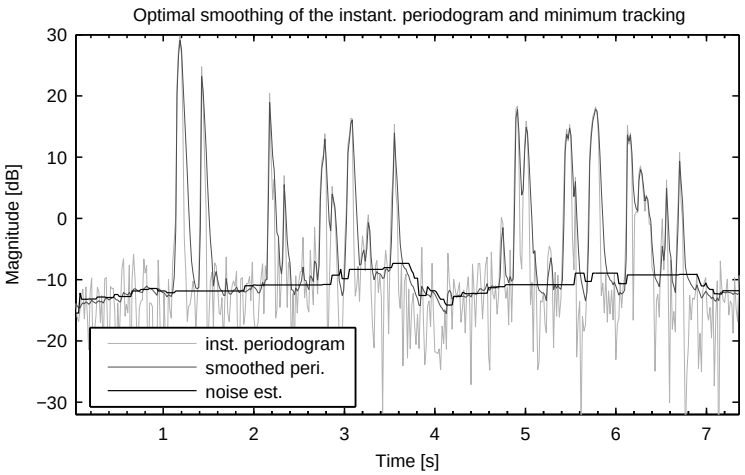


Figure 2.4: Power spectral magnitude as a function of time segments computed in the 16*th* frequency bin (1 kHz). The optimally smoothed periodogram mimics the presence of speech signal (i.e. instantaneous SNR) well. The noise estimation is seen to follow the minima of the smoothed power spectral magnitude.

To illustrate the tracking ability of the noise power spectral density estimation, we have applied the procedure to a noisy speech signal observation. The signal is a concatenation of the sentence: "*Good service should be rewarded with big tips*", uttered by a female, and the sentence: "*The boost is helpful, but inadequate*", uttered by a male speaker. The SNR used was $10\,\text{dB}$. Using a Hann window of 256 samples and 50% overlap, the sub-window parameters, U and V have been chosen to $V = 6$ samples and $U = 8$ window samples, which gives a tracking latency of $\big(UV \cdot 128\big)/f_s = 768$ ms. In Figure 2.4, the resulting instantaneous periodogram, the smoothed

Noise power spectral density estimation based on optimal smoothing and minimum statistics, Martin [59].

1. compute smoothing parameter $\hat{\alpha}(r,n)$
2. compute smoothed power $P(r,n)$
3. compute bias correction $B_{min}(r,n)$ and $B_{min\,sub}(r,n)$
4. compute $\overline{Q^{-1}} = \frac{1}{L}\sum_{n=0}^{L-1} Q^{-1}(r,n)$
5. set $k\_mod(n) = 0$ for all n
6. if $P(r,n)B_{min}(r,n)B_c(r,n) < actmin(U\_c,k)$

   $actmin(U\_c,n) = P(r,n)B_{min}(r,n)B_c(r,n)$

   $actmin\_sub(n) = P(r,n)B_{min\,sub}(k)B_c(r,n)$

   set $k\_mod(n) = 1$
7. if $subwc == V$

   if $k\_mod(n) == 1$ then $lmin\_flag(n) = 0$

   $P_{min\,u}(n) = \min(actmin(\text{all},n))$ (minimum for each frequency in the length-U circular buffer $actmin$)

   if $(\overline{Q^{-1}}(r) < 0.03)$ then $noise\_slope\_max = 8$

   elseif $(\overline{Q^{-1}}(r) < 0.05)$ then $noise\_slope\_max = 4$

   elseif $(\overline{Q^{-1}}(r) < 0.06)$ then $noise\_slope\_max = 2$

   else $noise\_slope\_max = 1.2$

   if $\{lmin\_flag(n)$ & $(actmin\_sub(n) < noise\_slope\_max P_{min\,u}(n))$ & $(actmin\_sub(n) > P_{min\,u}(n))\}$

   $P_{min\,u}(n) = actmin\_sub(n)$

   $actmin(\text{all},n) = actmin\_sub(k)$ (replace all buffer values in $actmin$ by the value in $actmin\_min$)

   $lmin\_flag(n) = 0$

   set $subwc = 1$ (increment sub-window counter)

   $U\_c = \mod(U\_c,U)+1$ (update pointer for circular buffer $actmin$)

   $actmin(U_c,n) = actmin_{max}(n)$ (re-initiate next buffer entry $actmin(U\_c,n)$ to its maximum value)

   $actmin\_sub(U_c,n) = actmin\_sub_{max}(n)$

   $\hat{\sigma}_N^2(r,n) = \hat{\sigma}_N^2(r-1,n)$
8. else

   if $subwc > 1$ then

   if $k\_mod(n) == 1$

   $l\_min\_flag(n) = 1$

   compute $\hat{\sigma}_N^2(r,n) = \min(actmin\_sub(n), P_{min\,u}(n))$

   set $P_{min\,u}(n) = \hat{\sigma}_N^2(r,n)$

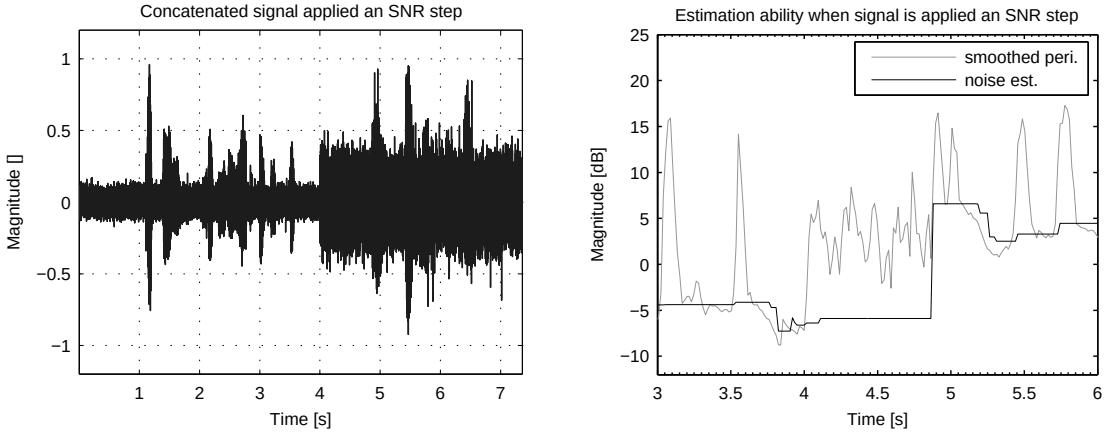
   else

   $\hat{\sigma}_N^2(r,n) = \hat{\sigma}_N^2(r-1,n)$

   $subwc = subwc + 1$ (increment sub-window counter)

Algorithm 1: The noise spectral density estimator based on optimal smoothing of the observed noisy speech signal instantaneous spectral density and tracking minimum statistics.

periodogram, and the bias-compensated, estimated noise power spectral density for $1\,\mathrm{kHz}$ ($16th$ bin of 256) is shown. The method is seen to track the minima.



(a) Concatenated signal of the sentences: "*Good service should be rewarded with big tips*" (female), and "*The boost is helpful, but inadequate*" (male), with an initial $\mathrm{SNR} = 5\,\mathrm{dB}$. After $4\,\mathrm{s}$ an SNR step of $-10\,\mathrm{dB}$ is applied.

(b) The tracking of the SNR-step seen in Figure 2.5(a). The latency is $780\,\mathrm{ms}$ due to the U and V parameters. The latency is observed to be in accordance with the expected duration.

Figure 2.5: Noise estimation in a dynamic noise environment (here a step). The tracking ability, and latency, is observed when the signal is applied an SNR step. In Figure 2.5(b) the $16th$ frequency bin is shown.

To facilitate a test of the ability to track dynamic changes, the same signal at $5\,\mathrm{dB}$ has been applied an SNR step of $-10\,\mathrm{dB}$ (to $-5\,\mathrm{dB}$) after a duration of $4\,\mathrm{s}$. The noisy time-signal is seen in Figure 2.5(a) while the resulting tracking is seen in Figure 2.5(b). The SNR step can easily be identified on the smoothed periodogram, while the noise estimation follows after some time, which is observed to correspond very well with the latency introduced by the sub-windowing. Circa $0.8\,\mathrm{s}$ after the SNR step is applied, the noise estimation follows to a floor corresponding to the minimum at that time instance. After yet some time, the estimation procedure is seen to follow the varying minimum of the smoothed periodogram, as seen in Figure 2.5(b), between $5\,\mathrm{s}$ and $6\,\mathrm{s}$.

The delay introduced by the sub-windowing of the algorithm clearly manifests itself in Figure 2.5(b). Had the background noise been rather complex, compared to the white noise used, say background babble noise, clearly the latency would have impeded the background noise tracking. This stems from the assumption that the often-occurring speech pauses can be exploited in order to estimate the background noise. If the noise is not stationary, for example another speaker, the noise is rather difficult to estimate. Clearly, the latency can be reduced by adjusting the number of sub-windows, but this will also reduce the accuracy of the minimum statistics approach as the minimum search is performed over a shorter time span. Speech signals can consist of long high-energy utterances, and the minimum procedure must be able to cope with this time span, which for most situations can be assumed less than $0.5 - 0.7\,\mathrm{s}$.

Having successfully described the estimation procedure facilitating optimal smoothing and minimum statistics, and described experimental results to verify the functionality, we apply the procedure to spectral subtraction, in order to compare the performance gain/loss to spectral subtraction using an ideal VAD. To accentuate the observations of previous paragraph, tests with spectral subtraction will be carried out for $5\,\mathrm{dB}$ white noise to gain insight into the quality of the estimation results.

2.2.1 Minimum Statistics Based Noise Estimation Applied to Spectral Subtraction

Referring to Section 2.1 on spectral subtraction, the purpose of this section is to compare the performance of the power spectral subtraction using an ideal VAD and the noise estimation procedure described in previous section. Tests will be carried out for the white noise, as well as for the pink noise, case.

In Table 2.3 the results from using either an ideal VAD and recursive smoothing ($\lambda = 0.96$), or the method of minimum statistics as noise estimation in order to carry out spectral subtraction. The empirical parameters, see Eq. (2.5b) in Section 2.1, has been found to be best at $\alpha = 5$, $\beta = 0.01$, $\gamma = 2$, which have been used for both noise estimators.

Noise estimate	SNR[dB]	ΔSNR[dB]	$\mathrm{SNR_{SEG}}$[dB]	WSSM[·]
Observation	5.0	0.0	-0.4	44.4
Ideal VAD	12.5	7.5	2.2	53.7
Martin2001	12.3	7.3	2.1	48.8

Table 2.3: Using Martins [59] method for noise estimation, the resulting spectral subtraction by the Berouti [6] method results in following numbers. As in previous section, the empirical parameters have been set to $\alpha = 5$, $\beta = 0.01$, $\gamma = 2$. The noise estimation procedure by Martin is observed to result in superior performance compared to using the ideal VAD and recursive smoothing.

The noise reduction performance is seen to be comparable for the two noise estimation procedures, albeit slightly lower for using the method of minimum statistics. Without audible noise reduction performance difference, the distortion measure, WSSM, indicates a slight difference of 5 points. The modified spectrogram, depicted in Figure 2.6, reveals however, that the difference is subtle compared to the modified spectrogram of the ideal VAD-based method, see Figure 2.1 in Section 2.1. Slightly less musical noise and less sporadic noise in the beginning is characteristic for the method of minimum statistics. This could explain the lowered WSSM, as this metric is sensitive to differences in spectral slopes. Informal listening tests among group members indicates that the current method using Martin's method is preferable.

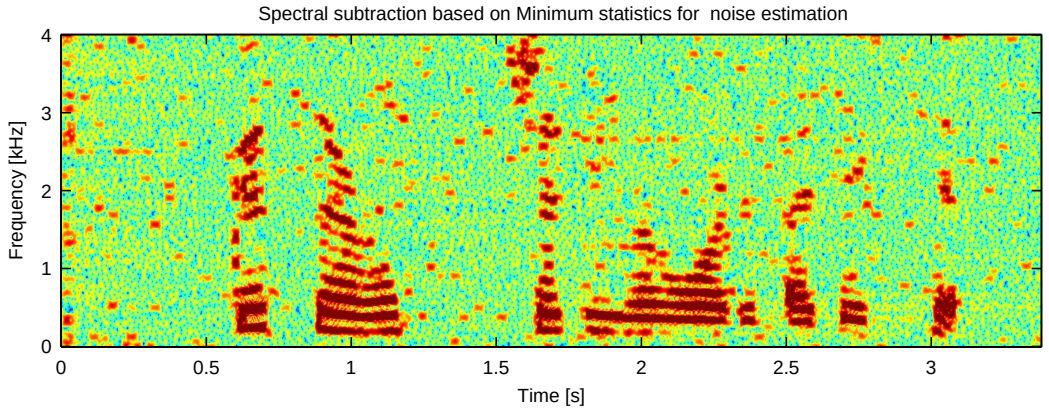


Figure 2.6: Modified spectrogram for applying the method of minimum statistics for noise estimation for spectral subtraction. The spectrogram is seen comparable to Figure 2.1 where an ideal VAD and recursive smoothing have been applied.

In order to compare the use of the noise estimation procedure presented in this section with the use of an ideal VAD and recursive smoothing, we have carried out the test underlying Figure 2.3 seen on page 29. The empirical parameters are set to $\beta = 0.01$, $\alpha = 5$, and $\gamma = 2$, respectively. In Figure 2.7 the results from a test on two speech signals with various SNRs, and white and pink noise.

By comparing Figure 2.7 and Figure 2.3 on page 29 subtle differences are seen in noise reduc-

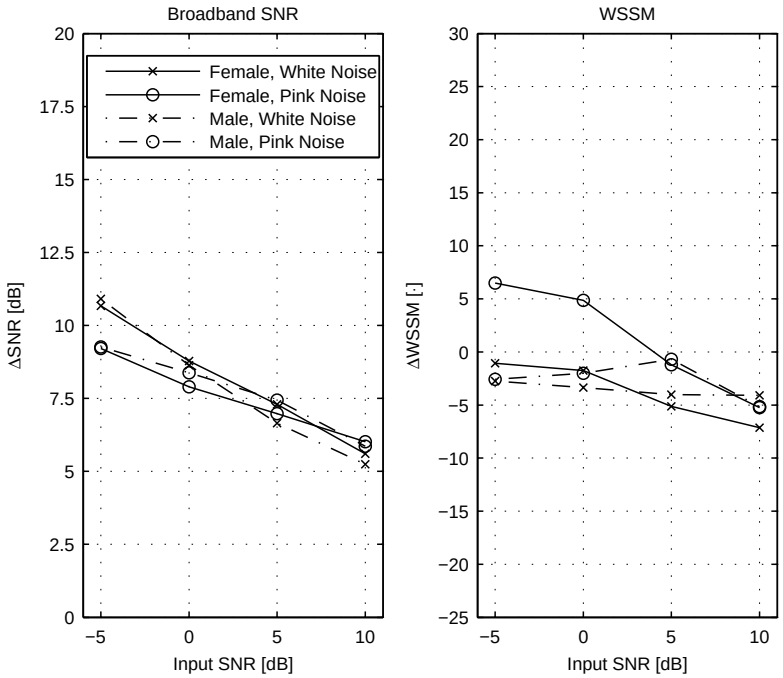


Figure 2.7: Noise reduction (SNR) and speech distortion (WSSM) for power spectral subtraction employing the minimum statistics approach for noise estimation. Empirical parameters $\beta = 0.01$, $\alpha = 5$, and $\gamma = 2$, is used. While noise reduction is comparable to the results using an ideal VAD, the speech distortion is several points better, cf. Figure 2.3.

tion performance, as earlier indicated by the results of Table 2.3. The speech distortion introduced using Martin's method for noise estimation results in lower speech distortion, cf. Figure 2.3. This is not surprising, as the noise estimation by Martin's method includes what is called optimal smoothing. This term is not justified by its formulation in (2.11), see [59], as it is not defined optimal in any sense. However, compared to the smoothing used in previous section, the smoothing is indeed *local in time and frequency*. This feature is presumed the source to the reduced speech distortion.

In the current section we have introduced a state-of-the-art method for noise estimation. Results applying the estimates of this procedure to spectral subtraction were presented, and indicated, that the minimum statistics estimation procedure is more local in time and frequency compared to the constant smoothing using a VAD. Using the minimum statistics approach for noise estimation was shown to reduce speech distortion, but not at the expense of noise reduction, which was comparable to the performance of the results in previous section employing a constant-smoothing parameter procedure using an ideal VAD. An SNR-step test revealed that the noise estimation procedure could track varying noise. However, increasing the responsiveness by reduced tracking latency would impede spanning high-energy speech utterances which are known to span up till $0.5 - 0.7\,\mathrm{s}$.

2.3 Log-Energy-Based Voice Activity Detector and Spectral Subtraction

In the preceding sections we have introduced spectral subtraction as a technique for noise reduction. Initially the noise was estimated using recursive smoothing of noise samples by identifying noise-dominated signal segments using an ideal VAD. In the previous section, a method to do advanced processing of the observation signal in order to obtain a better noise estimate was presented. This method is realistic in the sense that it operates on the observable signal, $y[k]$. The ideal VAD, on the other hand, is computed solely on the clean-speech signal, $s[k]$, which we do

not have available in a practical situation. In this section we seek to investigate a VAD algorithm due to Van Gerven and Xie [29], which is based on short-time smoothed log-energy distribution estimation. By modelling the local noise statistics, the instantaneous short-time log-energy can be distinguished from the smoothed noise log-energy distribution based on thresholding. The beauty of the algorithm lies in its simplicity.

By computing the instantaneous, short-time log-energy signal, e.g. by computing the STFT of the observable signal $y[k]$,

$$signal = \log_{10}\big(\sum_{n=0}^{N-1} Y(r,n)^2\big) \tag{2.12}$$

we calculate *speech onset* and *speech offset* based on comparison with the speech (speech onset), $T_S[r]$, and noise (speech offset), $T_N[r]$, thresholds. The thresholds are given by

$$\begin{aligned} T_S[r] &= \mu_N[r] + \alpha_{T_S}\sigma_N[r] \\ T_N[r] &= \mu_N[r] + \alpha_{T_N}\sigma_N[r] \end{aligned} \tag{2.13}$$

The thresholds depends on an estimate of the noise mean, $\mu_N[r]$, and the noise variance, $\sigma_N^2[r]$, at time instance k. Whenever the *signal* is below T_S, speech is detected, but the noise statistics, μ_N and σ_N is frozen. Should *signal* be below T_N, the adaptation, or smoothing, of the noise statistics are commenced. The estimation can conveniently be done using recursive smoothing. In Algorithm 2 the procedure is summarised.

Short-time, smoothed log-energy distribution-based VAD as proposed by Van Gerven and Xie [29], algorithm 3.

---

1. compute *signal* $= \log_{10}\big(\sum_{n=0}^{N-1} Y(r,n)^2\big)$
2. if (*signal* $\geq T_S[r-1]$)

   onset = true, else *offset* = true (speech detected)
3. if (*signal* $\leq T_N[r-1]$)

   offset = true, else *offset* = false (log-energy below noise threshold)
4. if ((!*onset* & *offset*) or ($r <$ *init*))

   $\mu_N[r] = \beta_{\mu_N}\mu_N[r-1] + (1-\beta_{\mu_N})$ *signal* (update noise statistics)

   $\sigma_N[r] = \beta_{\sigma_N}\sigma_N[r-1] + (1-\beta_{\sigma_N})\big|$ *signal* - $\mu_N[r]\big|$

   else

   $\mu_N[r] = \mu_N[r-1]$

   $\sigma_N[r] = \sigma_N[r-1]$
5. compute $T_N[r]$ and $T_S[r]$ according to (2.13)
6. *speech*[r] = true (VAD marking)

Algorithm 2: Voice activity detector based on modelling the local log-energy noise statistics and observing whenever the signal, *signal*, deviates from the estimated noise energy distribution.

For the theoretical implementation, one should compensate for the window energy added when forming $Y(r,n)$ by an STFT procedure. Assuming the window is $w[k]$, we can normalise by $\|w\|^2$, i.e. the 2-norm of the length-N window function. This is, however, not done in a practical implementation, as we are not estimating the spectral density as in previous section, rather we are making a noise-presence decision. Initially *init*-samples are bypassed to ensure adaption of the estimated distribution parameters.

For the adaptation we use a first-order recursive filter with coefficients β_{μ_N} and β_{σ_N}, for the mean and the standard deviation, respectively. To define the "upper"-threshold for speech *onset* and "lower"-threshold for speech *offset*, the factors α_{T_S} and α_{T_N} are used.

Hangover - Robustness improvement to Algorithm 2

7. if $\big($(*onset* = true) & (*speech*$[r-1]$ = false)$\big)$

   hangover = *hangover_cnst*

8. if $\big($(*onset* = false) & (*hangover* > 0)$\big)$

   hangover = *hangover* - 1

   speech$[r]$ = true

Algorithm 3: Hangover amendment in order to make the voice activity detector of Van Gerven and Xie [29] in Algorithm 2 more robust against sporadic changes. Whenever speech has been detected, the algorithm keeps detecting speech, though the statistics says otherwise.

In order to add robustness against sporadic jumps across the thresholds, we have added a hangover algorithm. This was not included in the original work by Van Gerven and Xie [29]. The motivation behind a hangover is to reduce sudden offsets (noise markings), which stems from the fact, that for speech enhancement (and other applications like speech recognition), marking speech as noise has a worse effect than marking noise as speech. Thus, whenever speech onset has been detected, we stay onsetting for, say, $100\,\mathrm{ms}$. Algorithm 3 is simply appended to Algorithm 2 in order to add the hangover functionality.

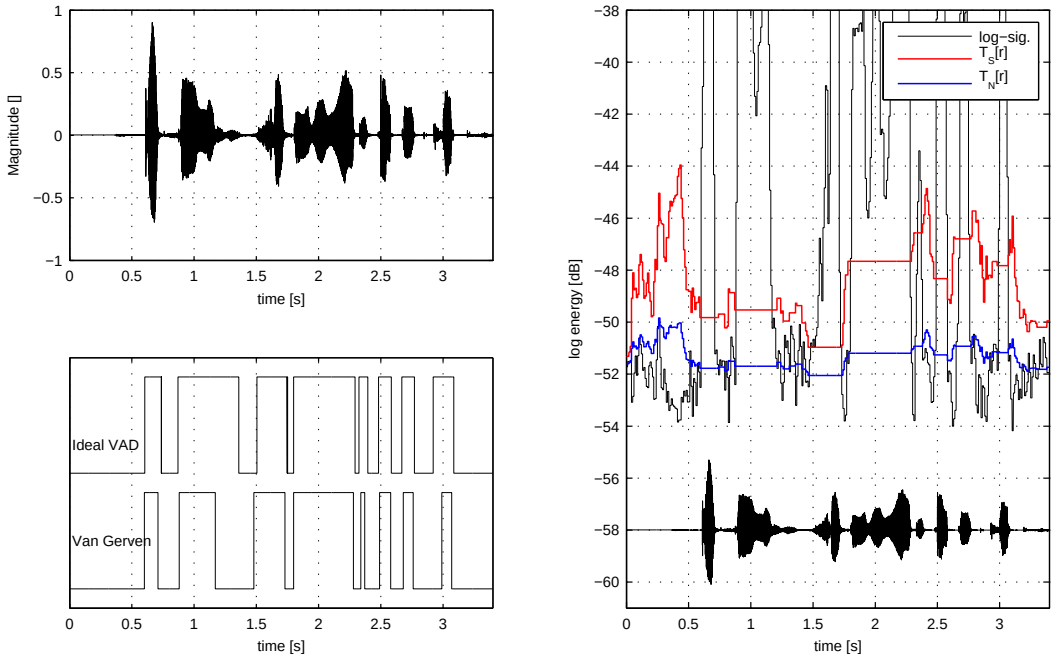


Figure 2.8: From top right, the clean-speech signal (the algorithm works on the observable signal, $y[k]$). Below, the resulting ideal VAD (processing $s[k]$) and the log-energy based algorithm. The decisions are seen to coincide, however, for low-energy utterances (e.g. around $1.25\,\mathrm{s}$ and $2.9\,\mathrm{s}$), the log-energy method is seen to fail. The observation was degraded by white noise at $\mathrm{SNR} = 5\,\mathrm{dB}$. In the right-hand figure, the log-energy distribution and the thresholds, $T_N[r]$ and $T_S[r]$, are seen. Whenever the log-energy distribution is below $T_S[r]$ the segment is marked as noise.

In order to demonstrate the functionality of Algorithm 2 and Algorithm 3, the algorithms have been implemented (`gerven1997algo3.m`) and applied to a signal. The clean-speech signal is degraded by white additive noise at $\mathrm{SNR} = 5\,\mathrm{dB}$ and the observation signal processed in segments of 160 Hann tappered samples at 50% overlap. Empirical parameters, for smoothing, $\beta_S = 0.76$ and $\beta_N = 0.96$, and for offsetting the thresholds, $\alpha_S = 5$ and $\alpha_N = 1.4$, have been used. The hangover constant is adjusted to $100\,\mathrm{ms}$ hangover. The thresholds and the log-energy distribution is depicted in Figure 2.8 (to the right). Whenever the log-energy distribution is above

$T_S[r]$, speech onset is marked, viz. signal segment is speech dominated. Whenever the log-energy distribution is below, a noise-dominated segment is detected. The log-energy distribution is updated using speech/noise decision based on the (slightly lower) threshold, $T_N[r]$. This ensures, that the distribution is updated in heavily noise-dominated segments only.

To the left in Figure 2.8, the clean-speech signal, $s[k]$, is depicted. Below the resulting VAD markings from the ideal VAD and log-energy based VAD are depicted. The ideal VAD corresponds very well with speech presence as the ideal VAD processes the clean-speech signal, not the observable signal, $y[k]$. The log-energy based VAD fails whenever the clean-speech signal is seen to consist of a low-energy utterance. This is expected, as the additive noise on $y[k]$ will drown the clean-speech signal and make the energy-based decision difficult for the present method.

We have successfully described an implementable, voice activity detection algorithm and demonstrated the functionality on a realistic observation signal. Next we apply the log-energy VAD to the recursive smoothing for spectral subtraction and compare the results with result using an ideal VAD and the minimum statistics estimation procedure described in previous section.

2.3.1 Log-Energy-Based VAD Applied to Spectral Subtraction

In order to compare the performance of the log-energy VAD to the ideal VAD, we have repeated the test of Section 2.2.1. Spectral subtraction with the empirical parameters, see Eq. (2.5b) in Section 2.1, set to $\alpha = 5$, $\beta = 0.01$, $\gamma = 2$, has been applied to a speech signal degraded by additive white noise at an SNR of $5\,\mathrm{dB}$. Although the spectral subtraction is done with Hann tappering of 256 samples, the present log-energy VAD algorithm is using 160 samples per segment. This is done, as it has been observed, that increasing the segment size substantially decreases the performance of the VAD algorithm.

Noise estimate	$\mathrm{SNR}[\,\mathrm{dB}]$	$\Delta\mathrm{SNR}[\,\mathrm{dB}]$	$\mathrm{SNR}_{\mathrm{SEG}}[\,\mathrm{dB}]$	$\mathrm{WSSM}[\cdot]$
Observation	5.0	–	-0.4	44.4
Ideal VAD	12.5	7.5	2.2	53.7
Martin2001	12.3	7.3	2.1	48.8
Log-energy VAD	12.5	7.5	2.3	51.0

Table 2.4: Using Martin's [59] method for noise estimation, the resulting spectral subtraction by the Berouti [6] method results in followin numbers. As in previous section, the empirical parameters have been set to $\alpha = 5$, $\beta = 0.01$, $\gamma = 2$ in both cases.

In Table 2.4 the results applying spectral subtraction to a speech signal degraded with additive white noise at an SNR of $5\,\mathrm{dB}$. The noise estimator is based on the method of minimum statistics, or recursive smoothing of noise-dominated samples using an ideal or a log-energy based VAD, respectively. The empirical parameters used in the setup of generalised spectral subtraction were $\alpha = 5$, $\beta = 0.01$, and $\gamma = 2$. Noise reduction performance is seen comparable to the use of an ideal VAD. Speech quality is surprisingly slightly better for the use of the non-ideal VAD (the log-energy method). Referring to the clean-speech signal in Figure 2.8 (top left corner), the segmental WSSM reveals few notable differences, except for the time segment $0.9-1.3\,\mathrm{s}$, where the ideal VAD based method shows slightly higher WSSM values. A spectrogram reveals increased musical noise (amount not magnitude) in this time segment, but inaudible when listening to both speech estimates.

We conclude that similar performance can be obtained for $\mathrm{SNR} = 5\,\mathrm{dB}$ when considering an ideal (infeasible) VAD to the present log-energy based VAD. This section is finished by evaluation spectral subtraction based on the log-energy based VAD for white and pink additive noise for various SNRs.

Figure 2.9 depicts the assessment metrics for noise reduction (SNR) and speech distortion (WSSM) for applying spectral subtraction to a number of different observations. The noise estimator is the log-energy VAD in combination with recursive smoothing. The same parameters as those described for Table 2.4 are used. Performance is seen comparable to using the ideal VAD,

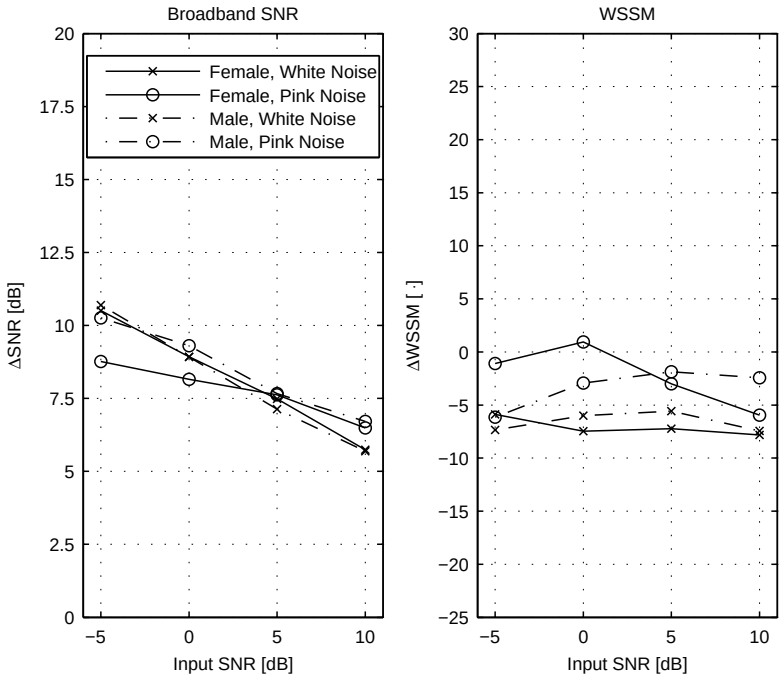


Figure 2.9: Noise reduction (SNR) and speech distortion (WSSM) for power spectral subtraction employing the log-energy dsitribution based VAD and recursive smoothing over the noise-dominated speech segments to obtain a noise estimate. Empirical parameters $\beta = 0.01$, $\alpha = 5$, and $\gamma = 2$, is used. Both noise reduction and speech distortion is comparable to the results obtained using an ideal VAD, albeit with slightly smaller difference (min-to-max) value of WSSM.

cf. Figure 2.3 on page 29. However, the speech distortion is slightly more limited, i.e. the span from max-to-min is less using this method. From the results we can conclude, that the presented log-energy based VAD is a good VAD for stationary environments.

We have presented a VAD algorithm and demonstrated the performance in a stationary environment. The log-energy distribution and the simple thresholding has been depicted to demonstrate the functionality. Finally, the performance using the VAD in a spectral subtraction functionality (under stationary conditions) was investigated. In the following section we will introduce another noise reduction functionality, the signal subspace approach.

2.4 Signal Subspace Techniques

The signal subspace method uses a transform that differs from the spectral subtraction technique, which was based on the Fourier transform. The signal subspace approach is based on a signal dependent transform and, for the white-noise case, it inherently provides a noise estimate.

Generally the signal subspace approach for single-microphone observations can be split in two; (1) EVD-based methods, and (2) SVD-based methods. Referring back to A3 model in Figure 1.13 on page 19, the signal subspace approach can be interpreted as the application, or functionality. Whereas the EVD and SVD-based methods can be considered algorithms, which implements the application. Common is the low-rank linear model of the speech signal, which translates the speech signals into a number of signal-dependent transform-domain coefficients. The low-rank property ensures, that at least some of these coefficients are zero. When observing a noisy speech signal embedded in additive noise, the number of transform-domain coefficients can be truncated and the noise residing in that subspace nullified. The remaining transform-domain coefficients represents noise and speech signal. Using a linear estimator of speech, the speech component in the remaining coefficient can be estimated.

In this section the low-rank model is presented, which constitutes the basis for this method.

It is shown that the noise reduction performance depends on correct estimating the rank of the speech signal subspace, therefore an adaptive rank-estimation method is investigated. When presenting the signal subspace approach it reveals that different classes of estimators are applicable. Four different estimators are then described and tested and lastly, a pre-whitening (or joint diagonalisation) step is presented in order to cope with non-white noise .

2.4.1 Linear Signal Model for Low-Rank Modelling of Speech Signals

Let a real-valued finite-length zero-mean signal drawn from a random process at time k be given by the vector

$$\mathbf{s}[k] = \begin{bmatrix} s[k] & s[k-1] & \dots & s[k-M+2] & s[k-M+1] \end{bmatrix}^T \tag{2.14}$$

and its auto-correlation matrix, $\mathbf{R}_{ss}$, be given, then we want to represent $\mathbf{s}[k]$ by the linear model (transformation)

$$\mathbf{c}[k] = \mathbf{A}^T \mathbf{s}[k] \tag{2.15}$$

where $\mathbf{A}$ is an orthonormal matrix, $\mathbf{A}^{-1} = \mathbf{A}^T$, then

$$\mathbf{s}[k] = \sum_{i=1}^{M} c_i[k]\mathbf{a}_i \quad \text{where} \quad \mathbf{a}_i^T \mathbf{a}_j = 0 \ , \ i \neq j \tag{2.16}$$

is called the *discrete Karhunen-Loève Expansion*[1] [46] of the signal vector, $\mathbf{s}[k]$. Of course, one can always represent the length-M signal vector by M expansion coefficients. Representing the speech signal of $\mathbf{s}[k]$ with fewer expansion coefficients, say K, is a linear model that successfully has been applied to speech signals [24, 39]. This is formally known as *low-rank modelling* of speech signals. Sinusoidal modelling is one of such techniques (which defines the form of the matrix $\mathbf{A}$), but it has been noted in Ephraim and Van Trees [24], that the exact underlying model is of no importance for signal enhancement. The relation $K < M$, however, is.

By reducing the rank of the linear model of (2.16) to K, that is $\mathrm{rank}(\mathbf{A})$, we introduce a model error. It can be shown that minimising the MSE of the model error

$$E\{\mathbf{e}_K^T \mathbf{e}_K\} = \sum_{i=K+1}^{M} E\left\{ |c_i[k]|^2 \right\} \mathbf{a}_i^T \mathbf{a}_i \tag{2.17}$$

$$\text{where} \quad \mathbf{e}_K = \mathbf{s} - \hat{\mathbf{s}} = \sum_{i=1}^{M} c_i[k]\mathbf{a}_i - \sum_{i=1}^{K} c_i[k]\mathbf{a}_i \tag{2.18}$$

corresponds to analyse the signal vector, $\mathbf{s}[k]$, with the eigenvectors of the auto-correlation matrix, $\mathbf{R}_{ss} \in \mathbb{R}^{M \times M}$ [46]. These eigenvectors are revealed by the diagonalisation of $\mathbf{R}_{ss}$ accomplished by the eigendecomposition (see Appendix A)

$$\begin{gathered} \mathbf{A}_s^T \mathbf{R}_{ss} \mathbf{A}_s = \mathbf{\Lambda}_s = \mathrm{diag}(\lambda_{s,1}, \lambda_{s,2}, \dots, \lambda_{s,M}) \\ \text{where} \quad \lambda_{s,1} \geq \lambda_{s,2} \geq \dots \geq \lambda_{s,M} \geq 0 \end{gathered} \tag{2.19}$$

thus, the eigenvalues, $\lambda_{s,i}$, are arranged in a decreasing order. For a low-rank model of, e.g. speech signals, $\mathbf{R}_{ss}$ will be of rank K and have $M - K$ zero eigenvalues. The spectral theorem

$$\mathbf{R}_{ss} = \mathbf{A}_s \mathbf{\Lambda}_s \mathbf{A}_s^T = \sum_{i=1}^{M} \lambda_{s,i} \mathbf{a}_i \mathbf{a}_i^T \tag{2.20}$$

[1] For the Karhunen-Loève Expansion, the upper limit of equation (2.16) would have been ∞. We, however, deal with a length-M signal, which can always be represented by M weighted orthonomal vectors.

states that the autocorrelation matrix can be represented by a set of distinct eigenvalues and a set of rank-one projection matrices - outer products of the orthonormal column eigenvectors, $\mathbf{A}_s = [\mathbf{a}_1\ \mathbf{a}_2\ \ldots\ \mathbf{a}_M]$, from the eigenvalue decomposition.

The relation sought in the expansion of (2.15) can be obtained by replacing $\mathbf{A}$ by the eigenvectors of $\mathbf{R}_{ss}$, $\mathbf{A}_s$, i.e.

$$\mathbf{c}[k] = \mathbf{A}_s^T \mathbf{s}[k] \tag{2.21}$$

This provides an orthonormal transformation of the signal vector, $\mathbf{s}[k]$, into a set of uncorrelated zero-mean random variables, $c_i[k]$. If we consider the energy of the signal vector and use (2.16), we obtain

$$\begin{aligned} \left\|\mathbf{s}[k]\right\|_2^2 &= \mathbf{s}[k]^T \mathbf{s}[k] \\ &= \left(\sum_{i=1}^{M} c_i[k]\mathbf{a}_{s_i}^T\right)\left(\sum_{j=1}^{M} c_j[k]\mathbf{a}_{s_j}\right) \\ &= \sum_{i=1}^{M} \left\|c_i[k]\right\|_2^2 \end{aligned}$$

which states that the coefficients together has an energy equal to that of the signal vector [39]. These coefficients are random variables whose mean value equals the eigenvalues of the autocorrelation matrix, as stated

$$E\left\{\, |c_i[k]\,|^2 \,\right\} = \lambda_{s,i} \quad \text{for } i = 1, 2, \ldots, M \tag{2.22}$$

This justifies the energy-decomposition property that the eigendecomposition is often attributed. An intuitive interpretation of the eigenvalues and their eigenvectors as a gain function and a frequency-domain filter is given in Section 2.4.3.

We now turn our attention to speech signal embedded in noise. In specific, how to use the linear model and the decomposition of the second order statistics, to split noise and speech energy.

2.4.2 Reconstruction of Speech Signal from Noisy Speech Observation

With the linear model of a speech signal presented in previous section in mind, let the length-M observation vector be given by (using the observation model of speech signal embedded in additive noise, see Section 1.2.4 on page 5)

$$\mathbf{y}[k] = \mathbf{s}[k] + \mathbf{b}[k] = \mathbf{A}_\mathbf{s}\mathbf{c}[k] + \mathbf{b}[k] \tag{2.23}$$

where the additive noise, $\mathbf{b}[k]$, is assumed zero-mean and uncorrelated with the speech signal, i.e. $E\{\mathbf{s}[k]\mathbf{b}[k]^T\} = 0$. Furthermore, the noise auto-correlation matrix, $\mathbf{R}_{bb} \in \mathbb{R}^{M\times M}$, is assumed known and equal to

$$\mathbf{R}_{bb} = E\{\mathbf{b}\mathbf{b}^T\} = \sigma^2 \mathbf{I}_M \tag{2.24}$$

meaning that we assume white noise with variance σ^2. The matrix, $\mathbf{I}_M$, is a rank-M identity matrix with ones on the diagonal. Signal subspace approaches are not restricted to white noise, but this assumption is useful in treating the underlying theory. Non-white noise cannot directly be treated by the signal subspace approach and needs either a pre-whitening prior to signal subspace processing, or an integrated pre-whitening approach, such as joint-diagonalisation of the empirical noisy speech and noise data matrices, $\mathbf{R}_{yy}$ and $\mathbf{R}_{bb}$, respectively. Put shortly, in the case of coloured noise, $\mathbf{R}_{ss}$ and $\mathbf{R}_{bb}$ cannot be diagonalised by the same eigenvectors. The coloured-noise case will be considered in Section 2.4.7. For now we continue along the line of white noise.

A property of the auto-correlation matrix of white noise is $\mathrm{rank}(\mathbf{R}_{bb}) = M$, i.e. a full rank matrix, see (2.24), and all eigenvalues non-zero. Let the eigendecomposition of the observation

vector be given by

$$\mathbf{Q}_y^T \mathbf{R}_{yy} \mathbf{Q}_y = \mathbf{\Lambda}_y = \text{diag}(\lambda_{y,1}, \lambda_{y,2}, \ldots, \lambda_{y,M}) \quad \text{where} \quad \text{rank}(\mathbf{R}_{yy} \in \mathbb{R}^{M \times M}) = M \tag{2.25}$$

or

$$\begin{aligned} \mathbf{R}_{yy} &\triangleq E\left\{\mathbf{y}\mathbf{y}^T\right\} && (2.26)\\ &= E\left\{\mathbf{Q}\mathbf{c}\mathbf{c}^T\mathbf{Q}^T + \mathbf{Q}\mathbf{s}\mathbf{b}^T + \mathbf{b}\mathbf{s}^T\mathbf{Q}^T + \mathbf{b}\mathbf{b}^T\right\} \\ &= \mathbf{Q}\mathbf{R}_{cc}\mathbf{Q}^T + \mathbf{R}_{bb} = \mathbf{R}_{ss} + \mathbf{R}_{bb} && (2.27) \end{aligned}$$

where (2.27) has been achieved by exploiting that the signal and noise are uncorrelated vectors $E\{\mathbf{s}[k]\mathbf{b}[k]^T\} = 0$. It follows from the white-noise assumption, that the eigenvectors diagonalising $\mathbf{R}_{yy}$ are also the eigenvectors of $\mathbf{R}_{ss}$ and $\mathbf{R}_{bb}$ [24]. This fact can easily be verified by inspecting the white-noise correlation matrix, $\mathbf{R}_{bb} = \sigma^2\mathbf{I}_M$. By the eigendecompositions of (2.19) and (2.24) we have

$$\lambda_{y,i} = \begin{cases} \lambda_{s,i} + \sigma^2 & 0 < i \leq K, \\ \sigma^2 & K+1 \leq i \leq M \end{cases} \tag{2.28}$$

As stated, the (white) noise fill in the entire Euclidean space $\mathbb{R}^M$, while speech signals can be modelled by a low-rank model, say $\mathbb{R}^K$. The subspace spanned by the first K eigenvectors of $\mathbf{R}_{yy}$ is denoted the *signal-plus-noise subspace*, or sometimes, the *signal subspace*, because it is dominated by the speech signal. The complementary subspace of dimension $M - K$ is denoted the *noise subspace*. This splitting of the subspaces of $\mathbf{R}_{yy}$ follows clearly from (2.28), and can be more explicitly written as

$$\mathbf{R}_{yy} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_1 & \mathbf{0}_{K\times(M-K)} \\ \mathbf{0}_{(M-K)\times K} & \mathbf{\Lambda}_2 \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1^T \\ \mathbf{Q}_2^T \end{bmatrix} \tag{2.29}$$

where $\mathbf{\Lambda}_1 \in \mathbb{R}^{K\times K}$ is given by the upper part of (2.28) and $\mathbf{\Lambda}_2 \in \mathbb{R}^{(M-K)\times(M-K)}$ is the lower part of (2.28). From (2.29) we expect a distinct gap, $\lambda_{y,K} > \lambda_{y,K+1} = \sigma^2$, to exist in the eigenvalues of $\mathbf{R}_{yy}$. The eigenvectors in the columns of $\mathbf{Q}_1$ equals the eigenvectors of $\mathbf{A}_s$ in (2.20). One should note the importance of this property, which means the eigendecomposition of $\mathbf{R}_{yy}$ in (2.29) is sufficient to recover $\mathbf{R}_{ss}$ consistently. That is, assuming we can successfully estimate the set of speech signal eigenvalues of (2.28).

The splitting of the eigenvalues allow us to give a possible definition of the signal-to-noise ratio related to the total M-dimensional space [66]

$$\text{SNR}_M = \left(\frac{\lambda_{s,1} + \lambda_{s,2} + \ldots + \lambda_{s,K}}{M\sigma^2}\right) \tag{2.30}$$

or, to define a signal-to-noise ratio related to the K-dimensional signal subspace

$$\text{SNR}_K = \left(\frac{\lambda_{s,1} + \lambda_{s,2} + \ldots + \lambda_{s,K}}{K\sigma^2}\right) \tag{2.31}$$

which leads to a potential SNR improvement factor of M/K. This theoretical results indicates, that we should choose M as large as possible, since the low-rank model of the speech signal should remain the same dimension. However, the size of M is limited by the assumptions on stationarity of speech.

To illustrate the usefulness of these relations and the subspace splitting, we consider the least-square approximation of $\mathbf{s}[k]$ from the observation, $\mathbf{y}[k]$. Given this observation, the speech wide-sense stationarity (WSS) and white-noise assumptions, and the knowledge of the eigendecomposition of $\mathbf{R}_{yy}$, we can obtain a least-squares estimate of the speech signal as [80]

$$\hat{\mathbf{s}}[k] = \mathbf{Q}_1(\mathbf{Q}_1^T\mathbf{Q}_1)^{-1}\mathbf{Q}_1^T\mathbf{y}[k] = \mathbf{Q}_1\mathbf{I}_K\mathbf{Q}_1^T\mathbf{y}[k] \tag{2.32}$$

The least-squares estimator corresponds to nulling the noise subspace, thus we remove the noise-only subspace. We do not modify the speech signal assuming the speech model order K is correctly estimated/known.

As can be seen from (2.28) we can do better. The signal-plus-noise subspace does also contain noise, which we could try to remove by estimating the signal-only eigenvalues. This corresponds somewhat to the approach taken in spectral subtraction where noise energy is subtracted from the observation signal in each frequency band.

In accordance with (2.29) the speech signal estimation corresponds to seeking a linear filter, $\mathbf{W} = \mathbf{Q}_1\mathbf{Q}_1^T$, to estimate the desired vector $\mathbf{s}[k]$, from the observation vector, $\mathbf{y}[k]$.

$$\hat{\mathbf{s}}[k] = \mathbf{W}\mathbf{y}[k] = \mathbf{W}\mathbf{s}[k] + \mathbf{W}\mathbf{b}[k] \tag{2.33}$$

A more efficient estimator, however, would not only project onto the column space of $\mathbf{R}_{yy}$, as the least squares estimator does, but also incorporate an estimator of the speech signal energy in the signal-plus-noise subspace, by employing a gain function, $f(\lambda_{y,i})$,

$$\begin{aligned} \mathbf{W} &= \mathbf{Q}_1\mathbf{F}\mathbf{Q}_1^T \\ \text{where} \quad \mathbf{F} &= \operatorname{diag}\big(f(\lambda_{y,i})\big) \quad 1 \le i \le K \end{aligned} \tag{2.34}$$

As will be shown later, several interesting signal subspace estimators can be constructed. Common is the need to estimate the eigenvalues of the speech signal auto-correlation matrix, $\mathbf{R}_{ss}$, which is done, according to (2.28), as

$$\hat{\lambda}_{s,i} = \max(\lambda_{y,i} - \hat{\sigma}^2, 0) \tag{2.35}$$

where the white noise variance estimate, $\hat{\sigma}^2$, is obtained from either averaging the eigenvalues in noisy-only periods (requires the use of a VAD), or from averaging the lower eigenvalues, as

$$\hat{\sigma}^2 = \frac{1}{M-(K+1)} \sum_{i=K+1}^{M} \lambda_{y,i} \tag{2.36}$$

The $\max()$ operator in (2.35) is used to assure positive-only eigenvalues, since estimates of $\lambda_{s,i}$ might otherwise be negative due to the use of empirical data [24]. This is a common problem in signal subspace methods.

To round the introduction to the theory of signal subspace approaches, we restate one of the original algorithms for noise reduction in speech applications by Ephraim and Van Trees [24] in Algorithm 4. The problem of estimating the auto-correlation matrix is covered in Appendix B, Section B.1. A number of linear signal subspace estimators are enumerated in Section 2.4.4.

Signal subspace noise reduction based on the eigendecomposition (Ephraim and Van Trees, `Ephraim95.m`) in Appendix D

---

1. Compute the eigendecomposition of $\hat{\mathbf{R}}_{yy} \in \mathbb{R}^{M\times M}$.
2. Compute the linear estimator, $\mathbf{F} = \operatorname{diag}\big(f(\lambda_{y,i})\big),\ 0 < i \le K$.
3. Remove the noise subspace by discarding the $M-K$ lowest eigenvalues (and eigenvectors) and construct the optimal filter, $\mathbf{W} = \mathbf{Q}\mathbf{F}\mathbf{Q}^T$
4. Reconstruct the estimated length-M signal vector $\hat{\mathbf{s}}[k] = \mathbf{W}\mathbf{y}[k]$

Algorithm 4: The signal subspace approach proposed in [24] based on the eigendecomposition of the auto-correlation matrix of $\hat{\mathbf{R}}_{yy}$.

Having established the low-rank linear model for speech, (2.19), and the theoretical approach for noise reduction and reconstruction, (2.33) and (2.34), we introduce a useful interpretation of the eigenvectors and eigenvalues. To that end, an eigen-domain to frequency-domain transformation is a useful aid. This is the topic of following section.

2.4.3 Frequency-to-Eigen-domain Transformation

In this section a few interpretative aspects of the signal subspace approach will be covered. In specific, a time-domain filtering interpretation and a frequency-to-eigendomain transformation.

Traditionally speech enhancement has been carried out in the time- or frequency domain. The subspace methods operate in the eigendomain. The experience and understanding of speech characteristics can be exploited only if we have some transformation or relation to the frequency domain. In order to incorporate residual noise shaping as to exploit masking effects, which are well-understood in the frequency domain, Jabloun et al. [47] have developed the frequency-to-eigendomain transformation. Given a noisy speech segment, $\mathbf{y}[k]$ and its power spectral density (PSD), $S_y(\omega)$, they defined the frequency-to-eigendomain transformation as

$$\lambda_{s,i} = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_y(\omega) |V_i(\omega)|^2 d\omega \quad 1 \leq i \leq M \tag{2.37}$$

$$\text{where} \quad V_i(\omega) = \sum_{p=0}^{N-1} q_i(p) e^{-j\omega N} \tag{2.38}$$

where $V_i(\omega)$ is the discrete-time Fourier transform of the vector in the i*th* column of $\mathbf{Q}$ from (2.29). This means, filtering the PSD with the i*th* squared magnitude filter, $|V_i(\omega)|^2$, and integrating over the entire spectrum equals the i*th* eigenvalue of $\mathbf{R}_{yy}$. Or put different, each eigenvalue represents the average energy of the stochastic process in the frequency band of the filter, $|V_i(\omega)|^2$. These column vector filters are in fact equal to eigenfilters, which are stochastic counterparts to the deterministic matched filters maximising the signal-to-noise ratio [39, chap. E.4]. The i*th* column vector of $\mathbf{Q}$, corresponding to the i*th* eigenvalue of $\mathbf{R}_{yy}$, is called an eigenfilter of $\mathbf{R}_{yy}$. The optimum, maximum SNR, eigenfilter is the one corresponding to the largest eigenvalue.

Now, it is clear, that the filterbank splits the signal into M subspaces with an energy equal to λ_i and a total energy equal to the sum of the eigenvalues, see (2.37). The filters are the zero-phase (squared magnitude) eigenfilters, which are computed in a signal adaptive fashion based on the second order statistics of the noisy observation signal. This interpretation can be used to understand the construction of the optimal filter in (2.34), based on an M-band signal-adaptive filterbank. In Figure 2.10, the power spectrum of the noisy signal is shown (bold line) together with the 20 eigenfilters (thin line) used in that example. An SNR of $10\,\mathrm{dB}$ was used. The 20 filters are ordered left to right, with the largest eigenvalue first. It can be seen, that the first formants are captured by the first two subbands. Third formant by filter 3-5, and so forth. It is readily seen, that truncation of the lower 9-8 subbands would not effect the speech spectrum significantly. The lower eigenvalue, the more the filters are located around non-formant areas, or broadband spectrum.

2.4.4 Linear Estimators of the Signal Subspace and the Speech Eigenvalues

By now we have introduced the low-rank model of speech signals, in Section 2.4.1, and a least-squares optimal filter for truncating the noise subspace, (2.34) in Section 2.4.2, which we assume holds no speech signal information. The truncation, however, requires the knowledge of the size of the signal subspace, K. Although the theory presented indicates a clear gap in the eigenvalues of the auto-correlation matrix, see (2.28) in Section 2.4.2, this gap is not trivial to measure, nor to approximate. A second issue is the estimation of the speech signal eigenvalues from the combined speech and noise eigenvalues residing in the signal subspace. This section is concerned with this estimation problem. In the following, we will assume the signal subspace dimension, K, known. The problem of estimating the subspace dimension is deferred to Section 2.4.5.

In the literature several estimators have been derived. Some straightforward implementable, others require fine-tuning of empirical parameters for controlling the trade-off between noise

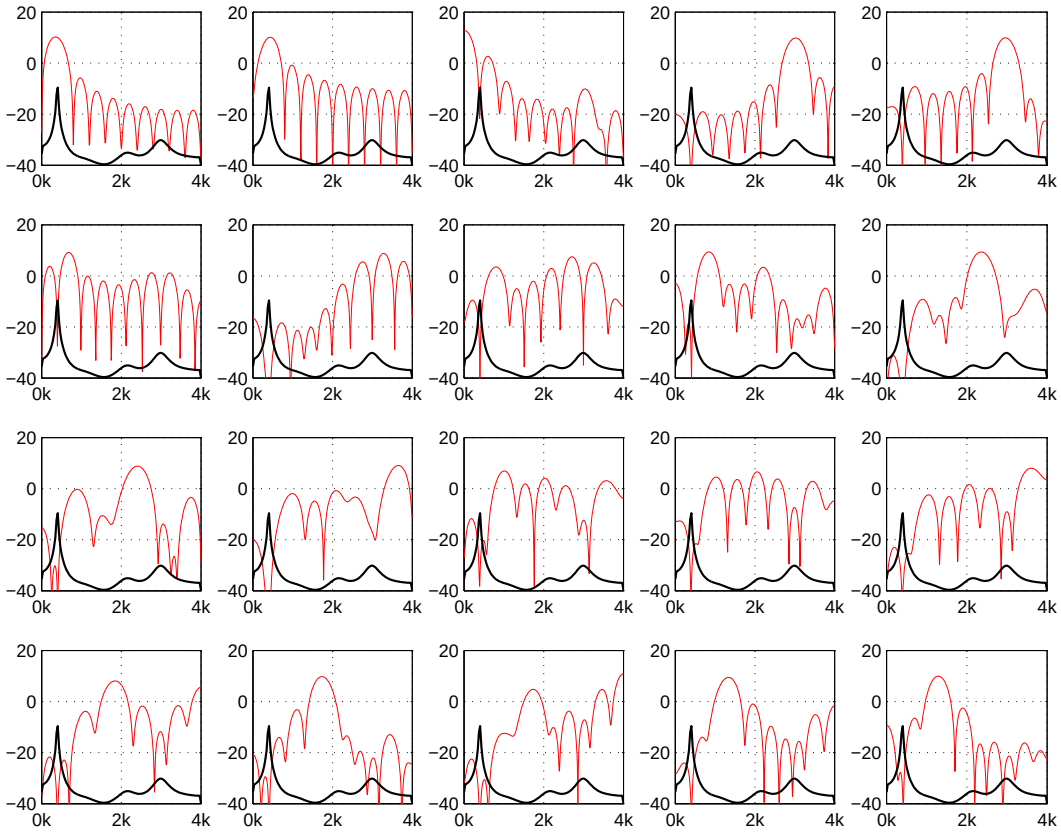


Figure 2.10: The bold line represents a PSD of an LPC model of the noisy speech signal, $S(\omega)$, obtained with Burgs PSD estimation method, see Appendix B. The other line(s) represent the magnitude-squared frequency response from the i*th* eigenfilter, $\mathbf{q}_i$ (see (2.25)), (left to right, top to bottom). The x-axis is frequency in Hz, while the y-axis is magnitude in dB.

reduction and speech distortion. To that end, it is useful to consider the *estimation residual*

$$\begin{aligned}\mathbf{r} &= \hat{\mathbf{s}} - \mathbf{s} = \mathbf{W}\mathbf{y} - \mathbf{s} \\ &= (\mathbf{W} - \mathbf{I})\mathbf{s} - \mathbf{W}\mathbf{b}\end{aligned} \tag{2.39}$$

where we can identify two contributions, the noise residual and the speech signal residual. Define the *noise residual* as

$$\mathbf{r}_b \triangleq \mathbf{W}\mathbf{b} \tag{2.40}$$

and the *speech residual* signal as

$$\mathbf{r}_s \triangleq (\mathbf{W} - \mathbf{I})\mathbf{s} \tag{2.41}$$

This can be seen as the part of the speech signal, which is projected onto the complement subspace to the signal subspace, that is, the noise subspace.

Four different estimators will be covered. The first is the widely used linear minimum mean square error (also known as minimum variance (MV)) estimator (LMMSE). The time-domain constrained (TDC) estimator is an extension of the MV estimator, and introduces additionally a scaling factor of the noise. The least squares (LS) method is the basic estimator using the gain function, $f_{\text{LS}} = \mathbf{I}_M$, and is not described further due to its simplicity. Lastly, the spectral-domain constrained (SDC) estimator is presented. This estimator tries to keep every spectral noise residual within a certain threshold.

Linear Minimum Mean Square Error (Minimum Variance)

The linear minimum mean square error is the optimisation criteria used in the derivation of the Wiener filter. It is also known as the minimum variance estimator. If the optimisation problem is defined as the minimum of the mean square residual error, $\mathbf{r}$, defined in (2.39), as

$$\begin{aligned} J_{MSE}(W) &= \min_{\mathbf{W}\in\mathbb{R}^{K\times K}} E\left\{\|\mathbf{r}\|^2\right\} \qquad (2.42a)\\ &= \min_{\mathbf{W}} E\left\{(\mathbf{s}-\mathbf{W}\mathbf{y})(\mathbf{s}-\mathbf{W}\mathbf{y})^T\right\} \\ &= \min_{\mathbf{W}} \left(E\{\mathbf{s}\mathbf{s}^T\} - E\{\mathbf{s}\mathbf{y}^T\mathbf{W}^T\} - E\{\mathbf{W}\mathbf{y}\mathbf{s}^T\} + E\{\mathbf{W}\mathbf{y}\mathbf{y}^T\mathbf{W}^T\}\right) \end{aligned}$$

the last equation can by differentiation w.r.t. $\mathbf{W}^T$ be set to zero, and we can find

$$\begin{aligned} \frac{\partial J_{MSE}}{\partial \mathbf{W}^T} &= -2E\{\mathbf{s}\mathbf{y}^T\} + 2\mathbf{W}E\{\mathbf{y}\mathbf{y}^T\} = 0 \qquad (2.42b)\\ &= -2\mathbf{R}_{sy} + 2\mathbf{W}\mathbf{R}_{yy} \end{aligned}$$

which, by assuming the inverse of $\mathbf{R}_{yy}$ exists, gives the optimal filter

$$\begin{aligned} \mathbf{W} &= \mathbf{R}_{sy}(\mathbf{R}_{yy})^{-1} \qquad (2.42c)\\ &= \mathbf{Q}\mathbf{\Lambda}_{\mathbf{s}}\mathbf{Q}^T(\mathbf{Q}\mathbf{\Lambda}_y\mathbf{Q}^T)^{-1} \\ &= \mathbf{Q}\mathbf{\Lambda}_s(\mathbf{\Lambda}_s + \mathbf{I}\sigma^2)^{-1}\mathbf{Q}^T \qquad (2.42d)\\ &= \mathbf{Q}\mathbf{F}_{MV}\mathbf{Q}^T \qquad (2.42e) \end{aligned}$$

where we have assumed that $\mathbf{R}_{sy} = \mathbf{R}_{ss}$ due to the orthogonality assumption between the noise and speech signals. The matrix gain is a diagonal matrix $\mathbf{F}_{MV} = \mathrm{diag}\big(f_{MV}(i)\big)$ with elements $f_{MV}(i)$ across the diagonals, defined as

$$f_{MV}(i) = \frac{\lambda_{s,i}}{\lambda_{s,i}+\sigma^2} = \frac{\lambda_{y,i}-\sigma^2}{\lambda_{y,i}} \qquad (2.43)$$

and zeros elsewhere. All the estimators in this section can be described by a gain function as (2.43), where the gain function should be placed along the diagonal to form an estimator matrix, as in (2.42e). The minimum-variance estimator, and other estimators of this section, can be found in Table 2.5 on page 47.

Time-Domain Constrained Estimator

If, instead of minimising the (total) residual $\mathbf{r} = \mathbf{r}_s + \mathbf{r}_b$, we minimise the residual speech, $\mathbf{r}_s$, while keeping the residual noise, $\mathbf{r}_b$, below some predefined threshold, the result is the time-domain constrained estimator (TDC), proposed in [24]

$$\min_{\mathbf{W}\in\mathbb{R}^{K\times K}} E\left\{\|\mathbf{r}_s\|^2\right\} \quad \text{subject to} \quad \frac{1}{M}E\left\{\|\mathbf{r}_b\|_2^2\right\} \le \alpha\sigma^2 \qquad (2.44)$$
$$\text{where} \quad 0 \le \alpha \le 1$$

This formulation reduces speech distortion, while reducing the residual noise within the threshold. By solving this constrained least-squares problem, we obtain following estimator [24]

$$\begin{aligned} \mathbf{W} &= \mathbf{R}_{ss}\left(\mathbf{R}_{ss} + \mu\sigma^2\mathbf{I}_K\right)^{-1} \qquad (2.45)\\ &= \mathbf{Q}_1\mathbf{\Lambda}_s\left(\mathbf{\Lambda}_s + \mu\sigma^2\mathbf{I}_K\right)^{-1}\mathbf{Q}_1^T \qquad (2.46)\\ &= \mathbf{Q}_\mathbf{1}\mathbf{F}_{TDC}\mathbf{Q}_1^T \qquad (2.47) \end{aligned}$$

where $\mathbf{F}_{TDC}$ is a diagonal matrix with diagonal elements

$$f_{TDC}(i,\mu) = \frac{\lambda_{s,i}}{\lambda_{s,i}+\mu\sigma^2} = \frac{\lambda_{y,i}-\sigma^2}{\lambda_{y,i}-(1-\mu)\sigma^2} \tag{2.48}$$

This is a Wiener filter with adjustable noise level, $\mu\sigma^2$. The α noise-level parameter in Eq. (2.44) can be shown to equal [47]

$$\alpha = \frac{1}{M}\sum_{i=1}^{K}\left(\frac{\lambda_{s,i}}{\lambda_{s,i}+\mu\sigma^2}\right)^2 \tag{2.49}$$

It is observed, that maximum α equals $\frac{K}{M}$ and occurs when $\mu = 0$. Special cases of the time-domain constrained estimator includes the least-squares estimator, $\mu = 0$, and the LMMSE estimator, $\mu = 1$.

Estimator	Gain matrix	Diag. elem.	SNR function
Least Squares (LS)	$\mathbf{I}_K$	1	1
Min. Variance (LMMSE/MV)	$\mathbf{F}_{MV}$	$\frac{\lambda_{y,i}-\sigma^2}{\lambda_{y,i}}$	$\frac{\zeta}{\zeta+1}$
Time-domain constr. (TDC)	$\mathbf{F}_{TDC}$	$\frac{\lambda_{y,i}-\sigma^2}{\lambda_{y,i}-(1-\mu)\sigma^2}$	$\frac{\zeta}{\zeta+\mu}$
Spectral-domain constr. (SDC)	$\mathbf{F}_{SDC}$	$\sqrt{\alpha_i}$	$\frac{-\nu}{\zeta}$

Table 2.5: Summary of the elements of the gain matrix for different linear signal estimators. The leftmost columns shows the matrix name, the middle column the diagonal entries of the matrices. The rightmost column is the middle column rewritten with the definition of SNR $= \lambda_{s,i}/\sigma^2 = \zeta$. Using the SNR function, we have plotted the gain function value in Figure 2.11.

Spectral-Domain Constrained Estimator

Another estimation formulation is to minimise speech distortion subject to keep every spectral component of residual noise within some predefined threshold. Considering the filter interpretation of Section 2.4.3, constraining the noise residual in the spectral domain seems a sound approach. The estimator is known as the spectral-domain constrained (SDC) estimator and formulated in [24] as

$$\min_{W\in\mathbb{R}^{K\times K}} E\left\{\|\mathbf{r}_s\|^2\right\} \quad \text{subject to} \quad \begin{cases} E\left\{|\mathbf{q}_i^T\mathbf{r}_b|^2\right\} \le \alpha_i\sigma^2 & 1\le i\le K \\ E\left\{|\mathbf{q}_i^T\mathbf{r}_b|^2\right\} = 0 & K+1\le i\le M \end{cases} \tag{2.50}$$

Solving the constrained optimisation problem leads to the following estimator

$$\mathbf{W} = \mathbf{Q_1}\mathbf{F}_{SDC}\mathbf{Q}_1^T \tag{2.51}$$

where the diagonal matrix, $\mathbf{F}_{SDC}$, has diagonals given by

$$f_{SDC}(i,\alpha_i) = \begin{cases} \alpha_i^{1/2} & 1\le i\le K \\ 0 & K+1\le i\le M \end{cases} \tag{2.52}$$

Left is the choice for α_i. It is noted by Jabloun et al. [47] that it is advantageously to define the gain function of Eq. (2.52) based on an SNR estimate, and in this fashion completely turn off spectral components with a low SNR, while keeping spectral components with high SNR. By defining the SNR as $\zeta = \lambda_{s,i}/\sigma^2$, a seemingly popular choice for α_i is [24, 37]

$$\alpha_i = \exp\left(\frac{-\nu}{\zeta}\right) = \exp\left(\frac{-\nu\sigma^2}{\lambda_{y,i}-\sigma^2}\right) \tag{2.53}$$

where ν is an experimentally chosen constant. This function satisfies $f_{SDC}(0)\to 0$ and $f_{SDC}(\infty)\to 1$. This is a desirable feature for all estimators of this section. In Figure 2.11, we have plotted the

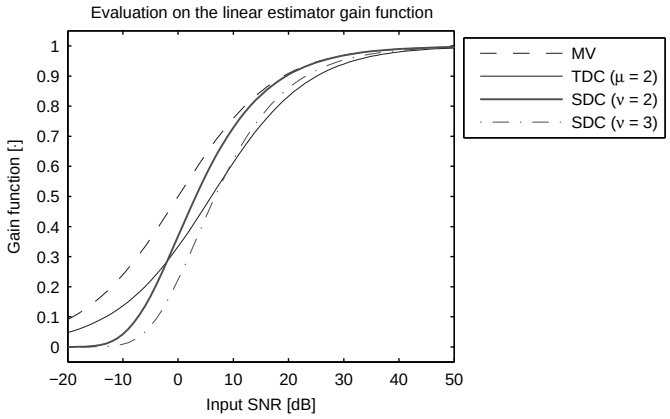


Figure 2.11: The gain function seen in third column of Table 2.5 evaluated for MV (dashed), TDC, $\mu = 2$ (full), SDC, $\nu = 2$ (full, fat), and SDC, $\nu = 3$ (dash-dotted). Each gain function is evaluated for the estimated input SNR, $\zeta = \lambda_{s,i}/\sigma^2$.

gain function for the MV, TDC, and SDC estimators as a function of the input SNR. One should realise, that the SNR mentioned, of course, refers to the estimated SNR by the definition of ζ.

The SDC estimator has been reported to give good results. This is likely due to the aggressive noise reduction, which can be observed in Figure 2.11. The estimators presented; LS, MV, TDC, and SDC, are summarised together with their $\mathrm{SNR} = \zeta$ definitions in Table 2.5. Having proposed several methods for estimation of the speech component in the signal subspace, we describe the problem of estimating, or approximating, the correct signal subspace dimension, K. Or, should we say, the correct trade-off between noise reduction and speech distortion, as the speech are not likely to be completely confined to the signal subspace, and truncation will inevitably introduce some speech distortion.

2.4.5 Estimators of the Dimension of the Signal Subspace

In the preceding sections focus has been laid on the underlying theory and derivations of the signal subspace approach. In previous section a number of important estimators for the speech component in the signal subspace have been presented. The low-rank modelling and inherit truncation of the noise subspace is of paramount importance to the noise reduction performance of the signal subspace approach. However, so far, we have left out the practical problem of estimating the order of the signal subspace, K. It has been noted, that the signal subspace approach employing low-rank speech modelling, is, in fact, the only known method to explicitly define noise reduction and speech distortion as an explicit trade-off parameter [9]. At one hand we seek to minimise the order to ensure a large noise reduction, while, at the other hand, we seek to prevent speech distortion by underestimation of the signal subspace. In this section we present some theoretical observations from Scharf et. al. [79] which clarifies this trade-off.

It is proposed to use the signal subspace dimension that minimises the residual energy [79], defined in (2.39), and restated here

$$\mathbf{r} = \mathbf{r}_s + \mathbf{r}_b$$

The energy in the speech residual, $\mathbf{r}_s$, can then be formulated as [37]

$$\epsilon_s^2 \triangleq \mathrm{tr}\big(E\{\mathbf{r}_s\mathbf{r}_s^T\}\big) = \mathrm{tr}\big((\mathbf{F}_M - \mathbf{I}_M)\Lambda_s(\mathbf{F}_M - \mathbf{I}_M)^T\big) \tag{2.54}$$

and energy in the residual noise, $\mathbf{r_b}$, can be formulated as

$$\epsilon_b^2 \triangleq \mathrm{tr}\big(E\{\mathbf{r}_b\mathbf{r}_b^T\}\big) = \sigma^2\,\mathrm{tr}\big(\mathbf{F}_M\mathbf{F}_M^T\big) \tag{2.55}$$

where $\mathbf{F}_M$ is an $M \times M$ version of $\mathbf{F} \in \mathbb{R}^{K\times K}$ (see Eq. (2.34)), i.e. a full rank estimator matrix. Using the LS estimator for the linear estimator, $\mathbf{F}_M = \mathbf{I}$, the residual signal energy will be the sum

of the eigenvalues that are not contained in the signal subspace

$$\epsilon_s^2 = \sum_{i=K+1}^{M} \lambda_i^2 \tag{2.56}$$

By reducing the rank of the signal space, not all eigenvalues will be used in the estimation of the speech signal, thus a (small) difference between the original speech signal and the estimated speech signal will be introduced. The residual speech energy, ϵ_s^2, will therefore decrease as K increases.

Continuing with the LS estimator, the residual noise energy will be the noise variance in each eigenvalue contained in the signal subspace

$$\epsilon_n^2 = \sigma^2 K \tag{2.57}$$

Since the observations are degraded with white noise, the noise variance will be distributed on all eigenvalues, as illustrated in (2.28). Therefore, as K increases, more noise is added to the speech signal. The residual noise energy, ϵ_n^2 can therefore be minimised by minimising K.

Using the residual speech energy, ϵ_s^2 and the residual noise energy, ϵ_n^2, the total estimation residual energy can be formulated, as

$$\epsilon^2 = \epsilon_s^2 + \epsilon_n^2 \tag{2.58}$$

The idea proposed by Scharf et al. in [79] for the LS estimator is illustrated in Figure 2.12(a). The optimum choice for the rank, K_{opt}, that will minimise ϵ^2 can be formulated as

$$K_{\text{opt}} = \arg\min_K(\epsilon^2) \tag{2.59}$$

In the figure, the residual noise (dashed), the speech residual (dash-dotted), and the sum of the two (bold), are depicted. As expected, the highest noise reduction is achieved for $K = 0$, while the least speech residual is when $K = M$. According to Eq. (2.59), the optimal model order, K_{opt}, is when the sum is the least.

If we use the LMMSE estimator, instead of the LS estimator, $\mathbf{F}_M$ is replaced by $\mathbf{F}_{MV}$ in the previous formulae. See Table 2.5 for a definition of $\mathbf{F}_{MV}$. Diagonal elements of $\mathbf{F}_{MV}$ from $K+1$ to M will be zero, thereby the residual energy for the LMMSE estimator can be described using (2.54) and (2.55) as

$$\epsilon_s^2 = \text{tr}\left((\mathbf{F}_{MV} - \mathbf{I}_M)\Lambda_s(\mathbf{F}_{MV} - \mathbf{I}_M)^T\right) \tag{2.60a}$$

$$\epsilon_b^2 = \sigma^2 \,\text{tr}\left(\mathbf{F}_{MV}\mathbf{F}_{MV}^T\right) \tag{2.60b}$$

$$\epsilon^2 = \epsilon_s^2 + \epsilon_b^2 \tag{2.60c}$$

The residual energy will not increase significantly if the signal dimension, K, is chosen large enough [37], as depicted in Figure 2.12(b). This is evident since the Wiener gain function tries to remove the noise on each eigenvalue. This conclusion can be extended to the TDC and SDC estimators as well [37].

We have introduced theoretical background to understand why the determination of the signal subspace dimension is important to find a trade-off point between noise reduction and speech distortion. The original paper by Ephraim and Van Trees. included a method for model-order estimation which was based on work by Merhav [61]. Basically, Ephraim and Van Trees noted, that it was important to prevent underestimation, which is verified by both Figure 2.12(a) and Figure 2.12(b). The residual error for the MV estimator is, however, seen to be invariant to the model order as long as the order is large enough (larger than 10 in Figure 2.12(b)) in order to keep the speech residual below the noise residual. This observation is supported by [49] where the output SNR is seen to be rather order invariant, when the signal subspace dimension, K, is large enough (above $K = 6$ out of $M = 20$ in [49]).

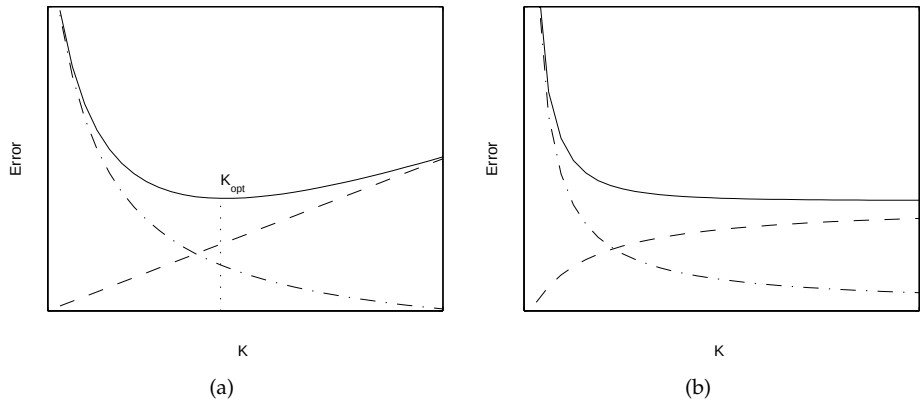


Figure 2.12: The bias (dash-dotted) and the variance (dashed) are plotted as function of the reduced rank, K. They together form the residual error (full line). The optimum choice of rank is where the residual error is minimised. Figure 2.12(a) uses an LS estimator, whereas Figure 2.12(b) uses the LMMSE/MV estimator. Both plots are sketches, freely made after [80, 37], for illustrating purposes.

Summary

In this section we have introduced the low rank model, which lead to the signal subspace method. A frequency-to-eigen-domain interpretation was then given in order to gain insight into the signal-dependent transform. Four different estimators and a rank-decision method to determine the dimension of the signal subspace were presented in order to inherently estimate the noise. It was shown that the SNR improvement is proportional to M/K, why M should be chosen $M > K$. However, so far, we have assumed the auto-correlation matrix known. This is not true for practical implementations.

As previously mentioned there generally exist two classes of estimators. Those we call statistical, which are based on forming the optimal filter by eigen-decomposition. No restrictions are laid upon these estimators. The other type we call empirical, because it is based on implicitly forming the auto-correlation matrix by computing an SVD of a data-matrix with a specific structure. For both methods, it is worth noting, that the length of the observation vector used for computing the auto-correlation matrix should be chosen as to obey the quasi-stationary property known to be attributed to speech.

2.4.6 Signal Subspace for Noise Reduction by Singular Value Decomposition

The signal subspace approach can be realised using the eigenvalue decomposition as shown by Ephraim and Van Trees and restated in Algorithm 4. In Jensen et al. [49] it is proposed to use the singular value decomposition (SVD, see Appendix A) on a Hankel data matrix. In this section we will introduce this method as it is numerically attractive and shares many characteristics with the multi-channel Wiener filter, described in Chapter 3.

A Hankel matrix has constant elements along the anti-diagonal, and the theory equally well applies to Toeplitz matrices with constant diagonals (related through the column-*exchange matrix*, see Appendix A). Assume a length-N observation signal $\mathbf{y}[k]$ known at time instance k. Let a $p \times m$ rectangular data matrix $\mathbf{Y}$ be given using the data from the observation vector, as

$$\mathbf{Y}[k] = \mathcal{T}(\mathbf{y}[k]) = \begin{bmatrix} \mathbf{y}^T[k-L+1] \\ \mathbf{y}^T[k-L+2] \\ \vdots \\ \mathbf{y}^T[k-1] \\ \mathbf{y}^T[k] \end{bmatrix} = \begin{bmatrix} y[k-L+1] & y[k-L] & \dots & y[k-N+1] \\ y[k-L+2] & y[k-L+1] & \dots & y[k-N+2] \\ \vdots & \vdots & \ddots & \vdots \\ y[k-1] & y[k-2] & \dots & y[k-M] \\ y[k] & y[k-1] & \dots & y[k-M+1] \end{bmatrix} \tag{2.61}$$

Such a matrix is called Toeplitz (see Appendix A). The Toeplitz data matrix forms the basis for the empirical auto-correlation estimator using the auto-covariance windowing method [39]. Let

us define the auto-correlation estimator as (see Appendix B)

$$\hat{r}_{yy}(l)[k] = \begin{cases} \frac{1}{2N} \sum_{n=-N}^{N} y[k+n]y[k+n+l] & 0 \leq l \leq M-1, \\ 0 & \text{otherwise} \end{cases} \tag{2.62}$$

then we can see the relation to the data matrix in Eq. (2.61), as the auto-correlation matrix formed using the auto-correlation estimator of Eq. (2.62) can be formed by the product

$$\hat{\mathbf{R}}_{yy} = \mathbf{Y}[k]^T \mathbf{Y}[k]/L \in \mathbb{R}^{M \times M} \tag{2.63}$$

A paramount property of the SVD when applied in signal processing applications is the implicit computation of the auto-correlation matrix of Eq. (2.63) by diagonalisation of the Toeplitz data matrix (2.61). It is possible to combine the linear speech signal estimators presented in Section 2.4.4 and the diagonalisation of the data matrix. The method proposed by Jensen et al. [49] does exactly that. In Algorithm 5 the *truncated SVD* (TSVD) is shown. By the end of the procedure, we end up with an enhanced signal matrix, $\hat{\mathbf{S}}$, from which we would like to extract the enhanced speech signal. In the statistical approach by Ephraim and Van Trees [24], it was possi-

Signal Subspace based on Truncated Singular Value Decomposition (TSVD, `Jensen95.m`)

1. Form the Toeplitz data matrix of (2.61).
2. Compute SVD of $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
3. Compute the linear estimator, $\mathbf{F} = \text{diag}\big(f(\sigma_{y,i})\big),\ 0 < i \leq K$.
4. Remove the noise subspace by discarding the $M - K$ lowest singular values (and singular vectors).
5. Reconstruct the data matrix by $\hat{\mathbf{S}} = \mathbf{U}_K(\mathbf{F}\mathbf{\Sigma}_K)\mathbf{V}_K^T$
6. Average along the diagonals to obtain $\hat{\mathbf{s}}[k]$, for example by (2.69).

Algorithm 5: The truncated singular value decomposition (TSVD) proposed by Jensen et al. [49]. It implicitly forms the auto-correlation matrix. The reconstructed signal must be retrieved from the diagonals of the reconstructed matrix, $\hat{\mathbf{S}}[k]$, which no longer holds the Toeplitz structure.

ble to consistently recover the auto-correlation matrix of the speech-only signal, $\mathbf{R}_{ss}$, see (2.29) on page 42. This is, however, not possible when implicitly forming the auto-correlation matrix $\mathbf{R}_{yy}$ by virtue of the signal value decomposition of the data matrix. Consider the the singular value decomposition of the speech-only auto-correlation matrix

$$\mathbf{R}_{ss} = \begin{bmatrix} \mathbf{U}_{s,1} & \mathbf{U}_{s,2} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{s,1} & \mathbf{0}_{K\times(M-K)} \\ \mathbf{0}_{(M-K)\times K} & \mathbf{0}_{(M-K)} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{s,1}^T \\ \mathbf{V}_{s,2}^T \end{bmatrix} \tag{2.64}$$

In De Moor [66], it is shown that the singular values and the right singular vectors (row space) of the speech auto-correlation matrix, $\mathbf{R}_{ss}$, can be recovered consistently from the singular value decomposition of $\mathbf{R}_{yy}$. This is, however, not true for the left singular vectors (column space), $\mathbf{U}_{s,1}$ and $\mathbf{U}_{s,2}$. Following the derivation in [66], the noisy speech signal can be decomposed using the singular value decomposition, as

$$\begin{aligned} \mathbf{Y} &= \mathbf{S} + \mathbf{B} && (2.65)\\ &= \mathbf{U}_{s,1}\mathbf{\Sigma}_{s,1}\mathbf{V}_{s,1}^T + \mathbf{B}\mathbf{V}_{\mathbf{s,1}}\mathbf{V}_{\mathbf{s,1}}{}^T + \mathbf{B}\mathbf{V}_{s,2}\mathbf{V}_{s,2}^T && (2.66)\\ &= \Big[(\mathbf{U}_{s,1}\mathbf{\Sigma}_{s,1} + \mathbf{B}\mathbf{V}_{s,1})\left(\mathbf{\Sigma}_{s,1}^2 + \sigma^2\mathbf{I}_K\right)^{-1/2} \quad \mathbf{B}\mathbf{V}_{s,1}\sigma^{-1}\Big] && (2.67)\\ &\quad \times \begin{bmatrix} \left(\mathbf{\Sigma}_{s,1} + \sigma^2\mathbf{I}_K\right)^{-1/2} & \mathbf{0}_{K-M} \\ \mathbf{0}_{M-K\times K} & \sigma\mathbf{I}_{M-K} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{s,1}^T \\ \mathbf{V}_{s,2}^T \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{U}_{y,1} & \mathbf{U}_{y,2} \end{bmatrix} \begin{bmatrix} \left(\mathbf{\Sigma}_{s,1} + \sigma^2\mathbf{I}_K\right)^{-1/2} & \mathbf{0}_{K-M} \\ \mathbf{0}_{M-K\times K} & \sigma\mathbf{I}_{M-K} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{y,1}^T \\ \mathbf{V}_{y,2}^T \end{bmatrix} && (2.68) \end{aligned}$$

where we have left out the time indices, k, for convenience. From this it can be seen, that $\mathbf{U}_{s,1} \neq \mathbf{U}_{y,1}$. It is further noted in [66], that the column space, $\mathbf{U}_{s,1}$, cannot be recovered asymptotically, as $M \rightarrow \infty$, where $\mathbf{Y} \in \mathbb{R}^{L \times M}$. This is the difference between the EVD and SVD based signal subspace approach.

By the TSVD algorithm, the input Toeplitz data matrix, $\mathbf{Y}[k]$ forms the basis for the rank-reduced matrix, $\hat{\mathbf{S}}[k]$. This matrix does not, in general, hold the Toeplitz structure from the data matrix due to perturbations by the gain matrix, $\mathbf{F}$. A common way to circumvent the signal reconstruction problem is arithmetic averaging along the diagonals (anti-diagonals for Hankel matrices) [21].

Assume we are interested in the enhanced speech signal vector, $\hat{\mathbf{s}}[k]$, with elements $s[k']$, where the index, $0 \geq k' \geq N-1$. Remember the input column data vector, $\mathbf{y}[k]$, was of length-N ($N = M + L - 1$). Now using the new time index, k', we define the diagonal-averaging process as

$$\begin{aligned} \hat{s}[k'] &= \frac{1}{\beta - \alpha + 1} \sum_{j=\alpha}^{\beta} \hat{\mathbf{S}}_{(j+k'-(M-1)),\, j} \\ \alpha &= \max(M - k', 1) \\ \beta &= \min(N - k', M) \end{aligned} \tag{2.69}$$

where $\hat{\mathbf{S}}_{i,j}$ denotes the (i,j)th element of the matrix, $\hat{\mathbf{S}}[k]$.

However, as we noted in the beginning of this section, the signal subspace approaches inherently form noise estimates in white noise, but they can not operate in non-white noise conditions. However, using a VAD and using joint diagonalisation (or pre-whitening) it is possible. This is shortly described in the following section.

2.4.7 Non-White Noise, Pre-Whitening, and Joint Diagonalisation

In this section we will extend the signal subspace techniques to handle non-white noise. The two signal subspace approaches operate on two different matrices. The EVD works on the auto-correlation matrices, whereas the SVD-based technique uses a Hankel structured data matrix and from this forms the empirical auto-covariance matrix.

However, having only the auto-covariance or auto-correlation matrices with non-white present the noise cannot be estimated. Using a VAD and assuming the noise is stationary for shorter periods of time an estimate of the noise correlation matrix or a noise sample matrix can be formed to use in the EVD or SVD correspondingly.

With this a priori information the EVD can handle different noise types using joint diagonalisation, which basically can be interpreted as solving a constrained least squares (CLS) problem. Joint diagonalisation is also known as the generalised eigenvalue decomposition (GEVD). The problem of pre-whitening the noise data in the SVD approach is solved using the generalised singular value decomposition (GSVD).

Restating the formulation of the EVD-based signal subspace approach as in (2.25) slightly rewritten we get

$$\mathbf{R}_{yy} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \tag{2.70}$$

using the noise estimate obtained using the VAD, $\mathbf{R}_{bb}$, in a GEVD we obtain (as defined in Appendix A)

$$\begin{cases} \mathbf{R}_{yy} = \mathbf{Q}\mathbf{\Lambda}_y\mathbf{X}^{-1} \\ \mathbf{R}_{bb} = \mathbf{Q}\mathbf{\Lambda}_b\mathbf{X}^{-1} \end{cases} \tag{2.71}$$

where $\mathbf{X}$ is an invertible, but not necessarily orthogonal matrix. Now estimating $\mathbf{R}_{ss}$ as $\mathbf{R}_{yy} - \mathbf{R}_{bb}$

using the signal subspace part of the decompositions we get

$$\mathbf{R}_{ss} = \mathbf{Q}_1 \mathbf{\Lambda}_{K,y} \mathbf{X}_1^{-1} - \mathbf{Q}_1 \mathbf{\Lambda}_{K,b} \mathbf{X}_1^{-1} \tag{2.72a}$$
$$= \mathbf{Q}_1 \left(\mathbf{\Lambda}_{K,y} - \mathbf{\Lambda}_{K,b} \right) \mathbf{X}_1^{-1} \tag{2.72b}$$

where subscript 1 denotes the signal subspace part of the decompositions. The filter matrix used in the signal subspace becomes, referring to (2.34)

$$\mathbf{W} = \left(\mathbf{Q}_1 \mathbf{\Lambda}_{K,y} \mathbf{X}_1^{-1} \right)^{-1} \mathbf{Q}_1 \mathbf{\Lambda}_{K,s} \mathbf{X}_1^{-1} \tag{2.73}$$
$$= \mathbf{X}_1 \mathbf{\Lambda}_{K,y}^{-1} \mathbf{Q}_1^{-1} \mathbf{Q}_1 \mathbf{\Lambda}_{K,s} \mathbf{X}_1^{-1} \tag{2.74}$$
$$= \mathbf{X}_1 (\mathbf{F} \mathbf{\Lambda}_g) \mathbf{X}_1^{-1} \tag{2.75}$$

where $\mathbf{F}$ is the estimator using the $K \times K$ generalised eigenvalue matrix, $\mathbf{\Lambda}_g$, where K is the size of the signal subspace.

The GSVD using the noise data matrix can be analogously described as (also defined in Appendix A.2)

$$\begin{cases} \mathbf{Y} = \mathbf{U}_Y \mathbf{\Sigma}_Y \mathbf{X}^{-1} \\ \mathbf{B} = \mathbf{U}_B \mathbf{\Sigma}_B \mathbf{X}^{-1} \end{cases} \tag{2.76}$$

where $\mathbf{X}$ is an invertible, but not necessarily orthogonal matrix. One can now obtain the estimated speech matrix, $\hat{\mathbf{S}}$, using the interpretation, that the matrix $\mathbf{X}^{-1}$ transforms the observed data into the signal-domain and inherently transforms the noise (based on the VAD markings) in the observed signal white, therefore the term, a pre-whitening step. After the pre-whitening step the method resembles the SVD approach for white-noise, described in Algorithm 5, that is

$$\hat{\mathbf{S}}[k] = \mathbf{U}_{1,Y} \mathbf{F} \mathbf{\Sigma}_{1,g} \mathbf{X}_1^{-1} \tag{2.77a}$$

where the estimator matrix, $\mathbf{F}$, uses the generalised singular values, $\mathbf{\Lambda}_{1,g}$ and subscript 1 indexes the signal subspace. The last step is averaging along the diagonals to obtain $\hat{s}[k']$ e.g. by using the method proposed in (2.69).

Having obtained signal subspace methods that are capable of removing non-white noise as well we investigate the signal subspace with respect to different noise types, the model-order decision method is examined, and a comparison of the EVD and SVD based approached is commented.

2.4.8 Experimental Results

In this section the two signal subspace techniques, proposed by [24] and [49], using the EVD and SVD respectively, are tested. We will use our standard test signal, a female uttering:"*Good service should be rewarded with big tips*", degraded with additive white noise with a noise level at $5\,\mathrm{dB}$. First we will compare the two methods, then test different linear estimators of the signal subspace. Finally we will test two different sentences degraded with additive white and pink noise at SNRs -5, 0, 5, and $10\,\mathrm{dB}$, respectively. In this test the two generalised methods presented in the previous section is used.

In the following simulations we will use a frame size of $32\,\mathrm{ms}$, which also were used in the spectral subtraction. The dimension of the full noise space is fixed at $M = 32$, and a fixed signal subspace is used, the dimension is set to $K = 14$. In some simulations, the model order of the system is estimated, thus the speech subspace dimension will be of varying size.

Comparing the EVD and SVD for Signal Subspace

The signal subspace technique has classically been implemented using two different methods. The method proposed in [24] is summarised in Algorithm 4 on page 43 and uses the EVD to estimate the eigenvalues and corresponding eigenvectors. Another method, proposed in [49], uses

the SVD, instead of the EVD, as summarised in Algorithm 5 on page 51. We have implemented and tested the two methods using our standard test setup. The method based on the SVD is implemented as described in Algorithm 5, where the MV linear estimator, $\mathbf{F}_{MV}$, is used. The method based on the EVD is implemented as described in Algorithm 4, also using the MV linear stimator. Forming the autocorrelation matrix, $\hat{\mathbf{R}}_{yy}$, is done using the empirical auto-covariance estimator, since it resembles the data matrix formed using the SVD. The singular values computed using the SVD will then correspond to the eigenvalues computed using the EVD. In [24] the empirical auto-correlation estimator is, however, preferable, since it implicitly involves windowing of the data matrix. These two methods are further discussed in Appendix B. However by choosing the empirical auto-covariance estimator. The two methods are similar, except for the reconstruction of the estimated speech signal. The method based on the SVD does not involve forming the autocorrelation, which therefore reduces the complexity, and numerical accuracy of this method compared with the EVD method. In Table 2.6 the results of the simulations are listed. From this we can see that the SVD can obtain a broadband SNR of $11.07\,\mathrm{dB}$, whereas the EVD obtains $10.66\,\mathrm{dB}$. Also the segmental SNR is higher using the SVD compared to the EVD. The WSSM is, for both the EVD and SVD, increased. Thus more speech distortion is introduced, which is particularly notable for the SVD. The SVD obtains better noise reduction, this is, however, at the expense of increased speech distortion.

Noise estimate	$\mathrm{SNR}[\,\mathrm{dB}]$	$\Delta\mathrm{SNR}[\,\mathrm{dB}]$	$\mathrm{SNR}_{\mathrm{SEG}}[\,\mathrm{dB}]$	$\mathrm{WSSM}[\cdot]$
Observation	5.0	–	-0.4	44.4
Method based on EVD	10.7	5.7	1.4	52.4
Method based on SVD	11.1	6.1	1.6	55.5

Table 2.6: The results are obtained using the method proposed in [24], based on the EVD, and the method proposed in [49], based on the SVD. Both methods uses the MV estimator and is tested on a female voice, degraded with white noise.

In Figure 2.13 the spectrograms of the estimated speech signals are depicted. In the upper figure, the signal subspace method based on the EVD, and, in the lower the method based on the SVD. By comparing them we see that the speech estimate obtained using the SVD has a slightly more blue background, thus more noise is removed, as seen in Table 2.6. No noticeable difference in the speech energy is seen.

The only difference between the two methods was the reconstruction of the estimated speech signal. In [49] averaging of the off-diagonal elements in the SVD is found better, which can explain the small difference between the two methods seen in the table.

Test of the Different Estimators for Signal Subspace

The subspace technique offers the ability to use different estimators. We have tested four different estimators, the LS, MV, TDC, and SDC, listed in Table 2.5, using the signal subspace method based on the SVD. Again we have used the standard test setup. We have tested the four different estimators using a fixed speech dimension, $K = 14$, however we have also tested the different estimators applying model order estimation, thus the speech dimension is estimated in each frame and is thereby time-varying. The speech dimension is estimated using the approach described in [61].

In Table 2.7 the results for the different estimators are listed. For each estimator the results are listed using both a fixed speech dimension and using the estimated speech dimension. In Section 2.4.5 the μ and ν were defined for the TDC and SDC estimator, respectively. We have found $\mu = 2.1$ and $\nu = 2.8$ preferable. The first row in Table 2.6 describes the observation, which is the standard test scenario.

From Table 2.7 we see that all estimators are able to remove noise from the observation. The SDC estimator is slightly better than the TDC and MV estimator, whereas the LS estimator only obtains a broadband noise reduction of 2.05 and $2.92\,\mathrm{dB}$ with and without model order estima-

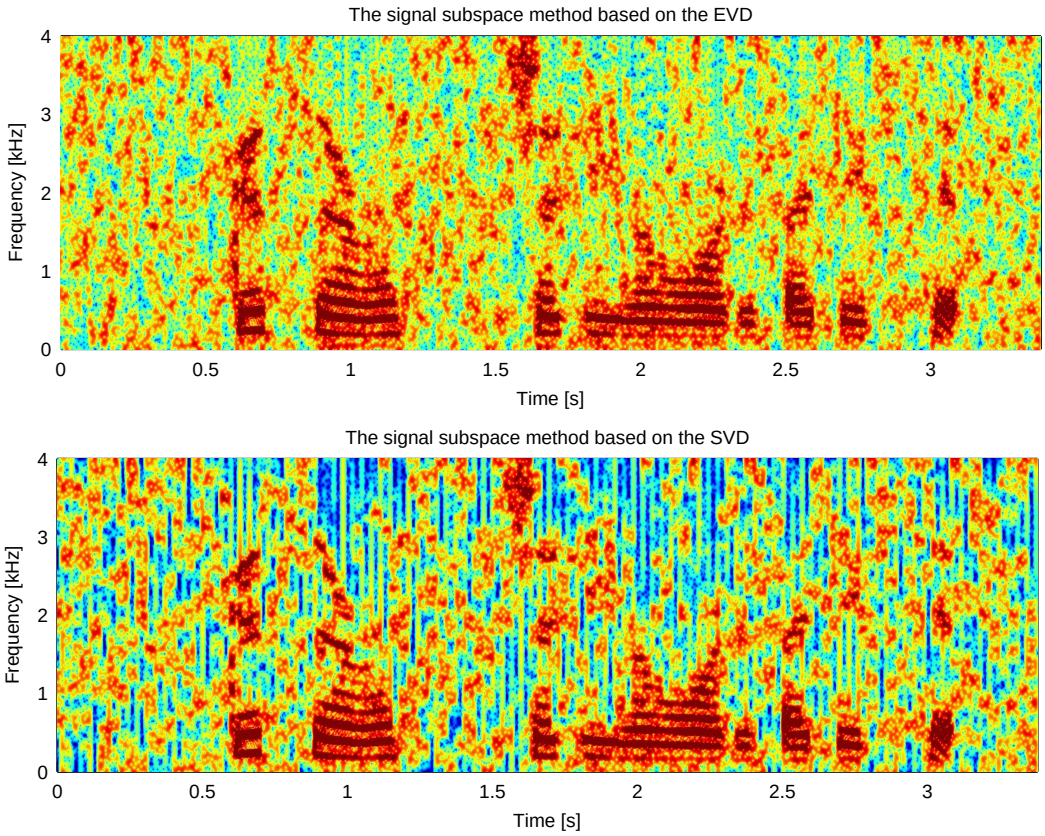


Figure 2.13: The upper spectrogram is obtained using the method proposed in [24], based on the EVD. The lower spectrogram is obtain using the method proposed in [49], based on the SVD. Both methods uses the MV estimator and is tested on a female voice, degraded with $5\,\mathrm{dB}$ white noise.

tion, respectively. The same tendency is apparent for the segmental SNR. For both noise reduction measures, broadband and segmental SNR, the model order estimation has no significantly influence on the results. Whereas for the speech distortion, the use of model order estimation seems to decrease the speech distortion, thus lowering the WSSM. This is however not detectable for the SDC estimator. The WSSM obtained for all estimators is, however, still higher than the observation, thus the noise reduction results in speech distortion.

By using model order estimation the speech dimension is estimated in each frame, when only noise is present the speech dimension is lowered. Whereas in frames containing speech the dimensions are increased, often to more, i.e. $K > 14$, thus the speech energy contained in the eigenvalues up to the estimated model order is preserved. If the estimated speech dimension is larger than 14 this will lead to less speech distortion compared with using a speech dimension of $K = 14$, and thereby a better WSSM score. The same tendencies were found in similar tests using the method based on the EVD.

Using a VAD with the Signal Subspace

In order for the signal subspace to handle other noise types than white noise, a pre-whitening step is needed. This is necessary, for securing the noise variance to be equally distributed on the eigen/singular-values. The pre-whitening however require that the autocorrelation of the noise is known, or a noise matrix can be formed. This can be done by acquiring the noise in noise only frames, detected by a VAD. The pre-whitening can then be obtained by computing the GSVD of the formed noise and data matrix.

Before testing the GSVD on other noise types, different speakers, and at other SNR levels, we have compared it with a standard SVD using $5\ \mathrm{dB}$ pink noise. Again we have used our standard test signal. The estimator used is the MV estimator without model order estimation.

Noise estimate	SNR[dB]	ΔSNR[dB]	$\mathrm{SNR}_{\mathrm{SEG}}$[dB]	WSSM[·]
Observation	5.0	–	-0.4	44.4
LS estimator without model order estimation	8.0	3.0	0.69	56.1
LS estimator with model order estimation	7.1	2.1	0.33	52.5
MV estimator without model order estimation	11.1	6.1	1.6	55.5
MV estimator with model order estimation	10.7	5.7	1.5	52.5
TDC estimator without model order estimation	11.5	6.5	1.8	55.3
TDC estimator with model order estimation	11.5	6.5	1.8	53.1
SDC estimator without model order estimation	11.7	6.7	2.0	55.5
SDC estimator with model order estimation	11.8	6.8	2.0	55.7

Table 2.7: The results are obtained using the method proposed in [49], based on the SVD, tested on a female voice degraded white noise. For the TDC estimator $\mu = 2.1$ was found preferable, in [24] it is advised to be between 2 and 3. $\nu = 2.8$ was chosen for the SDC estimator.

The obtained results are listed in Table 2.8. As expected we can see a significant difference due to the pre-whitening step.

Noise estimate	SNR[dB]	ΔSNR[dB]	$\mathrm{SNR}_{\mathrm{SEG}}$[dB]	WSSM[·]
Observation	5.0	–	-0.3	57.2
Method based on SVD	5.7	0.7	-0.1	61.2
Method based on GSVD	9.7	4.7	1.7	65.7

Table 2.8: The results are obtained using the SVD based signal subspace method tested with $5\,\mathrm{dB}$ additive white noise, using the MV estimators. The method based on the SVD is the same as in Table 2.6 where no VAD is used. The method based on the GSVD is now using an ideal VAD to form a noise matrix and thereby pre-whitening the signal.

In Figure 2.14 two spectrograms of the two speech estimates are depicted. The upper is obtained using the SVD, whereas the lower is the estimate from the GSVD. It is clearly seen that the GSVD-based method has a pre-whitening step. The noise is removed in the entire frequency range, whereas the SVD-based approach suffices when non-white noise is added. The method simply removes what it "believes" is white noise even though pink noise is present. The characteristic, low-frequency noise is still present in the output from this method.

The Signal Subspace Method for other Noise Types

The signal subspace will now be tested for both white and pink noise of noise levels at $-5, 0, 5$, and $10\ \mathrm{dB}\ \mathrm{SNR}$. Two different sentences have been deployed, a male and a female. The GSVD is used with the MV estimator without model order estimation. Using these settings we obtain the results depicted in Figure 2.15.

From the figure it is observable that the signal subspace can remove more white noise than pink noise, for all noise levels, despite the pre-whitening step. However, pink noise introduces less speech distortion than white noise especially for the male speaker. This tendency seems general, the more noise removed, the more speech distortion is introduced. Another general tendency is that more noise can be removed as the noise levels of the observations increase, which is also seen in Figure 2.15.

Conclusion

We have compared the EVD and the SVD and shown and has chosen this method throughout the remaining tests, because it manages to remove slightly more noise and that using a VAD is directly applicable by forming a noise matrix containing the non-VAD-marked samples. The SDC estimator shows to be the best estimator with respect to noise reduction as it was also shown in [24]. Using low-rank model estimation is seen to be insignificant when the signal subspace size

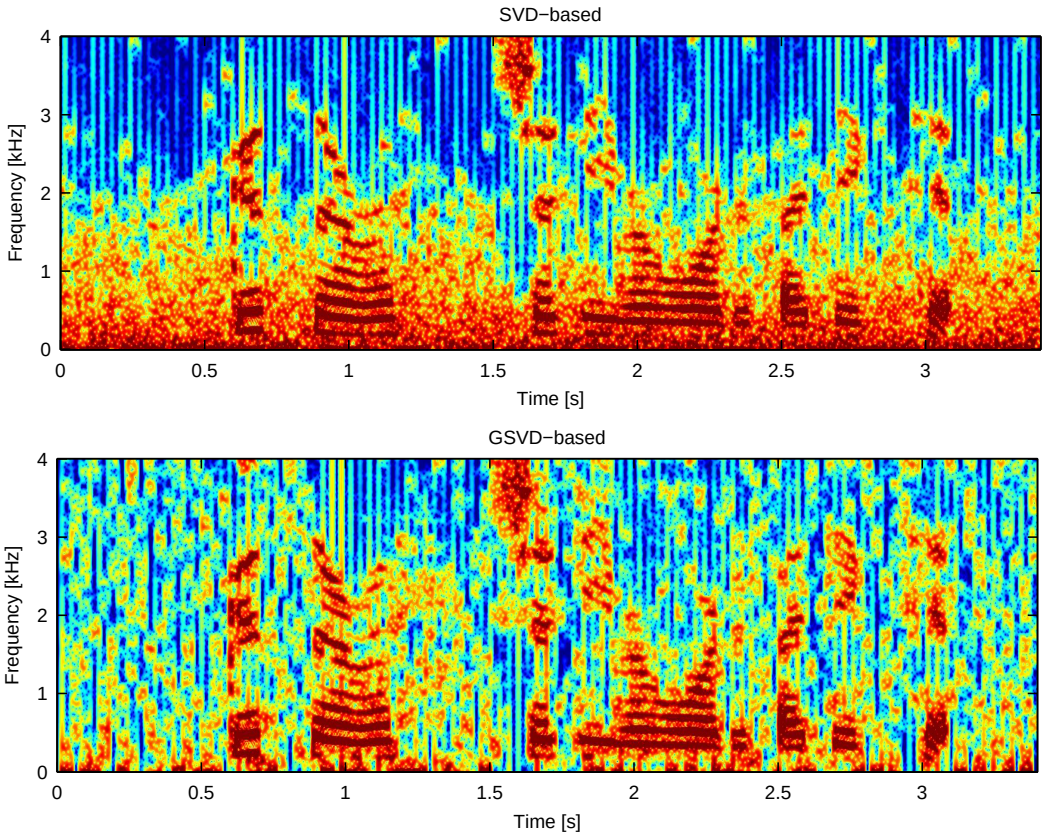


Figure 2.14: In the upper spectrogram the SVD based signal subspace method is tested with $5\,\mathrm{dB}$ additive pink noise using the MV estimator. In the lower spectrogram the same simulation is performed, but now using an ideal VAD to form a noise matrix and thereby using the GSVD. It is clearly seen that the GSVD-based approach pre-whitens the pink noise and the output resembles the SVD of $5\ \mathrm{dB}$ white noise as depicted in Figure 2.13.

K is greater than 6 and all but the LS estimator is used. Therefore, we have chosen not to use model order estimation, or the LS estimator. The GSVD was compared to the SVD-based signal subspace approach in a pink noise setup. The method was as expected able to reduce more noise than the basic SVD technique. In general it seemed that the higher noise reduction the more speech distortion was introduced.

In the last section of this chapter, we compare the spectral subtraction using three different noise estimation methods, with the GSVD-based signal subspace technique in different noise, speaker, and SNR-level setups. The methods are also tested with respect to reverberated signals.

2.5 Discussion and Conclusion

In this chapter we introduced single-channel speech enhancement using a simple, and well-known technique, spectral subtraction. However, it was quickly evident that the method lacked a noise estimate. We introduced the minimum-statistics method proposed by Martin [59], and a log-energy-based VAD proposed by Gerven and Xie [29], in order to have two different methods that were able to obtain a noise estimate. A low rank model of speech was then introduced, and it was shown how a signal-dependent transform resulted in another single-channel technique, the signal-subspace approach. The EVD based method by Ephraim and Van Trees, [24], and the SVD-based by Jensen et al. [49], was described. The methods were extended to deal with non-white noise as well. They were compared and the GSVD-based approach was chosen to investigate robustness with respect to different speakers, noise types, and noise levels.

In this section we compare the spectral subtraction and the signal subspace techniques. We

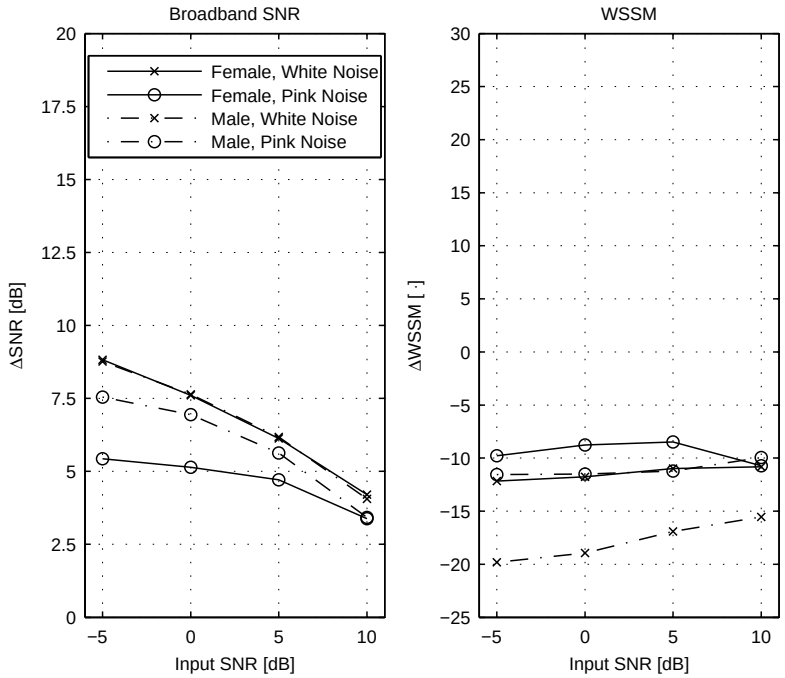


Figure 2.15: Noise reduction and speech distortion for the signal subspace method using GSVD where a noise matrix is formed based on an ideal VAD. The method is tested for two different sentences, degraded with white and pink noise, at $-5, 0, 5$, and 10 dB SNR.

have chosen two different test setups in order to compare the methods. In addition, spectral subtraction is used in combination with three different types of noise estimators; an ideal VAD, Martin's method, and the log-energy-based VAD. The GSVD-based signal subspace technique uses an ideal VAD, however.

For spectral subtraction we have used $\alpha = 5$ and $\beta = 0.01$. In both methods a window of 32 ms duration (256 samples at 8000 Hz) and an overlap of 50% has been used. In the signal subspace approach the decomposition size is $M = 32$ and the reconstructed signal subspace is truncated to $K = 14$.

The first comparison, shown in Figure 2.16, is made in a non-reverberated room with -5 to 10 dB additive pink noise present. The methods are in general seen able to remove noise when more noise is present in the observation. The spectral subtraction, however, is seen to be significantly better than the signal subspace approach both w.r.t. to SNR improvement and speech distortion. Using Martin's noise minima-tracking method speech distortion is seen to be better than using a VAD at low-level-SNR inputs. The log-energy-based VAD is observed to obtain at least as good results as the ideal VAD.

Referring to the discussion about speech quality in Section 1.3.1, we must conclude that even though noise has been removed, the WSSM measure indicates that the methods have introduced additional speech distortion compared to the observed signal. This is due to on-off switching of pure, low-level tones, known as musical noise, inevitable in both speech enhancement methods.

In Figure 2.17 the two single-channel speech enhancement techniques are compared in convolutional noise. Reverberation times from $100 - 600$ ms have been used. The additive noise is pink with an SNR $= 5$ dB. The single-channel techniques are seen unable to remove noise with reverberation present, and share similar performance. With respect to WSSM, Martin's method is slighty better than the other methods in spectral substraction, but not significantly. Again spectral subtraction is significantly better than the signal subspace approach.

In the last figure it was shown that single-channel methods were not capable of removing noise in reverberated environments. The methods are due to the "single-channel" not able to do spatial filtering. In the next section we introduce one additional microphone in order to examine the possibilities of spatial filtering, also known as beamforming.

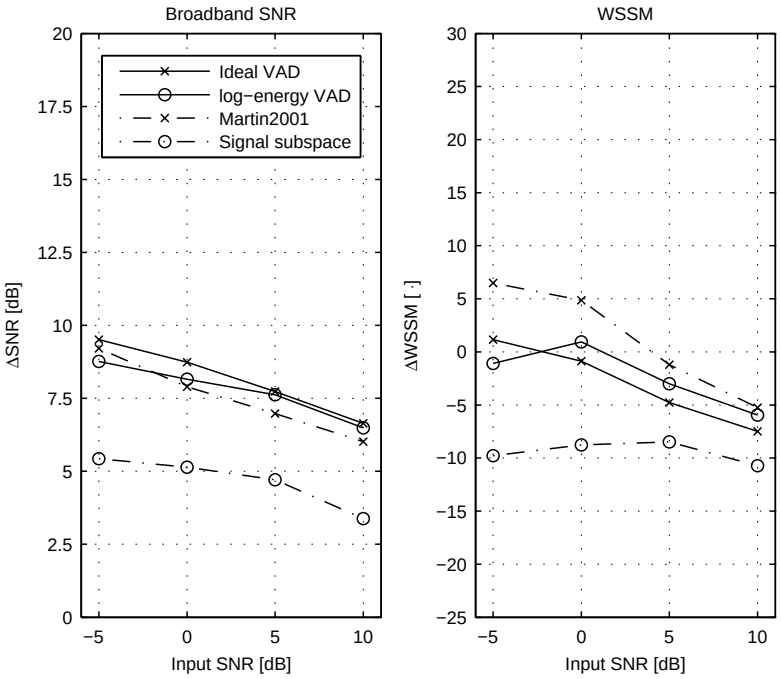


Figure 2.16: Comparision of the two single-channel speech enhancement techniques. No reverberation is used, and the additive noise is pink. Spectral subtraction is seen to be significantly better than the signal subspace approach. Using Martin's noise minima-tracking method speech distortion is seen to be better than using a VAD with low-level SNR input.

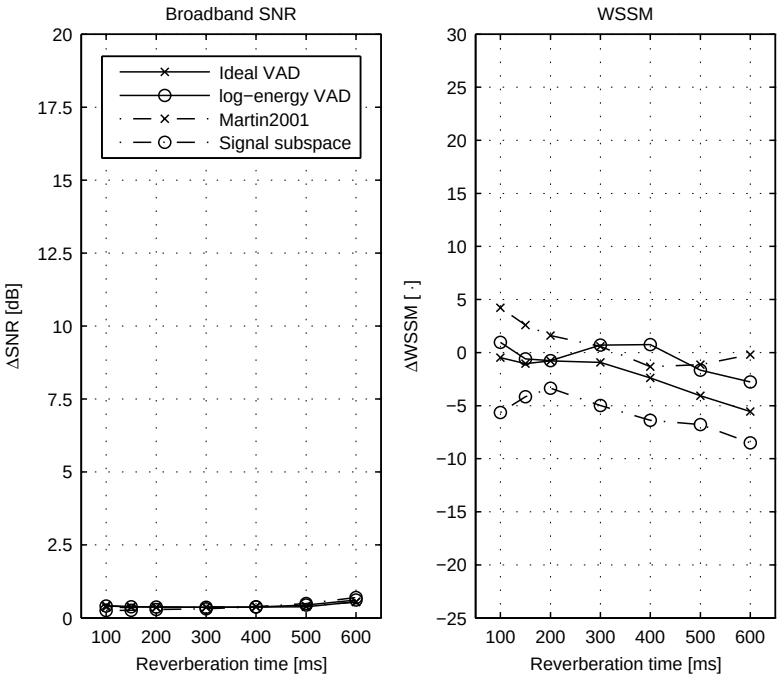


Figure 2.17: Comparision of the two single-channel speech enhancement techniques with convolutional noise. Reverberation times from $100 - 600$ ms have been used. The additive noise is pink with an $\mathrm{SNR} = 5$ dB. The single-channel techniques are seen unable to remove noise with reverberation present, and share similar performance. The WSSM score indicate that Martin's method is slighty better than the other methods in spectral substraction, but not significantly. Again spectral subtraction is significantly better than the signal subspace approach.

CHAPTER 3

Noise Reduction Techniques Based on Multi-Channel Observations

We will in this chapter introduce an additional microphone to the application, such that two observations are used instead of one. In Section 1.2.4 we formulated the observation model of which our problem is founded. In the previous chapter we reduced this model to only concern one channel and only noise reduction. By introducing a second microphone we can now reformulate our observation model back to the original proposal, thus our observation model is described as

$$y_m(t) = g_m(t) * s(t) + b_m(t) \qquad m = 1, \ldots, M \tag{3.1}$$

where $y_m(t)$ is the observation at microphone m, $M = 2$ is the number of microphones, and $g_m(t)$ is the room responses convolved with the speech signal. By adding the room response $g_m(t)$ to the observation model, we introduce spatial information to the system. Sampling the signals using more than one microphone, enables the spatial information to be exploited by spatial filtering, i.e. discrimination of signals based on the physical locations of their sources. If a speech signal is corrupted by interfering signals coming from other speakers then temporal filtering is not ideal since the desired signal and the interfering signals would reside in the same spectral bands. However, if the interfering talkers are located in different locations than the desired talker, these can be separated by exploiting spatial filtering. Introducing an additional microphone is also motivated by Spriet et al. [83] that observed using more microphones in behind-the-ear (BTE) hearing-aid devices is a clear benefit for the resulting speech intelligibility using speech enhancement techniques.

With this as motivation we will in this chapter examine different spatial filtering techniques for speech enhancement. We start out by describing the fundamental principles in spatial sampling. We introduce the delay-and-sum to illustrate the basics in beamforming techniques, which is easily extended to the filter-and-sum beamformer. We derive the LCMV beamformer in order to elegantly formulate an adaptive beamformer, the generalised side-lobe canceller (GSC). These techniques unfortunately requires some a priori information of the spatial position of the speech and noise signals in order to work.

Having introduced the fixed and adaptive beamforming techniques and gained understanding of the fundamental theory of spatial filtering, we introduce the multi-channel Wiener filter (MCWF), a more sophisticated and advanced spatial filtering technique. The method can be seen as a single-channel Wiener filter re-casted to a multi-channel scenario. Whereas the single-channel Wiener filter is based on short-time spectral information, the MCWF mainly functions as a beamformer, thus exploits the spatial information, e.g. using the inter-channel cross-correlation. This turns out to be advantageous, as the method is less sensitive to short-time stationarity assumptions, and thus can alleviate some of the musical noise and other distortion artifacts known in other speech enhancement techniques.

The MCWF method is directly applicable to a dereverberation method proposed by Affes et al. [1], such that both additive noise, $b_m(t)$, and convolutional noise, $g_m(t)$, can be reduced. The method, however, relies on priori information of a frequency-dependent ambiguity factor related to the acoustic room impulse response. We present this method in combination with a method we refer to as spectral addition, which is a novel approach, to our knowledge, firstly seen in this report. The method of spectral addition is used to determine this ambiguity factor. We will now start out by exploring the fundamentals in spatial sampling.

3.1 Fixed and Adaptive Beamforming

Spatial filtering requires that data are spatially sampled, for speech enhancement a microphone array is commonly used for sampling data, where the array consists of a number of microphones. Spatial sampling have some close parallels to the classical discrete sampling, we will in this section relate some of these similarity.

We will only consider the case where the microphones are placed on a straight line with a uniform distance, d, from each other. Furthermore we will only consider spatial sampling in the two dimensional plane, even though the sampling can be extended to three-dimensional sampling by placing microphones in a two-dimensional plane, instead of just a straight line [38].

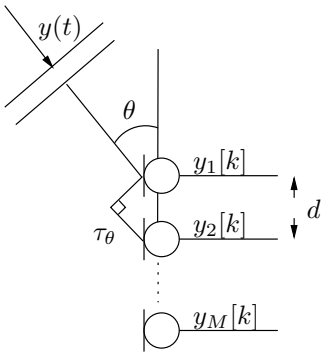


Figure 3.1: Illustration of a microphone array.

In Figure 3.1 a microphone array is depicted. A signal, $y_m(t)$, is impinging from an angle, θ, to the M microphones placed with a distance d. The angle, θ, of the impinging signal determines the time delay, τ_θ, for the input signal to propagate between two neighbouring microphones and is therefore referred to as the spatial sampling time. From Figure 3.1 it can be seen that spatial sampling time τ_θ can be expressed as

$$\tau_\theta = \frac{d}{c}\cos(\theta) \tag{3.2}$$

where c is the speed of sound in the propagating medium, for air at $20^{\text{o}}\,\text{Celcius}$ we will use $c = 340\,\text{m/s}$.

The impinging signal is sampled with a distance corresponding to the spatial sampling time, τ_θ. If the incoming signal is a sine wave this distance correspond to a phase shift. The phase shift, ϕ, depends not only on the spatial sampling time, but also on the frequency of the impinging sine wave, ω. The phase shift, ϕ, is given as

$$\phi = \omega\,\tau_\theta = \omega\,\frac{d}{c}\cos(\theta) \tag{3.3}$$

In order to avoid spatial aliasing the phase shift, ϕ should not exceed $\pi\,\text{rad/s}$, and the angle of the incoming signal is limited to $0^o \leq \theta \leq 180^o$ corresponding to $0 \leq \theta \leq \pi\,\text{rad/s}$. This can be expressed as

$$\pi > \phi > \omega\frac{d}{c}\cos(\theta) \tag{3.4}$$

where worst case occurs for $\theta = 0$ or $\theta = \pi\,\mathrm{rad/s}$ where $\cos(\theta) = 1$ and -1 respectively. We therefore have

$$\begin{aligned} \pi &> \omega \frac{d}{c} \\ \frac{c}{2\,f} &> d \end{aligned} \tag{3.5}$$

This relation is analogue to the Nyquist sampling theorem for electrical sampling. In Section 1.2.2, the speech spectrum was defined to be in the range of $300\,\mathrm{Hz}$ to $3400\,\mathrm{Hz}$, therefore, by using a maximum frequency of $3400\,\mathrm{Hz}$, a maximum distance of $5\,\mathrm{cm}$ between the microphones is required in order to avoid spatial aliasing. Furthermore it is required that the signals are impinging from the front of the microphone array, $0 \leq \theta \leq \pi$. This can not necessary be fulfilled for hearing aids, since noise sources of cause can be located at the rear of the hearing aid. This problem can be solved by using directional microphones, where the beam pattern is constrained within $0 \leq \theta \leq \pi$, this is however beyond the scope of this thesis, for further reading we refer to [36]. We will throughout this report assume that these requirements are fulfilled.

3.1.1 Conventional Beamforming (Delay-and-Sum Beamforming)

One of the most simple realisations of a beamformer is the conventional beamformer or the delay-and-sum (DS) beamformer [39]. The method uses coherent averaging, i.e. the observations are time/phase aligned and then summed up and optionally normalised. This is done by attaching a delay to each microphone signal, thereby compensating for the arrival-time differences of the speech signal to each microphone. The delayed microphone signals, which are time/phase aligned, are then summed coherently. Thereby the magnitudes of the two speech signals will add up, whereas the energy of the interfering noise sources add up. The output is then optionally normalised with respect to the number of microphones. In Figure 3.2 the delay-and-sum beamformer is depicted.

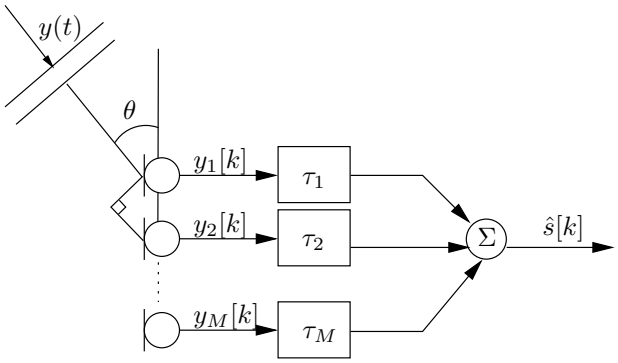


Figure 3.2: Illustration of a delay-and-sum beamformer.

The advantage of the DS beamformer is its simplicity, the only estimation needed is the delay estimation, which is, however, not trivial. This is most commonly done by finding the time lag that maximises the cross-correlation between the input signals [13]. Another method proposed by Benesty [5] uses the impulse responses between the source and the microphones in order to estimate the time delay. The eigenvector corresponding to the smallest eigenvalue of the cross-correlation matrix of the input signals will contain these impulse responses. The direct path from the source to the mth microphone will be represented as a dominant peak in the mth impulse response, by detecting the position of this peak, the mth delay can be estimated.

The disadvantage of the DS beamformer is the large number of microphones required to improve the SNR. A doubling of the number of microphones will ideally provide a $3\,\mathrm{dB}$ increase in

SNR [13]. Following the principles of coherent averaging [55] this can be seen mathematically

$$\begin{aligned} \mathrm{SNR}_{\text{improvement}} &= \mathrm{SNR}_{\text{out}} - \mathrm{SNR}_{\text{in}} \\ &= 10\log\frac{(s\cdot M)^2}{b^2\cdot M} - 10\log\frac{s^2}{b^2} \\ &= 10\log(M) \end{aligned} \tag{3.6}$$

where M is the number of microphones, s is the speech signal of interest, and b is the interfering noise source. The DS beamformer will add the energy of the M noise signals, but add the amplitude of the M speech signals, since the speech signal is time/phase aligned, which leads to an improvement of $3\,\mathrm{dB}$, if the number of microphones are doubled. This is, however, only obtained if the delay estimation is perfect, and the incoming noise signals in each microphone are uncorrelated [13]. This will not be the case when reverberation is present since the noise recorded in each microphone will be correlated and the assumption of coherent averaging becomes invalid, thus no improvement is obtained.

Another disadvantage is, that the delay-and-sum beamformer only enhances the signal in the direction of interest, it cannot suppress the direction of the noise. In the following section we will therefore refine the delay and sum beamformer by replacing the delays with filter coefficients, thereby obtaining a beamformer that holds the ability to form a specific beam pattern.

3.1.2 Super Directive Microphone Array

In the previously section we introduced spatial sampling by explaining the delay and sum beamformer. It, however, suffers from the ability to suppress a noise source from a known direction. We will therefore examine the super directive microphone array, which enables the possibility to form a desired beam pattern. Furthermore the super directive microphone array convey to gaining good understanding of spatial sampling. Finally the super directive microphone array is easily extended to broadband beamforming by combining temporal and spatial sampling.

We start out by replacing the delays in the DS beamformer with weighting coefficients. The beamformer can then be viewed as a spatial FIR filter, where the characteristics will depend on the direction of the impinging signal [38], as opposed to a temporal FIR filter, which depends on the signal frequency. In Figure 3.3 such a beamformer is depicted, where the signal, $y(t)$, is impinging at angle θ. The output at each of the M microphones is denoted $y_1[k], y_2[k], \ldots, y_M[k]$ and the output from the beamformer, $\hat{s}[k]$, is the estimated speech signal. The microphone signals are discrete sampled, the notation therefore changes from (t) to $[k]$. The weighting coefficients $w_1, w_2, \ldots, w_M$ describe the spatial FIR-filter.

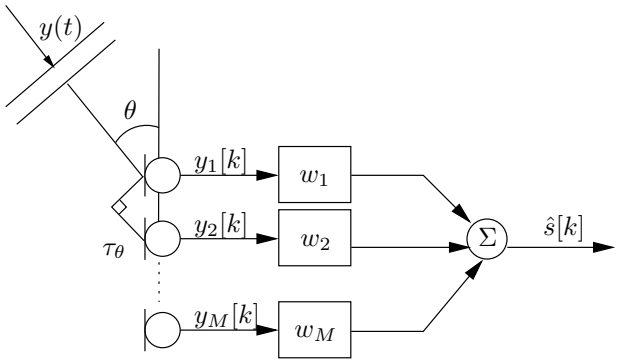


Figure 3.3: Illustration of a narrow-band beamformer.

Using the interpretation as a spatial FIR filter, the microphone output

$$\mathbf{y} = [y_1[k], y_2[k], ..., y_M[k]]^H \tag{3.7}$$

can be seen as a spatially sampled version of the input signal, $y(t)$, where the delay, τ_θ, between each microphone can be seen as the spatial sampling time. In [89] the beamformer output of the spatial FIR filter is given as

$$\hat{s}[k] = \sum_{m=1}^{M} w_m \cdot y_m[k] = \mathbf{w}^H \mathbf{y} \tag{3.8}$$

where $\mathbf{w} = [w_1, w_2, ...w_M]^T$ is the spatial FIR coefficient vector. Considering the impinging signal as a complex sine wave and for convenience let the phase be zero at the first microphone

$$y_1[k] = e^{j\omega k} \tag{3.9a}$$

$$y_m[k] = e^{j\omega(k-(m-1)\cdot\tau_\theta)} \tag{3.9b}$$

where ω is the frequency of the impinging complex sinusoid and τ_θ is the spatial sampling time in seconds. By using Equation (3.3), we can express the incoming signal at microphone m, as a phase shifted version of the first microphone

$$y_m[k] = y_1 e^{-j(m-1)\cdot\phi} \tag{3.9c}$$

Now using a complex sine wave as input to the beamformer, the output of the beamformer can then be expressed by substituting Equation (3.9c) into Equation (3.8)

$$\hat{s}[k] = y_1 \sum_{m=1}^{M} w_m e^{-j(m-1)\cdot\phi}$$

recall that ϕ is the phase shift for a frequency component between two neighbouring microphones and can be expressed as $\phi = \omega\tau_\theta$.

Let us now define a steering vector $\mathbf{d}(\phi)$, that depends on this phase shift

$$\mathbf{d}(\phi) = \begin{bmatrix} 1 & e^{-j\phi} & \dots & e^{-j(M-1)\cdot\phi} \end{bmatrix}^H \tag{3.10}$$

we can then express the beamformer output using (3.8) and (3.10)

$$\hat{s}[k] = y_1[k] \cdot \mathbf{w}^H \mathbf{d}(\phi) \tag{3.11}$$

where we see that the steering vector controls the direction and frequency of interest for the beamformer, since the phase shift depends on the frequency and the direction of imping signal, thus $\phi = \omega \cdot \tau_\theta = \omega \cdot \frac{d}{c}\cos(\theta)$. The beamformer response is therefore highly dependent on the frequency of the incoming signal, thus the beamformer is only optimal for a narrowband signal.

Since speech is a broadband signal, we wish to extend the beamformer to handle broadband signals. This can typically be done by sampling the incoming signal in both space and time by attaching a delay line to each microphone, as proposed in [89] and [38]. In Figure 3.4, a broadband beamformer is depicted where an impinging signal is sampled with M microphones and L samples per microphone. The incoming signal, in each microphone, is delayed $L \cdot T_s$ seconds from the input to the output, where T_s is the discrete sampling time. At any time instant, the impinging signal is thereby sampled at $M \cdot L = N$ points. The output $\hat{s}[k]$ of this beamformer can be expressed

$$\hat{s}[k] = \sum_{m=1}^{M} \sum_{l=1}^{L} w_{l,m} \cdot y_m[k-l] \tag{3.12}$$

where $w_{l,m}$ is the filter coefficient in the lth filter coefficient at the mth microphone. Now by defining a stacked weight vector $\mathbf{w}_{\text{stacked}}$

$$\mathbf{w}_{\text{stacked}} = \begin{bmatrix} \mathbf{w}_1^H & \mathbf{w}_2^H & \dots & \mathbf{w}_M^H \end{bmatrix}^H \tag{3.13a}$$

$$\mathbf{w}_m[k] = \begin{bmatrix} w_m[k] & w_m[k-T_s] & \dots & w_m[k-(L-1)T_s] \end{bmatrix}^H \tag{3.13b}$$

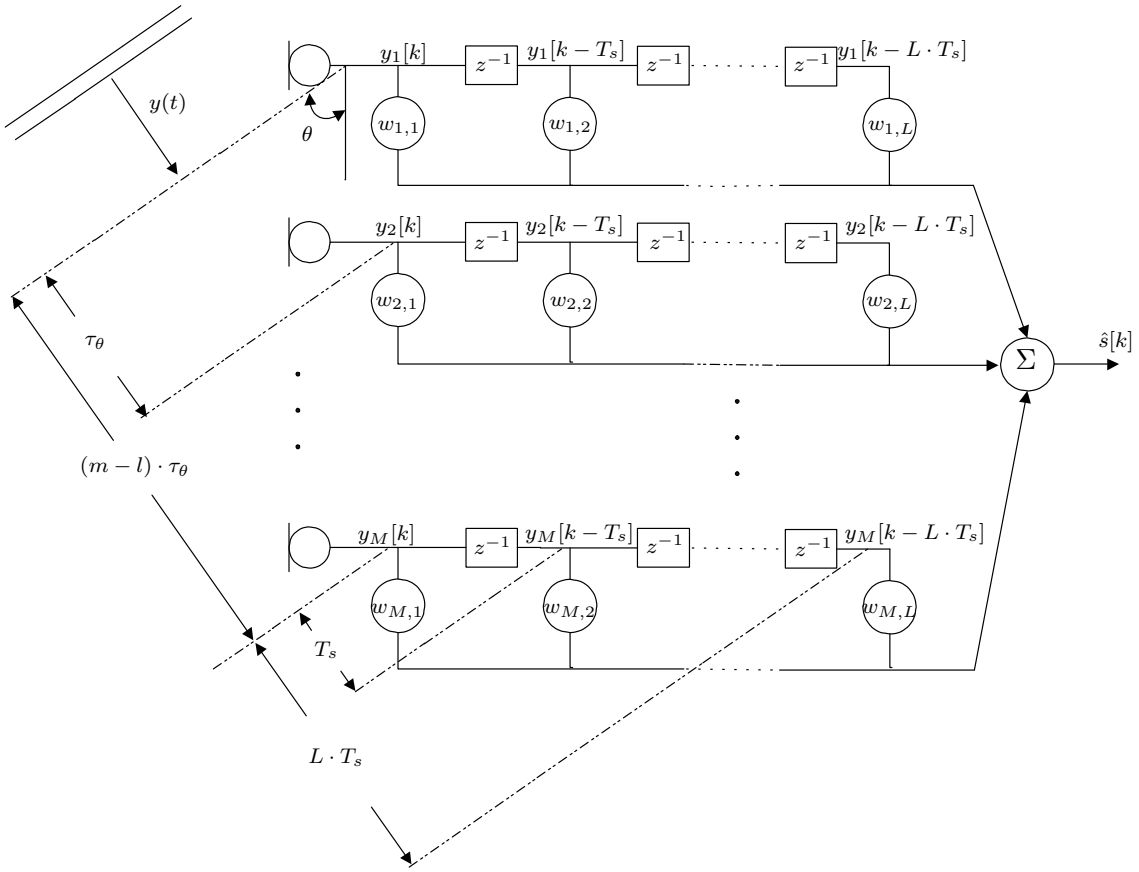


Figure 3.4: Illustration of a broadband beamformer.

thereby

$$\mathbf{w}_{\text{stacked}} = \big[\quad \big| w_{1,1} \quad w_{2,1} \quad \dots \quad w_{L,1} \big| \quad \big| w_{1,2} \quad w_{2,2} \quad \dots \quad w_{L,2} \big| \quad \dots \quad \big| w_{1,M} \quad w_{2,M} \quad \dots \quad w_{L,M} \big| \quad \big]^H \tag{3.13c}$$

and by defining a stacked input vector $\mathbf{y}_{\text{stacked}}[k]$ as

$$\mathbf{y}_{\text{stacked}}[k] = [\ \mathbf{y}_1^H[k] \quad \mathbf{y}_2^H[k] \quad \dots \quad \mathbf{y}_M^H[k]\ ]^H \tag{3.14a}$$

$$\mathbf{y}_m[k] = \big[\ y_m[k] \quad y_m[k-T_s] \quad \dots \quad y_m[k-(L-1)\cdot T_s]\big]^H \tag{3.14b}$$

thus

$$\mathbf{y}_{\text{stacked}} = [\big| y_{1,1} \quad y_{2,1} \quad \dots \quad y_{L,1} \big| \quad \big| y_{1,2} \quad y_{2,2} \quad \dots \quad y_{L,2} \big| \quad \dots \quad \big| y_{1,M} \quad y_{2,M} \quad \dots \quad y_{L,M} \big|\ ]^H \tag{3.14c}$$

we can express the beamformer output using the stacked vectors as

$$\hat{s}[k] = \mathbf{w}_{\text{stacked}}^H \mathbf{y}_{\text{stacked}}[k] \tag{3.15}$$

Again assume that the impinging signal is a complex sine wave, and for convenience let the phase be zero at the first microphone, we thereby have that $y_1[k] = e^{j\omega k}$ and for $y_m[k]$ we have from Equation (3.9a)

$$y_m[k] = e^{j\omega(k-(m-1)\cdot\tau_\theta)} \tag{3.16}$$

then by extending to $y_n[k]$, where $y_n[k]$ is the lth sample at the mth microphone, we have

$$\begin{aligned} y_n[k] &= y_m[k-(l-1)\cdot T_s] && \text{(3.17a)} \\ &= e^{j\omega(k-(m-1)\cdot\tau_\theta-(l-1)\cdot T_s)} && \text{(3.17b)} \\ &= e^{j\omega k}\cdot e^{-j\omega((m-1)\cdot\tau_\theta+(l-1)\cdot T_s)} && \text{(3.17c)} \\ &= e^{j\omega k}\cdot e^{-j\psi_k(\theta,\omega)} && \text{(3.17d)} \end{aligned}$$

where $\psi_n(\theta,\omega)$ is the phase shift from the first microphone at the first delay tap to the mth microphone at the lth delay tap, thus

$$\psi_n(\theta,\omega) = \omega((m-1)\tau_\theta + (l-1)T_s)$$

$\psi_n(\theta,\omega)$ depends on the time delay $\tau_m(\theta)$, due to propagation from the first to the mth microphone, and depends on the time delay $l\cdot T_s$, due to l tap delays.

We can now express the beamformer output by substituting into Equation (3.12)

$$\hat{s}[k] = \sum_{m=1}^{M}\sum_{l=1}^{L} w_{l,m}\cdot e^{j\omega k}\cdot e^{-j\psi_n(\theta,\omega)} \tag{3.18}$$

and by redefining the steering vector

$$\mathbf{d}(\theta,\omega) = \begin{bmatrix} 1 & e^{-j\psi_2(\theta,\omega)} & e^{-j\psi_3(\theta,\omega)} & \dots & e^{-j\psi_N(\theta,\omega)} \end{bmatrix}^H \tag{3.19}$$

we can write

$$\hat{s}[k] = y_1[k]\cdot \mathbf{w}_{\text{stacked}}^H \mathbf{d}(\theta,\omega) \tag{3.20}$$

The beam pattern is thereby determined by both the filter coefficients $\mathbf{w}_{\text{stacked}}$ and the steering vector $\mathbf{d}(\theta,\omega)$ which now both affect temporal and spatial response of the beamformer.

The filter coefficient can be chosen in many ways, resulting in different beamformer characteristics. A well studied beamformer is the linearly constrained minimum variance (LCMV) beamformer, where the weight factors are chosen such that they minimises the output variance of the beamformer, but constrained so the signal from the direction of interest is passed with specified gain and phase.

3.1.3 Linearly Constrained Minimum Variance-Beamforming

The primary motivation for deriving the LCMV beamformer, is, however, that we from this derivation elegantly can formulate the generalised sidelobe canceller, the GSC, beamformer, which is better suited for an adaptive implementation.

The LCMV beamformer minimises the output variance of the beamformer, with the direction of interest constrained so this passes with specified gain and phase. This can mathematically be formulated as

$$\min\left(E\{\hat{s}^2\}\right) = \min\left(E\{\mathbf{w}_{\text{stacked}}^H \mathbf{y}_{\text{stacked}} \mathbf{y}_{\text{stacked}}^H \mathbf{w}_{\text{stacked}}\}\right) \min\left(\mathbf{w}_{\text{stacked}}^H \mathbf{R}_{yy} \mathbf{w}_{\text{stacked}}\right) \tag{3.21}$$

subject to the constraint

$$\mathbf{w}_{\text{stacked}}^H \mathbf{d}(\theta,\omega) = g \tag{3.22}$$

where $\mathbf{R}_{yy}$ is the correlation matrix, $E\{\mathbf{y}\mathbf{y}^H\} = \mathbf{R}_{yy}$, and g is the gain in the direction of interest. This optimisation problem can be solved using Lagrange multipliers yielding the following optimum filter

$$\mathbf{w}_{\text{stacked}} = \frac{\mathbf{R}_{yy}^{-1}\mathbf{d}(\theta,\omega)g}{\mathbf{d}(\theta,\omega)^H \mathbf{R}_{yy}^{-1}\mathbf{d}(\theta,\omega)} \tag{3.23}$$

The LCMV beamformer is easily extended to multiple constraints by expanding the steering vector $\mathbf{d}(\theta, \omega)$ to contain J directions

$$\mathbf{C} = \begin{bmatrix} | & | & & | \\ \mathbf{d}_1(\theta,\omega) & \mathbf{d}_2(\theta,\omega) & ... & \mathbf{d}_J(\theta,\omega) \\ | & | & & | \end{bmatrix} \tag{3.24}$$

where $\mathbf{C}$ is an $N \times J$ matrix with rank J and the gain g is extended to a vector, $\mathbf{g}$. For example, if an interfering source location is known at direction θ_2 then it may be desirable to force zero gain in that direction, and θ_1 is then the direction of interest. The multiple constrains would then be

$$\mathbf{C}^H \mathbf{w}_{\text{stacked}} = \mathbf{g} \tag{3.25}$$

$$\begin{bmatrix} \mathbf{d}_1^H(\theta,\omega) \\ \mathbf{d}_2^H(\theta,\omega) \end{bmatrix} \mathbf{w}_{\text{stacked}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.26}$$

and the filter is then given as

$$\mathbf{w}_{\text{stacked}} = {\mathbf{R}_{\mathbf{yy}}}^{-1} \mathbf{C} (\mathbf{C}^H {\mathbf{R}_{\mathbf{yy}}}^{-1} \mathbf{C})^{-1} \mathbf{g} \tag{3.27}$$

we thereby obtain the LCMV beamformer for the specified beam pattern. If the direction of interest is constraint to a distortionless response as the only constraint,the LCMV is often referred the minimum-variance distortionless response, MVDR, beamformer, as described in [39].

We have now derived the formulation of the LCMV beamformer, this derivation can, however, lead to an algorithm more suited for adaptive filtering, which will be the subject for the following section.

3.1.4 The Generalised Sidelobe Canceller

An alternative formulation of the LCMV-beamformer can be obtained with the generalised sidelobe canceller, GSC. Basically the GSC is a method for changing the constrained minimisation problem into an unconstrained form [89], thereby obtaining a better suited method for an adaptive implementation.

We derive the GSC using the approach in [38] where the GSC is derived from the minimisation problem of the LCMV-beamformer. The idea is to decompose the filter coefficients, $\mathbf{w}_{\text{stacked}}$ in the following $\mathbf{w}_s$, into a constrained part, fulfilling the constraints, and an unconstrained part, that minimises the variance of the output.

The multiple linear constraint was formulated in Equation (3.25), where J independent constraints result in the constraint matrix $\mathbf{C}$ to have dimension $N \times J$ and be of rank J. Recall that $ML = N$. If $J < N$ the problem is underdetermined and will have infinitely many solutions. Using an orthogonal projection matrix, $\mathbf{P}_c$, that projects the $N \times 1$ $\mathbf{w}_s$-vector onto the row space of $\mathbf{C}^H$ and where $(\mathbf{I} - \mathbf{P}_c)$ is the orthogonal complement projection matrix that project $\mathbf{w}_s$ onto the nullspace of $\mathbf{C}^H$. The filter coefficients $\mathbf{w}_s$ can then be decomposed into two projections

$$\mathbf{w}_s = \mathbf{P}_c \mathbf{w} + (\mathbf{I} - \mathbf{P}_c)\mathbf{w} \tag{3.28a}$$

where we will use the notation

$$\mathbf{w}_q = \mathbf{P}_c \mathbf{w}_s \tag{3.28b}$$

Substituting this notation into Equation (3.28a) we can write

$$\mathbf{w}_s = \mathbf{w}_q + (\mathbf{I} - \mathbf{P}_c)\mathbf{w}_s \tag{3.28c}$$

The projection matrix $\mathbf{P_c}$ is given as

$$\mathbf{P}_c = \mathbf{C}(\mathbf{C}^H \mathbf{C})^{-1} \mathbf{C}^H \tag{3.29}$$

where $\mathbf{P}_c^H \mathbf{P}_c = \mathbf{P}_c$. By using the SVD on the constraint matrix $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ the projection matrix $\mathbf{P}_c$ can be expressed as

$$\begin{aligned}
\mathbf{P}_c &= \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H && \text{(3.30a)}\\
&= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \underbrace{(\mathbf{V}\mathbf{\Sigma}\mathbf{U}^H\mathbf{U}\mathbf{\Sigma}\mathbf{V}^H)^{-1}}_{\mathbf{V}^{-H}\mathbf{\Sigma}^{-2}\mathbf{V}^{-1}=\mathbf{V}\mathbf{\Sigma}^{-2}\mathbf{V}^H} \mathbf{V}\mathbf{\Sigma}\mathbf{U}^H && \text{(3.30b)}\\
&= \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H\mathbf{V}\mathbf{\Sigma}^{-2}\mathbf{V}^H\mathbf{V}\mathbf{\Sigma}\mathbf{U}^H && \text{(3.30c)}\\
&= \mathbf{U}\mathbf{U}^H && \text{(3.30d)}
\end{aligned}$$

Using this notation the nullspace projection $\mathbf{I} - \mathbf{P}_c$ can be expressed as

$$\mathbf{I} - \mathbf{P}_c = \mathbf{I} - \mathbf{U}\mathbf{U}^H \tag{3.31}$$

where $\mathbf{I} - \mathbf{U}\mathbf{U}^H = \tilde{\mathbf{U}}\tilde{\mathbf{U}}^H$ is the orthogonal complement matrix to $\mathbf{U}\mathbf{U}^H$. Thereby Equation (3.28c) can be written as

$$\mathbf{w}_s = \mathbf{w}_q + \tilde{\mathbf{U}}\tilde{\mathbf{U}}^H\mathbf{w}_s \tag{3.32}$$

Using the notation $-\mathbf{C}_a = \tilde{\mathbf{U}}$ and $\mathbf{w}_a = \tilde{\mathbf{U}}^H\mathbf{w}_s$. Where $\mathbf{C}_a$ is an $N \times (N-J)$ orthogonal complement matrix to $\mathbf{C}$ and $\mathbf{w}_a$ is an $(N-J)\times 1$ vector. Then finally Equation (3.28c) can be reformulated as

$$\mathbf{w}_s = \mathbf{w}_q - \mathbf{C}_a\mathbf{w}_a \tag{3.33}$$

The multiple linear constraint was formulated in Equation (3.25) as

$$\mathbf{C}^H\mathbf{w}_s = \mathbf{g} \tag{3.34}$$

then by substituting $\mathbf{w}_s$ with Equation (3.33) the multiple linear constraint can be expressed as

$$\begin{aligned}
\mathbf{C}^H(\mathbf{w}_q - \mathbf{C}_a\mathbf{w}_a) &= \mathbf{g} && \text{(3.35)}\\
\mathbf{C}^H\mathbf{w}_q - \mathbf{C}^H\mathbf{C}_a\mathbf{w}_a &= \mathbf{g} && \text{(3.36)}\\
\mathbf{C}^H\mathbf{w}_q &= \mathbf{g} && \text{(3.37)}
\end{aligned}$$

since $\mathbf{C}^H$ and $\mathbf{C}_a$ are complement matrices $\mathbf{C}^H\mathbf{C}_a = 0$ which shows that $\mathbf{w}_a$ is not dependent on the constraints in $\mathbf{C}$ and is therefore projected to the nullspace. Whereas $\mathbf{w}_q$ is the vector that satisfy $\mathbf{C}$ which is projected to the rowspace of $\mathbf{C}^H$. This decomposition of $\mathbf{w_s}$ is illustrated in Figure 3.5.

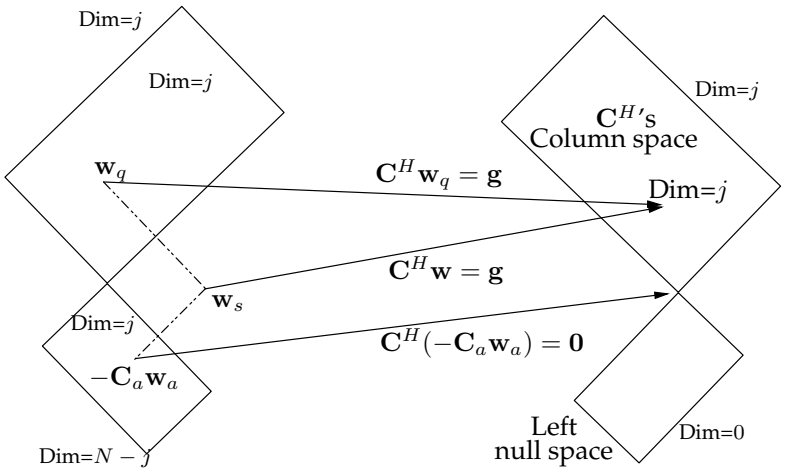


Figure 3.5: A graphical interpretation of the projection of the $\mathbf{w}$ into $\mathbf{C}^H$'s rowspace and nullspace.

The minimisation problem described in Equation (3.21) for the LCMV beamformer, can now be expressed for the GSC using the decomposed filter coefficients from (3.33) giving

$$\min\left(\mathbf{w}_s^H\mathbf{R}_{\mathbf{yy}}\mathbf{w}_s\right) = \min\left((\mathbf{w}_q - \mathbf{C}_a\mathbf{w}_a)\mathbf{R}_{\mathbf{yy}}(\mathbf{w}_q - \mathbf{C}_a\mathbf{w}_a)\right) \tag{3.38}$$

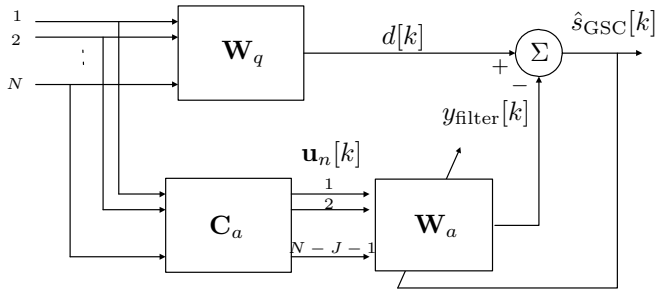


Figure 3.6: Illustration of a GSC.

A graphical interpretation of the decomposition of the filter coefficient $\mathbf{w}_s$ can be seen in Figure 3.6 where a block diagram of the GSC beamformer is depicted. The block diagram can be seen as a fixed beamformer $\mathbf{w}_q$, with adaptive noise cancellation, $\mathbf{C}_a$ and $\mathbf{w}_a$. Where $\mathbf{C}_a$ is referred as the *Blocking Matrix* since it tries to block the signal of interest and thereby providing a noise reference. The output from the GSC beamformer can be found as

$$\hat{s}_{\,\mathrm{GSC}} = \mathbf{w}_s^H \mathbf{y} \tag{3.39a}$$
$$= (\mathbf{w}_q - \mathbf{C}_a \mathbf{w}_a)^H \mathbf{y} \tag{3.39b}$$
$$= \mathbf{w}_q^H \mathbf{y} - \mathbf{w}_a^H \mathbf{C}_a^H \mathbf{y} \tag{3.39c}$$

where $\mathbf{y}$ is the observations, $\mathbf{w}_q^H \mathbf{y}$ is the speech reference and $\mathbf{w}_a^H \mathbf{C}_a^H \mathbf{y}$ is the noise reference. We can obtain the classical adaptive formulation by defining

$$d_n = \mathbf{w}_q^H \mathbf{y} \tag{3.40}$$
$$\mathbf{u}_n = \mathbf{C}_a^H \mathbf{y} \tag{3.41}$$

we can now rewrite Equation (3.39c):

$$\hat{s}_{\,\mathrm{GSC}} = d_n - \mathbf{w}_a^H \mathbf{u}_n \tag{3.42}$$

This formulation is well suited for an adaptive filtering approach. In a classical adaptive filter, the filter output would be:

$$y[k] = \mathbf{w}^H[k]\mathbf{u}[k] \tag{3.43}$$

in our case this would then be:

$$y_{\,\mathrm{filter}}[k] = \mathbf{w}_a^H[k]\mathbf{u}_n[k] \tag{3.44}$$

The error signal, $e[k]$, is obtained by subtracting the filter output, $y_{\,\mathrm{filter}}[k]$, from the desired signal, $d[k]$

$$e[k] = d_n - \mathbf{w}_a^H \mathbf{u}_n \tag{3.45}$$

However in our case the error signal, $e[k]$, is the estimated speech signal, $\hat{s}_{\,\mathrm{GSC}}[k]$. In the classical adaptive filter we now used the error signal, speech estimate, to update the filter,

$$\mathbf{w}_a[k+1] = \mathbf{w}_a[k] + \mu \mathbf{u}_n[k] \hat{s}_{\,\mathrm{GSC}}[k] \tag{3.46}$$

thereby obtaining the Least-Mean-Square adaptive filter. Where μ is the the step size, which can be normalized with the input as:

$$\tilde{\mu}[k] = \frac{\mu}{\| \mathbf{u}_n[k] \|^2} \tag{3.47}$$

which leads to a time-varying step-size parameter, $\tilde{\mu}[k]$, thus the Normalised-LMS adaptive filter. The basic theory for spatial sampling has been presented. First a simple delay and sum beamformer was described, then the theory for the LCMV-beamformer was introduced in order to derive the adaptive beamformer in form of the GSC. We will now test how these spatial filtering techniques can be used in speech enhancement for noise reduction.

3.1.5 Experimental Results

In the previous section we examined the fundamental theory for spatial filtering, we will now investigate the ability of these spatial filtering techniques to enhance degraded speech signals. We will primarily focus on the GSC beamformer, but the performance of DS-beamformer will shortly be reviewed.

We have chosen two different spatial test situations. In the first test situation, the speaker is located in front of the listener, and a noise source is placed on the left side (90°) of the listener, for both sources, only the direct path from the source to the two microphones exits. This corresponds to an anechoic chamber, where no reflections of the sources are present. We have generated this scenario by delaying the the noise signal recorded in the right microphone one sample compared to the noise signal in the left microphone. The desired signal is not delayed between the two microphones, thus the speaker is located in front of the listener. We will in this thesis not consider the shadowing-effect caused by placing the two microphones behind the ear, we refer to [36] for further information. Delaying the noise source one sample in the generated test signal corresponds to having a spatial sampling time equal the discrete sampling time, $\tau_\theta = T_s$. Then by setting the distance between the microphones to $5\,\mathrm{cm}$, we can compute the angle the noise source

$$\tau_\theta = T_s \Rightarrow \theta = \arccos\left(\frac{T_s \cdot c}{d}\right) = 32^\mathrm{o} \tag{3.48}$$

This test signal is not a realistic scenario, since a room always will reflect the signal to some extent. We will, however, use this test signal to test the different beamformers since this scenario constitute the ideal situation, and therefore indicates best possible results obtainable with the method under test.

In the other test signal, we use the image method to generate a room impulse response. This test setup is further described in Section 1.4, where the room, and spectrograms of speech signals and observations recorded in this room, are depicted. We will use this room in order to test realistic scenarios and see how this influence the performance.

Using this room corresponds to having the speaker placed approximately $2\,\mathrm{m}$ in front of the listener 14 degrees on the left-hand side, and the noise source placed approximately $2\,\mathrm{m}$ in front of the listener, but 14 degrees on the right-hand side.

The evaluation of the estimated speech signal, consists of three different assessment techniques, broadband-SNR, segmental-weighted-SNR and weighted spectral slope measure (WSSM), described in section 1.3.2. These were also used in the previous chapter for assessment of the single-channel techniques.

The DS beamformer consists of a microphone array of two microphones, each attached with a delay, in order to time align the impinging signals. The delay is found by cross-correlating the two signals and finding the time lag that corresponds to the maximum correlation.

The GSC was original proposed by Griffiths and Jim in [34], therefore the GSC is also referred as the Griffiths-Jim beamformer, GJBF. The fixed beamformer in the GSC originally consisted of a simple delay-and-sum beamformer, and the blocking matrix of a simple delay-and-subtract beamformer, often referred to as the DS-GSC. Thereby the fixed beamformer forms a beam in the look direction and the blocking matrix forms a null in the look direction. However, by using an ANC stage, the GSC is capable of forming nulls in in the direction of the interference. We extended the LCMV to the GSC since, adaptive beamformers generally achieve better interference suppression than fixed beamformers [44].

We have implemented the GSC beamformer with two microphones, and using a priori knowledge, thus the steering vector is formed using the known location of the speaker (see `gsc_with_steering.m` in Appendix D). Thereby the optimal fixed beamformer, $\mathbf{w}_q$, and blocking matrix, $\mathbf{C}_a$, can be formed. We have expanded the DS-GSC, by using the super directive, or filter-and-sum beamformer. We have however not used the broadband approach since the ANC-stages is capable of extracting the target signal and suppress the interference source, although the directivity is frequency dependent [44]. We have implemented the adaptive filter in the ANC-stage

using the NLMS adaptive filter, with a filter length of 256 filter taps as default, since this is found preferable, as concluded by testing the filter length. The filter is only updated in noise-only periods, in the literature often referred to as *mode control*. Using mode control the filter only updates when no speech is present in the noise estimate. The noise only periods are found using an ideal VAD. The test signal is concatenated with $3\,\mathrm{s}$ noise, in order to avoid initial ringing effects when the adaptive filter initialises. These $3\,\mathrm{s}$ are not included in the evaluation of the GSC beamformer.

The DS and GSC Beamformer

In the first simulation we will test the DS (see `delay_sum_ideal.m` in Appendix D) and GSC beamformer on the standard test signal a female voice, degraded with white noise at $5\,\mathrm{dB}$ SNR, for spectrograms and time plots of the clean speech signal and the observation we refer to Section 1.4, where these are depicted. We will use the scenario where only the direct path are used, thus no reverberation. The signal of interest is impinging at $\theta = 90^{\mathrm{o}}$, and the noise source is located at what corresponds to $\theta = 32^{\mathrm{o}}$

In Table 3.1 the results for the DS and GSC beamformer are listed. The DS beamformer obtains a broadband SNR improvement of $3\,\mathrm{dB}$, which is expected, if the coherent addition is successful. We can then conclude that the time-alignment is done correctly, which of cause is easy since this is already the case. The DS beamformer, introduces a small speech distortion, thus the WSSM is increased a few points.

The GSC obtains an improvement in broadband SNR of $13.7\,\mathrm{dB}$ in the ideal scenario with the speaker location known. Thereby obtaining good speech and noise reference to the ANC stage. Furthermore this noise reduction is not at the expense of increased speech distortion. The WSSM is decreased with 13.2 points.

Noise estimate	$\mathrm{SNR}[\,\mathrm{dB}]$	$\Delta\mathrm{SNR}[\,\mathrm{dB}]$	$\mathrm{SNR}_{\mathrm{SEG}}[\,\mathrm{dB}]$	$\mathrm{WSSM}[\cdot]$
Observation	5.0		-0.4	44.4
DS beamformer	8.0	3.0	0.8	46.6
GSC beamformer	18.7	13.7	7.6	31.2

Table 3.1: The results are obtained using the DS and GSC beamformer tested with $5\,\mathrm{dB}$ additive white noise, with only the direct path as room response, which corresponds to an anechoic chamber. The location of the speaker is known in advance.

In Figure 3.7 the spectrograms of the two speech estimates for the DS and GSC beamformer are depicted. The upper spectrogram is the speech estimate obtained using the DS beamformer. It removes $3\,\mathrm{dB}$ additive noise which also was expected, since the test scenario is ideal. The speech energy seems well preserved, which corresponds to the nearly unchanged WSSM. The noise is not spatial white, thus the noise is not removed in the entire frequency range. The method becomes frequency dependent and at $4\,\mathrm{kHz}$ the two noise signals cancels, because one delay corresponds to π, thus noise is reduced at this frequency.

In the lower part of Figure 3.7, the spectrogram for the speech estimate of the GSC is depicted. The noise is removed, except for very low frequencies, which when played is notable as some distant rumbling noise. The noise is below $300\,\mathrm{Hz}$, thus it does not decrease the intelligibility. The beamformer suffers from reducing low frequency noise, however, this is a general tendency in beam patterns that the direction of interest can not be narrow for very low frequencies.

Comparing this spectrogram with the spectrogram of the original speech signal, see Figure 1.8 on page 15, it is seen that the speech energy is well preserved, no speech distortion is introduced. This is also supported by the decrease in WSSM, which indicates a significant improvement in the output speech compared to the observed signal.

A filter length of 256 filter taps has been used in the ANC-stage. This choice is motivated by the test in the following section, where the filter length is tested.

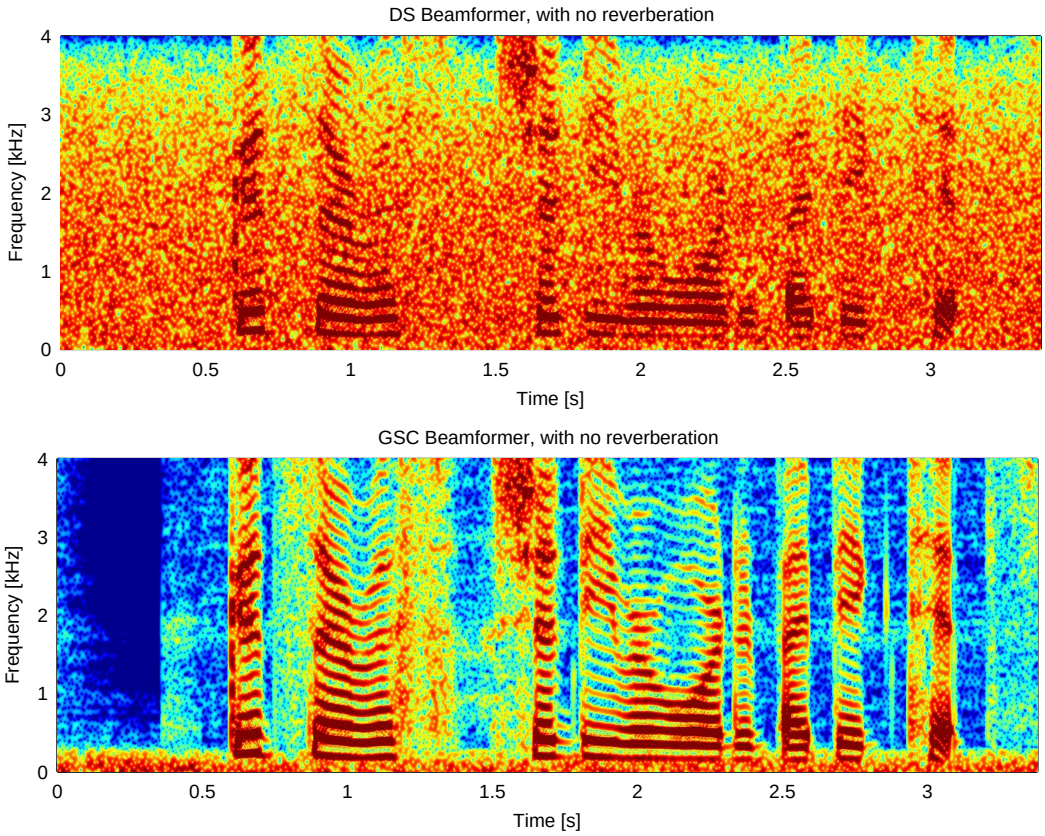


Figure 3.7: The upper spectrogram showing the estimated speech signal obtained using the DS-beamformer. The lower spectrogram showing the estimated speech signal obtained using the GSC beamformer. Both methods are tested using the standard test signal, with direct paths as room responses.

Test of Filter Length in the GSC Beamformer

The GSC is implemented using an NLMS filter with mode control. The length of the filter will in this section be tested. Again the standard test signal, a female speaker degraded with additive noise at a broadband SNR of $5\,\mathrm{dB}$ has been used. In addition to the standard setup we have used both white and pink noise. The GSC is tested using filter lengths of 32 to 1000 taps.

In Figure 3.8 the results from these simulations are depicted. For pink noise we see that the best broadband SNR improvement is obtained using a filter length of around 400 taps. For the speech distortion it seems that the highest values for pink noise is obtained for a filter length between 200 and 300 taps.

For white noise the most noise reduction is obtained using a filter length of 100 filter taps, whereas the speech quality shows the same tendencies as pink noise, the best WSSM improvement is obtained with a filter length between 200 to 300 taps. The size of the filter length is related to the stationarity of speech. The NLMS is an adaptive stochastic approach of the Wiener filter, which is build on the assumption that the signals to be filtered are stationary. Both pink and white noise is stationary, the speech signal, however, can only be assumed stationary for frames of 20 to $40\,\mathrm{ms}$, as described in Section 1.2.2 on page 4. This corresponds to a filter length between 160 and 320 filter taps. Thus, as the stationarity criteria is not fulfilled, the performance decreases.

The GSC is capable of removing significantly more white noise than pink noise. This is a consequence of the GSC's incapability of removing low frequency noise, which was seen from Figure 3.8. Pink noise has a high concentration of energy in the low frequencies, thus less noise energy is removed from the observation. The noise energy is still below $300\,\mathrm{Hz}$ and has therefore not influenced the speech distortion. This can be seen in Figure 3.9, where the spectrogram of the speech estimate from the pink noise observation, female, using a filter length of 256 samples

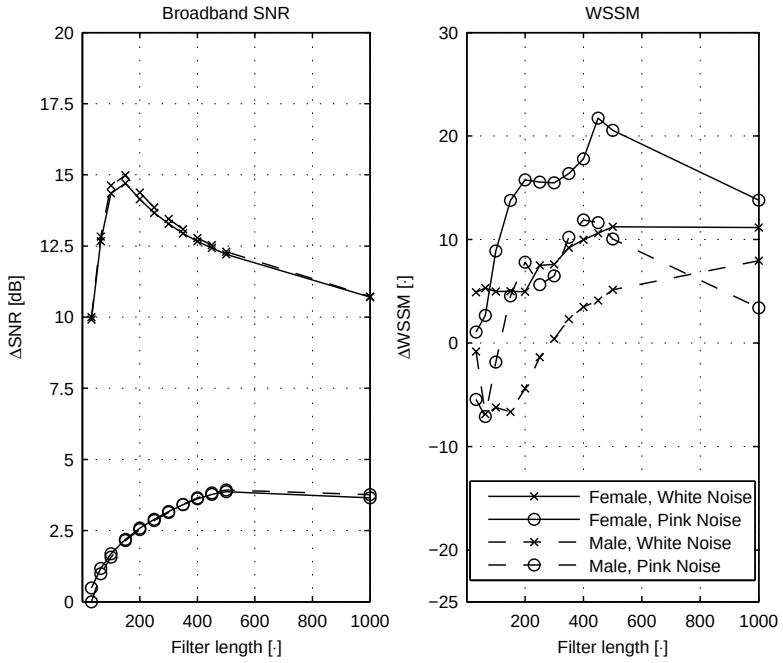


Figure 3.8: Results obtained using the GSC-beamformer. Test of the filter length of the NLMS, with only the direct path as room response. The observation is the standard test signal.

is depicted. By comparing this figure with the speech estimate from the white noise observation in Figure 3.7 we can see that they are both nicely recovered, in spite of the significant difference in SNR improvement. Informal listening tests among the group members identify that the SNR improvement is hardly audible.

In an overall evaluation of the filter length it seems that a reasonable choice of filter length is between 200 and 300 taps. This is a trade-off between noise reduction for white and pink noise obtained WSSM score.

In this section a significant difference with white and pink noise was observed in broadband noise reduction. In the next section this is investigated further.

Test of the GSC using Other Types of Noise

The GSC will now be tested for both white and pink noise of noise levels at $-5, 0, 5$ and $10\,\mathrm{dB}$ SNR. Two different sentences have been deployed; a male and a female speaker.

From Figure 3.10 we see that the noise reduction depends on the noise type only. The GSC is capable of improving the broadband SNR more than $13\,\mathrm{dB}$ for white noise and $3\,\mathrm{dB}$ for pink noise, independent of noise levels and speaker.

For the WSSM, the improvement seems to dependent on noise type, noise level, and speaker. It is noticeable that as the input SNR of the observation increases the WSSM improvement decreases. The tendency seems general, however more distinct for the female speaker. The female speaker obtains higher WSSM improvement than the male speaker for both pink and white noise, and for all noise levels.

Generally we see that the GSC can reduce the noise from the observation and improve the speech quality. However there are a significant difference in the noise reduction for pink and white noise, and this is general for all noise levels. We saw, however, in the previous section that this is caused by the lacking ability of the GSC to remove low frequency noise. Since pink noise contains more energy in the low frequency bands, this will naturally lead to less improvement for pink noise. In the lower part of Figure 3.7 and in Figure 3.9 the speech estimates from observations degraded with white and pink noise, respectively, are depicted. By comparing the two figures, it is seen, that the noise reduction for the speech estimate recovered from pink noise is located in

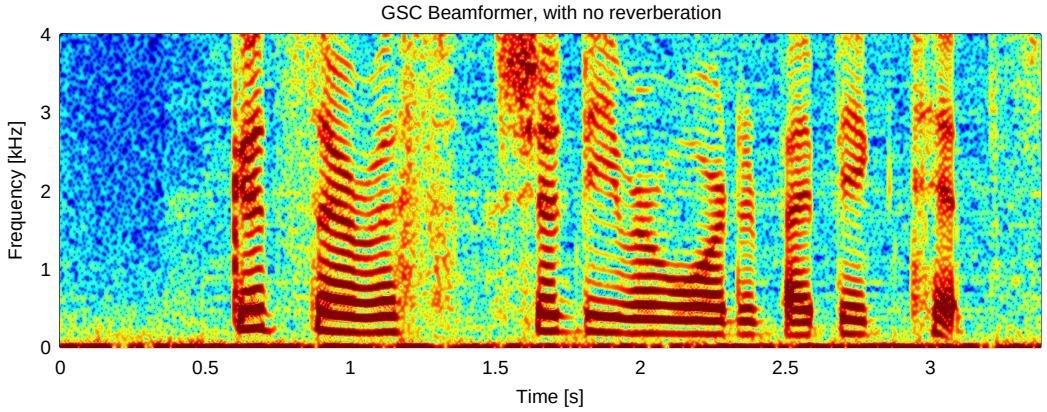


Figure 3.9: Spectrogram of the speech estimate for the GSC, using a filter length of 256 taps. The test signal is the standard setup, using only direct path. However the female voice is degraded with $5\,\mathrm{dB}$ *pink* additive noise. By comparing this speech estimate with the one using white noise, Figure 3.7, we can see that they are both nicely recovered. Informal listening tests among the group members identify that the SNR improvement is hardly audible.

the frequency bands below $300\,\mathrm{Hz}$.

The GSC has been tested under ideal conditions, in the following section the GSC is tested in a reverberated setup.

Test of the GSC Beamformer in Reverberation

In the simulation performed so far, the room impulse response has only contained a direct path, corresponding to a test setup in an anechoic chamber. This is however not a realistic setup, therefore reverberation is added in this test.

We use the scenario depicted in Section 1.4 on page 16, where the speaker is placed approximately $2\,\mathrm{m}$ in front of the listener 14 degrees on the left-hand side, and the noise source approximately $2\,\mathrm{m}$ in front of the listener, but 14 degrees on the right-hand side.

Noise estimate	SNR[dB]	ΔSNR[dB]	$\mathrm{SNR}_{\mathrm{SEG}}$[dB]	WSSM
Observation with 100 ms reverberation time	3.6		-1.5	58.4
Estimate with 100 ms reverberation time	2.6	-1.0	-1.8	59.4
Observation with 200 ms reverberation time	2.1		-2.1	63.4
Estimate with 200 ms reverberation time	1.2	-0.9	-2.5	67.4
Observation with 300 ms reverberation time	1.3		-2.5	64.2
Estimate with 300 ms reverberation time	0.6	-0.7	-2.8	66.1
Observation with 400 ms reverberation time	0.7		-2.7	64.4
Estimate with 400 ms reverberation time	0.3	-0.4	-2.9	65.1
Observation with 500 ms reverberation time	0.4		-2.8	64.9
Estimate with 500 ms reverberation time	0.0	-0.0	-3.0	64.4
Observation with 600 ms reverberation time	0.1		-3.2	63.6
Estimate with 600 ms reverberation time	-0.4	-0.5	-3.1	65.2

Table 3.2: The results are obtained using the GSC tested with $5\,\mathrm{dB}$ additive white noise using rooms with different reverberation time.

In Table 3.2 the results using different reverberation times are listed. From this we see that when reverberation is present the GSC is not cable of removing noise, nor is the speech distortion

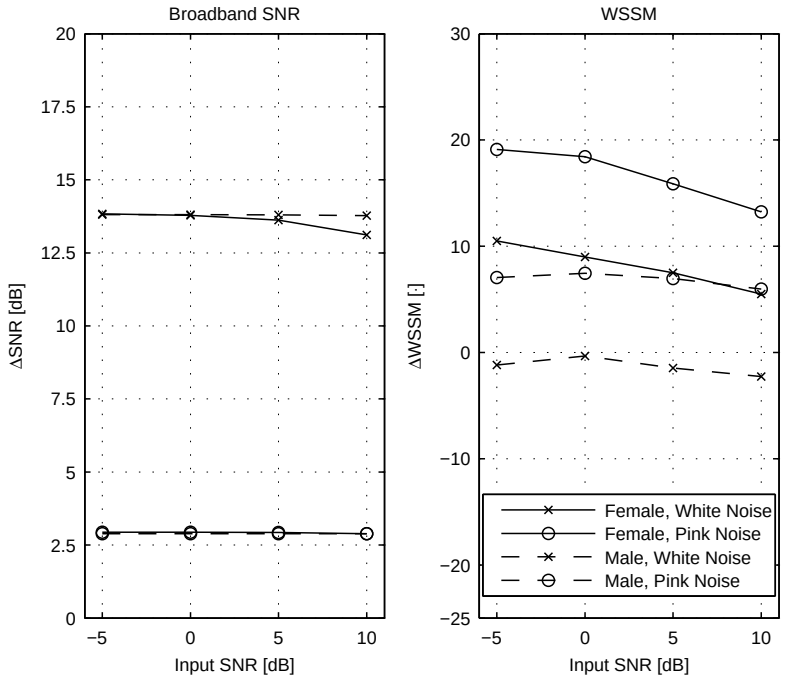


Figure 3.10: Results obtained using the GSC-beamformer, with different noise types. Almost similar performance is obtained between pink and white noise. The difference is dependent of the speaker.

improved.

The blocking matrix can not remove the speech signal from the observation, which provides a bad noise estimate to the ANC-stages. That is, when no reverberation is present the blocking matrix is capable of cancelling out all desired speech signal by cancelling the direction of interest. However, when reverberation is present the speech signal will be reflected in the room, thus impinging to the microphones from other directions than the direction of interest.

This can be confirmed by measuring the blocking capabilities of the blocking matrix. One approach is to measure the correlation between the blocking matrix output and the reverberated speech signal. In Figure 3.11 the normalised cross-correlation between the output from the blocking matrix and the speech signal is depicted. In the right figure no reverberation is used and it shows no correlation, thus the blocking matrix removes the noise from the observation and obtains a solid noise reference. In the left figure a room with $300\,\mathrm{ms}$ reverberation time is used. Here a clear correlation between the two signals is observed, thus blocking out speech signal from the noise reference fails due to reverberation. When reverberated speech is present in the noise estimate and the mode control allows the filter to adapt, the filter will erroneously remove the correlation between the speech and the noise estimate and in the output from the beamformer. This explains the lacking noise reduction in Table 3.2.

3.1.6 Conclusion

We have in this section introduced the fundamental theory of spatial filtering. The starting point was simple delay-and-sum beamforming, which was extended to super directive beamforming. The LCMV beamformer was derived in order to elegantly formulate the generalised sidelobe canceller.

The GSC was capable of removing noise by exploiting spatial information, if no reverberation was present and the direction of interest known. The noise reduction was significantly higher for white noise than pink noise, since the GSC lacks the ability to remove noise energy in the low frequency bands below $300\,\mathrm{Hz}$. This has, however, little influence on the audible results.

The noise reduction is not, as seen for single-channel techniques, obtained at the expense

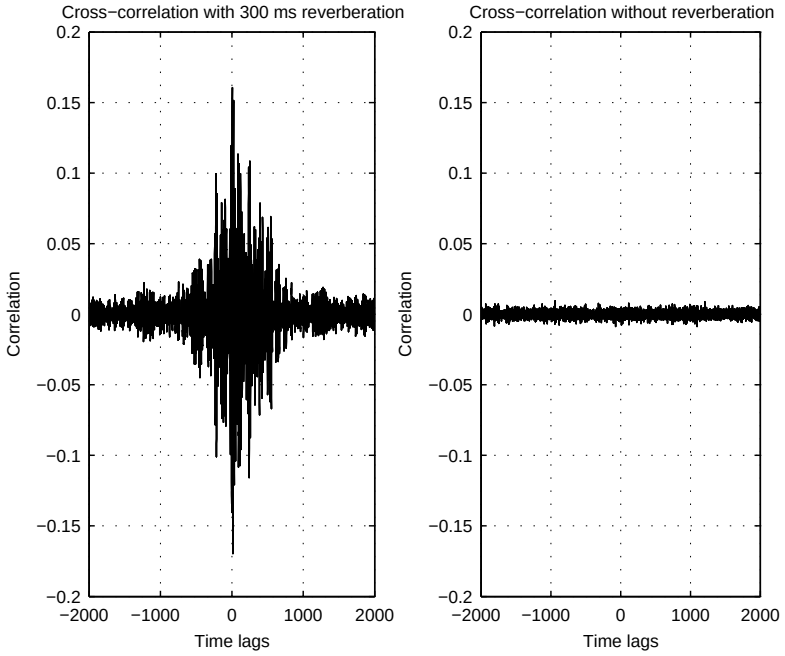


Figure 3.11: Normalised cross-correlation between the output from the blocking matrix and the reverberated speech signal. In the left figure a noise estimate with reverberated speech has been correlated with the output from the beamformer. From this it is evident that these are correlated and thus speech will erroneously be removed in the output. In the right figure, only the direct path is used as the room impulse response. Thereby obtaining a noise reference without correlation to the speech signal.

of speech distortion. The speech distortion is improved considerably compared to the observed signal. By comparing spectrograms of the estimated speech signals from the GSC with the original speech signal it was seen that the speech energy was well preserved.

The GSC was, however, not capable of removing noise when reverberation was present. No noise reduction was obtainable with reverberation, which was caused by lacking blocking abilities. The blocking matrix only removed the signal in the direction of interest, however, when reverberation was present speech leakage caused poor noise reduction and introduced little speech distortion.

Much effort has been done in order to construct robust adaptive beamformers, thus beamformes which not suffer from the observed speech leakage. Among many, the proposal of using leaky adaptive filters (leaky NLMS algorithm [39]) in the blocking matrix and the noise cancellers [44] should be mentioned. A more advanced method consist of using norm-constrained adaptive filters in connection with coefficient-constrained adaptive filters, which has shown good results even for slightly reverberated rooms ($0.3\,\mathrm{s}$) [44]. Yet another promising proposal by Doclo and Moonen in [19] is based on GSVD-based optimal filtering, i.e. a multi channel Wiener filter is used to exploit both temporal and spatial information..

The primary motivation in this section was to gain insight and provide understanding for spatial filtering, in order to proceed to more complex and advanced speech enhancement techniques exploiting spatial information. We will in the next section examine the spatial filter technique proposed by Doclo and Moonen in [19]. This method has shown promising noise reduction results even for reverberated room [18]. Another motivation for investigating this method is that it provides spatial filtering without a priori knowledge of the location of the speaker.

3.2 Multi-Channel Wiener Filtering

Having investigated conventional types of beamformers in the previous section a different approach based on the well-known Wiener-Hopf equations is derived. In the context of this section

the method is referred to as a multi-channel Wiener filter (MCWF) as proposed by Doclo et al. [19], even though as we will see, the method is a combined beamformer and a traditional Wiener filter.

The motivation for this method is that in the previous section, the beamformer was dependent of a priori information, i.e. the steering vector had to be determined before hand. The MCWF, however, is derived in a different way using none *a priori* information whatsoever. Thus there is no need of determining a steering vector and Doclo et al. also show that the MCWF is more robust to reverberated conditions than an adaptive beamformer. Furthermore, Doclo [16] showed a method on how to in cooperate a dereverberation procedure into the MCWF using a frequency domain interpretation.

This section will firstly introduce the concept of the MCWF using a signal subspace approach that can be interpreted as an extension of the method described in Section 2.4 on page 39. Then a time domain solution is provided using the GSVD. The time-domain solution is followed up by simulation results. The frequency domain interpretation of the MCWF is then presented in order to use it in the next section about dereverberation and noise reduction.

The basic idea in the MCWF is to exploit both spatial and temporal information using inter-channel correlations. To present the multi-channel optimum (Wiener) filter, we start by introducing the single-channel optimum filter and extend this to the multi-channel case in an intuitively manner. Consider for a moment a finite single-channel optimum filter, $\mathbf{w}[k]$, which is a length-L FIR filter defined as

$$\mathbf{w}[k] = \begin{bmatrix} w^0[k] & w^1[k] & \dots & w^{L-1}[k] \end{bmatrix}^T \tag{3.49}$$

with time index k. It is a realisable, truncated version of the theoretical filter obtained from solving the Wiener-Hopf equations [39, 81]. Now consider the filter depicted in Figure 3.12, where the single-channel input is defined as

$$\mathbf{y}[k] = \begin{bmatrix} y[k] & y[k-1] & \dots & y[k-L+1] \end{bmatrix}^T \tag{3.50}$$

One way to formulate the optimum (Wiener) filter is by the following discrete-time, matrix-vector formulation of the solution to the well-known Wiener-Hopf equations

$$\mathbf{w}_{\mathrm{WF}} = E\left\{\mathbf{y}[k]\mathbf{y}^T[k]\right\}^{-1} E\left\{\mathbf{y}[k]x[k-\Delta]^T\right\} = \mathbf{R}_{yy}^{-1}\mathbf{r}_{yx} \tag{3.51}$$

where $\mathbf{r}_{yx}$ is the cross-correlation vector. The right-hand side has been formulated using the expectation operator, $E\{\}$, on the input data vector at time instance k, $\mathbf{y}[k]$, and the desired (scalar) response, $x[k-\Delta]$. By including the integer-delay operator, Δ, we have made explicit that we delay the desired response in order to make a non-causal Wiener filter realisable (move to the causal domain). This is easily seen by assuming $\Delta \geq 1$ and inspecting (3.51) and Figure 3.12. The filtering is done by

$$\hat{x}[k] = \mathbf{w}_{\mathrm{WF}}^T\mathbf{y}[k] \tag{3.52}$$

but since we have defined the optimum filter according to a delayed response, the output is non-causal.[1] For a finite-length filter, see (3.49), we can introduce 0 to $L-1$ delays. To distinguish these L different optimum filters with increasing amount of delay/non-causality, we introduce the notation $\mathbf{w}_l[k]$ for a length-L filter with l number of non-causal taps ($0 \leq l \leq L-1$).

We will now introduce the multi-channel Wiener filter by extending concepts from the single-channel optimum filter. According to (3.52), the output of a single-channel filtering can be expressed as the inner product of a data vector and a filter coefficient vector. In order to keep this useful notation, we will in the following make extensive use of a stacked data (and filter) vector notation in order to transform multi-dimensional filtering (actually filtering and beamforming) into the inner product of two stacked vectors.

Now consider the Wiener filter scenario depicted in Figure 3.12. If we denote the number of channels by M, we can define the stacked vector of input data, $\mathbf{y}[k]$, as a concatenation of

[1] It should be noted, that any non-causality in Equation (3.51) can be accounted for in the filtering by introducing a delay, that is, non-causal filtering is implemented by introducing a delay - as is shown in Figure 3.12.

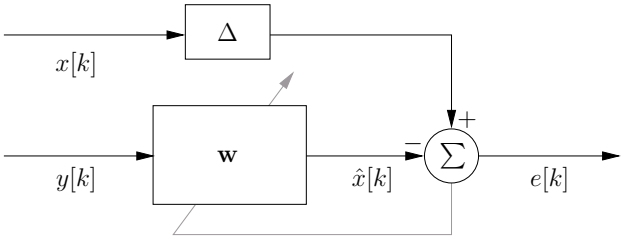


Figure 3.12: The concept of a optimum (Wiener) filter. The Δ-box is inserted to make explicit the number of non-causal taps realised by the optimal finite filter.

M, length-L data vectors, , such that we obtain the same stacked vector as firstly defined in Section 3.1.2.

$$\mathbf{y}[k] = \begin{bmatrix}\mathbf{y}_1^T[k] & \mathbf{y}_2^T[k] & \dots & \mathbf{y}_M^T[k]\end{bmatrix}^T \tag{3.53a}$$

$$\mathbf{y}_m[k] = \begin{bmatrix}y_m[k] & y_m[k-1] & \dots & y_m[k-L+1]\end{bmatrix}^T \tag{3.53b}$$

and the optimal filter with causality l for channel m defined as

$$\mathbf{w}_{m,l}[k] = \begin{bmatrix}w_{m,l}^0[k] & w_{m,l}^1[k] & \dots & w_{m,l}^{L-1}[k]\end{bmatrix}^T \tag{3.54a}$$

$$\mathbf{w}_l[k] = \begin{bmatrix}\mathbf{w}_{1,l}^T[k] & \mathbf{w}_{2,l}^T[k] & \dots & \mathbf{w}_{M,l}^T[k]\end{bmatrix}^T \tag{3.54b}$$

Using these redefinitions, the optimal filter vector can be expanded to matrix notation such that $\mathbf{W}[k]$ is an $N \times N$ $(N = ML)$ matrix with columns corresponding to (3.54b). The actual structure of $\mathbf{W}[k]$, we will return to shortly. As seen in Figure 3.12 the output of filtering the stacked input vector we obtain the estimated output

$$\hat{\mathbf{x}}[k] = \mathbf{W}[k]^T\mathbf{y}[k] \qquad , \hat{\mathbf{x}}[k] \in \mathbb{R}^N \tag{3.55}$$

which we try to optimise as close as possible to the stacked desired response vector

$$\mathbf{x}[k] = \begin{bmatrix}\mathbf{x}_1^T[k] & \mathbf{x}_2^T[k] & \dots & \mathbf{x}_M^T[k]\end{bmatrix}^T \tag{3.56a}$$

$$\mathbf{x}_m[k] = \begin{bmatrix}x_m[k] & x_m[k-1] & \dots & x_m[k-L+1]\end{bmatrix}^T \tag{3.56b}$$

By defining the output error vector, $\mathbf{e}[k] = \mathbf{x}[k] - \hat{\mathbf{x}}[k]$, we can compute the optimum filter by minimising the minimum mean-square error vector

$$\begin{aligned} J_{MSE}\left(\mathbf{W}_{\mathrm{WF}}\right) &= E\left\{\|\mathbf{e}\|_2^2\right\} \\ &= E\left\{\left\|\mathbf{x} - \mathbf{W}^H\mathbf{y}\right\|_2^2\right\} \\ &= E\left\{-\mathbf{x}\mathbf{x}^H - \mathbf{W}^H\mathbf{y}\mathbf{x}^H - \mathbf{x}\mathbf{y}^H\mathbf{W} + \mathbf{W}^H\mathbf{y}\mathbf{y}^H\mathbf{W})\right\} \\ &= \mathbf{R}_{xx} - \mathbf{W}^H\mathbf{y}\mathbf{x}^H - \mathbf{x}\mathbf{y}^H\mathbf{W} - \mathbf{W}^H\mathbf{y}\mathbf{y}^H\mathbf{W} \end{aligned} \tag{3.57}$$

Where we have denoted the optimum (Wiener) filter by $\mathbf{W}_{\mathrm{WF}}$. By setting the derivative of the cost function, $\partial J_{MSE}\left(\mathbf{W}_{\mathrm{WF}}\right)/\partial\,\mathbf{W}_{\mathrm{WF}}^H$, to zero, we can obtain an expression for the Wiener filter

$$\mathbf{W}_{\mathrm{WF}} = \mathbf{R}_{yy}^{-1}\mathbf{R}_{yx} \tag{3.58}$$

By the assumptions of no correlation between the speech signal, $\mathbf{x}[k]$, and the noise signal, $\mathbf{b}[k]$, that is

$$\mathbf{R}_{xb} = E\left\{\mathbf{x}\mathbf{b}^H\right\} = 0 \tag{3.59}$$

which is a assumption common to most optimum filters, we can reduce expression (3.58) to

$$\mathbf{W}_{\mathrm{WF}} = \mathbf{R}_{yy}^{-1}\left(\mathbf{R}_{yy} - \mathbf{R}_{bb}\right) \tag{3.60}$$

were we have made use of the definitions of the auto-correlation matrices for the observation signal, $\mathbf{R}_{yy}$, and the noise signal, $\mathbf{R}_{bb}$. Note, the multi-channel Wiener filter is similar as to the single-channel Wiener filter with the exception that matrices are used instead of vectors for $\mathbf{W}$ and $\mathbf{R}_{yx}$. To illustrate the effect consider a two-channel example, $\mathbf{R}_{yy}$ will then equal

$$\mathbf{R}_{yy} = \begin{bmatrix} \mathbf{R}_{y_1 y_1} & \mathbf{R}_{y_1 y_2} \\ \mathbf{R}_{y_2 y_1} & \mathbf{R}_{y_2 y_2} \end{bmatrix} \tag{3.61}$$

which is seen to hold a block-Toeplitz structure. The main diagonal holds the auto-correlation (temporal-only) information, while the other entries holds spatial information as well (cross-terms).

The two quantities to be estimated in order to compute the Wiener filter is seen in (3.59). In order to compute the filter matrix $\mathbf{W}_{\mathrm{WF}} \in \mathbb{R}^{ML \times ML}$, we, by assuming the correlation matrices known, do a joint-diagonalisation of $\mathbf{R}_{yy}$ and $\mathbf{R}_{bb}$ by the generalised eigenvalue decomposition (as done in single-channel in Section 2.4.7)

$$\begin{cases} \mathbf{R}_{yy} &= \mathbf{Q}\mathbf{\Lambda}_y\mathbf{Q}^T \\ \mathbf{R}_{bb} &= \mathbf{Q}\mathbf{\Lambda}_b\mathbf{Q}^T \end{cases} \tag{3.62}$$

where $\mathbf{\Lambda}_y = \mathrm{diag}(\lambda_{y,i}), 1 \leq i \leq N$ and $\mathbf{\Lambda}_b = \mathrm{diag}(\lambda_{b,i}), 1 \leq i \leq N$, and $\mathbf{Q} \in \mathbb{R}^{N \times N}$ is an invertible, but not necessarily orthogonal matrix [2]. By insertion into (3.60), we obtain

$$\mathbf{W}_{\mathrm{WF}} = \mathbf{Q}^{-T}\,\mathrm{diag}\left(1 - \frac{\lambda_b}{\lambda_y}\right)\mathbf{Q}^T \tag{3.63}$$

an $N \times N$ matrix.

Before explaining the meaning of the $N \times N$ optimum filter matrix let us define the structure of $\mathbf{W}_{\mathrm{WF}}$. By (3.54b) we have L different length-L optimum filters, $\mathbf{w}_l[k]$, each with a *different* delay of l samples on the desired input in order to make a filter, with l taps working in the non-causal domain, realisable. As we can optimise the Wiener filter to each of the microphones, this leaves us with $N = ML$ different optimum filters. These filters constitute the filter matrix, $\mathbf{W}_{\mathrm{WF}}$. First L columns corresponds to the optimum filters w.r.t. the speech component in channel 1, the next L columns to speech component in channel 2, etc. Using the notation $\mathbf{w}_{m,l}^{m_{\mathrm{speech}}}$, where m_{speech} refers to the microphone channel which contains the speech component we are optimising against, we can write $\mathbf{W}_{\mathrm{WF}}$ as

$$\mathbf{W}_{\mathrm{WF}} = \begin{bmatrix} \mathbf{w}_{1,l}[k] & \mathbf{w}_{2,l}[k] & \cdots & \mathbf{w}_{M,l}[k] \end{bmatrix} \tag{3.64}$$

$$\mathbf{W}_{\mathrm{WF}} = \left[\begin{array}{cccc|cccc|c} \mathbf{w}_{1,0}^1 & \mathbf{w}_{1,1}^1 & \cdots & \mathbf{w}_{1,L-1}^1 & \mathbf{w}_{1,0}^2 & \mathbf{w}_{1,1}^2 & \cdots & \mathbf{w}_{1,L-1}^2 & \cdots \\ \mathbf{w}_{2,0}^1 & \mathbf{w}_{2,1}^1 & \cdots & \mathbf{w}_{2,L-1}^1 & \mathbf{w}_{2,0}^2 & \mathbf{w}_{2,1}^2 & \cdots & \mathbf{w}_{2,L-1}^2 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \\ \mathbf{w}_{M,0}^1 & \mathbf{w}_{M,1}^1 & \cdots & \mathbf{w}_{M,L-1}^1 & \mathbf{w}_{M,0}^2 & \mathbf{w}_{M,1}^2 & \cdots & \mathbf{w}_{M,L-1}^2 & \cdots \end{array}\right. \left.\begin{array}{cccc} \mathbf{w}_{1,0}^M & \mathbf{w}_{1,1}^M & \cdots & \mathbf{w}_{1,L-1}^M \\ \mathbf{w}_{2,0}^M & \mathbf{w}_{2,1}^M & \cdots & \mathbf{w}_{2,L-1}^M \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{M,0}^M & \mathbf{w}_{M,1}^M & \cdots & \mathbf{w}_{M,L-1}^M \end{array}\right] \tag{3.65}$$

From (3.65) it can be seen, that we should pick out the column corresponding to the desired speech component and amount of filter causality. For example with four non-causal taps, $l = 4$, and

[2] MATLAB provides the function `eig` to do the GEVD computation as in `[V,D] = eig(A,B)`, where $\mathtt{V} = \mathbf{Q}^{-T}$ and $\mathtt{D} = \mathrm{diag}(\frac{\lambda_A}{\lambda_B})$

estimation of the speech component in microphone 2, we should pick out column $L\cdot(m-1)+l = L\cdot(2-1)+4$, which we can note $\mathbf{w}_{\mathrm{WF},\,l=0}^{m=2}$ (stacked vector from column $L+4$ in (3.65)). In order to make this choice of column explicit, we can restate (3.57) and (3.60) to the speech component in channel m with delay l as

$$\mathbf{w}_{\mathrm{WF},\,l}^{m} = \mathbf{R}_{yy}^{-1}\big(\mathbf{R}_{yy}-\mathbf{R}_{bb}\big)\mathbf{e}_{m,l} = \mathbf{W}_{\mathrm{WF}}\mathbf{e}_{m,l} \tag{3.66a}$$

where $\mathbf{e}_{m,l}$ is an $N\times 1$ vector, defined as

$$\mathbf{e}_{m,l} = \begin{bmatrix}0 & 0 & \dots & 0 & 1 & 0 & \dots & 0\end{bmatrix}^T \tag{3.66b}$$

and the 1 is placed on the $\big((m-1)\cdot L+l\big)$'th position $(0\le l\le L-1\ ,1\le m\le M)$. If we compute the optimum filter matrix by (3.63) and apply $\mathbf{e}_{m,l}$, we will pick out the correct column of $\mathbf{W}_{\mathrm{WF}}$ in (3.65). This perspective of the multi-channel Wiener filter is depicted in Figure 3.13. In $\mathbf{e}_{m,l}$, the choice of channel, m, and delay, l, resides. In Figure 3.13 this is shown as the choice of $x_m[k]$ as desired response and with a delay of $\Delta = l$. Compared to Figure 3.12, Figure 3.13 shows the estimate of the speech component in channel m with delay l. The filter vector, $\mathbf{w}$, has been replaced by the stacked filter vector $\mathbf{w}_{\mathrm{WF},\,l}^{m}$.

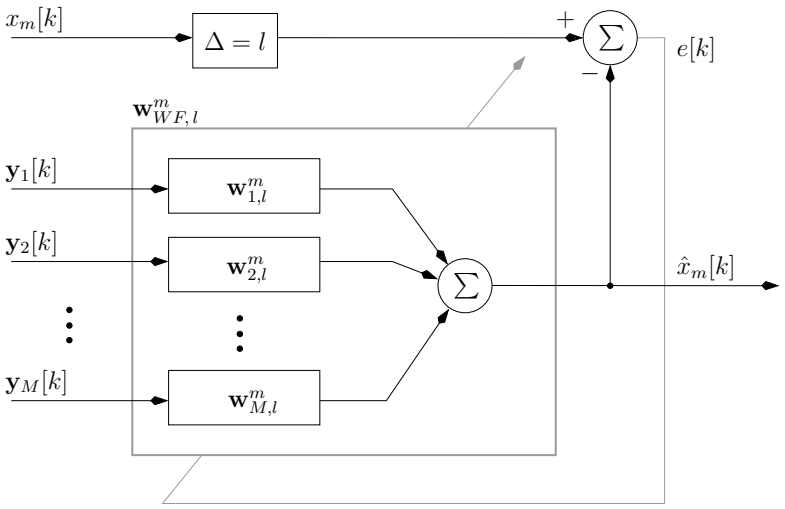


Figure 3.13: A multi-channel Wiener filter which estimates the speech component in channel m with delay l, $x_m[k-l]$. The length-L FIR filters, $\mathbf{w}_{i,l}^{m}$ $(1\le i\le M)$, are computed by applying the $\mathbf{e}_{m,l}$ operator in (3.66b). Compared to Figure 3.12, the filter matrix, $\mathbf{W}$, has been replaced by the stacked filter vector $\mathbf{w}_{\mathrm{WF},\,l}^{m}$.

Doclo et al. [19] show that choosing a filter which has an equal amount of causal and non-causal taps is the optimal choice. Considering a smoothed Kalman filter (which not is to be discussed in this report), which is the optimal filter compared to the non-smoothed Kalman. The smoothing in Kalman filtering is obtained by introducing a filter with half of the filter length non-causal. Thus, the delay introduced in the multi-channel Wiener filter can be seen as realising a smoothed Wiener filter.

Having derived the MCWF using the GEVD it can directly be interpreted as a multi-channel signal subspace approach. Thus, one could also consider such things as low-rank modelling and/or different estimators as already described in Section 2.4 for single-channel. These topics is seen to apply directly to the MCWF if Eq. (3.63) is rewritten to

$$\mathbf{W}_{\mathrm{WF}} = \mathbf{Q}^{-T}\mathbf{F}_{MV}\mathbf{Q}^{T} \tag{3.67}$$

which is similar to (2.34) (derived in Section 2.4.2) for the signal subspace approach, but with different matrix dimensions. The multi-channel Wiener filter is seen to use the minimum-variance estimator inherently. However, low-rank modelling and different estimators in combination with the multi-channel Wiener filter have not been fields of investigation in this project.

3.2.1 Practical Computation Using the Singular Value Decomposition

The multi-channel Wiener filter can be computed by a procedure similar to the one applied for the signal subspace approaches based on generalised singular value decomposition [49]. The multi-channel extension is presented by Doclo et al. [19, 16] and coined GSVD-based multi-channel Wiener filter. It is based on the derivation of the Wiener estimator in the previous section and further assumptions on noise stationarity.

The correlation matrices, $\mathbf{R}_{yy}$ and $\mathbf{R}_{bb}$, are formed implicitly by the empirical auto-covariance estimator described in Appendix B.1 based on a data block-Toeplitz matrix. Frames of L samples for all M channels, see (3.53a), are stacked in the $p \times N$ block-Toeplitz data matrix

$$\mathbf{Y}[k] = \begin{bmatrix} \mathbf{y}^T[k-p+1] \\ \vdots \\ \mathbf{y}^T[k-1] \\ \mathbf{y}^T[k] \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1[k-p+1] & \mathbf{y}_2[k-p+1] & \dots & \mathbf{y}_M[k-p+1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_1[k-1] & \mathbf{y}_2[k-1] & \dots & \mathbf{y}_M[k-1] \\ \mathbf{y}_1[k] & \mathbf{y}_2[k] & \dots & \mathbf{y}_M[k] \end{bmatrix} \tag{3.68}$$

which by pre-multiplication by itself transposed becomes the empirical auto-covariance matrix, $\hat{\mathbf{R}}_{yy}[k] = \mathbf{Y}^T[k]\mathbf{Y}[k]/p$. It can be seen, that p frames ($\mathbf{y}[k]$) are used to form $\mathbf{Y}[k]$.

The noise correlation matrix is in [19] proposed to be estimated using a VAD. Assuming the noise statistics sufficiently stationary to be estimated during speech pauses, indicated by the VAD marking, we can estimate the noise covariance matrix by

$$\hat{\mathbf{R}}_{bb}[k] = \mathbf{R}_{bb}[k'] = \mathbf{B}^T[k']\mathbf{B}[k']/q \tag{3.69}$$

where $\mathbf{B}[k']$ is defined in a similar manner to (3.68), where q frames of $\mathbf{y}[k']$ are used. The starred time index, k', is used to indicate, that the time indexes used for noise estimation, i.e. picking out noise frames, are different than the time indexes, k, used to estimate the observation correlation matrix, $\mathbf{R}_{yy}$.

In order to compute the Wiener filter matrix, $\mathbf{W}_{\mathrm{WF}}$, the block-Toeplitz data matrices are diagonalised using the generalised singular value decomposition (GSVD), see Appendix A,

$$\begin{cases} \mathbf{Y}[k] &= \mathbf{U}_y \mathbf{\Sigma}_y \mathbf{Q}^T \\ \mathbf{B}[k'] &= \mathbf{U}_b \mathbf{\Sigma}_b \mathbf{Q}^T \end{cases} \tag{3.70}$$

where $\mathbf{\Sigma}_y = \mathrm{diag}(\sigma_{y,i})$ and $\mathbf{\Sigma}_b = \mathrm{diag}(\sigma_{b,i})$ are $p \times N$. $\mathbf{U}_y \in \mathbb{R}^{p\times p}$ and $\mathbf{U}_b \in \mathbb{R}^{q\times q}$ are orthogonal matrices, and $\mathbf{Q} \in \mathbb{R}^{M\times M}$ is an invertible, but not necessarily orthogonal, matrix comprising the generalised singular vectors[3]. The matrix $\mathbf{W}_{\mathrm{WF}}$ is then given by

$$\mathbf{W}_{\mathrm{WF}} = \mathbf{Q}^{-T}\, \mathrm{diag}\left(1 - \frac{p\,\sigma_{b,i}^2}{q\,\sigma_{y,i}^2}\right) \mathbf{Q}^T \tag{3.71}$$

where p and q is the number of different samples used in the empirical estimation of the covariance matrices. The relationship between these two variables determine the relationship in the energy of the estimated covariances, thus a constant factor is multiplied to each eigenvalue.

For the white-noise case, the joint diagonalisation is not necessary, and one can use the ordinary singular value decomposition to diagonalise $\mathbf{Y}[k]$. For the non-white noise case, the joint diagonalisation is necessary, as for the signal subspace method described in Section 2.4.7. A MATLAB implementation, `mcwf_time_batch.m`, is described in Appendix D.

The next section will discuss the choices of p and q, which directly affects the auto-covariance estimates.

[3] MATLAB provides the function `gsvd` to do a GSVD computation as in `[U,V,X,C,S] = gsvd(A,B)`, where $\mathtt{U} = \mathbf{U}_a$, $\mathtt{V} = \mathbf{U}_b$, $\mathtt{X} = \mathbf{Q}$, $\mathtt{C} = \mathbf{\Sigma}_a$, and $\mathtt{S} = \mathbf{\Sigma}_b$.

3.2.2 Long-Term Estimates and Stationarity

In the preceding sections, the multi-channel Wiener filter and a GSVD-based approach for practical implementation has been presented. This GSVD-based MCWF is based on implicit forming the auto- and cross-correlation matrices, which stems for the relationship between the block-Toeplitz data matrices and the empirical estimators of the correlation matrices, see Appendix B. To that end, the amount of data in the data matrices have to be considered. In particular, we have to consider the estimation problem as regards noise and speech stationarity.

To facilitate a discussion on the amount of data used in the estimation of the correlation matrices, which is eventually determined by the p and q parameters, see Eq. (3.68), we use the single-channel Wiener filter as starting point. The Wiener filter is a stochastic optimisation problem and relies on good estimates of the underlying process in order to achieve the desired noise suppression. To that end, we note, that when we refer to stationarity, we mean the quality of a process in which the statistical parameters do not change over time. Usually we refer to the first two moments, mean and variance. In order to successfully apply Wiener filtering for noise reduction in speech signals, assumptions has to be made regarding the stationarity of the noise and speech process. Speech signals are very dynamic, however, they can generally be assumed stationary within a $20 - 40\,\mathrm{ms}$ time frame. Thus, ideally, an optimal filter will exist for each time frame.

Due to the restricting stationary assumption of speech only a limited amount of samples can be used to estimate the process parameters. The background noise can, depending on the application, be assumed more stationary than the speech signal. If the stationarity assumptions are violated it leads to performance degradations.

In the former considerations, stationarity refers to temporal, or spectral, stationary. Unlike single-channel Wiener filtering, the multi-channel Wiener filter relies on spectral as well as spatial stationarity assumptions. Spatial stationarity of speech can safely be assumed up to several seconds, which is much longer than the short-time stationarity normally attributed to speech signals. It turns out that the performance of the multi-channel Wiener filter heavily depends on long-term spatial and spectral estimates of the underlying speech and noise stochastic processes, rather than short-term estimates [16]. This brings about the desirable feature, that no musical-noise effects are observed using the multi-channel Wiener filtering, as opposed to the annoying artifacts observed in spectral subtraction and signal subspace which were based solely on short-term estimates.

Remember, the height of the data matrix, p, in (3.68), determines the amount of data which is used to estimate the implicitly-formed correlation matrix. In line with the above considerations on long-term estimates, p should be chosen rather large. In order to elucidate this assertion, a series of simulations have been carried out.

A standard test signal of a $3.5\,\mathrm{s}$ sample with a female voice has been applied to the GSVD-based MCWF method using no reverberation (no convolutional noise, direct path) and 40 filter taps in two channels ($N = 2 \cdot 40$). The VAD employed was an ideal VAD applied to the clean-speech signal. In Table 3.3 the results from the simulations using a number of combinations of data matrix heights p and q, for the observation matrix, $\mathbf{Y}[k]$, and noise matrix, $\mathbf{B}[k]$, respectively, are shown. For combinations of p and q, the output SNR and output WSSM are tabulated in the two tables. Consistent with our assertions the maximum length of p and q, corresponding to 8000 samples for both parameters, yields the highest SNR improvement and lowest distortion (lowest WSSM).

The results in Table 3.3 are based on simulations on a test signal at $8\,\mathrm{kHz}$. This means, that the best performance is obtained when the long-term estimates approach $1\,\mathrm{s}$, which is a much longer time segment than which can be used in the single-channel Wiener filter, where short-term stationarity of the speech signal has to be obeyed. Simulations for the GSVD-based MCWF are done using MATLAB and their are rather computational heavy (on a workstation, one simulation takes more than 24 hours). The tendency of increasing performance using long-term stationarity could be more extensively examined. In specific, using longer p and q values, for long-term estimates up till, say $5\,\mathrm{s}$, would strengthen the conclusions. It is expected, that increasing the estimate to span a few seconds, $2 - 3\,\mathrm{s}$, would improve performance. This is confirmed in work by Doclo

Observation SNR $= 5.0\,\mathrm{dB}$					
$q\backslash p$	256	1000	2000	4000	8000
160					13.3
256	-29.1				
500					16.8
1000		-15.4			18.6
2000			-3.1		20.0
4000				20.4	21.2
8000					**21.8**

Observation WSSM $= 44.2$					
$q\backslash p$	256	1000	2000	4000	8000
160					42.4
256	97.3				
500					38.8
1000		50.7			40.3
2000			53.6		36.3
4000				30.5	33.3
8000					**28.0**

Table 3.3: Results from simulations using the GSVD-based MCWF applied to a clean-speech signal degraded by white noise ($8\,\mathrm{kHz}$). In order to indicate that the performance of the method depends heavily on long-term estimates of the underlying stochastic processes, rather than short-term estimates, different values for p and q are tabulated. In the left table the output SNR is shown, while in the right table, the output WSSM is shown. The results indicates that the higher p and q, thus the longer segments used in the estimation procedure, the better assessment scores.

[16]. Depending on the application one could expect that using longer time spans for the estimate of the background noise process, while keeping the estimation of the observation process at, e.g. $0.5 - 1\,\mathrm{s}$, would be the better choice. This is, of course, dependent on the application and the characteristics of the background noise process. In Doclo [16], the observation matrix is estimated on a time span of $0.3 - 0.5\,\mathrm{s}$, while the noise on a time span of $1 - 2.5\,\mathrm{s}$, but no explicit simulations or investigation of this matter has, to our knowledge, been carried out.

3.2.3 Results using the GSVD-based MCWF method

The MCWF method is tested using the same signals as the previously investigated methods. A series of simulations are performed in order to clarify the effect of adjusting the parameters, that is, the effect of increasing the filter length with and without reverberation, the performance under different reverberated conditions, and robustness are tested. Robustness is tested with respect to two different noise types, two different speech signals, and four levels of background noise. The filter length and reverberation tests are performed using the standard test signal "*Good service should be rewarded with big tips*".

First the filter length has been examined such that the filter length used in the following simulations can be optimal for the test signals used in this project. From Table 3.4 it is clearly seen that long filters improve the performance with reverberation present. With no reverberation an optimal length exists, in this case it is 40 taps. These 40 taps are thus SNR and WSSM optimal in a spatial and temporal sense such that the output error is minimised.

Measure \ Filter length	Observation	10	20	40	80	160
SNR No Rev. [dB]	5	16.2	19.2	**20.7**	18.8	16.5
SNR Rev. 600 ms [dB]	0.1	0.8	0.8	0.9	**1.0**	0.9
WSSM No Rev.	44.4	41.3	35.5	**30.7**	33.0	34.3
WSSM Rev. 600 ms	63.7	61.7	66.5	67.4	63.8	**57.0**

Table 3.4: The MCWF method applies to test signals with or without reverberation. The SNR is $5\,\mathrm{dB}$ and white noise has been used. Different filter lengths has been examined. It is clearly seen that long filters improve the performance with reverberation present. With no reverberation an optimal length exists, in this case it is 40 taps. $p = q = 4000$ samples has been used in both matrices, $\mathbf{Y}[k]$ and $\mathbf{B}[k]$.

Reverberation (convolutional noise) is added to the test signal to investigate the robustness of the MCWF-method regarding reverberated rooms using $p = q = 4000$ and a filter length of $L = 40$. The results are shown in Table 3.5

The results in Table 3.5 indicate that the noise reduction increases when reverberation time increases. An ideal VAD mark based on the true speech signal ($s(t)$ in the observation model) is used. This introduces reverberated speech in the noise matrix and thus more reverberated speech

(i.e. the reverberated signal in the original speech pauses) is removed. Having a longer reverberation time results in more reverberated speech, thus more can be removed. One should notice, though, that this increase is less than 1 dB. The WSSM has been improved at low reverberation times and worsened at more reverberation ($500 - 600$ ms). This indicate that the method is robust with respect to some reverberation. Note, that the filter length used to obtain these results was 40 taps, whereas the results in Table 3.4 indicate that using a filter length of $L = 160$ would increase performance compared to the results obtained in this test. As in the stationarity test the computational times increase in cubic time, thus, testing for increasingly filter lengths the execution time grows rapidly and thus the reverberation test with very long filters has not been performed in this section.

Measure \ Reverberation time [ms]	100	200	300	400	500	600
Observation 5 dB white noise SNR [dB]	3.6	2.1	1.3	0.7	0.4	0.1
Estimated output SNR [dB]	3.6	2.2	1.4	0.9	0.9	0.9
Observation 5 dB white noise WSSM	58.5	63.5	64.2	64.4	64.9	63.7
Estimated output WSSM	53.1	56.1	62.0	63.8	65.8	67.4

Table 3.5: The performance with respect to $s(t)$ improves in both SNR and WSSM, which proves that the method is robust under reverberated conditions.

The test results in Figure 3.14 show the performance using different signals, different noise levels, and different types of noise without reverberation. A very high noise reduction is obtained in all tests and the WSSM was improving above 0 dB of noise, exept for the male speaker in pink noise. The figure shows that pink noise heavily influences the performance for both test signals. The performance for the male speaker is in general worse than for the female speaker. However, in the SNR measure shows an overall, constant improvement, whereas the WSSM measure is different for each test signal. The WSSM is unfortunately in this case not very reliable. The three spectrograms in Figure 3.15 show a high SNR (5 dB), white noise, male speaker test setup output and the same test with a low SNR (-5 dB), pink noise test setup. From the spectrograms it is difficult to see the big span of approximately 20 WSSM points, which can be seen in Figure 3.14 between SNR $= 5$ dB, white noise, and SNR $= -5$ dB, pink noise, male speaker (dashed lines). Listening test (among the group members), too, reveal that the performance of the low SNR test setup is much better than the WSSM measure indicates.

One reason that can explain the behaviour of the WSSM method is visible at very low frequencies in the lower subfigure of Figure 3.15. The pink noise in very noisy setup will inherently drown the low frequencies in the test signal and thus noise is still present in the estimated speech signal (seen between 0.5 and 1.5 seconds). The WSSM measure is based on Bark filter banks, which weigh small low-frequency bands more than higher frequency bands. The small amount of noise present in this band thus affects the overall WSSM measure and yields misinforming results with pink noise. Furthermore, the MCWF removes one of the lower speech traces in the spectrogram from 2 to 2.5 seconds. This valley will rise big differences in the spectral slope difference as well.

3.2.4 Conclusion

The GSVD-based MCWF method proved the results that were expected. In reverberated rooms long filter lengths showed good results whereas one could cope with shorter filter length without reverberation present. 40 filter taps was apparently a good comprise between no and 600 ms reverberation. Using a second (8000 samples, $f_s = 8000$ Hz) of samples when forming the data matrices showed that a good empirical auto-covariance was computed for non-reverberated rooms, but the tests was performed using only 4000 samples and a short filter length due to very long-termed simulation times. Even with 4000 samples the method was capable of improving the assessment scores in rooms with a reverberation time less than 500 ms.

The robustness test showed that the performance of the method was generally high, but pink

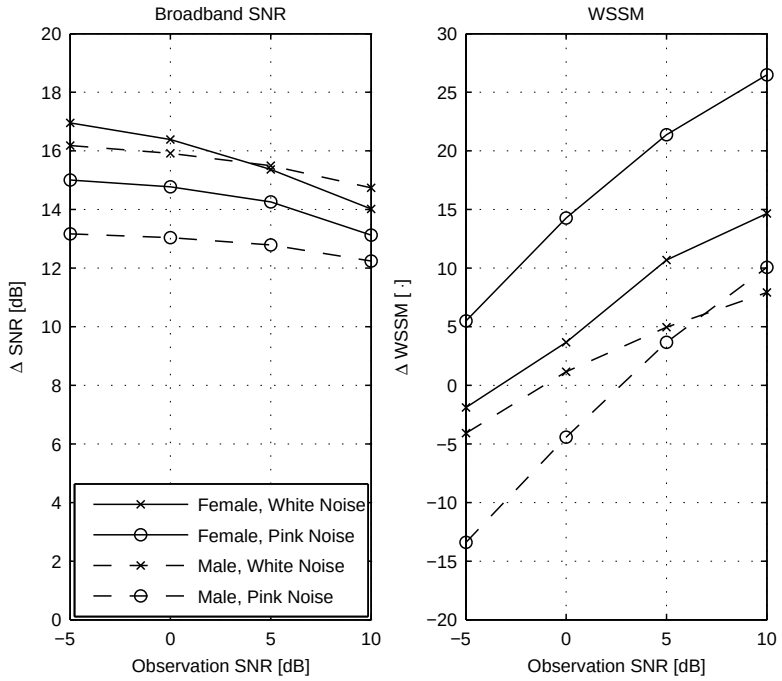


Figure 3.14: Test performance with two different speakers, two types of noise and four different noise levels. Pink noise heavily influences the performance for the female speaker. The male speaker does not manage just as good results as the female speaker.

noise setups caused some troubles for the assessment method, WSSM. A very high noise reduction was however obtained in all tests and the WSSM was improving above 0 dB of noise, except for the male speaker in pink noise. This score was unfortunately not (compared to listening tests among the group members) representative for the actual output.

3.2.5 Frequency-Domain Multi-Channel Wiener Filter using GSVD or GEVD

In the previous section we saw one way of implementing the multi-channel Wiener filter in the time domain. In the next section a dereverberation procedure is presented which is derived using a frequency domain representation. Therefore, a frequency domain based method of computing the MCWF is presented.

In order to present the procedure in the frequency domain, we either use a GSVD-based method, which is analogous to the GSVD-based time-domain implementation, or we use the GEVD to joint diagonalise the power spectral density matrices, or estimates hereof, $\mathbf{S}_{yy}(\omega)$ and $\mathbf{S}_{bb}(\omega)$.

The GSVD-based frequency-domain multi-channel Wiener filter can be implemented by doing joint diagonalisation of the data matrices $\mathcal{Y}(\omega)$ and $\mathcal{B}(\omega)$ which are frequency-domain counterparts to $\mathcal{B}[k']$ and $\mathcal{Y}[k]$, see (3.68). The observation model in the frequency domain, thus becomes

$$Y_m(\omega) = G_m(\omega)S(\omega) + B_m(\omega) \qquad , m = 1 \dots M \tag{3.72a}$$

$$\mathbf{Y}(\omega) = \mathbf{G}(\omega)S(\omega) + \mathbf{B}(\omega) \tag{3.72b}$$

$$\mathbf{Y}(\omega) = \mathbf{X}(\omega) + \mathbf{B}(\omega) \tag{3.72c}$$

where $\mathbf{Y}(\omega)$ is an $M \times 1$ matrix, where M is the number of microphones used. $\mathbf{G}(\omega)$ is the stacked (frequency-wise) room frequency response, $S(\omega)$ is the speech spectrum and $\mathbf{B}(\omega)$ is the stacked

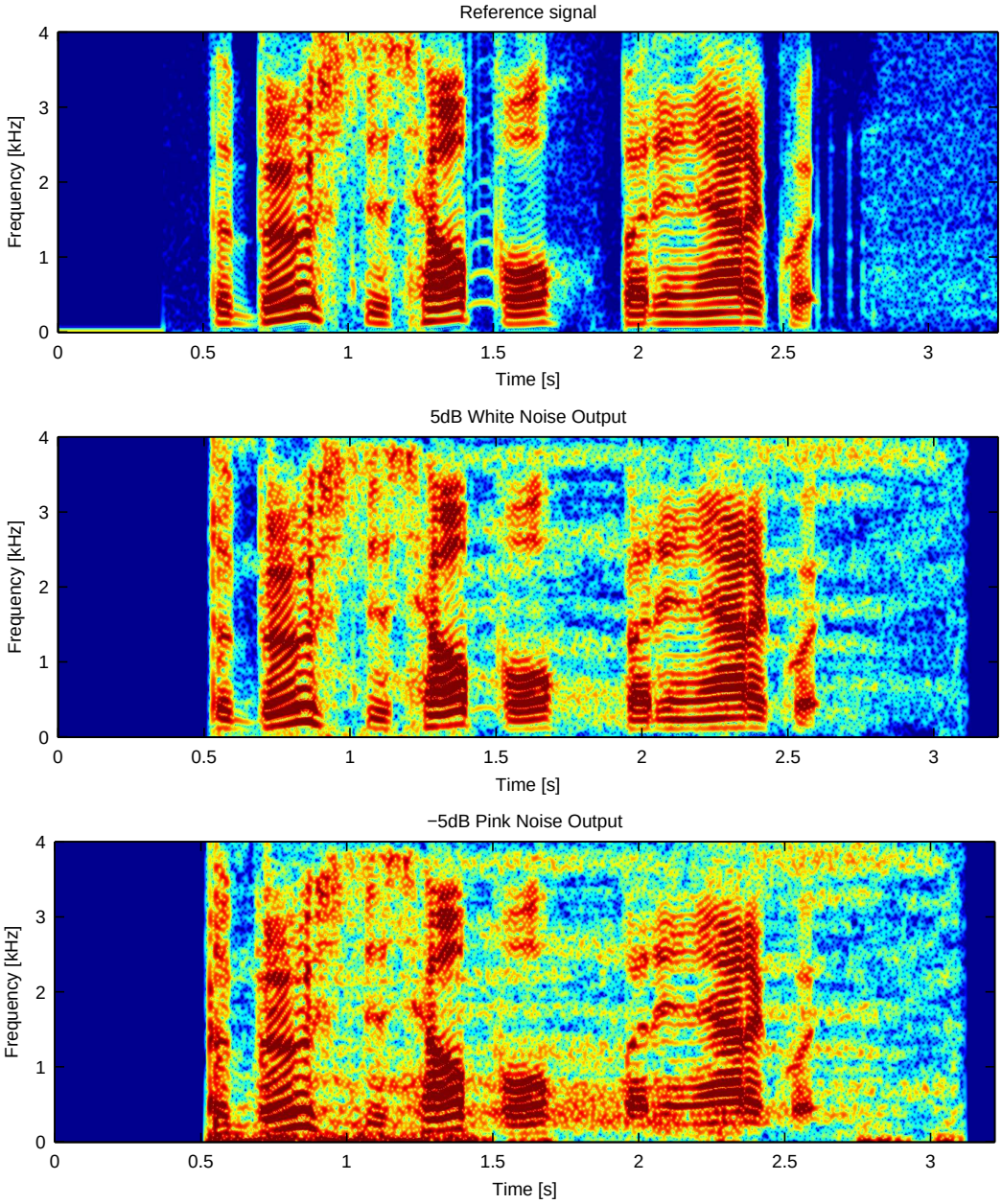


Figure 3.15: The reference signal is shown in the top. The male speaker test signal at $5\,\mathrm{dB}$ white noise and $-5\,\mathrm{dB}$ pink noise, correspondingly. The two output look almost similar, however, a difference of 20 WSSM points exist. The pink noise setup has more noise present in the output at low frequencies and a speech trace has been removed by the filter between 2 and 2.5 seconds.

noise spectrum. The Wiener solution is the well-known solution

$$\mathbf{W}_{\mathrm{WF}}(\omega) = E\left\{\mathbf{Y}(\omega)\mathbf{Y}^H(\omega)\right\}^{-1} E\left\{\mathbf{Y}(\omega)\mathbf{X}^H(\omega)\right\} \tag{3.73}$$

$$= \mathbf{S}_{yy}^{-1}(\omega)\mathbf{S}_{xx}(\omega) \tag{3.74}$$

which using a generalised eigenvalue decomposition (GEVD) as in (3.62) can be rewritten to

$$\mathbf{W}_{\mathrm{WF}} = \mathbf{Q}^{-H}(\omega)\,\mathrm{diag}\left(1 - \frac{\sigma_b^2(\omega)}{\sigma_y^2(\omega)}\right)\mathbf{Q}^H(\omega) \tag{3.75}$$

which is analogous to (3.63), but implies that the GEVD now yields a frequency-dependent decomposition, thus a GEVD is used at every frequency.

This representation (using ω as index) is very convenient, when deriving the Wiener filter, but is impractical to implement. Implementing the frequency domain MCWF, thus requires time to be considered as well. In order to describe the difference between time-domain version and the frequency-domain MCWF we introduce the more practical notation using the short-time Fourier transform (STFT), such that the stacked observation vector can be described as (using a rectangular window)

$$Y_m[r,n] = \sum_{n=0}^{N-1} y_m[k + rR]e^{-j2\pi kn/N} \tag{3.76}$$

where $Y_m[r,n]$ refers to the nth frequency component for block-time index, r, in the mth channel. For each block-time index, the signal, $y_m[k]$, is incremented by R samples (overlap is $N - R$). If we use (3.76) to form a data matrix similar to (3.68), we obtain

$$\mathcal{Y}[r,n] = \begin{bmatrix} \mathbf{y}^H[r-p+1,n] \\ \vdots \\ \mathbf{y}^H(r-1,n] \\ \mathbf{y}^H[r,n] \end{bmatrix} \tag{3.77}$$

$$\text{where} \qquad \mathbf{y}[r,n] = \begin{bmatrix} Y_1[r,n] & Y_2[r,n] & \dots & Y_M[r,n] \end{bmatrix}^T$$

with p stacked frequency-domain vectors of the input signal in the nth frequency bin. The noise data matrix, $\mathcal{B}[r,n]$, can be defined in a similar manner.

The data matrices $\mathcal{Y}[r,n]$ and $\mathcal{B}[r,n]$ can be diagonalised as

$$\begin{cases} \mathcal{Y}[r,n] &= \mathbf{U}_y[r,n]\mathbf{\Sigma}_y[r,n]\mathbf{Q}^T[r,n] \\ \mathcal{B}[r,n] &= \mathbf{U}_b[r,n]\mathbf{\Sigma}_b[r,n]\mathbf{Q}^T[r,n] \end{cases} \tag{3.78}$$

thus the filter matrix is now time-varying and frequency-dependent

$$\mathbf{W}_{\mathrm{WF}} = \mathbf{Q}^{-H}[r,n]\,\mathrm{diag}\left(1 - \frac{\sigma_b^2[r,n]}{\sigma_y^2[r,n]}\right)\mathbf{Q}^H[r,n] \tag{3.79}$$

such that $\mathbf{W}_{\mathrm{WF}}$ is an $M \times M \times N$ cube, by which we mean that N such square matrices exist. The filter corresponding to the mth microphone can be picked as mth column vector. Whereas the time delay (causality of filter) of the desired response, $x_m[k - \Delta]$, is explicit in the time-domain procedure, this is not explicitly defined in the frequency domain. As in other frequency-domain methods, we have traded frequency-discrete computations for time-discrete computations. This does, however, not mean, that non-causal filtering is not possible.

In the time domain, non-causal filtering is achieved by introducing a delay of the input signal, thus timely aligning filter taps and input signal differently. Delaying the input signal or time advancing the filter coefficients is an interchangeable operation (real-time considerations disregarded). The idea is to exploit a property of the discrete Fourier transform in order to advance the filter coefficients. We note, that the time delay/advance corresponds to a linear phase shift

(constant group delay) in the frequency domain [67]. This means multiplication of the discrete Fourier transform by a linear phase factor, $e^{-j2\pi n\Delta/L}$, corresponds to the desired time advance. By a property of the discrete Fourier transform can the phase shift equivalently be achieved by a circular shift of the time coefficients obtained from the inverse Fourier transform of the frequency-domain filter coefficients.

To explain these concepts we will defer from linear algebra vector notation. If we denote the discrete Fourier transform of the filter coefficients as

$$W[n] = \mathcal{F}\{w[k]\} \qquad , 0 \leq k, n \leq L-1 \tag{3.80}$$

where n are the discrete-frequency indexes. The discrete Fourier transform of the time delayed (phase shifted) filter coefficients, $w'[k]$ are then

$$e^{-j2\pi n\Delta/L} W[n] = \mathcal{F}\{w'[k]\} \tag{3.81}$$

with a delay of Δ samples. The relation between $w[k]$ and $w'[k]$ is then given by [67]

$$w'[k] = w[k'] \qquad , k' \equiv k - \Delta \pmod{L} \quad , 0 \leq k \leq L-1 \tag{3.82}$$

from which we can see, that the time-shifted filter coefficients, $w'[k]$, are computed by doing a time-circular shift of the non-shifted filter coefficients, $w[k]$. As we were looking for a time advance, we just use negative Δ-values.

To illustrate the concept of time delaying, we use a length-6 sequence, $w[k]$, which we transform by $\Delta = 2$ to the delayed sequence $w'[k]$ using (3.82). The result is seen in Figure 3.16. The circular shift of the sequence $w[k]$ can easily be observed in the figure. For the practical im-

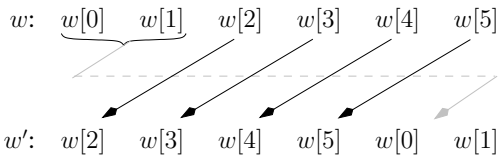


Figure 3.16: Time delaying a length-6 sequence by $\Delta = 2$ using (3.82). By a property of the discrete Fourier transform, the time delay is seen to employ a circular shift form.

plementation of the multi-channel Wiener filter, the GSVD-based implementation suggested in Section 3.2.1 is defined in the time domain making it a logical choice to do the actual filtering with $\mathbf{w}_{\mathrm{WF}}[k]$ in the time domain. However, for the frequency-domain methods, that being GSVD based or GEVD based, it seems logical to keep the computations, including the filtering, entirely in the frequency domain (A MATLAB implementation, `mcwf_freq_batch.m`, is described in Appendix D). This can be achieved using the overlap-save method [67, 73].

3.3 Combined Noise Reduction and Dereverberation

In the previous section a frequency domain interpretation of the multi-channel Wiener filter was described with respect to noise reduction. In this section a method to do dereverberation using the frequency-based MCWF is presented. Recall that the problem of speech enhancement initially was described as obtaining the true speech signal, $s(t)$, from the observations described in (1.3) in Section 1.2.4, i.e.

$$y_m(t) = g_m(t) * s(t) + b_m(t) \qquad m = 1, \ldots, M$$

Obtaining $s(t)$ from $y_m(t)$ requires a complete noise reduction (removing $b_m(t)$) and a full deconvolution of $g_m(t) * s(t)$, thus the degrading effects of the room, $g(t)$ are removed.

A room can be characterised by the reverberation in the room. Reverberation is the effect of adding up the direct path signal and the multiple delayed and attenuated versions caused by

reflections from large surfaces. One way of interpreting an acoustic impulse response is to first recognise the direct-path signal shortly followed by the reflections of large nearby surfaces. This is called early echoes. The early echos are followed by dense, smaller echoes. This is called diffuse reverberation. We refer dereverberation to a full deconvolution, obtaining the true clean speech signal, ideally.

One method to remove $g(t)$ is to do inverse filtering. However, this solution yields a number of problems. First, if the impulse response is very long, then inverse filtering introduces a very long delay. Secondly, room responses are mixed-phase systems, which causes problems with stability in the inverse filter. Furthermore, inverse filtering takes a priori information of the room, i.e. a measure between the speaker and the listener, which in practise is unobtainable and when either the speaker or the listener moves around, a new measurement is needed. Thus, inverse filtering is very position-dependent.

As stated above it requires very long impulse responses to remove the echos. At least to remove those that we hear. Others of these echos are not perceived as echos, but rather as a energy boost at certain frequencies introducing colouring (like a corner-placed loudspeaker seems to boost the bas). These early echos can be seen as the way the room alters the power spectrum of the signal, how it colours the signal.

Removing the colouring of the signal (decolouring) is easier than doing inverse filtering, because shorter filters are required and no a priori impulse response is required. However, some a priori knowledge is still needed.

Decolouring can be seen as obtaining a flat power spectrum of the room, whereas inverse filtering obtains a flat frequency response. To obtain a flat power spectrum of the room one must either know the power spectrum of the signal or the power spectrum of the room. This is almost as difficult as obtaining the inverse impulse response in the inverse filtering case. However, the signal is known a priori to be speech and the power spectrum of the speech can be estimated using an average human speech spectrum as reference.

This chapter will first show how a frequency domain interpretation of the observation model can be used to find a dereverberation filter. The method found although, depends on a frequency-dependent scalar that corresponds to the power spectrum of the present room. This factor is unknown. However, a method referred to as "spectral addition" is presented. This method estimates this ambiguity factor and proves to yield almost as good results as when using the real room power, i.e. using a priori knowledge.

3.3.1 Dereverberation using Signal Subspaces

As stated above, inverse filtering is very position dependent, because it takes the inverse of the path from point A to point B. However, Affes et al. [1] has proposed an adaptive method, which approximates the room impulse, if it is quasi-stationary, i.e. the method can track a slowly varying room. If the frequency-domain observation model presented in (3.72c) in Section 3.2.5 is filtered with a well-defined filter the convolution noise can be removed by this filter, $\mathbf{W}(\omega)$, as shown in

$$\begin{aligned}\mathbf{Y}(\omega) &= \mathbf{G}(\omega)S(\omega) + \mathbf{B}(\omega) \qquad (3.83a)\\ &= \mathbf{X}(\omega) + \mathbf{B}(\omega)\\ \mathbf{Z}(\omega) &= \mathbf{W}^H\mathbf{Y}(\omega) = \mathbf{W}^H(\omega)\mathbf{G}(\omega)S(\omega) + \mathbf{W}^H(\omega)\mathbf{B}(\omega) \qquad (3.83b)\end{aligned}$$

where $\mathbf{G}(\omega)$ contains the acoustical transfer functions from the source to the corresponding microphones

$$\mathbf{G}(\omega) = \begin{bmatrix} G_1(\omega) \\ \vdots \\ G_M(\omega) \end{bmatrix} \qquad (3.84)$$

where M is the number of microphones in the setup. As seen in (3.83b) the new transfer function, $\mathbf{F}(\omega)$, is altered by how $\mathbf{W}(\omega)$ is defined.

Dereverberation narrows down to the design criteria $F(\omega) = \mathbf{W}_d^H(\omega)\mathbf{G}(\omega) = 1$, thus obtaining the dereverberated speech signal $S(\omega)$. An inverse filter can be interpreted as a matched[4] filter

$$\mathbf{W}_d^H(\omega) = \frac{1}{\mathbf{G}(\omega)} = \frac{\mathbf{G}^H(\omega)}{\mathbf{G}^H(\omega)}\frac{1}{\mathbf{G}(\omega)} = \frac{\mathbf{G}^H(\omega)}{\|\mathbf{G}(\omega)\|_2^2} \tag{3.85}$$

is the evident solution to this problem.

The method proposed to approximate the dereverberation filter is based on a signal subspace approach proposed by Affes et al., [1], and used by Doclo, [17]. In [1] it is shown that the eigenvector corresponding to the highest generalised eigenvalue approximates the room response down to a scalar ambiguity and a phase shift. That is,

$$\mathbf{G}(\omega) = \frac{\|\mathbf{G}(\omega)\|}{\|\mathbf{q}(\omega)\|}\mathbf{q}(\omega)e^{j\phi(\omega)} \tag{3.86}$$

where $\mathbf{G}(\omega)$ is the acoustic transfer function, $\mathbf{q}(\omega)$ is the eigenvector corresponding to the highest generalised eigenvalue of the decomposition of the auto-covariance matrices shown in (3.75)

$$\begin{cases} \mathbf{S}_{yy}(\omega) = \mathbf{Q}(\omega)\mathbf{\Lambda}_y(\omega)\mathbf{Q}^H(\omega) \\ \mathbf{S}_{bb}(\omega) = \mathbf{Q}(\omega)\mathbf{\Lambda}_b(\omega)\mathbf{Q}^H(\omega) \end{cases} \tag{3.87}$$

with $\mathbf{Q}(\omega)$ an $M \times M$-dimensional invertible, but not necessarily orthogonal matrix and the relationship to the time-domain is [81, p. 145]

$$\mathbf{S}_{yy}(\omega) = \mathcal{F}\{\mathbf{R}_{yy}\} \tag{3.88}$$

The phase shift $e^{j\phi(\omega)}$ is considered non-audible to the human auditory system [1], thus $\mathbf{G}(\omega)$ can be determined down to an uncertainty of a frequency-dependent scalar, $\|\mathbf{G}(\omega)\|^2$. A method to estimate this ambiguity is proposed in the next section.

3.3.2 Decolouring by Spectral Addition

The decolouring technique proposed in this section is covering two aspects. Firstly, it is a method that can be used as an estimator to the scalar ambiguity from the previous section. Secondly, it can be used directly as a decolouring method that decolours the room, but does not do inverse filtering.

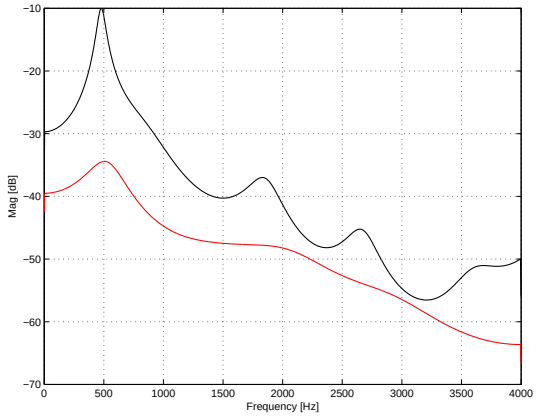
Decolouring is basically an energy estimate of the room. This estimate can be computed using some a priori knowledge of human speech. The idea proposed here is based on finding an average human speech power spectrum density. Thus, by subtracting this average from the observed power spectrum density, the unknown frequency-dependent scalar is estimated

$$\log|X_m(\omega)| = \log|G_m(\omega)| + \log|S(\omega)| \qquad 1 \le m \le M \tag{3.89a}$$

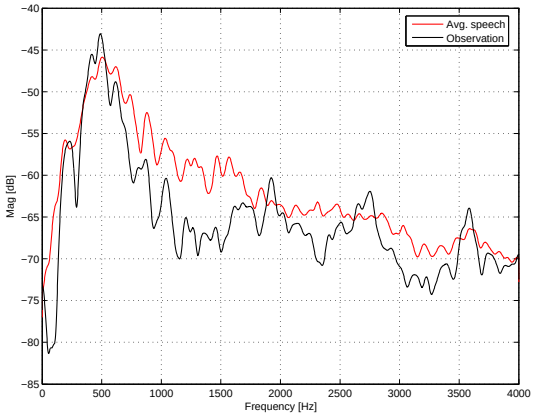
$$\log|\overline{G}(\omega)| = \log|\overline{X}(\omega)| - \log|S_{\mathrm{AVG}}(\omega)| \tag{3.89b}$$

where $\overline{X}(\omega)$ denotes the mean of all the noise-free inputs, which practically can be obtained using a VAD. Notice that the main idea is to obtain an average room energy estimate, $\overline{G}(\omega)$, which can be used as an estimate for all acoustical transfer functions.

In Figure 3.17(a) the concept is illustrated. The difference between the observed signal and an average speech spectrum, both obtained using Burg's estimation to emphasise the formants, will identify the room energy down to some energy leak due to the actual difference between the speech average and the present observation. A human speech average spectrum is obtained by averaging the speech samples from the TIMIT-database [27, samples 160, 190, 305, 338, 357, 751]. Then one of the voice samples has been used as an observation. The periodograms in Figure 3.17(b) are calculated using Welch averaging. If the two periodograms almost overlap the



(a) Conceptual idea of how the energy estimate between average human speech and the observation can be obtained. This is a long speech signal `sent.wav` (30 seconds of speech) as the average human speech (red, lowest) and `305.wav` as the observed PSD (black, top-most). The example is computed using Burg's estimation method to emphasise the formants.



(b) Conceptual idea of how the energy estimate between average human speech and the observation can be obtained. The average is based on the samples in the TIMIT database and the observation is one of these voice-samples. Welch averaging: 160 samples window, 75% overlap, 4096 FFT points.

Figure 3.17: The conceptual ideal and practical illustration of spectral addition.

concept holds and the difference from a real observation will yield an estimated power spectrum. So, to describe the final method, the observed signal spectrum (black) is coloured to the spectrum of average speech using a method we refer to as *spectral addition*. This method is based on the same ideas as used in spectral subtraction, which can be interpreted as a magnitude-only filtering procedure[5]. Thus, we simply add (addition is convolution in the log spectral domain) the PSD of the observed signal and the difference to the human average spectrum to obtain the estimated speech spectrum. The original phase part of the speech spectrum is then used to get back to the non-log domain.

$$\log|S_{\text{Coloured}}(\omega)| = \log|X(\omega)| - \log|\overline{\text{G}}(\omega) \tag{3.90a}$$

$$= \log|X(\omega)| + \underbrace{\left(\log|S_{\text{AVG}}(\omega)| - \log|\overline{X}(\omega)|\right)}_{\text{Spectral Addition}} \tag{3.90b}$$

$$S_{\text{Coloured}}(\omega) = |S_{\text{Coloured}}(\omega)|\angle S(\omega) \tag{3.90c}$$

where the spectra are in log domains. The time signal is obtained by using the inverse Fourier transform on the coloured spectrum and the original speech phase.

In Table 3.6 the spectral addition method is applied at different reverberation time setups.

Measure \ Reverberation time [ms]	100	200	300	400	500	600
Observation 5 dB white noise WSSM	22.9	37.3	44.5	49.5	54.2	55.0
Estimated output WSSM	26.8	41.2	47.3	51.3	53.7	56.2

Table 3.6: The objective measures improvements when only using spectral addition as speech enhancement. The WSSM measure indicates that using spectral addition does not improve the speech quality nor does it degrade it.

The WSSM measure in Table 3.6 indicates that decolouring the room using spectral addition does not degrade performance, since it deviates less than 4 WSSM points. The method can be used to estimate the ambiguity factor, but as an independent function it does not improve speech quality with respect to the assessment measures.

[4]However, in this context the "matched" part refers to matching the room impulse response and not the signal of interest [1]. Thus, it is not used as a correlator as in conventional matched filtering [73].

[5]The main difference here is that we exploit the logarithmic cumulative property, whereas spectral subtraction uses the non-logarithmic data.

3.3.3 Combining Dereverberation and Spectral Addition

Previously, it was shown that the room response could be determined down to an unknown frequency-dependent scalar using the dereverberation method. The spectral addition method estimated this scalar. This section combines these two methods.

The noise reduction result is however different from the results obtained in the time-domain. This is due to two things. Firstly, is this method based on a filter length of $L = 1024$ and secondly, these tests were conducted using the true noise data, thus every noise sample is known a priori. In addition the results were computed using only one filter update, such that only one GSVD computation was necessary. This saves an overlap-save method implementation and the results should still give an indication of whether the method dereverberated the signal or not. This gave a good sampling of the underlying stochastic process, such that a good dereverberation filter could be computed due to stationarity of the acoustic environment (all samples used in the GSVD).

In Figure 3.18 the effects of dereverberation using different methods are shown. In the figure it is shown that the estimate obtained using spectral addition is quite good compared to using the real room energy. However, the WSSM improvement is negative for both methods even the one using the real room energy estimate. The deconvolution method which uses the real room impulse response does not reconstruct the true speech signal. This is due to the fact that the FFT length used is not exactly equal to the true filter length. Filtering using the overlap-save procedure introduces additional zeros which renders the frequency response less exact, such that the output from the deconvolution does not equal the true signal. This effect is noticeable in Figure 3.18 where the "real" deconvolution method also drops off in performance as the room impulse length increase and thus also increases the reverberation time. The results in Figure 3.18

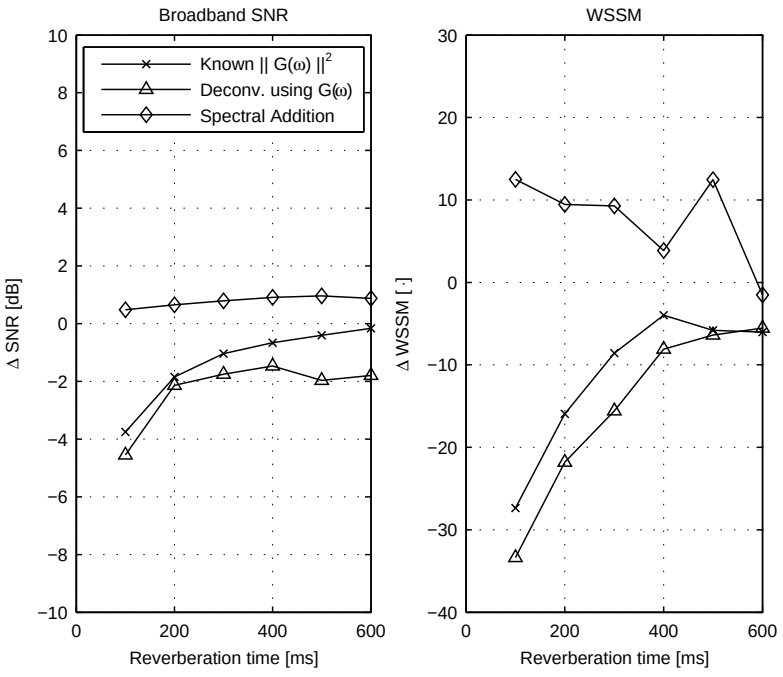


Figure 3.18: The effects of dereverberation using different methods. The diamond-mark is the matched filtering using the real room impulse response. The cross-marked method refers to the filter found in (3.86) using the L_2-norm of the real room impulse response. The last method, "spectral addition", refers to the same method as cross-marked, but uses spectral addition to estimate the L_2-norm. The graphs show that the estimate obtained using spectral addition is quite good compared to using the real room energy.

show that the estimate obtained using spectral addition is quite good compared to using the real room energy. However, the method seems not to deconvolve the room.

3.3.4 Combining Dereverberation using Signal Subspaces and MCWF Noise Reduction

Doclo, [17], shows how the dereverberation method can be combined with the multi-channel Wiener filter (MCWF) to obtain the optimal combined filter. The method, however, is evaluated to be impractical due to the unknown scalar factor. However, we presented the spectral addition method to determine this scalar and used it in combination with the dereverberation method. Combining the method with noise reduction, thus results in a complete speech enhancement method, which is capable of obtaining the speech signal from a number of noisy (additive and convolutional) observations.

Dereverberation and noise reduction is a combined filter, $\mathbf{W}_c(\omega)$, which is a trade off between dereverberation and noise reduction. Thus, having a very good dereverberation at the expenses of increased noise and good noise reduction at the expense of speech distortion. Using the observation model in the frequency-domain as defined in Section 3.2.5 an optimal combined filter, in an MSE-sense, minimises the total MSE with respect to $S(\omega)$, i.e.

$$\min_{\mathbf{W}_c(\omega)} \left(E\{e(\omega)e(\omega)^*\}\right) = \min_{\mathbf{W}_c(\omega)} \left(S_{ee}(\omega)\right) \tag{3.91a}$$

where the error vector is

$$e(\omega) = \mathbf{W}_c^H(\omega)\mathbf{Y}(\omega) - S(\omega) \tag{3.91b}$$

The solution to this minimisation problem is solved by minimising the cost function

$$\begin{aligned} J_{MSE}\left(\mathbf{W}_c\right) &= E\{e(\omega)e^*(\omega)\} && (3.92a)\\ &= E\left\{\left(\mathbf{W}_c^H(\omega)\mathbf{Y}(\omega) - S(\omega)\right)\left(\mathbf{W}_c^H(\omega)\mathbf{Y}(\omega) - S(\omega)\right)^H\right\} && (3.92b)\\ &= E\{-\mathbf{W}_c^H(\omega)\mathbf{Y}(\omega)S^*(\omega) + S(\omega)S^*(\omega) + \mathbf{W}_c^H(\omega)\mathbf{Y}(\omega)\mathbf{Y}^H(\omega)\mathbf{W}_c(\omega) \\ &\qquad - S(\omega)\mathbf{Y}^H(\omega)\mathbf{W}_c(\omega)\} && (3.92c)\\ &= P_{ss}(\omega) + \mathbf{W}_c^H(\omega)\mathbf{S}_{yy}(\omega)\mathbf{W}_c(\omega) - \mathbf{W}_c^H(\omega)P_{ss}(\omega)\mathbf{G}(\omega) \\ &\qquad - P_{ss}(\omega)\mathbf{G}^H(\omega)\mathbf{W}_c(\omega) && (3.92d) \end{aligned}$$

where * denotes complex conjugation and $P_{ss}(\omega) = E\{S(\omega)S(\omega)^*\}$ is the speech spectrum and uses a different notation, because it is only a scalar. Minimising the error gives

$$\begin{aligned} \nabla J_{MSE}\left(\mathbf{W}_c\right) = 0 \Leftrightarrow \frac{\partial \mathbf{S}_{ee}(\omega)}{\partial \mathbf{W}_c^H} = 0 \Leftrightarrow \mathbf{S}_{yy}(\omega)\mathbf{W}_c(\omega) - P_{SS}(\omega)\mathbf{G}(\omega) = 0 && (3.93a)\\ \mathbf{W}_c(\omega) = P_{ss}(\omega)\mathbf{S}_{yy}^{-1}(\omega)\mathbf{G}(\omega) && (3.93b) \end{aligned}$$

which is the combined filter that reduces both noise and reverberation. The noise reduction filter and the dereverberation method now are combined in order to show that they yield this optimal solution.

The MMSE estimate with respect to noise reduction was given

$$W_{WF}(\omega) = \mathbf{S}_{yy}^{-1}(\omega)\mathbf{S}_{xx}(\omega) \tag{3.94}$$

thus, using (3.94) to remove noise and the matched filter (3.85) to remove reverberation the combined filter is found to be

$$\begin{aligned} \mathbf{W}_c(\omega) &= W_{WF}(\omega)\mathbf{W}_d(\omega) && (3.95a)\\ &= \left\{\mathbf{S}_{yy}^{-1}(\omega)\mathbf{S}_{xx}(\omega)\right\}\left\{\frac{\mathbf{G}(\omega)}{\|\mathbf{G}(\omega)\|^2}\right\} && (3.95b)\\ &= \mathbf{S}_{yy}^{-1}(\omega)\mathbf{G}(\omega)P_{ss}(\omega)\mathbf{G}^H(\omega)\frac{\mathbf{G}(\omega)}{\|\mathbf{G}(\omega)\|^2} && (3.95c)\\ &= P_{ss}(\omega)\mathbf{S}_{yy}^{-1}(\omega)\mathbf{G}(\omega) && (3.95d) \end{aligned}$$

which proves equal to the derived optimal combined filter in (3.93b). Recall that $E\{\mathbf{Y}(\omega)\mathbf{Y}^H(\omega)\} = \mathbf{S}_{yy}(\omega)$, that $\mathbf{S}_{xx}(\omega) = \mathbf{G}(\omega)P_{ss}(\omega)\mathbf{G}^H(\omega)$ and $P_{ss}(\omega) = E\{S(\omega)S^*(\omega)\}$.

According to (3.95d) the combined noise reduction and dereverberation filter can be seen as a dereverberation filter with a post- or pre filter (a scalar multiplication) that reduces noise.

Using the property, that $\mathbf{S}_{xx}(\omega)$ has rank one, which is shown in [1], and using (3.86) and (3.75), (skipping the phase)

$$\begin{aligned}\mathbf{S}_{xx}(\omega) &= \mathbf{S}_{yy}(\omega) - \mathbf{S}_{bb}(\omega) \\ &= \mathbf{Q}(\omega)\big(\mathbf{\Lambda}_y(\omega) - \mathbf{\Lambda}_b(\omega)\big)\mathbf{Q}^H(\omega) \\ &= \mathbf{q}(\omega)\left(\lambda_{y_1}(\omega) - \lambda_{b_1}(\omega)\right)\mathbf{q}^H(\omega) \\ &= \hat{\sigma}_x^2(\omega)\mathbf{q}(\omega)\mathbf{q}^H(\omega) \end{aligned} \tag{3.96}$$

where $\mathbf{q}(\omega)$ corresponds to the eigenvector with the largest singular value of $\mathbf{Q}(\omega)$ (rank 1 matrix) and $\hat{\sigma}_{x_1}^2 = \sigma_{y_1}^2 - \hat{\sigma}_{b_1}^2$. The Wiener Filter can then using the GSVD be expressed as

$$\mathbf{W}_{WF}(\omega) = \mathbf{Q}^{-H}(\omega)\mathbf{\Sigma}_Y^{-2}(\mathbf{\Sigma_Y}^2 - \mathbf{\Sigma}_B^2)\mathbf{Q}^H(\omega) = \frac{\sigma_x^2}{\sigma_{y1}^2}\overline{\mathbf{q}}(\omega)\mathbf{q}(\omega)^H \tag{3.97}$$

where $\overline{\mathbf{q}}(\omega)$ denotes the column eigenvector of $\mathbf{Q}^{-H}(\omega)$ that corresponds to the largest generalised singular value, σ_{y1}.

Using (3.86), skipping the phase though, (3.96), and (3.95b) the combined Wiener filter can be expressed as

$$\begin{aligned}\mathbf{W}_c(\omega) = \mathbf{W}_{WF}(\omega)\mathbf{W}_d(\omega) &= \mathbf{S}_{yy}^{-1}(\omega)\mathbf{S}_{xx}(\omega)\frac{\mathbf{G}(\omega)}{\|\mathbf{G}(\omega)\|^2} \\ &= \frac{\sigma_x^2}{\sigma_{y_1}^2}\overline{\mathbf{q}}(\omega)\mathbf{q}^H(\omega)\frac{\|\mathbf{G}(\omega)\|}{\|\mathbf{G}(\omega)\|^2\|\mathbf{q}(\omega)\|}\mathbf{q}(\omega) \\ &= \frac{\|\mathbf{q}(\omega)\|}{\|\mathbf{G}(\omega)\|}\frac{\sigma_x^2}{\sigma_{y_1}^2}\overline{\mathbf{q}}(\omega) \end{aligned} \tag{3.98}$$

This combined method has been tested using the same setup has in previous dereverberation test (Figure 3.18). However, 5 dB of white noise (thus also convolved at the microphone) has been used to test noise reduction and dereverberation in combination. The test results are shown in Figure 3.19. The results show that the dereverberation technique is able to improve under noisy conditions either. The methods are able to increase the noise reduction slightly, but it is clearly observed that the method without dereverberation performs better, thus the dereverberation actually decreases the performance with the difference between the noise-reduction-only estimate and the combined methods. The noise reduction result is however different from the results obtained in the time-domain. This is due to two things. Firstly, is this method based on a filter length of $L = 1024$ and secondly, these tests were conducted using the true noise data, thus every noise sample is known a priori. In addition the results were computed using only one filter update, such that only one GSVD computation was necessary. This save overlap-save method implementation and the results should still give an indication of whether the method dereverberated the signal or not. This gave a good sampling of the underlying stochastic process, such that a good dereverberation filter could be computed due to stationarity of the acoustic environment. The noise reduction filter however would suffer from this long (batch) window and yield a noise reduction filter without dynamics corresponding to those of the speech.

3.3.5 Conclusion and Remarks

In this section we introduced a dereverberation method based on using the eigenvector corresponding to the largest eigenvalue as a room frequency response. This estimate was indeterminable down to a linear phase shift and an unknown frequency-dependent factor. We presented a method, so-called *spectral addition*, to estimate this ambiguity factor.

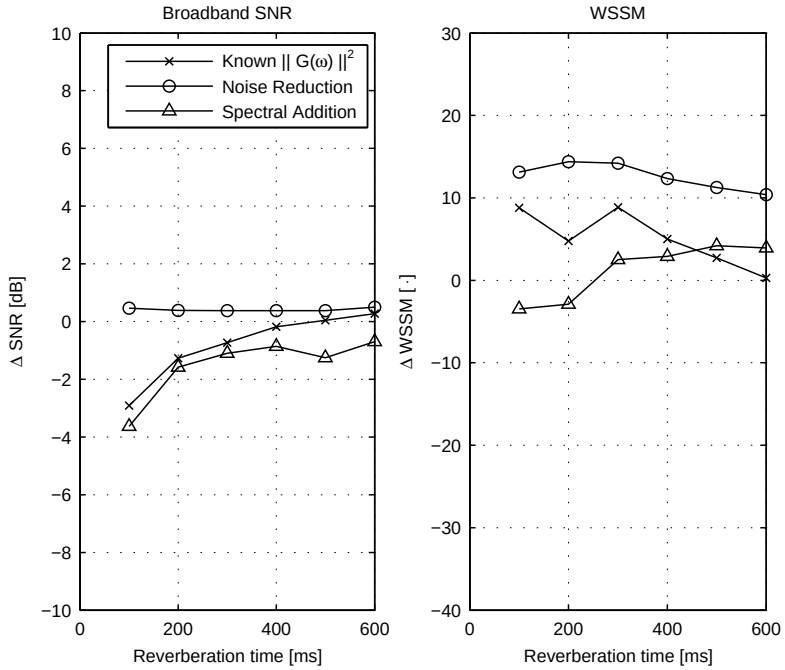


Figure 3.19: The effects of noise reduction and dereverberation using different methods at an $\mathrm{SNR} = 5\,\mathrm{dB}$. Notice, that noise reduction is preferable over combined dereverberation and noise reduction.

Based on Doclo's work we combined the dereverberation method using signal subspaces with the frequency-domain version of the multi-channel Wiener filter and spectral addition. The results showed that noise reduction alone obtained better noise reduction (unaltered) and improved the WSSM score.

The reader should be informed that the method proposed by Affes et al. for dereverberation was using 16 microphones from distances less than one meter and Affes et al. states that the method fails when the microphones are too closely spaced such that the signals becomes highly correlated. In addition we have limited us to two microphones. These facts are an indication of the lacking performance which was shown in this section.

3.4 Conclusion

In this chapter we have introduced an extra microphone and thereby spatial information to our observation model. We investigated spatial sampling and spatial filtering by examining the delay and sum beamformer and the adaptive beamformer, a GSC. Under ideal situations (no reverberation) the DS beamformer obtained a noise reduction of $3\,\mathrm{dB}$ without affecting speech distortion. The GSC obtained good noise reduction ($13.7\,\mathrm{dB}$), and improved the WSSM score (13.2 points). This was a significantly improvement compared to the single channel methods. For other noise levels and other noise types the GSC also obtained good scores. The noise was decreased and the speech quality improved, which was obtained by exploiting both temporal and spatial information.

The GSC was tested using an artificial constructed room, with varying reverberation times. In this test setup no noise reduction could be obtained, because reverberation introduces correlation in the ANC stage since reflections from the surroundings was impinging from many different directions and thereby can not be totally eliminated from the noise estimate.

The fixed and adaptive beamforming was introduced in order to gain understanding of spatial filtering and it was extended to a more sophisticated and advanced method, the multi-channel Wiener filter (MCWF). An advantage of this method was that it did not require a priori knowledge of the location of the speaker. The MCWF disregarded the quasi-stationary assumptions for

speech and thereby mainly functioned as a beamformer by exploiting the spatial information in the inter-channel cross-correlation.

This method was also tested under ideal conditions as the GSC. The MCWF obtained an SNR improvement of $15.7\,\mathrm{dB}$ and an improvement in WSSM of 13.7 points, which was better than the GSC.

The MCWF was also tested in more realistic environments, thus adding reverberation. When reverberation was present the performance of the MCWF decreased drastically, at best a noise reduction of just below $1\,\mathrm{dB}$ was obtained. The speech signal was however not distorted, an improvement in WSSM of a few points was attained. Concluding that the MCWF obtained better results than the GSC both with and without reverberation present.

Motivated by the lacking performance of the MCWF in reverberated rooms, we investigated a method for dereverberating the recorded signal proposed by Affes et al.[1]. This method, however, relied on priori information of a frequency-dependent ambiguity factor related to the impulse response of the room. Therefore a method, *spectral addition* was proposed, to estimate this ambiguity factor. Using the method proposed by Affes [1] for decolouring we did not find any improvement compared to the observation, by using either, the ambiguity factor estimated by spectral addition, nor by using the true impulse response of the room.

Finally we combined the decolouring method with the MCWF by deriving the MCWF in the frequency domain. From the simulations the best result was found for noise reduction alone, thus without using dereverberation. Overall we can conclude that the MCWF was found as the best speech enhancement method w.r.t. noise reduction, for both reverberated rooms and rooms without reverberation, where especially the latter showed promising results.

Motivated by the results obtained using this method the MCWF has been chosen to be analysed with respect to implementation considerations. This is the topic of the next chapter.

CHAPTER 4

Implementation Considerations for the Multi-Channel Wiener Filter

In previous chapters single- and multi-channel methods for speech enhancement have been presented. It was concluded in Chapter 3 that using the multi-channel Wiener filter in a dual-channel context resulted in substantial noise reduction, while keeping the speech distortion below a reasonable level. Even in reverberated environments, the multi-channel Wiener filter proved itself capable of enhancing the intelligibility.

The computations which formed the basis for the multi-channel Wiener filter (MCWF), where based on the generalised singular value decomposition. This rests on the exposition by Doclo et al. [19], which proved that the MCWF could be computed using this linear algebra technique, which in many ways resembles the computations in the signal subspace methods proposed by Jensen et. al [49] re-casted to a multi-channel implementation. Although the MCWF can be computed using an GSVD-based algorithm, this is not the only way to realise a MCWF. Furthermore, the computations in our MATLAB implementation of the GSVD-based MCWF (`mcwf_time_batch.m`, see Appendix D) are very computational heavy. For example, for a $3.5\,\mathrm{s}$ signal at $8\,\mathrm{kHz}$ to be processed with $p = q = 4000$, referring to Eq. (3.68) in Section 3.2.1, and a filter length of 40 taps in two channels, the entire simulation took around 10 hours on a modern Pentium-based workstation running Linux.

In Chapter 3 the multi-channel Wiener filter *functionality* was examined by means of an GSVD-based implementation. The MATLAB simulations revealed that the computations involved in computing the optimal filter were quite complex and required heavy computing power. Motivated by the promising speech enhancement performance of the GSVD-based MCWF, we will in this chapter focus on the *implementation* of the MCWF *functionality*. That is, we will investigate alternative algorithms for an efficient implementation. The overall goal is reduced computational complexity. However, before commencing an exposition on low-complexity alternatives and related concepts, we digress slightly into a methodology, the *rugby meta model*, well-known in the HW/SW design community. This is to facilitate a precise description and motivation for the activities of this project with respect to the algorithm domain in the A3 paradigm presented in Section 1.6.

An example of a meta model, which is a methodology to describe models and design activities, is the Y-chart (see e.g. [90]). This is a classical, however, somewhat outdated meta model. More recently Jantsch et al. [48] have introduced the rugby meta model. As opposed to the Y-chart, the rugby model explicitly defines representation of abstraction of the domains *computations*, *data*, *time*, and *communication*. The model is depicted in Figure 4.1. If we define the idea or concept to the left, we define the goal or physical product on the right-hand side. We move left to right from high to low level of abstraction.

For large industrial projects, activities, and iterations over all domains must be specified. For

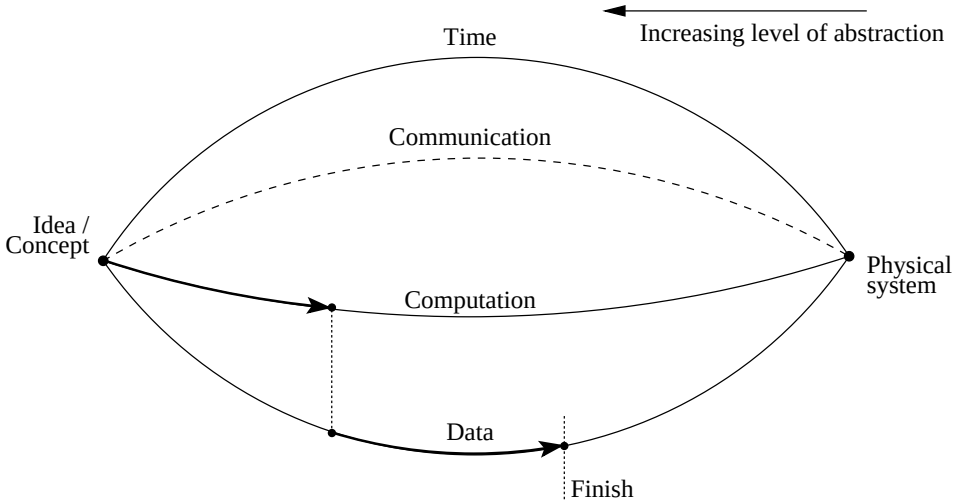


Figure 4.1: The rugby meta model which is a methodology or framework, in which design activities and design tools can be studied. The model is known due to its explicit model of the four shown domains, time, communication, computation, and data, which are concepts in real-life models, which have to be considered. The motivation for including the rugby model here, is to illustrate that we use the multi-channel Wiener filter as starting point for this chapter. We aim at study of algorithms for implementation of the MCWF and choose a superior method for implementation.

our work we have confined ourselves to the computational and data domain. This is shown in Figure 4.1 with bold lines. The starting point is the multi-channel Wiener filter. In Chapter 3 we opted for a GSVD-based solution. This implicitly defined a specific algorithm to realise the MCWF functionality and thus lowered the level of abstraction. In other words, we moved one step from the *idea* along the *computational* axis refining the model and lowering the level of abstraction. The MATLAB model used in Chapter 3 was very computational heavy. In this chapter we focus on converting this model to a more efficient implementation. We reduce the overall complexity of the multi-channel Wiener filter subject to retain the functionality of the MATLAB model. It is the goal, after a survey of possible algorithms, to implement the method of choice in the C programming language using floating-point arithmetic to facilitate real-time processing.

Following the floating-point implementation, we aim at considering the problem of data refinement. Referring to Figure 4.1, we move along the *data* axis by refining the C-based floating-point model towards fixed-point precision by fixed-point design-space exploration. In combination with the SystemC methodology, fixed-point arithmetic is simulated and integrated with the floating-point model to gradually convert each group of computations to fixed-point. Although not explicitly accounted for, the goal of the refinements in the computational and data domains are subject to limitations in the time domain. We implicitly aim for a real-time implementation of the multi-channel Wiener filter.

In the following sections we will firstly focus on alternative algorithms to implement the computation of the optimal filter for the MCWF formulation. The actual filtering is of little concern as it constitute only a fraction of the computations involved in the sample-by-sample adaptive filtering. In Section 4.1.1 to Section 4.1.5 we will describe a number of alternative algorithms, which leads to the further investigation of an recursive GSVD-based formulation based on interlaced QR-updating and diagonalisation using an Jacobi structure. In Section 4.2.1 the speech enhancement performance, or rather, the performance degradation is investigated. In the following Section 4.3, it is investigated how sub-sampling the estimate of the underlying stochastic process can further reduce the computational complexity, while keeping the performance fixed using an adaptive noise cancellation stage with an NLMS adaptive filter. Lastly, in Section 4.4 considerations on how to convert the floating-point C-based model of the GSVD-based MCWF to a fixed-point implementation are done.

4.1 Computational Complexity of the Multi-Channel Wiener Filter

As mentioned, the multi-channel Wiener filter can be implemented using the generalised singular value decomposition. In Chapter 3 the functionality of the MCWF was examined using a GSVD-based MATLAB implementation (`mcwf_time_batch.m`, see Appendix D). However, in this section we present a survey of candidate algorithms to efficiently implement the multi-channel Wiener filter. We start out by pin-pointing our problem at hand. We desire an estimate of the optimal clean-speech signal, which we intend to achieve by optimal filtering in a multi-channel scenario. This comes down to two facts

1. Compute the multi-channel Wiener filter

2. Only the noisy observation is available

These observations will guide our survey of algorithms to implement the multi-channel Wiener filter. It should be further noted, that since the desired signal is not available, it has been proposed to rely on an estimate of noise-dominated time segments by use of a VAD. This means that one way to obtain the optimal speech estimate is by joint diagonalisation of an observation and a noise data matrix. It is not possible to rely on white-noise assumptions.

In the following, algorithms with low complexity will be our goal. In order to assess different algorithms, we need to define a measure of complexity. We have chosen to count the operations of an addition, a multiplication, a division, and a square root all as one operation. We call this operation a FLOP or a floating-point operation. This choice is despite the fact that most modern architectures are more efficient w.r.t. certain application specific operations. One should be aware of the FLOP counting is a crude approach to measure algorithmic complexity as it ignores indexing, memory traffic and branching overhead, and other processor-/architecture-dependent overhead. We will make no assumptions about any architecture, thus we should be careful not to infer too much from algorithmic complexity based on FLOP counting. Referring back to the A3 model in Figure 1.13 on page 19, this chapter will focus on the algorithm part, with no concerns of an implementation to a specified architecture. In the following we will present a survey of different algorithms which efficiently implement our functionality, the multi-channel Wiener filter.

4.1.1 Survey of Low-Complexity Alternatives for Implementing the Multi-Channel Wiener Filter

In the literature various classifications have been introduced. However, not all are of orthogonal concepts and some methods fall under several classes. Generally we identify three paramount classes; *numerical methods*, *subspace tracking techniques*, and *steepest-descent based methods* (or LMS-like methods). Before the actual survey, we introduce the reference MATLAB -based GSVD computation, which falls under the numerical methods class.

The reference GSVD-based method for computing the multi-channel Wiener filter in MATLAB is referred to as the "full GSVD" in this text. MATLAB makes use of the LAPACK library, in which the GSVD computations are based on methods by Luk [54] and Charlier et al. [8], which is improved by Bai and Demmel [4]. The overall complexity is $\mathcal{O}(n^3)$, using the big-$\mathcal{O}$ notation. To compute the GSVD, the LAPACK library relies on two QR decompositions (QR) and a singular value decomposition. The matrices to be diagonalised are assumed to be of same column-width. An estimate of the complexity of the LAPACK library's GSVD, based on the assumption of $p = q$, is approximately $6n^2(m - n/3)$ (for two QRDs), see [32, p.227], and approximately $6.67n^3$ for one SVD of a square matrix [2]. With the parameters, $m = 4000$ and $n = 40$, determined in the

previous chapter, and $f_s = 8\,\mathrm{kHz}$, the complexity is higher than

$$\begin{aligned} \mathrm{FLOPS} &= \text{complexity per sample } \times f_s && \text{(4.1a)} \\ &= \left(6n^2(m - n/3) + 6.67n^3\right) f_s \\ &= 1250\ \mathrm{GFLOPS} && \text{(4.1b)} \end{aligned}$$

which is confirmed by an estimate in [20].

We would like to introduce the reader to the concept of *updating* in this context. In the full GSVD a decomposition of the data matrices, $\mathbf{Y}[k]$ and $\mathbf{B}[k]$ are done at each time step. It would, from a computational perspective, be advantageous to *update* the decomposition by adding the new row (or column) and removing the oldest row (or column). This operation is generally referred to as updating. However, adding followed by removing corresponds to an rectangular time windowing, for matrix $\mathbf{Y}[k]$ of length p. Some computational efficient algorithms relies on *recursive updating* employing an exponential time window. This recursive smoothing, or filtering, circumvents the down-dating connected with removing a row (or column), but comes with problems of its own, e.g. filter stability.

By this short introduction to the concepts of "full" decompositions, e.g. full GSVD, updating and the reference GSVD-method in MATLAB , we introduce computational efficient alternatives.

Methods with the objective of tracking one or more, but not all, of the eigenvalues (singular values) and/or eigenvectors (singular vectors) belongs to the subspace tracking class. Generally these methods are characterised by that they are devised in order to track a few sinusoids, and that they are based on a white-noise assumption, i.e. they do not employ joint diagonalisation as offered by the full GSVD. Yang [92] derives an RLS-like method to track one or more eigenvalues. By a deflation technique (PASTd), the eigenvectors can be sequentially estimated. Another, less complex algorithm, is proposed by Rabideau [76] and is called fast subspace tracking (FST). It is, however, only capable of tracking eigen/singular vectors, not -values. Recently Tufts et al. [88] and Real et al. [78], have shown significant speed-up compared to the PAST algorithm by Yang in specific setups. Their method, fast approximate subspace tracking (FAST), is capable of tracking singular values as well as singular vectors, and has its roots in column-updating of a data matrix and fast diagonalisation using the fast Fourier transform. Cooley et al. [14] has proposed a single-channel, highly optimised version of the FAST algorithm.

The above-mentioned methods for subspace tracking are concerned with tracking only a few singular values. This is of little interest for computing the multi-channel Wiener filter, as it requires the complete diagonalisation and computation of both singular vectors and singular values. When the methods in this class are used to do compute the complete decomposition, i.e. all singular vectors and singular values, the result is, in most cases, a complexity comparable to the one of the method proposed by Moonen et al. [65]. It is possible, albeit out of scope here, to incorporate low-rank modelling of speech signals into the multi-channel Wiener filter. By using a sufficient filter order, L, the above-mentioned subspace tracking methods might prove themselves superior to complete decompositions.

Benesty et al. has derived a generalised, frequency-domain multi-channel adaptive filter which is based on Toeplitz data matrices, as $\mathbf{Y}[k]$. The novel approach is based on forming circulant matrices from the Toeplitz structured matrices. Circulant matrices are known to be diagonalisable by the (fast) Fourier transform. He proposes both time-constrained and unconstrained versions, which are extensions of the well-known single-channel frequency-domain adaptive filters, see e.g. [82]. To alleviate the delay generally introduced by frequency-domain processing, he proposes the generalised multi-delay filter. Although the method seems highly promising it has one drawback w.r.t. multi-channel Wiener, it relies on two observation signals. The desired and the noise-degraded signal. This is feasible for e.g. echo-cancellation applications, but infeasible in the present context.

Recently Spriet [84] has proposed a stochastic gradient (LMS-like) method which is used to implement a multi-channel Wiener filter. Although the LMS-like approach generally ensue low-complexity algorithms, the basic functionality of the proposed system is slightly different than

that of the multi-channel Wiener filter under consideration. As with the methods of Benesty et al., a reference signal is available, however, Spriet proposes to use a fixed beamformer and the MCWF as multi-channel, post-processing stage.

In order to solve the SVD or GSVD problem many different approaches of the class of numerical methods exists. Common is that they are well-founded in linear algebra, that being bi-diagonalisation, power iterations, or using Householder or Givens matrices [31, 65, 54, 85]. Generally Givens rotations are preferred in updating schemes because of its simplicity, matrix locality, and the fact that it lends itself easily to parallel implementations. The most important class using Givens rotations seem to be the QR Jacobi-like algorithms, which is covered by Moonen et al. in [64, 65] for the SVD case and in [63] for the case of the joint diagonalisation. This method is also the choice which Doclo et al. has followed [20].

Interesting work has been presented by Pango and Champagne [70] w.r.t. devise efficient schemes which maximises the effect of the Givens rotations by applying the procedures on subregions. The idea stems from the fact that stationarity of data, or lack thereof, entering the data matrix upon updating will directly reflect the dynamics and this off-diagonal "energy" in subregions. Thus applying data annihilation on local regions with high dynamics will maximise the effect of each Givens rotation. This approach builds on work by Moonen et al. [65], but the skipping behaviour of the application of Givens rotations renders the parallelism inherent in the QR Jacobi-type algorithms difficult to exploit.

Of the three types, we have chosen to focus on the numerical class, in specific the recursive GSVD-based method, proposed by Moonen et al. [63]. Before describing the method, we will describe the test setup used to evaluate the performance of the complexity-reduced implementation.

4.1.2 Assessment Techniques for Measuring Acceptable Performance Degradation

Lowering the complexity is a goal by itself, however, we also need to refer to the MATLAB based "Full GSVD" method we have developed. We will use the standard signal with a female voice uttering: "*Good service should be rewarded with big tips*", a filter length of 40 taps in each of the two channels, and empirical correlation matrix estimators based on data matrices of length $p = q = 4000$ samples. The performance parameters, SNR improvement and WSSM in-output relation, will be our golden reference for the complexity reduction. Based on our experience, we have chosen the specification of demands to be $1\,\mathrm{dB}$ SNR and 2 WSSM. In this chapter we will focus on the complexity of the MCWF alone, and consider the (ideal) VAD available.

The chapter will consist of two major parts. The first part covers the steps taken going from an iterative SVD procedure to a recursive GSVD algorithm. The second part mainly focuses on the numerical properties of the obtained algorithm, but also covers complexity reducing parts such as decreasing the filter length of the MCWF and introducing an adaptive noise cancellation (ANC) stage.

As mentioned in the previous section computing the full GSVD requires a lot of operations. One way to reduce this complexity is by introducing a recursive update of the GSVD instead of computing a full GSVD for each new incoming sample. In order to introduce recursion an iterative method, which has it roots in an iterative SVD is presented first.

4.1.3 Computing an Iterative Singular Value Decomposition

Several methods on how to compute the SVD in an iterative manner exist. Methods that rely on a Jacobi-like method are beneficial when moving on to implementation. Methods like Gaussian elimination, Gram-Schmidt factorisation, Givens rotations, and Householder reflections have been used to solve linear equations, where Householder reflections are preferable due to its lower complexity.

However, in this section methods based on Givens rotations, which is the basis in Jacobi-like methods, is presented, because when moving on to an update of the SVD (as we will see) only one matrix element needs to be annihilated and thus Givens is preferred over Householder's, which annihilates all but the first component of a vector.

First an introduction to Givens rotations is given, then the cyclic-Jacobi method of diagonalising a matrix using Givens rotation is presented and it is shown that the complexity is rather high.

The goal is to save computations and preserve accuracy, thus after the Jacobi method is presented we move on to present one of the approximating methods. Then this method is computed using recursion, which also reduces the computational load.

Introduction to Givens Rotations and QR Decomposition

Givens rotations is a planar rotation of a 2-dimensional vector. It rotates the plane through an angle, θ, often in order to annihilate one of the scalar entries. One can interpret this as a change of basis, the Givens rotation is a special type of identity matrix, and it turns out the Givens rotation has a neat relation to the SVD (which also changes the basis) for a 2×2 matrix.

The method of Givens rotations is often applied, because the method lends itself nicely for parallel implementation in e.g. special architectures such as the systolic array. Haykin, [39], among others has presented the method of Givens rotations in Appendix D [F] of his book. A short description is given here as an introduction to the following methods, which all relies heavily on Givens rotations. We have, after each important method, added a short calculation of the specific method. This is to facilitate the calculation of the complexity of algorithms presented farther in this section.

The Givens matrix, $\mathbf{\Theta}$ is given by

$$\mathbf{\Theta} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4.2}$$

Given a vector $\mathbf{a} = \begin{bmatrix} a_{11} & a_{21} \end{bmatrix}^T$, we can derive formulae for calculating the angle of rotation in order to annihilate the lower entry, a_{21}, of the vector $\mathbf{a}$. This annihilation scheme is done using pre-multiplication (column computations), also known as left-hand side multiplications

$$\mathbf{\Theta}^T \mathbf{a} = \mathbf{a}' = \begin{bmatrix} a'_{11} \\ 0 \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} \tag{4.3}$$

where the desired output vector, $\mathbf{a}'$, contains a zero in the lower position. Algebraically, it follows that

$$1 = c^2 + s^2 \tag{4.4a}$$

$$0 = -sa_{11} + ca_{21} \tag{4.4b}$$

solving these two equations w.r.t. c and s we obtain

$$\begin{cases} s = \sqrt{\dfrac{1}{(1 + \left(\dfrac{a_{11}}{a_{21}}\right)^2}} \\ c = s\dfrac{a_{11}}{a_{21}} \end{cases} \tag{4.4c}$$

We can compute the complexity of Eq. (4.4c) to two multiplications, two divisions, one addition, and one square root. This method is seen to transform the vector, $\mathbf{a}$, to an upper triangular form. Quite like the form resulting from a QR decomposition. Indeed, by modifying the 2×2 Givens

rotation matrix into the so-called Jacobi structure, the method of left-hand side multiplication can be used to implement a QR decomposition. The following Jacobi structure can be used to zero out any element of an $M \times N$ matrix

$$\mathbf{G}_{ij} = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & c_{ii} & \dots & s_{ij} & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & -s_{ji} & \dots & c_{jj} & & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \tag{4.5}$$

where the notation $\mathbf{G}_{ij}$ refers to that the Jacobi matrix of (4.5) contains the sine and cosine elements on the four positions given by combinations of i and j. For the QR-method, the Jacobis are applied sequentially (cyclic-by-rows) on the left hand side [32] of an $M \times N$ matrix $\mathbf{A}$ as

$$\mathbf{Q} = \mathbf{G}_{12}\mathbf{G}_{13} \dots \mathbf{G}_{n-1,n}\mathbf{G}_{nn} \tag{4.6a}$$

$$\mathbf{R} = \mathbf{Q}^T\mathbf{A} \tag{4.6b}$$

The sequence results in a full QR decomposition (QRD) of $\mathbf{A}$. From Eq. (4.6a) it can be seen that $n(n-1)/2$ positions are to be zeroed in order to transform $\mathbf{A}$ to upper triangular form. The transformation is carried out in two steps; (1) compute the rotation parameters from (4.4c) for each entry to be zeroed, and (2) update the matrix by pre-multiplication by $\mathbf{G}_{ij}$. As can be seen from the Jacobi structure only two columns needs to be updated for each annihilation. The update can be written as $\mathbf{R}_{2\times1} = \mathbf{G}_{2\times2}^T\mathbf{A}_{2\times1}$ for annihilating the (i,j)th position of matrix $\mathbf{A}$. The subscripts are to indicate, that we refer to only a part of the updating procedure. It can be seen that the updating procedure involves four multiplications and two additions per column which is updated.

	Operation	$\times$	$+$	$/$	$\sqrt{}$	FLOP's	Times executed
1.	Comp. c and s (4.4c)	2	1	2	1	6	$n(n-1)/2$
2.	$\mathbf{R}_{2\times1} = \mathbf{G}_{2\times2}^T\mathbf{A}_{2\times1}$	4	2	-	-	6	$\frac{1}{6}(2n^3 + 10n^2 + 5n + 1)$
	Total complexity						$2n^3 + 13n^2 + 2n + 1$

Table 4.1: A full QRD computed using the Jacobi-structure Givens rotation procedure in a cyclic-by-rows manner. The two steps involved are, compute the c and s parameters, and, update the affected columns. The total complexity is seen in the lower rightmost row.

In Table 4.1 the total complexity of the QRD updating scheme based on Givens rotation is shown. The subscripts $\{\cdot\}_{2\times2}$ are to remind the reader that we refer to element updating. The rightmost column tabulate the number of times the operation is executed. The complexity is assessed in FLOP's as earlier mentioned (FLOP's are not to be mistaken for FLOPS, which are FLOP per second). The total complexity (total FLOP count over all iterations) is, according to [32, p. 227], $\mathcal{O}(2n^3)$, which corresponds well to our complexity calculation.

The described method for computing the QR decomposition by applying a sequence of Givens matrices can be extended computing the singular value decomposition.

Using Givens Rotations for Computing the Singular Value Decomposition

The scheme described for the QR decomposition can be applied in a two-sided fashion known as the two-sided Jacobi. We observed that by pre-multiplication by the Givens matrix we could annihilate an element of matrix $\mathbf{A}$. Imagine we apply the same procedure using post-multiplication.

The result would be annihilation of one element in the 1×2 row vector. If we apply the same matrix on both sides to a symmetric 2×2 matrix, both elements will be annihilated. The clever trick behind two-sided Jacobi's lies in symmetrisation of the 2×2 matrix followed by pre- and post-multiplication. The down side is that the method only applies to square matrices. Luckily this can be circumvented.

In this section we intend to show a method to compute the singular value decomposition on a matrix $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}_a\mathbf{V}^T$. We start by introducing the 2×2 SVD using Givens rotations in a scheme known as two-sided Jacobi, and further extend this to the $N \times N$ case.

The two-sided Jacobi, which is a 2×2 diagonalisation of matrix $\mathbf{A}$, is given by

$$\left(\mathbf{\Theta}^T\mathbf{\Theta_1}^T\right)\mathbf{A}\mathbf{\Theta} = \mathbf{\Phi}^T\mathbf{A}\mathbf{\Theta} = \mathbf{\Sigma}_A \tag{4.7}$$

The above procedure is seen to firstly symmetrise the matrix, by $\mathbf{\Theta_1}$, and then diagonalise, by matrix $\mathbf{\Theta}$. The result is the diagonal matrix, $\mathbf{\Sigma}_A = \mathrm{diag}(d_1, d_2)$. The procedure works as follows

$$\mathbf{A}' = \mathbf{\Theta}_1^T\mathbf{A} = \begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix}^T \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} a'_{11} & a'_{12} \\ a'_{12} & a'_{22} \end{bmatrix} \qquad \text{(Symmetrisation)} \tag{4.8}$$

$$\mathbf{\Theta}^T\mathbf{A}'\mathbf{\Theta} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a'_{11} & a'_{12} \\ a'_{12} & a'_{22} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix} \qquad \text{(Diagonalisation)} \tag{4.9}$$

The symmetrisation step is similar to a step in the QR method, however, the angle (c and s parameters) are different. The entries in the matrix $\mathbf{\Theta}_1$ can be determined using the following equations

$$\begin{aligned} a'_{12} &= a'_{21} && \text{(by Eq. (4.8))} && (4.10a)\\ &= c_1a_{12} - s_1a_{22} \\ &= s_1a_{11} + c_1a_{21} \\ 1 &= c_1^2 + s_1^2 && && (4.10b) \end{aligned}$$

which we solve w.r.t. c_1 and s_1. By defining

$$\rho = \frac{a_{22} + a_{11}}{a_{12} - a_{21}} = \frac{c_1}{s_1}$$

we obtain

$$\begin{cases} s_1 = \dfrac{\mathrm{sgn}(\rho)}{\sqrt{1+\rho^2}} \\ c_1 = s_1\rho \end{cases} \tag{4.10c}$$

The total computation in determining the c_1 and s_1 is two multiplications, two divisions, three additions, and one square root. The second step, the one of diagonalisation of $\mathbf{A}'$ is somewhat more cumbersome due to the restrictions imposed by the two-sided multiplication. We start by noting that

$$\begin{aligned} a'_{12} &= 0 && \text{(by Eq. (4.9))} && (4.11a)\\ &= s \cdot (ca'_{11} + sa'_{12}) + c \cdot (ca'_{12} + sa'_{22}) \\ &= c \cdot (sa'_{11} + ca'_{12}) - s \cdot (ca'_{22} + sa'_{12}) \\ 0 &= t^2 + 2\zeta - 1 && && (4.11b) \end{aligned}$$

by introducing the definition of

$$t = \frac{s}{c} = \tan(\theta) \tag{4.11c}$$

and

$$\zeta = \frac{a'_{22} - a'_{11}}{2a'_{12}} \tag{4.11d}$$

we obtain

$$\begin{cases} t = \dfrac{\operatorname{sgn}(\zeta)}{|\zeta| + \sqrt{1+\zeta^2}}, \quad t < 1 \qquad \text{(inner rotation)} \\ c = \dfrac{1}{\sqrt{1+t^2}} \\ s = ct \end{cases} \tag{4.11e}$$

This is known as the usual second order polynomial equation solution. In the context of Givens rotations, it is referred to as *inner rotations* as the angle, θ, is restricted by $0 \leq \theta < \pi/4$, which eventually result in that the $\tan(\theta)$ resides in the inner part of the solution in Figure 4.2, viz. $-1 \leq \tan(\theta) \leq 1$. The *outer rotation* where $\pi/4 \leq \theta < \pi/2$ and $\tan(\theta) > 1$ is obtained by

$$t = -\operatorname{sgn}(\zeta)(|\zeta| + \sqrt{1+\zeta^2}), \quad t > 1 \qquad \text{(outer rotation)} \tag{4.12}$$

The two types of solutions are depicted in Figure 4.2. The parameters involved in an inner rotation can be computed using four multiplications, four additions, three divisions, and two square roots, using Eq. (4.11d) and Eq. (4.11e). For the outer rotation one division can be skipped, and the parameters can be computed by four multiplications, four additions, two divisions, and two square roots.

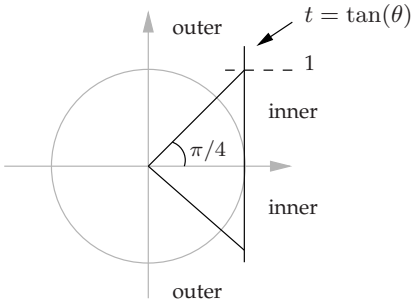


Figure 4.2: Unit circle with the tangent, $\tan(\theta)$, and the four regions. The regions are divided in *inner rotations* and *outer rotations*, which resides in $\left[-\pi/2, \pi/2\right]$ and $\left[\pm\pi/4, \pm\pi/2\right]$, respectively.

Using the two-sided Jacobi structure a square matrix can be diagonalised by choosing the right sequence. This sequence using Givens rotations is known as the *cyclic-Jacobi* algorithm and forms the singular value decomposition, $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}_A\mathbf{V}^T$.

$$\mathbf{\Sigma} = \mathbf{\Phi}^T\mathbf{A}\mathbf{\Theta} \tag{4.13a}$$

$$\mathbf{U} = \mathbf{\Phi}_{12}\mathbf{\Phi}_{13}\dots\mathbf{\Phi}_{n-1,n}\mathbf{\Phi}_{nn} = (\mathbf{\Theta_1\Theta})_{12}(\mathbf{\Theta_1\Theta})_{13}\dots(\mathbf{\Theta_1\Theta})_{n-1,n}(\mathbf{\Theta_1\Theta})_{nn} \tag{4.13b}$$

$$\mathbf{V} = \mathbf{\Theta}_{12}\mathbf{\Theta}_{13}\dots\mathbf{\Theta}_{n-1,n}\mathbf{\Theta}_{nn} \tag{4.13c}$$

This method, however, has to iterate (often referred to as *sweeps*) 4-10 times before obtaining convergence ([39, App. F p. 842]), where convergence is defined as when the off-diagonal elements of $\mathbf{\Sigma}_A$ has reached a certain precision below a given threshold [43, 39], i.e.

$$\operatorname{off}(\mathbf{\Sigma}) = \sum_i \sum_{\substack{j \\ i \neq j}} \sigma_{ij} < \epsilon \tag{4.14}$$

	Operation	$\times$	$+$	$/$	$\sqrt{}$	FLOP's	Times executed
1.	Symm. c_1 and s_1, (4.10c)	2	3	2	1	8	$n(n-1)/2$
2.	$\mathbf{A}'_{2\times 1} = \mathbf{\Theta}_{\mathbf{1}2\times 2}^T \mathbf{A}_{2\times 1}$	4	2	-	-	6	$\frac{1}{6}(2n^3+10n^2+5n+1)$
3.	Diag, c and s (4.11e)	4	4	3	2	13	$n(n-1)/2$
4.	$\mathbf{R}_{2\times 1} = \mathbf{\Theta}_{2\times 2}^T \mathbf{A}'_{2\times 1}$	4	2	-	-	6	$\frac{1}{6}(2n^3+10n^2+5n+1)$
6.	$\mathbf{\Sigma}_{1\times 2} = \mathbf{R}_{1\times 2}\mathbf{\Theta}_{2\times 2}$	4	2	-	-	6	$\frac{1}{6}(2n^3+10n^2+5n+1)$
	Total complexity						$3(2n^3+10n^2+5n+1) + \frac{21}{2}(n^2-n)$

Table 4.2: Calculation of the complexity of one sweep of a full SVD using the cyclic-by-row Jacobi algorithm. The c and s entries in the Givens rotations are computed using Eq. (4.10c) and Eq. (4.11e) using inner rotations.

The total complexity of one sweep is given in Table 4.2 which is based on Eq. (4.13)(a-c), which describe the symmetrisation step and the row-update associated with that step, the table also shows the diagonalisation stage which uses row-updates and column-updates.

The total complexity is as expected, greater than just computing the QR decomposition. However, the method is only suitable for square matrices. When combining the two methods, the QR decomposition and the SVD, any $M \times N$ matrix can be diagonalised. A QR decomposition of $\mathbf{A} = \mathbf{QR}$ generates a square, upper triangular matrix $\mathbf{R}$, which then is diagonalised by an SVD procedure.

A benefit of combining the two methods is that the SVD does not need to be computed on all off-diagonal elements, but only on the upper triangular part, which reduces the overall complexity. An implicit saving occur when working on tall matrices, as e.g. the Toeplitz data matrices presented in Chapter 2 and 3. The reduction in complexity stems from the QR decomposition. Recall that $\mathbf{A} = \mathbf{QR}$, and that $\mathbf{A}$ is $M \times N$, where we assume $M > N$. Notice that $\mathbf{R} \in \mathbb{R}^{N\times N}$, which means taking the SVD of $\mathbf{R}$ after the QR decomposition of $\mathbf{A}$ is preferable.

Several methods use variants of this combination due to the low complexity and the suitability of extending the method to subspace updating. The concept of updating refers to adding (and possibly removing) a row from the matrix under consideration. This corresponds very well with adding a new data sample, and thus a new row, to the Toeplitz data matrix.

To further reduce the complexity, Stewart [85] proposed to using a more straightforward choice of pivot indices. He, however, still makes use of the two-sided Jacobi structure, but now only working along the diagonal of an upper triangular matrix. This facilitates an algorithm with much inherent parallelism, which can be exploited in parallel signal processing architectures such as the systolic array. The upper-triangular assumption by Stewart is undesirable, but can be circumvented by applying a QR decomposing before the two-sided Jacobi. Along the lines of Stewart, Moonen et al. [64] showed how to update an SVD by interlacing QR decomposition and a two-sided Jacobi on the resulting upper triangular matrix.

Iterative diagonal-Jacobi, which is an iterative interpretation of Moonen's SVD algorithm to compute $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}_A\mathbf{V}^T$

1. $\mathbf{R} = \mathbf{Q}^T\mathbf{A}$ (from Eq. (4.6b))
2. do until convergence (off$(\mathbf{A}) < \epsilon$)

   for $i = 1, 2, \ldots, n-1$ do

   $\mathbf{R} \leftarrow \mathbf{\Phi}_{i,i+1}^T \mathbf{R} \mathbf{\Theta}_{i,i+1}$ (applying (4.13) along the diagonal)

Algorithm 6: Our pseudo algorithm for computing an SVD in an iterative manner. The algorithm is re-casted from the presentation of a recursive version in Moonen et al. [64]. The computation of the entries in the Givens matrices are done using *outer rotations* in order to ensure convergence.

In order to investigate the trade-off associated with working along the diagonals of an upper triangular matrix, we present a pseudo algorithm based on the work by Stewart [85] and Moonen et al. [64], but applied in an iterative manner (as opposed to the recursive approach in Moonen). The complexity discussion above motivates interest in these types of algorithms, as they facilitate

reduced complexity per sweep. However, we will use this pseudo algorithm to argue that the incurred loss of precision within the first number of sweeps is within an acceptable limit.

In Algorithm 6, the pseudo algorithm is seen. It consists of a complete QR decomposition by Eq. (4.6b) followed by two-sided Jacobis, see (4.13), along the diagonal of the upper triangular matrix. We will denote this algorithm the *iterative diagonal-Jacobi*.

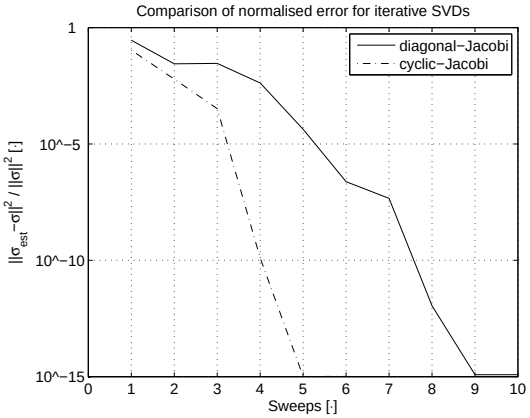


Figure 4.3: The $\mathcal{L}_2$-norm of the difference between the diagonal elements of the two methods and the true singular values obtained in MATLAB . After one sweep a relative error of less than 10% is obtained. The example is computed on a 5×5 magic square.

For the filtering operation using the multi-channel Wiener filter, we intend to use the diagonalisation of the observations matrices to compute the optimal filter (see Section 3.2.1). Therefore we have investigated the normalised difference between using the cyclic-SVD method with the proposed iterative diagonal-Jacobi from Algorithm 6. Denote σ as the singular values obtained from the full SVD procedure in MATLAB , and σ_{est} as the singular values obtained using the cyclic-Jacobi or diagonal-Jacobi algorithm. We evaluate the normalised difference, $\frac{\|\sigma_{\text{est}}-\sigma\|_2^2}{\|\sigma\|_2^2}$, as a function of the number of sweeps.

As expected the convergence rate is lower than using the diagonal-Jacobi algorithm, such that the method requires additional sweeps in order to obtain same precision as the cyclic-Jacobi algorithm. The figure indicates that for few sweeps the two methods have almost similar performance regarding approximation of the singular values. After one sweep, the error is less than 10%.

Complexity of computing an SVD using a QR decomposition and an iterative interpretation of Moonen's algorithm (see Algorithm 6) is calculated in Table 4.3. The calculation is per sweep. As expected the complexity is seen to be lower compared to the cyclic-Jacobi method. The iterative SVD step is only of $\mathcal{O}(n^2)$ complexity.

	Operation	$\times$	$+$	$/$	$\sqrt{}$	FLOP's	Times executed
1.	Comp. c and s (4.4c)	2	1	2	1	6	$\frac{1}{2}(n^2-n)$
2.	$\mathbf{A}'_{2\times1} = \mathbf{G}^T_{2\times2}\mathbf{A}_{2\times1}$	4	2	-	-	6	$\frac{1}{6}(2n^3+10n^2+5n+1)$
3.	Comp. c_1 and s_1 (4.10c)	2	3	2	1	8	$(n-1)$
4.	$\mathbf{R}_{2\times1} = \mathbf{\Theta}_1{}^T_{2\times2}\mathbf{A}'_{2\times1}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
5.	Comp. c and s (4.11e)	4	4	2	2	12	$(n-1)$
6.	$\mathbf{R}'_{2\times1} = \mathbf{\Theta}^T_{2\times2}\mathbf{R}_{2\times1}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
7.	$\mathbf{\Sigma}_{1\times2} = \mathbf{R}'_{2\times2}\mathbf{\Theta}_{2\times2}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
	Total complexity						$2n^3+22n^2+31n-19$

Table 4.3: Calculation of the complexity of the diagonal-SVD with QR decomposition, which is described in pseudo-code in Algorithm 6. The rotations used are outer rotations, see Equations (4.4c), (4.10c), (4.11e), and (4.12).

The fallout of this section is, that the diagonal-Jacobi was seen to approximate the cyclic-Jacobi w.r.t. estimating the singular values in the first sweeps. The complexity was calculated and the diagonal-Jacobi was unsurprisingly favourable over the cyclic-Jacobi. In the following section we will show how to incorporate the diagonal-Jacobi in an updating scheme, which is capable of updating, rather than re-computing the SVD after each new sample is provided.

4.1.4 Recursive Updating SVD by a QR Jacobi-Type Algorithm

Updating of the SVD is a natural extension of the iterative diagonal-Jacobi algorithm presented in previous section. An update procedure for the QR is interlaced with two-sided Jacobi's along the diagonal. The QR factorisation is especially efficient due to the Hessenberg structure involved with the data update (adding a row).

In the following paragraphs, we will introduce the QR of an Hessenberg, which forms the basis for a recursive updating SVD. This is one step towards the recursive GSVD, which will be covered in the next section.

Assume we have an upper triangular matrix, $\mathbf{R}$, a QR-update step for this matrix then corresponds to the operation of appending a row vector in the top matrix. The updated matrix holds a Hessenberg structure, which is an upper/lower triangular structure with non-zero elements adjacent (lower/upper) to the diagonal, see Appendix A. After appending a row the matrix will have the following structure

$$\mathbf{R}_{\text{append}} = \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ 0 & x & x & x \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \end{bmatrix} = \mathbf{R}_{\text{update}} \begin{bmatrix} x' & x' & x' & x' \\ 0 & x' & x' & x' \\ 0 & 0 & x' & x' \\ 0 & 0 & 0 & x' \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.15}$$

where the leftmost matrix corresponds to the QR updated matrix. For each new sample a new vector is appended and the QR needs updating. However, the Hessenberg structure facilitates a large reduction in complexity compared to computing the one-sided Jacobi-based QR decomposition described in previous section. The updated is performed along the diagonal adjacent to the main diagonal. With this observation, the QR updating can be performed in only $3n^2 + 9n$ flops instead of $\mathcal{O}(n^3)$ as in a Jacobi-based QR-decomposition. The complexity of a QR decomposition of a Hessenberg matrix is described in Table 4.4

	Operation	$\times$	$+$	$/$	$\sqrt{}$	FLOP's	Times executed
1.	Comp. c and s (4.4c)	2	1	2	1	6	n
2.	$\mathbf{A}'_{2\times 1} = \mathbf{G}^T_{2\times 2}\mathbf{A}_{2\times 1}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
	Total complexity						$3n^2 + 9n$

Table 4.4: Calculation of the complexity of a QRD updating for a Hessenberg structured matrix. The Jacobi operation is done using inner rotations.

A QR-update can then be computed by zeroing the element below the diagonal using Givens rotations on the left hand-side. The result is a large reduction in the operations needed in the update step, as seen when comparing Table 4.4 and Table 4.1. The QR update for a Hessenberg structured matrix is interesting for us only in the context of an SVD updating procedure. Next, we will show how we can exploit the Hessenberg structure to arrive at an efficient Jacobi-type algorithm for recursive computation of the SVD.

The updated singular value decomposition can be described to time index, $k+1$, as

$$\overbrace{\mathbf{A}[k+1]}^{k+1\times n} = \overbrace{\mathbf{U}[k+1]}^{k+1\times n}\overbrace{\mathbf{\Sigma}_A[k+1]}^{n\times n}\overbrace{\mathbf{V}^T[k+1]}^{n\times n} \tag{4.16}$$

which is the well-known SVD structure. The main problem of the update procedure of the SVD is that the row size of $\mathbf{U}$ grows infinitely. However, in our application, computing the multi-channel Wiener filter, it is not necessary to compute this matrix explicit [20, 64].

Now, before explaining the SVD, we need to introduce the concept of recursive updating. One could use a rectangular window for updating a matrix, that is each new sample is associated with an append of the newest vector and a removal of the oldest. It is, however, possible to use a recursive smoothing by employing a forgetting parameter. This approach eliminates the problem of storing the old vectors for later removal.

Assume the update step is done using a forgetting factor, λ, and that we append the new sample (vector), $\mathbf{a}[k+1]$, at time instance $k+1$, then updating the data matrix,$\mathbf{A}[k]$, is done as

$$\mathbf{A}[k+1] = \begin{bmatrix} \mathbf{a}[k+1] \\ \lambda\, \mathbf{A}[k] \end{bmatrix} \tag{4.17}$$

and by rewriting the updated decomposition we arrive at

$$\mathbf{A}[k+1] = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}[k] \end{bmatrix} \begin{bmatrix} \mathbf{a}[k+1]\mathbf{V}[k] \\ \lambda\, \mathbf{\Sigma}_A[k] \end{bmatrix} \mathbf{V}^T[k] \tag{4.18}$$

where in the middle expression, the diagonal matrix $\mathbf{\Sigma}_A$ has been forced one row down and rendered the entire matrix Hessenberg. Adding a QR update step of this Hessenberg structured matrix we produce the basis for the SVD of a triangular matrix

$$\mathbf{A}[k+1] = \underbrace{\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}[k] \end{bmatrix} \mathbf{Q}[k+1]}_{\mathbf{U}[k+1] \quad k+1\times n+1} \underbrace{\begin{bmatrix} \overbrace{\mathbf{R}[k+1]}^{n\times n} \\ \underbrace{\mathbf{0}}_{1\times n} \end{bmatrix}}_{n+1\times n} \underbrace{\mathbf{V}^T[k]}_{n\times n} \tag{4.19}$$

$$= \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}[k] \end{bmatrix} \mathbf{Q}[k+1] \underbrace{\begin{bmatrix} \mathbf{I}_{n\times n} \\ \mathbf{0} \end{bmatrix}}_{n+1\times n} \mathbf{R}[k+1]\, \mathbf{V}^T[k] \tag{4.20}$$

$$= \tilde{\mathbf{U}}[k+1]\tilde{\mathbf{R}}[k+1]\, \mathbf{V}^T[k] \tag{4.21}$$

where we see that the rightmost matrix, $\mathbf{V}[k]$, is not affected by the update. The matrix $\mathbf{U}[k]$ is modified to $\tilde{\mathbf{U}}[k+1]$, which holds the $\mathbf{Q}$ factorisation from the QR updating step. It only remains to diagonalise the upper triangular matrix, $\tilde{\mathbf{R}}[k+1]$, which can be done by the two-sided diagonal-Jacobi algorithm from previous section. It is worth noting that $\tilde{\mathbf{U}}[k+1]$ and $\mathbf{Q}$ is not computed explicitly.

The described updating SVD is in a class of *QR Jacobi-type* algorithms [70], which are traditionally employed in applications were updating or tracking is of prominent interest. The algorithms are all characterised by low complexity.

The complexity of the SVD updating algorithm is described in Table 4.4. The calculation is split in two. The QR update of an Hessenberg structured matrix followed by the diagonal-Jacobi type algorithm. The iterations from the diagonal-Jacobi are now done over time, which eventually means, that the algorithm is tracking a matrix which changes over time, much like the RLS schemes known in adaptive filtering. In the table, it can be observed that a reduction of more than $2n^3$ operations per update is obtained, yielding a complexity of only $\mathcal{O}(n^2)$.

Having described the QR Jacobi-type SVD and shown that a relatively large amount of computations are saved, we are ready to introduce the generalised singular value decomposition (GSVD) and an efficient algorithm, which has its roots in the described QR Jacobi-type SVD.

	Operation	$\times$	$+$	$/$	$\sqrt{}$	FLOP's	Times executed
1.	Comp. c and s (4.4c)	2	1	2	1	6	n
2.	$\tilde{\mathbf{R}}_{2\times1} = \mathbf{Q}_{2\times2}^T \boldsymbol{\Sigma}'_{2\times1}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
3.	Comp. c_1 and s_1 (4.10c)	2	3	2	1	8	$(n-1)$
4.	$\mathbf{R}'_{2\times1} = \boldsymbol{\Theta}_{\mathbf{1}2\times2}^T \tilde{\mathbf{R}}_{2\times1}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
5.	Comp. c and s (4.11e, 4.12)	4	4	2	2	12	$(n-1)$
6.	$\mathbf{R}'_{2\times1} = \boldsymbol{\Theta}_{2\times2}^T \mathbf{R}'_{2\times1}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
7.	$\boldsymbol{\Sigma}_{1\times2} = \mathbf{R}'_{2\times2} \boldsymbol{\Theta}_{2\times2}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
	Total complexity						$12n^2+38n-20$

Table 4.5: Calculation of the complexity of one time-update of the QR Jacobi-type SVD proposed by Moonen et al. [64]. The method is based on outer rotations for the Jacobi-steps.

4.1.5 Recursive Updated Generalised Singular Value Decomposition with Implicit Matrix Inversion

The updating procedure for the SVD in previous section forms the basis for an GSVD updating procedure with implicit matrix inversion. Generally the GSVD problem can easily be solved insofar inverting a large matrix is of no concern. However, this brings about a number of computational as well as numerical instability problems, and is generally not the preferred solution [32]. The method proposed by Paige [69], which implicitly forms the inverse matrix, has resulted in numerous variations of stable algorithms for computing the GSVD. The resulting algorithms are generally well suited for parallel implementation [63, 54]

In this section we first introduce the generalised singular value decomposition and our notation. Then, by implicit matrix inversion, we extend the SVD updating scheme from previous section to apply to the GSVD.

The SVD and GSVD are related mathematical problems. The SVD relates to least squares. The generalised singular value decomposition relates to constrained least squares. By defining the rectangular $p \times m$ matrix $\mathbf{Y}$ and the rectangular $q \times m$ matrix $\mathbf{B}$, the GSVD of $\mathbf{Y}$ and $\mathbf{B}$ is defined as [32]

$$\begin{cases} \mathbf{U}_Y^T \mathbf{Y} \mathbf{Q} = & \boldsymbol{\Sigma}_Y \mathbf{R} \\ \mathbf{V}_B^T \mathbf{B} \mathbf{Q} = & \boldsymbol{\Sigma}_B \mathbf{R} \end{cases} \tag{4.22}$$

where $\boldsymbol{\Sigma}_Y = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_m)$ and $\boldsymbol{\Sigma}_B = \mathrm{diag}(\beta_1, \beta_2, \ldots, \beta_m)$. We have left-out the time indices, which we used in describing the recursive procedures, to digress into the definition of the GSVD. By reordering (4.22) and equating $\mathbf{R}$, we obtain following

$$\mathbf{U}_Y^T \mathbf{Y} \mathbf{Q} = \left(\boldsymbol{\Sigma}_Y \boldsymbol{\Sigma}_B^{-1}\right) \mathbf{V}_B^T \mathbf{B} \mathbf{Q} \tag{4.23}$$

which shows, that $\mathbf{U}_Y^T \mathbf{Y} \mathbf{Q}$ and $\mathbf{V}_B^T \mathbf{B} \mathbf{Q}$ have parallel rows with row-scaling factor $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_Y \boldsymbol{\Sigma}_B^{-1} = \mathrm{diag}(\alpha_1/\beta_1, \alpha_2/\beta_2, \ldots, \alpha_m/\beta_m)$. These quotients are called the *generalised singular values*, $\sigma_i = \alpha_i/\beta_i$.
By reordering (4.23) we obtain

$$\mathbf{U}_Y^T \left(\mathbf{Y}\mathbf{B}^{-1}\right) \mathbf{V}_B = \boldsymbol{\Sigma} \tag{4.24}$$

which emphasises, that the GSVD of $\mathbf{Y}$ and $\mathbf{B}$ corresponds to the SVD of $\mathbf{Y}\mathbf{B}^{-1}$ [54]. Lastly, one can rewrite equation (4.22) to a widely used form, which resembles two separate SVDs, as

$$\begin{cases} \mathbf{Y} = & \mathbf{U}_Y \boldsymbol{\Sigma}_Y \mathbf{Q}^T \\ \mathbf{B} = & \mathbf{U}_B \boldsymbol{\Sigma}_B \mathbf{Q}^T \end{cases} \quad \text{where} \quad \mathbf{Q}^T = \mathbf{R}\mathbf{V}^T \tag{4.25}$$

Having upper triangular matrices, Paige [54, 69] showed how to compute the GSVD with implicit inversion of $\mathbf{B}$, which is numerically preferable.

$$\mathbf{C}_{ij} = \mathbf{Y}_{ij} (\mathbf{B}_{ij})^{-1}, \quad (\mathbf{B}^{-1})_{ij} \triangleq (\mathbf{B}_{ij})^{-1} \tag{4.26}$$

where (i, j) indicate pivot indices in $\mathbf{C}$, i.e. $\mathbf{C}_{i,j}$ is a 2×2 sub-matrix containing the pivots of $\mathbf{C}$ at positions, (i, i), (i, j), (j, i), and (j, j) . This property holds only for $\mathbf{Y}$ and $\mathbf{B}$ upper triangular matrices.

Using the SVD updating procedure from previous section, we re-cast the GSVD formalisation to a sample-by-sample recursive updating algorithm. The algorithm and results from the following section forms the basis for a C-based floating-point implementation.

Updating the GSVD

The method for updating (and computing) the generalised singular value decomposition is, as was the SVD updating, based on the QR Jacobi-like algorithm of the preceding section. Recall that the savings were based on the Hessenberg structure assumption. This we will exploit in the GSVD updating, as well. Compared to the SVD updating, the most distinct difference lies in the fact, that we are jointly diagonalising two matrices instead of one. We start by defining the updating of two data matrices, $\mathbf{Y}[k]$ and $\mathbf{B}[k]$, at time instance, k, as

$$\mathbf{Y}[k+1] = \begin{bmatrix} \mathbf{y}[k+1] \\ \lambda_Y \, \mathbf{Y}[k] \end{bmatrix} \qquad \mathbf{B}[k+1] = \begin{bmatrix} \mathbf{b}[k+1] \\ \lambda_B \, \mathbf{B}[k] \end{bmatrix} \tag{4.27}$$

The GSVD of theses matrices is then given by

$$\begin{cases} \overbrace{\mathbf{Y}[k+1]}^{k+1\times n} & = \overbrace{\mathbf{U}_Y[k+1]}^{k+1\times n} \overbrace{\mathbf{R}_Y[k+1]}^{n\times n} \overbrace{\mathbf{V}^T[k+1]}^{n\times n} \\ \mathbf{B}[k+1] & = \mathbf{U}_B[k+1] \, \mathbf{R}_B[k+1] \, \mathbf{V}^T[k+1] \end{cases} \tag{4.28}$$

where the matrices $\mathbf{R}_Y$ and $\mathbf{R}_B$, are square upper triangular matrices. Notice that the diagonal matrices $\mathbf{\Sigma}_A$ and $\mathbf{\Sigma}_B$ from Eq. (4.24), are not computed explicitly. Rather the aforementioned upper triangular matrices are row parallel, thus the quotient of the diagonal of $\mathbf{R}_Y$ and $\mathbf{R}_B$ equals the generalised singular values. Using Eq. (4.25) we can rewrite Eq. (4.28) as

$$\begin{cases} \mathbf{Y}[k+1] & = \mathbf{U}_Y[k+1]\mathbf{\Sigma}_Y[k+1]\mathbf{R}[k+1]\mathbf{V}^T[k+1] \\ \mathbf{B}[k+1] & = \mathbf{U}_B[k+1]\mathbf{\Sigma}_B[k+1]\mathbf{R}[k+1]\mathbf{V}^T[k+1] \end{cases} \tag{4.29}$$

which makes explicit, that the quotient of the upper triangular matrices, $\mathbf{R}_Y$ and $\mathbf{R}_B$, equals the generalised singular values, and that $\mathbf{R}_Y = \mathbf{\Sigma}_Y\mathbf{R}$ and $\mathbf{R}_B = \mathbf{\Sigma}_B\mathbf{R}$.

The updating, or tracking, procedure consists of three main steps. The first two are similar to the update procedure in the SVD case, while the last step concerns computation of the matrix $\mathbf{V}$ in Eq. (4.28). In Algorithm 7, the entire GSVD-updating step for adding a new row (sample) is outlined. We have made extensive use of the $\{\cdot\}_{i,j}$-notation, explained earlier, as the algorithm relies heavily on looping along the pivots (diagonal) of $\mathbf{R}_Y[k+1]$ and $\mathbf{R}_B[k+1]$. The algorithm as such has not previously, to our knowledge, seen so explicitly written. The algorithm is found described in Moonen et. al [64, 63] and with application to the multi-channel Wiener filter in Doclo et al. [20].

The algorithm stands by itself, however, in order to explain the algorithm in short steps, we will now step through the algorithm in words. Update the upper triangular matrices, $\mathbf{R}_Y[k]$ and $\mathbf{R}_B[k]$ (if sample is determined noise-dominated) with new sample (vector), which renders the updated matrix, $\mathbf{R}'_Y[k+1]$, Hessenberg. Re-triangularise the matrix and form the matrix, $\big(\mathbf{Y}[k+1]\mathbf{B}^{-1}[k+1]\big)$, implicitly using the equality from Paige [69], which only holds for triangular matrices. Diagonalise the resulting 2×2 matrix using an SVD updating step, as explained in previous section, and update $\tilde{\mathbf{R}}_Y[k+1]$ and $\tilde{\mathbf{R}}_B[k+1]$ (disregarding the VAD marking) to have parallel rows. The matrices are now (again) upper Hessenberg and needs to be re-triangularised. The updating step is performed from the right (right-hand one-sided Jacobi update), as opposed

GSVD sample-by-sample updating, due to Moonen et al. [63] (`mcwf_recursive_gsvd_mat.m`) see Appendix D.

New sample

- $\tilde{\mathbf{y}}[k+1] = \mathbf{y}[k+1]\mathbf{V}[k]$
- update the upper triangular matrix $\mathbf{R}_Y$ (and $\mathbf{R}_B$, if VAD mark is true)

$$\mathbf{R}'_Y[k+1] = \begin{bmatrix} \tilde{\mathbf{y}}[k+1] \\ \lambda\,\mathbf{R}_Y[k] \end{bmatrix}$$

- re-triangularise matrix $\mathbf{R}'_Y[k+1]$ (and $\mathbf{R}'_B[k+1]$) by QR updating along main diagonal

  for i = 1 to $n-1$

$$\tilde{\mathbf{R}}^Y_{i,i+1}[k+1] = \mathbf{\Theta}^T_{i,i+1}\mathbf{R}'^Y_{i,i+1}[k+1]$$

  end

for i = 1 to $n-1$

Stage 1 - step 1

- find $\mathbf{U}^C_{i,i+1}$ symmetrising the 2×2 matrix $\mathbf{C}_{i,i+1}$

$$\mathbf{C}_{i,i+1} = \tilde{\mathbf{R}}^Y_{i,i+1}\cdot(\tilde{\mathbf{R}}^B_{i,i+1})^{-1} = \begin{bmatrix} \tilde{R}^Y_{i,i} & \tilde{R}^Y_{i,i+1} \\ 0 & \tilde{R}^Y_{i+1,i+1} \end{bmatrix}\begin{bmatrix} \tilde{R}^B_{i,i} & \tilde{R}^B_{i,i+1} \\ 0 & \tilde{R}^B_{i+1,i+1} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{\tilde{R}^Y_{i,i}}{\tilde{R}^B_{i,i}} & \frac{-\tilde{R}^B_{i,i+1}\cdot\tilde{R}^Y_{i,i}}{\tilde{R}^B_{i+1,i+1}\cdot\tilde{R}^B_{i,i}} + \frac{\tilde{R}^Y_{i,i+1}}{\tilde{R}^B_{i+1,i+1}} \\ 0 & \frac{\tilde{R}^Y_{i+1,i+1}}{\tilde{R}^B_{i+1,i+1}} \end{bmatrix}$$

- update $\tilde{\mathbf{C}}_{i,i+1} = \mathbf{U}^C_{i,i+1}\mathbf{C}_{i,i+1}$

Stage 1 - step 2

- diagonalise $\tilde{\mathbf{C}}_{i,i+1}$ by annihilating $\tilde{c}_{i+1,i}$ and $\tilde{c}_{i,i+1}$ using outer rotations, such that

$$\mathbf{\Theta}^T_{i,i+1}\tilde{\mathbf{C}}_{i,i+1}\mathbf{\Theta}_{i,i+1} = \mathbf{\Sigma}_{i,i+1}$$

  where $\mathbf{\Sigma}$ is a diagonal matrix containing the generalised singular values.

- update $\mathbf{R}'^Y_{i,i+1}[k+1]$ and $\mathbf{R}'^B_{i,i+1}[k+1]$ using $\mathbf{U}^C_{i,i+1}$ and $\mathbf{\Theta}_{i,i+1}$

$$\mathbf{R}'^Y_{i,i+1}[k+1] = \left(\mathbf{\Theta}_{i,i+1}\right)^T\left(\mathbf{U}^C_{i,i+1}\right)^T\tilde{\mathbf{R}}^Y_{i,i+1}[k+1]$$
$$\mathbf{R}'^B_{i,i+1}[k+1] = \left(\mathbf{\Theta}_{i,i+1}\right)^T\tilde{\mathbf{R}}^B_{i,i+1}[k+1]$$

- The two triangular matrices, $\mathbf{R}'^Y_{i,i+1}[k+1]$ and $\mathbf{R}'^B_{i,i+1}[k+1]$, now have parallel rows, but has become Hessenberg structured (if considered outside the for loop).

Stage 2

- Force Hessenberg $\mathbf{R}'^Y_{i,i+1}[k+1]$ and $\mathbf{R}'^B_{i,i+1}[k+1]$ upper triangular
  - annihilate $R'^Y_{i+1,i}$ and $R'^B_{i,i+1}$ (the Hessenberg pivots) using givens matrix $\mathbf{Q}$ and update $\mathbf{V}[k]$
  - if $(R'^Y_{i,i} == 0)$ compute $\mathbf{Q}$ from $\mathbf{R}^B_{i,i+1}$ otherwise use $\mathbf{R}^Y_{i,i+1}$ (notice the left-hand side Jacobi)

$$\mathbf{R}^Y_{i,i+1}[k+1] = \mathbf{R}'^Y_{i,i+1}[k+1]\mathbf{Q}$$
$$\mathbf{R}^B_{i,i+1}[k+1] = \mathbf{R}'^B_{i,i+1}[k+1]\mathbf{Q}$$
$$\mathbf{V}_{i,i+1}[k+1] = \mathbf{V}_{i,i+1}[k]\mathbf{Q}$$

  End of GSVD updating step for one pivot position. A new sample has been added and upper triangular structure restored - matrix, $\mathbf{V}_{i,i+1}[k+1]$ is also updated.

end

Algorithm 7: Recursive GSVD updating based on QR updating interlaced with two-sided Jacobi's. Computations are done along the diagonal due to the Hessenberg structured matrices.

	Operation	$\times$	$+$	$/$	$\sqrt{}$	FLOP's	Times executed
1.	$\mathbf{y}[k+1]\mathbf{V}[k]$	n	$n-1$	-	-	$2n-1$	$n(\times 2)$
2.	Multiply by λ	1	-	-	-	1	$n^2(\times 2)$
3.	QR updates, Givens rotation	2	1	2	1	6	$n(\times 2)$
4.	$\tilde{\mathbf{R}}^Y_{2\times 1} = (\mathbf{Q}^Y_{2\times 2})^T\mathbf{R}^Y_{2\times 1}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
5.	$\tilde{\mathbf{R}}^B_{2\times 1} = (\mathbf{Q}^B_{2\times 2})^T\mathbf{R}^B_{2\times 1}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
6.	SVD, Givens rot. (incl. $\mathbf{C}$), Stage 1	2 (+3)	3 (+1)	2 (+3)	1	8 (15)	$(n-1)$
7.	$\mathbf{C}'_{2\times 1} = (\mathbf{U}^C_{2\times 2})^T\mathbf{C}_{2\times 1}$	6	2	-	-	8	n
8.	SVD, Givens rot., Stage 2	4	4	2	2	12	$(n-1)$
9.	$\mathbf{U}^Y_{2\times 1} = (\mathbf{U}^C_{2\times 2})^T\mathbf{U}^B_{2\times 2}$	8	4	-	-	12	n
10.	$\mathbf{R}'_{Y,2\times 1} = (\mathbf{U}^Y_{2\times 2})^T\tilde{\mathbf{R}}^Y_{2\times 1}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
11.	$\mathbf{R}'_{B,2\times 1} = (\mathbf{U}^B_{2\times 2})^T\tilde{\mathbf{R}}^B_{2\times 1}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
12.	GSVD part (find $\mathbf{Q}$)	2	1	2	1	6	n
13.	$\mathbf{R}^Y[k+1]_{1\times 2} = \mathbf{R}'^Y_{1\times 2}\mathbf{Q}_{2\times 2}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
14.	$\mathbf{R}^B[k+1]_{1\times 2} = \mathbf{R}'^B_{1\times 2}\mathbf{Q}_{2\times 2}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
15.	$\mathbf{V}[k+1]_{1\times 2} = \mathbf{V}[k]_{1\times 2}\mathbf{Q}_{2\times 2}$	4	2	-	-	6	$\frac{1}{2}(n^2+n)$
	Total complexity						$27n^2+84n-27$

Table 4.6: Calculation of the total complexity of an GSVD update operation by the GSVD-updating algorithm proposed by Moonen et al. [63]. The calculation is divided in four parts, which can be recognised in Algorithm 7 on page 114.

to the left-hand update in the QR updating in preceding section. The QR updating is also applied to the $\mathbf{V}[k]$ matrix, which thereby is updated. This concludes the GSVD-update step.

The complexity for an updating GSVD procedure is described in Table 4.6. The SVD step differs slightly when using it to compute the GSVD. The numbers in parentheses are indicating the operations needed to compute $\mathbf{C}_{2\times 2}$ implicitly. Another difference is the row-updates, which is performed on both matrices. Lastly, the GSVD step is computed to maintain an upper triangular structure. The computational cost is as expected slightly higher than for an SVD update on only one matrix.

In Algorithm 7 we have distinguished between noise-dominated and speech-dominated samples, because $\mathbf{R}_B[k]$ only needs a row added when the sample is determined to be noise-dominated. In Doclo et al. [20] they propose to only update one of the matrices to keep a so-called balanced design. This means, keeping equal amount of computations in each loop, disregarding the VAD marking. This is interesting for the case of efficient parallel implementation on e.g. systolic arrays, as noted by Moonen et. al [63]. This leads to a slight degradation in performance, as the observation data matrix is not kept up-to-date. We have not considered this balanced design (although the differences are subtle), thus the calculated complexity in Table 4.6 is based on the worst-case amount of operations per sample, viz. both observation and noise data matrix update.

Having presented an algorithm for computing the generalised singular value decomposition on a recursive updated sample-by-sample basis, we have found a feasible solution to the problem of finding the optimal filter, see Section 3.2.1. However, one still needs to do the actual filtering using this filter. This we will cover in the following section.

4.2 Recursive GSVD-Based Multi-Channel Wiener Filtering

The multi-channel Wiener filter at time instance $k+1$ can be computed using the generalised singular value decomposition of the two data matrices, $\mathbf{Y}[k]$ and $\mathbf{B}[k]$, as derived in Section 3.2.1

on page 82 to

$$\mathbf{W}_{\mathrm{WF}}[k] = \mathbf{Q}^{-T}[k]\,\mathrm{diag}\left(1 - \frac{p\,\sigma^2_{b,i}[k]}{q\,\sigma^2_{y,i}[k]}\right)\mathbf{Q}^{T}[k] \tag{4.30}$$

As pointed out by Doclo et al. [20], the upper triangular, row-parallel matrices, $\mathbf{R}_Y$ and $\mathbf{R}_B$, are sufficient to find the generalised singular values needed in the computation of the optimal filer matrix. By Eq. (4.29), we can see that

$$\frac{R^B_{i,i}[k]}{R^Y_{i,i}[k]} = \frac{\Sigma^B_{i,i}[k]}{\Sigma^Y_{i,i}[k]} = \frac{\sigma_{b,i}[k]}{\sigma_{y,i}[k]} \tag{4.31}$$

where the latter relation is to elucidate the relationship to Eq. (4.30). We define the relation between the forgetting factor, λ_Y, and the rectangular windowing factor, p, as $\lambda_Y = \big(1 - 1/p\big)$. Using this definition, Eq. (4.28) and Eq. (4.29), we arrive at the following expression for the optimal (Wiener) filter

$$\mathbf{W}_{\mathrm{WF}}[k] = \mathbf{V}[k]\mathbf{R}_Y^{-1}[k]\,\mathrm{diag}\left(1 - \frac{\big(1-\lambda_B\big)\,\big(R^B_{i,i}[k]\big)^2}{\big(1-\lambda_Y\big)\,\big(R^Y_{i,i}[k]\big)^2}\right)\mathbf{R}_Y[k]\mathbf{V}^T[k] \tag{4.32}$$

Recall from Section 3.2 that in order to compute a set of filters for the MCWF only one column of $\mathbf{W}_{\mathrm{WF}}[k]$ is needed. Assume we are interested in the ith column of $\mathbf{W}_{\mathrm{WF}}[k]$ and let us denote that column matrix by $\mathbf{w}_{i,\mathrm{WF}}[k]$. By re-arrangement, this filter can be computed by

$$\mathbf{R}_Y[k]\underbrace{\mathbf{V}^T[k]\mathbf{w}_{i,\mathrm{WF}}[k]}_{tildemathbfw[k]} = \underbrace{\mathrm{diag}\left(1 - \frac{\big(1-\lambda_B\big)\,\big(R^B_{i,i}[k]\big)^2}{\big(1-\lambda_Y\big)\,\big(R^Y_{i,i}[k]\big)^2}\right)\mathbf{R}_Y[k]\mathbf{v}_i[k]}_{\tilde{\mathbf{v}}[k]} \tag{4.33}$$

where $\mathbf{v}_i[k]$ denotes the ith column vector of $\mathbf{V}^T[k]$. We can circumvent the inversion of $\mathbf{R}_Y[k]$ by exploiting its upper triangular structure and solve for $\mathbf{w}_{i,\mathrm{WF}}[k]$ by back-substitution. First compute

$$\mathbf{R}_Y[k]\tilde{\mathbf{w}}[k] = \tilde{\mathbf{v}}[k] \tag{4.34}$$

then solve w.r.t. $\tilde{\mathbf{w}}[k]$ by back-substitution and

$$\mathbf{w}_{i,\mathrm{WF}}[k] = \mathbf{V}[k]\tilde{\mathbf{v}}[k] \tag{4.35}$$

which gives the optimal filter in the ith column of $\mathbf{W}_{\mathrm{WF}}[k]$. In Algorithm 8 a procedure for computing the back-substitution is shown.

The complexity of computing the filter is summarised in Table 4.7. The total complexity for computing a sample using the recursive GSVD-based MCWF is also put into the table to give an overview of the final complexity. It is seen that computing the filter output only takes up about one-sixth of the total complexity.

By assuming 40 filter taps per channel and 2 channels, $n = 2 \cdot 40 = 80$, and a sampling frequency of $8\,\mathrm{kHz}$, we obtain a total complexity per second

$$f_s\big(32n^2 + 87n - 27\big)\Big|_{n=80} = 1.7\ \mathrm{GFLOPS} \tag{4.36}$$

this is as stated earlier an underestimate, since all operations, square roots and divisions are counted as if they only take one FLOP operation. The FLOPS count has been reduced by a factor of 735 from 1250 GFLOPS to 1.7, thus in order to ensure that the approximation keeps the performance close to the reference-GSVD different simulations have been performed using the recursive version.

Back-substitution algorithm due to Higham [43, p.140]

1. $w_n = \tilde{v}_n / R^Y_{n,n}$
2. for $i = n-1 : -1 : 1$

   $tmp = \tilde{v}_i$

   for $j = i+1 : n$

   $tmp = tmp - R^Y_{i,j} w_j$

   end

   $w_i = tmp / R^Y_{i,i}$
3. end

Algorithm 8: Pseudo-code which shows how back substitution can be computed on the upper triangular matrix, $\mathbf{R}^Y$, and vector $\tilde{\mathbf{v}}$.

	Operation	$\times$	$+$	$/$	$\sqrt{}$	FLOP's	Times executed
1.	Computing $\tilde{\mathbf{v}}$	n	$n+1$	1	-	$2n+2$	n
2.	Back-subst., inner loop	1	1	-	-	2	$\frac{1}{2}(n^2-n)$
3.	Back-subst., outer loop	-	-	1	-	1	n
4.	$\mathbf{w}_{i,\mathrm{WF}} = \mathbf{V}\tilde{\mathbf{w}}$	1	1	-	-	2	n^2
5.	Filtering, $\hat{s}[k+1] = \mathbf{y}[k+1]^T \mathbf{w}_{i,\mathrm{WF}}$	1	1	-	-	2	n
	Total complexity						$5n^2+3n$
	Total complexity of the GSVD						$27n^2+84n-27$
	Total complexity including the GSVD						$32n^2+87n-27$

Table 4.7: Operations overview of computing the Wiener filter and the overall complexity for the recursive GSVD-based multi-channel Wiener filter including computing the filtered output.

4.2.1 Results using a Recursive GSVD-Based Multi-Channel Wiener Filter

In this section we will investigate the performance w.r.t. noise reduction and speech distortion of an C-based implementation of the recursive GSVD-procedure in combination with filtering using the resulting optimal filter. We will compare the results with the performance of the full GSVD (our golden reference data) computed using our MATLAB model, as described in Section 3.2.3.

The forgetting factors, λ's, determines the amount of timely smoothing w.r.t. updating of the data matrix, i.e. they roughly corresponds to the functionality of the p and q factors used to determine the height of the observation and noise matrices, $\mathbf{Y}[k]$ and $\mathbf{B}[k]$. We have calculated the forgetting factors to have approximate the same smoothing as p and q by

$$\lambda_Y = \lambda_B = 1 - 1/p = 1 - 1/4000 = 0.99975 \tag{4.37}$$

The filter length has been set to 40 taps per channel. The standard test signal of a female voice uttering "*Good service is rewarded with big tips*" has been concatenated with itself to allow the recursive version to settle, i.e. avoid initial ringing effects when adapting the triangular matrices $\mathbf{R}_Y$ and $\mathbf{R}_B$. No convolutional noise has been added to the observation, only direct-path observations.

The results in Table 4.8 is obtained using two different sweep-settings. This refers back to the discussion on whether trading off accuracy (speed of convergence) for speed was acceptable. In that section we discussed the incurred loss of precession by updating along the diagonal only compared to doing an update by the cyclic-by-row scheme. It can be observed in the table, that sweeping more than once is unnecessary when considering noise reduction and speech distortion measured by SNR and WSSM, respectively.

Measure \ Method	Observation	full GSVD	Rec. GSVD 1 sweep	Rec. GSVD 5 sweep
SNR	5.0	20.7	20.5	20.9
WSSM	44.4	30.7	27.5	27.9

Table 4.8: Performance for the multi-channel Wiener filter when applied the standard test signal at an SNR of 5 dB and white additive noise. The full GSVD (by means of MATLAB simulations) yields little difference when compared to the recursive GSVD-based MCWF. Additional sweeps, in this case 5, are seen to gain little extra performance.

The performance results in Table 4.8 show that the recursive GSVD-based MCWF has similar performance to the full GSVD. Actually, the performance is seen to be slightly better for several sweeps of each updating step in the recursive version. This is due to a slightly larger window size, i.e. the approximated p and q sizes by the forgetting factor, λ, yields better, but non-significant performance. The table also shows that additional sweeps in the GSVD approximation does not lead to an improvement.

4.2.2 Conclusion

Givens rotations were introduced to describe QRDs and SVDs. A full QRD was shown to have a complexity of $\mathcal{O}(n^3)$. The cyclic-Jacobi based method also had a complexity of that order. Combining Moonen's method with the QR reduced the complexity of the SVD step to $\mathcal{O}(n^2)$. Modifying the SVD procedure to a diagonal-SVD when using the QR update procedure the complexity was lowered to an order of $\mathcal{O}(n^2)$. Paige's method on how to compute the GSVD was then described and the recursion based GSVD was presented. Introducing the recursion-based GSVD reduced the computational cost with a factor of $1250/1.7 = 735$. Even though this many operations were saved for computing one output sample, the performance of the functionality was not changed. In addition it was shown that using only one sweep in the GSVD step, the complete algorithm maintained similar performance as our golden reference from MATLAB.

4.3 Sub-Sampling and ANC Post-Processing Stage

Even though, the complexity of the MCWF is reduced by a factor of more than 700 in the previous section, the complexity can be further reduced. We will investigate the possibility of imposing an additional reduction.

Firstly, we will examine the effect of computing the GSVD only every n_s samples, referred to as sub-sampling. We expect due to stationarity, discussed in Section 3.2.2, that sub-sampling is feasible. This is described in the following section.

Secondly, we expect to see a decrease in performance when introducing sub-sampling due to sub-sampling of the underlying stochastic process. In order to overcome this effect an adaptive noise cancellation (ANC) stage is added such that we obtain a GSC structure, where the MCWF acts as the beamformer.

4.3.1 Sub-Sampling the MCWF

The sub-sampling term and the effect on the MCWF output is explained in this section. Recall the recursion-based GSVD described in Table 4.7. Every new incoming sample 4 steps were conducted

1. The updating step (adding a row)
2. Update the GSVD
3. Compute the filter
4. Filtering

The last step is computed for every sample in order to obtain an output sample. The filter computation only needs to be computed every time we wish to update the filter and the first two steps has to be executed for every incoming sample if the method should approximate the full GSVD.

The idea of sub-sampling is to only perform the first two steps every n_s time a new sample arrive. Computing the filter obviously becomes redundant in between the n_s samples and is thus also only updated when a new GSVD approximation is ready[1].

The observation matrix which is used in the computation of the MCWF was described in Section 3.2.1. Modifying the observation matrix to only update (append a new row) every n_s sample changes the content of the matrix to

$$\mathbf{Y}_{\text{sub}}[k] = \begin{bmatrix} \mathbf{y}^T[k - p\,n_s + 1] \\ \vdots \\ \mathbf{y}^T[k - n_s] \\ \mathbf{y}^T[k] \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1[k - p\,n_s + 1] & \mathbf{y}_2[k - p\,n_s + 1] & \dots & \mathbf{y}_M[k - p\,n_s + 1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_1[k - n_s] & \mathbf{y}_2[k - n_s] & \dots & \mathbf{y}_M[k - n_s] \\ \mathbf{y}_1[k] & \mathbf{y}_2[k] & \dots & \mathbf{y}_M[k] \end{bmatrix} \tag{4.38}$$

thus p is the unaltered row size, but now spans over $p\,n_s$ samples and not p. Assuming spatial stationarity as discussed in Section 3.2.2, the auto-correlation is not altered but is *sub-sampled*. The interesting part is that the auto-correlation sequence for time lag, $0, n_s, 2n_s, \dots$ is based on the same samples as usual, thus these data are not modified. The time lags in between n_s lags are skipped, though. Therefore the term "sub-sampling". The effect of modifying the observation matrix (and eventually also the noise matrix) is illustrated in (4.39) which shows the sub-sampled

[1] In the full GSVD it is only necessary to update the GSVD every time one wants a new filter. The filtering process is then also computed frame-wise, why it is often referred to as *batching*. However, in the complexity-reduced, recursion-based GSVD, batching is infeasible. The recursive method needs to update every sample due to the update procedure based on the QRD of a Hessenberg matrix.

auto-correlation matrix,

$$\mathbf{Y}_{sub}^H \mathbf{Y}_{sub} = \tilde{\mathbf{R}}_{yy} = \begin{bmatrix} \tilde{r}_{yy}(0) & \tilde{r}_{yy}(n_s) & \dots & \tilde{r}_{yy}(M-1) \\ \tilde{r}_{yy}(n_s) & \tilde{r}_{yy}(0) & \dots & \tilde{r}_{yy}(M-1-n_s) \\ \tilde{r}_{yy}(2n_s) & \tilde{r}_{yy}(n_s) & \dots & \tilde{r}_{yy}(M-1-2n_s) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{r}_{yy}(M-1) & \tilde{r}_{yy}(M-1-n_s) & \dots & \tilde{r}_{yy}(0) \end{bmatrix} \tag{4.39}$$

The effect of updating the GSVD only every n_s sample then affects the filter too.The filter now longer uses any information of the statistics between time lags 0 and n_s. The expectation is that the spatial information is almost unaltered in the statistics, but some temporal information is lost due to the "missing" time lags. In order to overcome this short-coming additional noise reduction is performed using a GSC-structure with an adaptive noise cancellation stage.

The benefit of using sub-sampling is seen from an example of e.g. using a factor 10. Then the complexity reduces to $1700/10 = 170$ MFLOPS, which can compute in real-time[2] on a Celeron 1 GHz.

4.3.2 Merging the MCWF and the GSC-Structure

The basic idea in introducing extra noise reduction in order to cancel the negative effect of sub-sampling was first proposed by Doclo et. al [20]. To this purpose a GSC-structure using the MCWF as beamformer is used. We did already describe the GSC-beamformer in Section 3.1.4. The adaptive noise cancellation setup is illustrated in Figure 4.4. The basic idea is to exploit that the output from the MCWF is the optimal noise estimate in an MMSE sense. In order to be able implement non-causal filtering, a delay element has been introduced in the forward path of $\hat{s}[k]$. The delay element and summation in the lower path corresponds to the blocking matrix in the GSC. The delay counterfeits the delay introduced internally in the MCWF.

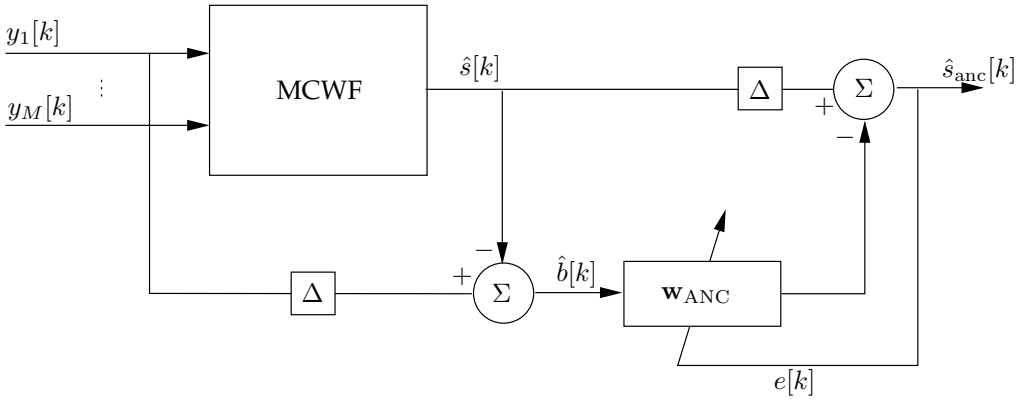


Figure 4.4: GSC-structure using the MCWF as beamformer. We also refer this setup as the ANC post-processing stage. The two delay elements are used to align the time signals.

This adaptive filter is implemented using an NLMS adaptive filter that only updates the error, $e[k]$, when the VAD detects noise-only samples. This is often described as a mode-controlled ANC. Using mode control ensures that the filter is not adapted in speech segments, $\hat{b}[k]$, if however speech is leaked (e.g. with reverberation present) into the noise reference the performance after the ANC stage will undesirably decrease.

Using a low-complexity filter like the NLMS does not affect the complexity much due to its

[2]Here defined as 1:1, i.e. 1 second of signal yields 1 second of execution time in a non-RT OS.
`Linux 2.6.18-gentoo-r6 #3 PREEMPT i686 Intel(R)`
`Celeron(TM) CPU 1133MHz GenuineIntel GNU/Linux`

linear complexity $\mathcal{O}(n)$ as summarised in Table 4.9. The NLMS filter estimation is

$$\mathbf{w}_{\text{anc}}[k] = \mathbf{w}_{\text{anc}}[k-1] + \frac{\mu}{\|\mathbf{b}[k]\|^2}(\mathbf{b}[k](\hat{s}[k-\Delta] - \mathbf{w}_{\text{anc}}^T[k-1]\mathbf{b}[k])) \tag{4.40}$$

where the delay optionally implements a non-causal adaptive filter.

	Operation	$\times$	$+$	$/$	$\sqrt{}$	FLOP's	Times executed
1.	Blocking matrix (find $\hat{\mathbf{b}}[k]$)	-	1	-	-	1	n
2.	Filter computation (4.40)	$3n+1$	$2n+1$	1	-	1	1
3.	Filtering, $\hat{b}_{\text{anc}}[k] = \tilde{\mathbf{b}}[k]^T\mathbf{w}_{\text{anc}}$	1	1	-	-	2	n
4.	Subtraction, $\hat{s}_{\text{anc}}[k] = \hat{s}[k] - \hat{b}_{\text{anc}}[k]$	-	1	-	-	1	1
	Total complexity						$7n+4$

Table 4.9: Complexity of the entire ANC stage using an NLMS adaptive filter. n is the adaptive filter length and has nothing to do with the filter lengths in the MCWF.

The complexity in Table 4.9 shows that using sub-sampling is possible at a lower complexity, e.g. if sub-sampling with a factor of 10 and using a 40 filter tap ANC-stage, the complexity is lowered to

$$\frac{1700}{10}\ \text{MFLOPS} + (7 \cdot 40 + 5)f_s = 172.3\ \text{MFLOPS} \tag{4.41}$$

increasing the complexity with only 2.3 MFLOPS.

The influence sub-sampling has on the performance at different sub-sampling rates is investigated and the robustness of the ANC stage is tested in a non-reverberated environment using the log-energy VAD. The NLMS filter used in the ANC stage computes 40 filter taps and a μ-value of 0.5 has been used. A delay of 21 taps has been introduced in order to realise a non-causal noise reduction filter, this small delay only introduces an I/O-latency of 2.6 ms.

In Figure 4.5 the ANC post-processing stage and the sub-sampled recursive multi-channel Wiener filter are tested in a non-reverberated environment and using the ideal VAD. The standard test signal with $\text{SNR} = 5$ dB white noise has been used. The signal has been concatenated 15 times and the assessment has been carried out on the last 15th part of the output signal. Testing on this long signal ensures convergence of the statistics even when increasing the sub-sampling factor significantly. Two results are shown. The first is when only using sub-sampling and the other includes the ANC stage.

The test was however conducted using a constant forgetting factor such that the effective window size (number of samples spanned) actually increases when increasing the sub-sampling factor. Keeping a constant effective window length (alway p samples) was observed to give results that yielded very poor performance. This is due to a less good auto-correlation estimate in the MCWF. As in a conventional GSC structure, the method is very dependent on a good noise estimate, which is obtained on the basis of a reliable output from the beamforming part. If a wrong noise estimate, such as when sub-sampling and using a short window, is obtained the ANC stage is not capable of improving the output. When sub-sampling the statistics, the spatial part (mainly) of the MCWF was not able to obtain a precise spatial filter. However, keeping the forgetting factor constant at all sub-sampling rates increased the effective window length, but still the performance decreased when the ANC stage is not active as seen in the figure.

The effective window length has major influence on the MCWF. The beamforming part of the MCWF is now seen to qualify the noise estimate in the ANC stage to be able to remove noise. Increasing the effective window size also requires the quasi-stationarity assumption to be increased from 0.5 seconds to 20 times more. Thus, 10 seconds when sub-sampling with a factor of 20. However, we assume this is true for the spatial stationarity in most applications. That the speaker does not move that often or that fast. A time-varying positioned noise source is though more likely,

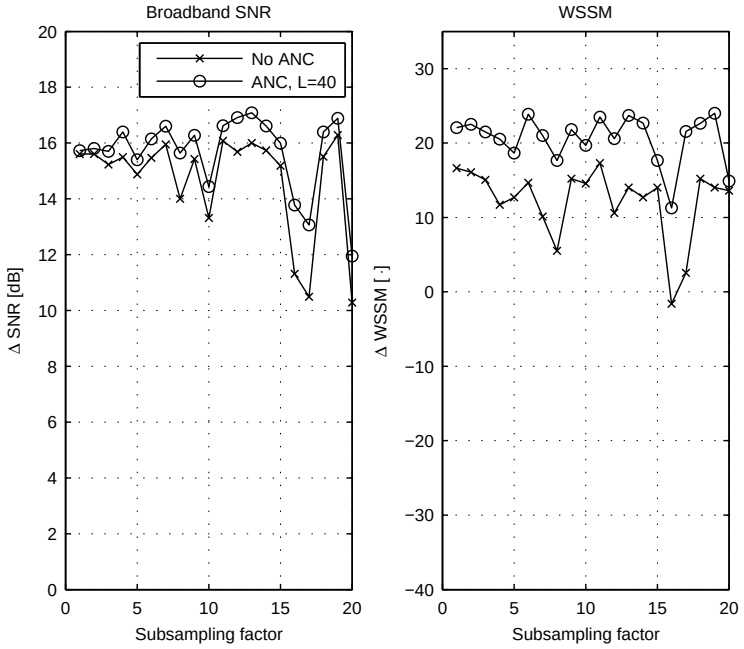


Figure 4.5: The ANC post-processing stage and the sub-sampled recursive multi-channel Wiener filter tested in a non-reverberated environment and using an ideal VAD. The standard test signal with $\mathrm{SNR} = 5$ dB white noise has been used. It is seen that a constant performance is maintained even though sub-sampling is introduced and that the post-processing stage improves noise reduction and lowers speech distortion, even such that better results than with no sub-sampling is obtained.

like a bypassing car. The importance is though that the spatial directivity is mostly concerned with the position of the speaker and not the interfering noise.

The interesting observation in Figure 4.5 is that for no sub-sampling ($n_b = 1$) the ANC stage results in even better results, however at a slightly increased computational cost (in-significant at that sub-sampling rate). For higher sub-sampling rates performance in both SNR and WSSM is maintained, which is a clear indication of that using sub-sampling and ANC post-processing lowers the overall complexity and maintains good performance.

The reason why the ANC stage impose even better assessment scores is illustrated with three spectrograms in Figure 4.6. Three figures are shown to explain the difference in performance using an ANC stage with sub-sampling compared to using the regular recursive MCWF. The upper figure shows the original speech signal as a reference. The MCWF with sub-sampling uses a factor of 3 and an ANC stage with a filter length of 40. This method obtains better objective assessment scores than the regular one. Firstly, one can notice that more noise is removed in the entire spectrogram, and secondly the noise is reduced even more in speech pauses, which is expected due to the mode controlled ANC-stage. The noise in the upper left corner of both spectrograms is left from the previous test signal, i.e. it is edge effects due to the concatenation of the test signals.

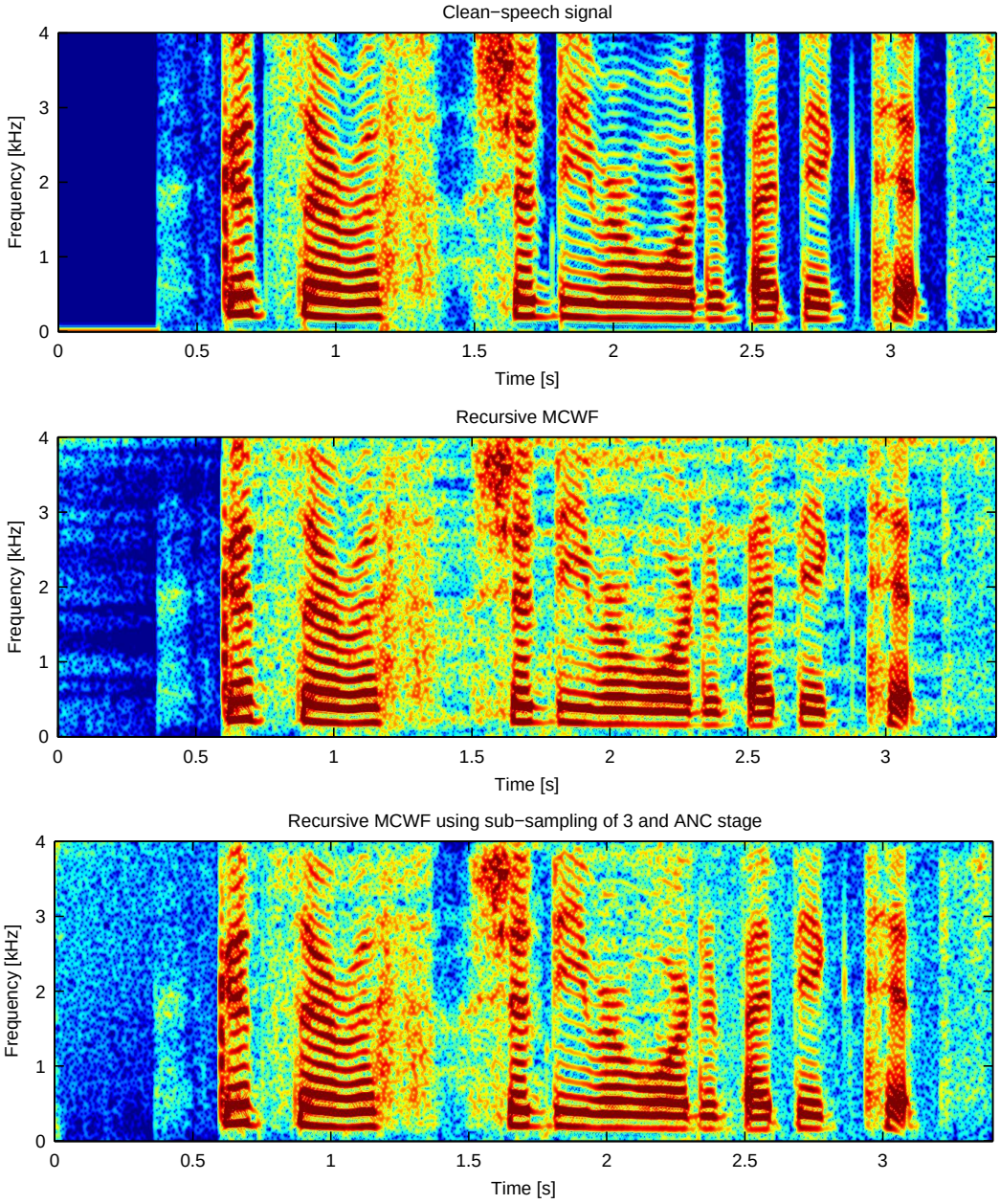


Figure 4.6: The three figures is illustrated to explain the better performance using an ANC stage and sub-sampling compared to using the regular recursive MCWF. The upper figure shows the original speech signal, "Good service should be rewarded with big tips". Then the output from the regular MCWF is shown and lastly, the method using sub-sampling with a factor of 3 and an ANC stage with a filter length of 40 is plotted. The latter method obtains better objective assessment scores than the regular one. Firstly, one can notice that more noise is removed in the entire spectrogram, and secondly the noise is reduced even more in speech pauses, which is expected due to the mode controlled ANC-stage.

In order to investigate the robustness of the ANC stage with respect to a bad mode control, the log-energy VAD is used instead of the ideal VAD in Figure 4.7. The same test signal as in the previous figure has been used. The figure shows that the log-energy VAD-based MCWF does not perform as good as the ideal VAD, unsurprisingly. Without sub-sampling ($n_b = 1$) the performance drops from a $\Delta\mathrm{SNR} \sim 15.5$ dB to almost $15\,\mathrm{dB}$, which is not a very significant difference. The distortion however drops with several WSSM-points (from 16 to 10). This is evident due to wrongly marked noise samples in the VAD, such that speech is detected as noise and then untendedly removed. One should though notice that the performance using the non-ideal VAD still decreases noise and speech distortion. The entire ANC stage is also seen to be robust at this noise level. The mode control ensures even when using the non-ideal VAD that additional noise is removed and thus both assessment scores increase to better scores than the higher-complexity non-sub-sampled recursive MCWF without ANC stage.

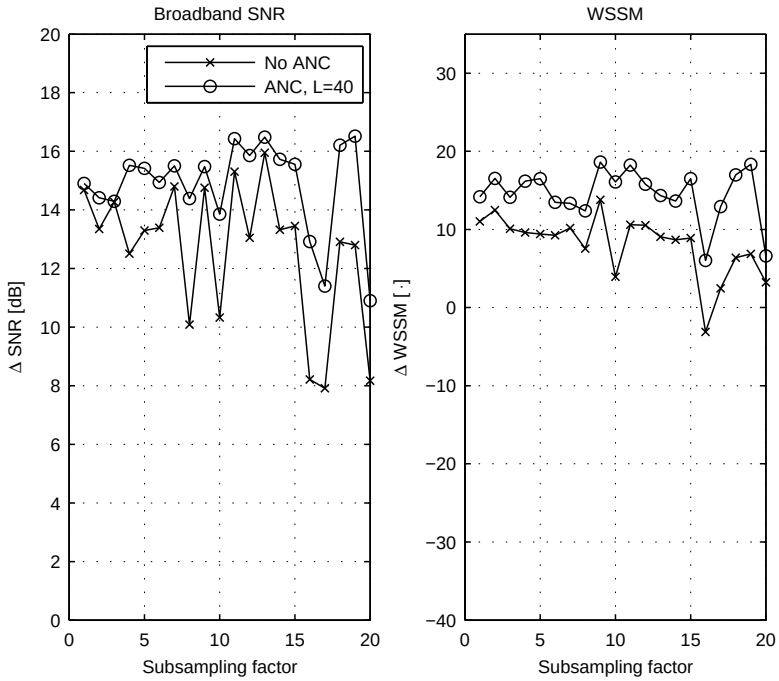


Figure 4.7: The figure shows robustness of the ANC stage with respect to a bad mode control. The log-energy VAD is used instead of the ideal VAD. The same test signal as in the previous test has been used. The figure shows that the log-energy VAD-based MCWF does not perform as good as the ideal VAD, unsurprisingly.

The last test combines two things. It investigated the effect of using more filter taps and it examines how adding reverberation to the test signal influences the output of the sub-sampled MCWF with an ANC post-processing stage. The test signal is the same as in the last tests, albeit a reverberation time of 600 ms has been added in one of the tests. Different filter lengths between 4 and 500 taps have been examined. Figure 4.8 indicate that a filter length around 50 is sufficient, and that the ANC stage is robust w.r.t. to reverberation even at short filter lengths.

4.3.3 Conclusion

In this section it was proved under certain spatial stationarity assumptions it was possible to use sub-sampling in the recursive MCWF. Merging the sub-sampled MCWF with an adaptive noise cancellation stage showed performance was held almost constant or even slightly better, considering speech distortion, even with a sub-sampling factor of 20. The ANC stage had a low complexity, thus sub sampling with a factor of 10 yielded a complexity reduction from 1700 MFLOPS to 172.3 MFLOPS which was computable in real-time on a Pentium-M.

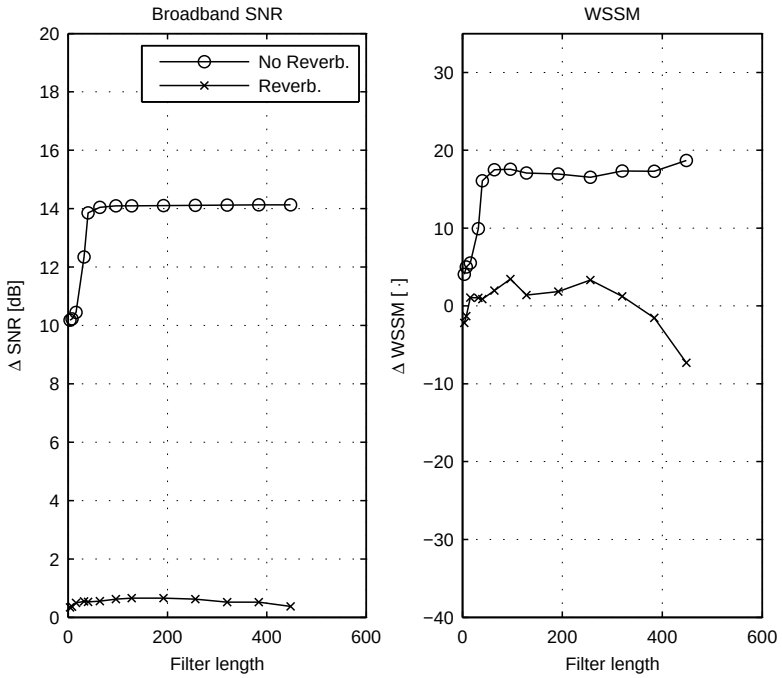


Figure 4.8: ANC filter length analysis and ANC robustness w.r.t. reverberation. A reverberation time of 600 ms has been used. The results indicate that a filter length around 50 is sufficient, and that the ANC stage is robust w.r.t. to reverberation even at short filter lengths.

In addition, the methods were tested using a non-ideal VAD to examine the robustness of the ANC stage when using a log-energy-based VAD. This test showed that the entire system, log-energy VAD, recursive MCWF, sub-sampling, and ANC post-processing maintained a similar performance as compared to the results obtained with an ideal VAD.

Lastly, the optimal (under certain test conditions) filter length was found to be about 40 filter taps, which also maintains low complexity in the ANC stage and low I/O-delay. Adding reverberation showed that the ANC was robust regarding reverberation, but poorer results was obtained as expected. However, the method did not worsen the output compared to the observation, i.e. the assessment improvements were positive.

4.4 Word-Length Considerations of the Recursive GSVD-Based MCWF

In the preceding sections we have reduced the computational complexity of the recursive GSVD-Based multi-channel Wiener filter, with the aim of achieving a real-time C-based implementation. This section is concerned with limited-precision arithmetics of the C-based implementation. Starting from the C-based implementation, we intent to carry out fixed-point design-space exploration in order to give some indications on the sensitivity of the algorithm regarding finite word-length implementation. Referring back to the chapter introduction, we are concerned with the conversion of a C reference model in floating-point arithmetic towards a fixed-point precision model. This type of work resides in the *data* domain of the rugby meta model [48], see Figure 4.1 on page 100. To facilitate this data type refinement, we make use of the SystemC C++ class library which extends C++ w.r.t. simulation of arbitrary size and mode of fixed-point data types.

To keep the analysis as close as possible to the algorithm at hand, we have chosen to skip any considerations regarding sub-sampling and the use of an adaptive noise cancellation stage. We have focused entirely on the recursive GSVD-based multi-channel Wiener filter (Appendix D, `flpt/mcwfgsvd.cxx`), which for the simulations in the preceding sections was implemented entirely using 64 bit floating-point. Before presenting the model-conversion method and results,

we define the problem at hand and fix the specifications of demand.

We define the problem of carrying out a fixed-point refinement as finding a satisfying solution w.r.t. hardware architecture and performance. Performance is defined as an *acceptable level of performance degradation* compared to the golden specification. The golden specification is set to be our simulations using the full GSVD in MATLAB . We will make no assumption on the target architecture, however, we are aware that todays hearing-aid devices are low-power designs. Therefore we assume that a fixed-point implementation is a pre-requisite for seeing the MCWF technique in a hearing-aid device/product.

The initial C-based implementation was fixed in $64\,\mathrm{bit}$ floating-point precision. We have initially converted this implementation to $32\,\mathrm{bit}$ floating-point, as we find the activities involved in fixed-point model conversion rather superfluous if the algorithm cannot perform well in $32\,\mathrm{bit}$ floating point. The fixed-point model conversion is done following ideas from Sung et al. [87], who proposes to measure the degrading effects of quantisation using a floating-point reference model and a representative data set. We have chosen the MATLAB implementation which updates a matrix on a sample-by-sample basis followed by a full GSVD computations as described in Section 3.2.1 on page 82. The data set used is the standard female speech signal of a female uttering "*Good service should be rewarded with big tips*" degraded by additive white noise and no reverberation. This cannot be characterised as a representative data set, neither considering the representativeness of the short sentence as regards the application of speech enhancement, nor statistically regarding the number material. We have, however, chosen to use this data set, in doing so the performance using the MATLAB implementation can directly be compared with the performance of the fixed point implementation.

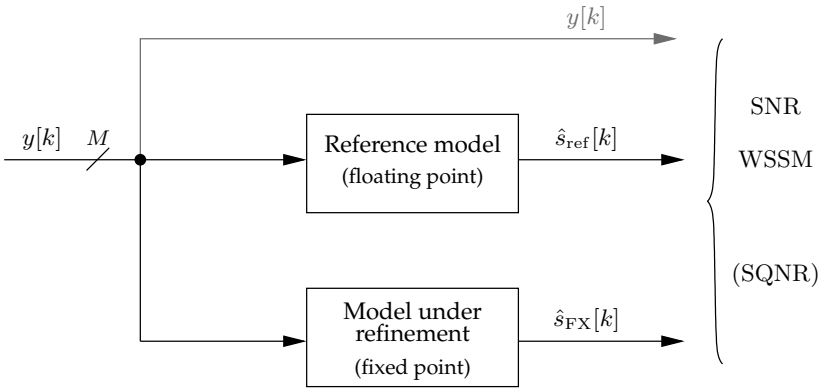


Figure 4.9: The model used as the basis for the fixed-point refinement. The reference model, here a MATLAB -based implementation, is used to compute the golden specification, $\hat{s}_{\mathrm{ref}}[k]$, which is compared to the output from the model under data-refinement. The output signals can be compared by means of the SQNR, or one can use the broadband SNR and WSSM performance parameters usually employed in this project.

In Figure 4.9, the procedure proposed by Sung et al. [87] is recasted in the context of this project. By feeding a representative data set to a reference model (floating-point arithmetic), the reference output, or golden specification, $\hat{s}_{\mathrm{ref}}[k]$, is obtained. The quantised signal, $\hat{s}_{\mathrm{FX}}[k]$, is obtained by executing the dual model, shown below the reference model in Figure 4.9. This model gradually refined one computation, or group of computations, at a time, towards a fixed-point implementation. The gradual refinement ensures, that we at all times have a floating-point and a fixed-point executable model, where the fixed-point model can be turned back into floating-point model for a sort of backward compatibility. The actual refinement is done iteratively in a number of steps; (1) determine word-length, quantisation mode, and overflow mode for a group of computations, (2) simulate, (3) compare the quantised output with the reference data.

It has not been feasible to aim at an working implementation in hardware, such as an FPGA or DSP implementation. Rather we aim at simulating a few scenarios on a workstation. The simulation-based fixed-point refinement requires the use of a programming language capable

of expressing arbitrary fixed-point data types. To that end, we have chosen to work with the SystemC methodology. It consists of a formal methodology, which is of little interest here, and a C++ class library to provide HW-oriented extensions to the C++ language. In specific, SystemC introduces a notion of time, hierarchy, communication, and fixed-point data types. Of which the latter is of interest for us. SystemC can be used to, qouting the SystemC (v.2.0.1) User's Guide [68]: "*create a cycle-accurate model of software algorithms, hardware architecture, and interfaces of your SoC (System On a Chip) and system-level designs*". We have used the data types defined in the C++ class library to extend our working C/C++ reference model with fixed-point data types.

Starting with the double-precision implementation (`mcwfgsvd.cxx`), we simulate the performance (degradation) using $64\,\mathrm{bit}$ and $32\,\mathrm{bit}$ floating-point arithmetics. We compare the results with the performance of the MATLAB based simulations. Furthermore, the implementation is combined with SystemC to allow fixed-point simulation. It was only possible to use limited efforts on this model conversion. We therefore chose a pragmatic approach. In the floating-point implementation many symbols (terms) defined in Algorithm 7 on page 114 were kept in a single variable, as it is not necessary to explicitly define variables for all symbols in the given formula's (especially not in floating-point precision with large dynamic range). For the fixed-point refinement we chose to consider each variable, which can span many computations, in one data group. Initially all variables were kept in $64\,\mathrm{bit}$ floating-point while we gradually converted the data groups to fixed-point data types using SystemC's class library. Simulations were drastically slowed down as each native data type gradually was changed into an C++ object representation as each computation (before being a native data type) involved function calls and extensive masking and casting.

Before presenting the fixed-point refinement and the performance results we have left the problem of which performance parameter to employ open. This we will cover first, then return to the results.

In Sung et al. [87] they employed the signal-to-quantisation ratio as metric for degradation by their automatic fixed-point refinement for FIR filters. Referring to Figure 4.9, we can define the SQNR as

$$\mathrm{SQNR} = \frac{\hat{s}_{\mathrm{ref}}[k]}{\hat{s}_{\mathrm{ref}}[k] - \hat{s}_{\mathrm{FX}}[k]} \tag{4.42}$$

where $\hat{s}_{\mathrm{ref}}[k]$ is the reference output data (golden specification) and $\hat{s}_{\mathrm{FX}}[k]$ any refined model under consideration, for example our $32\,\mathrm{bit}$ floating-point implementation. Another measure, as we have opted for, to use the performance of the speech enhancement technique, SNR and WSSM, and define a criterion for acceptable degradation based on the output performance of MATLAB simulations (golden specification) and the data-type refined model under consideration. We have chosen the following specification

$$\mathrm{SNR}(\hat{s}_{\mathrm{ref}}) - \mathrm{SNR}(\hat{s}_{\mathrm{FX}}) \quad < 1\,\mathrm{dB} \tag{4.43}$$

$$\mathrm{WSSM}(\hat{s}_{\mathrm{ref}}) - \mathrm{WSSM}(\hat{s}_{\mathrm{FX}}) \quad < 2 \tag{4.44}$$

which we by practical experience have found on the verge of audible. In our simulations we have also observed the SQNR which was around $29 - 30\,\mathrm{dB}$ for a simulation with performance within the above criteria. Another problem of consideration, which we have not covered in our analysis, is the convergence of the adaptive filtering. For adaptive filters not only the long-term performance under stationary conditions, but also the speed of convergence is affected by a data-type refinement. In our simulations we have used 3 concatenated signals of approx. $3.5\,\mathrm{s}$ each, and chosen the last segment for evaluation , where we assume that the filters are adapted to an optimal point for the rather stationary environment. If we had been interested in also the speed of convergence, the filter residual from the adaptive filters (reference model and data-type refined) should be compared.

In Figure 4.10 the simulation results for four different implementations are seen. In order to evaluate the data-type refinement, we have plotted the SNR and WSSM improvement for -5 to $10\,\mathrm{dB}$. We first describe the reference data and the results from the $64\,\mathrm{bit}$ and $32\,\mathrm{bit}$ floating-point

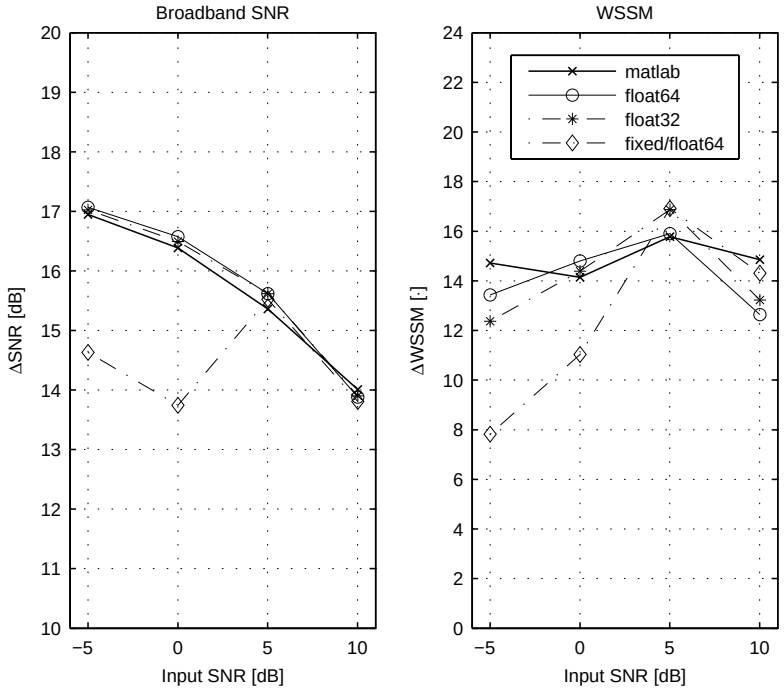


Figure 4.10: Comparison of the performance of the MCWF in different implementations. The C-based floating-point implementation is compared with the MATLAB implementation. Both 64 and $32\,\mathrm{bits}$ is tested. On top, the results using a partial fixed (partial floating-point $64\,\mathrm{bit}$ precision) model is compared.

simulations followed by explanations on the fixed-point refinement and the performance attained using this implementation.

The SNR performance for the reference data (MATLAB simulation) and the $64\,\mathrm{bit}$ and $32\,\mathrm{bit}$ floating-point implementation is comparable. The degradation for reducing the bit-length of the floating-point implementation is seen to reduce the WSSM improvement in a consistent manner, see the results at $-5\,\mathrm{dB}$ and $10\,\mathrm{dB}$. It is notable that the $32\,\mathrm{bit}$ floating-point implementation is performing within the $1\,\mathrm{dB}$ and 2 WSSM-point limit. Thus, using the current recursive GSVD-based implementation, the performance could be kept within the specification of demands in an architecture which can do $32\,\mathrm{bit}$ floating-point arithmetics.

From Figure 4.10 it is noticed that for the observation of $5\,\mathrm{dB}$ the WSSM is slightly better for the fixed and $32\,\mathrm{bit}$ floating-point implementation compared to the MATLAB implementation. This can possibly be caused by some low-level noise, introduced by the fixed and $32\,\mathrm{bit}$ floating-point implementation, which might be favourable for the WSSM.

The fixed-point implementation is rather a partial fixed-point implementation, as we have not move all variable groups to fixed point and some symbols and computations remain in $64\,\mathrm{bit}$ floating-point precision. Without presenting the actual C-code, we will try to explain which parts of the computations that are in fixed point.

The C-based implementation is largely based on Algorithm 7 and 8, and Eq. (4.33). Of course an interface with MATLAB , dynamic memory allocation, etc. has been added. The test signal used was the before-mentioned clean-speech signal of a female person. In order to reduce the computations the tests were performed at $5\,\mathrm{dB}$ only. Referring to the symbols in the specified formulae, the resulting fixed-point conversion status is as stated in Table 4.10. We adopted the SystemC notation of word length (wl) and integer word length (iwl), which means that we have a total of wl bits and hereof iwl are integer (before the decimal point) bits.

The input signal is in $12\,\mathrm{bit}$, which is seen in Table 4.10, while already after the first operation, updating by $\mathbf{V}[k]$, the required bit length is increased by $6\,\mathrm{bits}$. The following bit lengths regarding the $\mathbf{R}[k]$-matrices, which all are the same variable in the implementation, is seen to be $36\,\mathrm{bits}$.

Operation	Symbol	(wl, iwl)
New sample	$\mathbf{y}[k]$, $\mathbf{b}[k]$	$(12, 1)$
Updated vector	$\tilde{\mathbf{y}}[k] = \mathbf{y}[k+1]\mathbf{V}[k]$, and $\tilde{\mathbf{b}}[k]$	$(18, 3)$
λ-updated	$\mathbf{R}'_Y[k]$, $\mathbf{R}'_B[k]$	$(36, 12)$
QR update	$\tilde{\mathbf{R}}_Y[k]$, $\tilde{\mathbf{R}}_B[k]$	$(36, 12)$
SVD and QR update	$\tilde{\mathbf{R}}'_Y[k]$, $\tilde{\mathbf{R}}'_B[k]$	$(36, 12)$
Back-subst. output	$\mathbf{w}_{\mathrm{WF}}[k]$	$(19, 2)$

Table 4.10: The fixed-point values used referring to Algorithm 7 and 8, and Eq. (4.33). Integer word length = iwl, and word length = wl. We have used truncating quantisation and wrapping overflow for all variables.

The large integer bit length of $12\,\mathrm{bit}$ is very likely due to the intensive activity of Jacobi's on the diagonal of the $\mathbf{R}[k]$-matrices. The optimal filter resides in $\mathbf{w}_{\mathrm{WF}}[k]$, which is the output of the back-substitution procedure. The length is seen to be $19\,\mathrm{bits}$.

The rotation parameters for the Jacobi's have also been implemented in fixed-point precision, however, it turn out, that these variable were extremely sensitive w.r.t. accuracy. It should be noted that *all* the c and s (entries in the planar rotation matrices) were implemented in the same variable. We could expect that further splitting this group of variables could gain insight into the needed bit length and that it seems unlikely, that all rotations need to be done in equal precision. This observation also directly applies to the $\mathbf{R}[k]$-matrices, on which the Jacobi's (also) work. As shown in Section 4.1.5, the relationship (quotient) between the $\mathbf{R}_Y[k]$ and $\mathbf{R}_B[k]$ equals the generalised singular values. The large bit-length ($36\,\mathrm{bit}$) which were needed in the simulations might stem from the need for a large dynamic range, not from sensitivity. One way to investigate this is to split the many different symbols (stages in Algorithm 7) in different variables to better follow the computations and adjust the bit length to the necessary length. It is also expected that the generalised singular values have a large range from minimum quotient to maximum. It might be possible to reduce the bit length, assuming the problem is dynamic range, by splitting the noise and data matrices, $\mathbf{R}_Y[k]$ and $\mathbf{R}_B[k]$, which are numerator and denominator, respectively, in the quotients which equal the generalised singular values. Traditionally, when computing an SVD, the singular values can be assumed to be sorted (which is actually done by a sorting algorithm), however, the generalised singular values from the quotient is not sorted, so this cannot be exploited.

From the discussion above, it can be seen, that further work towards a fixed-point implementation needs to be done. It remains, however, to comment on the results from the fixed-point simulation. In Figure 4.10 it is clearly seen that the partial fixed-point simulation is optimised against the signal with $5\,\mathrm{dB}$ additive noise, as the performance is, at that level, kept within the specification of demands. If we, however, observe any of the two points at lower SNRs, we observe a performance degradation w.r.t. SNR and WSSM. For SNR improvement around $2.5\,\mathrm{dB}$ performance degradation and for WSSM improvement $4 - 6.5$ points. This stresses the need for representative data sets to measure the performance degradation. The advantage of the present simulation-based fixed-point methodology is its simplicity. By using SystemC to simulate the functionality of fixed-point precision variables it is easy to evaluate the performance degradation, however, the object-oriented approach gives rise to a (possibly) long simulation time depending on the amount of simulated variables.

We have in the preceding text presented a methodology to perform fixed-point design space exploration. Using SytsemC in combination with the methodology, it is possible to convert a floating-point C-based code/systems to an executable fixed-point precision model. Results were presented which indicated that the recursive GSVD-based MCWF could perform within the specification of demands (degradation of $\mathrm{SNR} = 1\,\mathrm{dB}$ and $\mathrm{WSSM} = 2$ point) compared to a reference MATLAB -based implementation. This was ascertained for a short time signal using additive white noise for both an $64\,\mathrm{bit}$ and $32\,\mathrm{bit}$ implementation.

4.5 Conclusion

In this chapter, the complexity of computing the multi-channel Wiener filter was attended. Motivated by the promising noise reduction performance of the MCWF shown in the previous chapter, we investigated alternative algorithms to implement the MCWF functionality. Throughout our work, the MATLAB -based MCWF, based on the GSVD, was our reference implementation, which provided the golden reference data set. To guide the study, we settled for a specification of demand based on an acceptable level of degradation: $1\,\mathrm{dB}$ lower noise reduction performance and $2\ \mathrm{WSSM}$ points.

It was ascertained that the MATLAB implementation of the MCWF actually was an algorithm for implementation of the MCWF functionality, referring to the A3 paradigm. The activities in this chapter were split i three: Find alternative ways to compute the MCWF; from the chosen algorithm make an executable C-based model in floating-point precision; and investigate the needed word-length, i.e. convert the floating-point model towards a fixed-point implementation.

In the survey of alternative algorithms to compute the MCWF, we discovered that although several algorithms existed, some complexity-reducing algorithms were based on certain assumptions on the application, e.g. the generalised multi-channel, frequency-domain adaptive filter, suited for adaptive echo cancellation, or the PAST and FST methods, which were suited for estimating sinusoids embedded in noise, such as the low-rank model employed in the signal subspace approach. We decided for a GSVD-based algorithm which was based on recursive updating and reducing complexity by trading-off convergence. We introduced Givens rotations and one- and two-sided Jacobi's. These were used to describe an updating QRD and updating SVD. Generally QR updating steps were interlaced with two-sided diagonal-Jacobi's in order to do recursive updating of the GSVD. The upper Hessenberg structured formulation of our data matrix formed the basis for the complexity reductions. Each sample-update step involved several re-triangularisations from the Hessenberg structure.

The chosen algorithm lowered the complexity of the full GSVD (MATLAB) from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$, and the estimated reduction, in terms of FLOPS under a number of assumptions, was approx. a factor 700. This measure is based on a crude FLOP count where we ascribe even importance to square roots as to additions. One might argue that this is unrealistic, as few architectures handles these operations in an equal amount of time. When taking the C-based model to the next level, implementation on a certain architecture, one might be interested in less square-root intense algorithms. A number of square-root free Jacobi's have been proposed [28, 45], of which some also have been applied to computations of the GSVD [63, 20]. In [35] a square-root and division free algorithm is devised.

Although the computations of the optimal filter were drastically lowered, the complexity per sample remained quite high. Motivated by the observations by Doclo et al. [16], it was investigated of whether sub-sampling the estimate of the stochastic processes would deteriorate the noise reduction or speech distortion performance. By introducing the sub-sampling, the effective window length for the estimation was extended while the sample-distance of the stochastics were lowered. Surprisingly, the results indicated acceptable performance. Even by sub-sampling of a factor larger than 4 (recall that the updating steps for the entire GSVD operates at a speed of f_s/n_s). By the introduction of an ANC stage, the noise reduction lost due to the process of sub-sampling, was restored.

In the last section we were concerned with floating and fixed-point implementation of the MCWF. It turned out, that $32\,\mathrm{bit}$ floating-point precision was sufficient to keep within the acceptable amount of degradation. Investigation of the needed fixed-point precision for several variables (symbols) was carried out, and indicated that more research was needed w.r.t. the dynamic range of the Hessenberg matrices and the rotation parameters, c and s.

CHAPTER 5

Discussion and Results

In the preceding chapters we have presented and examined several single- and multi-channel speech enhancement techniques. In this discussion we compare a selection of four different speech enhancement techniques. Namely, spectral subtraction using Martin's method of minimum statistics for noise estimation [59], the GSVD-based signal subspace method proposed by Jensen et al. [49], the mode-controlled GSC-beamformer using an a priori based steering vector, and finally the sub-sampled recursive GSVD-based multi-channel Wiener filter employing an mode-controlled ANC post-processing stage.

We will elucidate the functionality for the aforementioned single- and multi-channel methods using three different test setups, of which one are yet unseen. The three setups are, firstly, additive white and babble noise without convolution noise (direct-path observations), and, secondly, additive white noise in a reverberated environment using a simulated room-impulse response. The first setup, using additive white noise in a non-reverberated environment, is to provide the setting for further discussion of the more complex cases. We remind the reader that multi-channel methods have the benefit of multi-channel observations, which is a different physical setup, nevertheless it is interesting to compare the performance of the functionalities.

In all simulations in this chapter, the methods are evaluated using fixed parameters, such that spectral subtraction is applied using $\alpha = 5$, $\beta = 0.01$, and $\gamma = 2$ (power spectral subtraction). For the noise estimation method of minimum statistics, the search length is set to $U = 8$ and $V = 6$, which means the method can span 768 ms of high-energy utterances (tracking latency). For the GSVD-based signal subspace approach the minimum variance (MV) estimator is used, no model order estimation, and a decomposition size of $M = 32$ and signal subspace size of $K = 14$. For the above-mentioned methods, a window of 256 Hanning-tappered samples are used with 50% overlap, except for the signal subspace method which uses no overlap and no tappering.

The multi-channel methods are all based on two-channel observations. The GSC comprise a steering vector determined using a priori information about the direction-of-arrival for the desired (speaker) signal. The used ANC stage comprise 256 filter taps, of which half is realised non-causal. This size is chosen as a good trade-off between noise reduction in both reverberated and non-reverberated setups based on simulations in Chapter 3. The multi-channel Wiener filter is computed using the recursive GSVD-based method using a sub-sampling factor of 3 and a 40-tap ANC stage. The filter length in the MCWF is 40 and realised with half the taps filtering in the non-causal domain. For both the GSC and the MCWF, the ANC stage is operated by a mode control to prevent adaptation during periods which are likely to contain signal leakage in the noise reference signal.

Where needed – ANC, MCWF, and signal subspace approach – an ideal VAD has been employed. We have not opted for the log-energy based VAD as it is based on modelling the noise distribution (mean and variance), a method that is deemed to fail when applied to an observation comprising clean-speech and additive babble noise. This is confirmed in simulations (not shown).

The first test is a comparison of the four chosen speech enhancement methods when applied

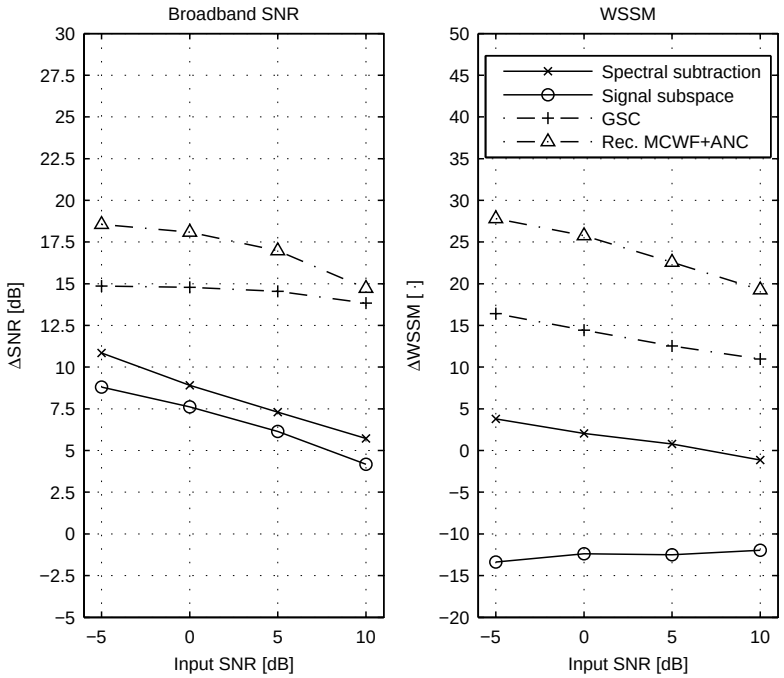


Figure 5.1: Comparison of four speech enhancement methods. The test setup is white noise in a non-reverberated environment (direct path) using various SNR-levels. The clean-speech signal is approx. $3.5\,\mathrm{s}$ long, recorded at $8\,\mathrm{kHz}$, and the speaker is female.

in a additive white-noise test setup. In Figure 5.1 the comparison is shown. The figure shows what was also concluded in the preceding chapters. The signal subspace approach is the method obtaining the least noise reduction while also introducing the most speech distortion, w.r.t. the WSSM measure, than any other of the methods. This stems from the large amount of musical noise. The spectral subtraction method is able to remove a few dBs and obtain an increased assessment score regarding speech distortion, in -5, 0, and 5 dB input SNR. Musical noise is also present for spectral subtraction, but the noise floor ensures better WSSM scores, because the spectral peaks (musical noise) are drowned by the background noise and thus weighted less in the WSSM score, which measures spectral slope differences.

It is worth noting, that although the signal subspace would be modified to include a spectral noise floor to drown the annoying effects of musical noise, as is the case with spectral subtraction, it will still not perform comparable to the spectral subtraction method, noise-reduction-wise. This indicate that the added complexity associated with the signal subspace methods are not worthwhile.

The GSC adaptive beamformer obtains near-constant noise reduction at all four input levels. The WSSM score slightly decreases, though. The multi-channel Wiener filter obtains good noise reduction and is overall best performing in this setup. The method also shows superior w.r.t. speech distortion, as it has the best WSSM improvement score. The MCWF is clearly the choice among the four presented methods for removing additive white noise in a non-reverberated environment.

In the second test we substitute white noise with babble noise. Babble noise is characterised by having the same spectral characteristic as speech and considered much more difficult to remove than white noise. Much less can be assumed about babble noise since the noise stems from speech signals and are likely non-stationary. Often the dynamics of babble noise differs from the desired speakers and the noise can be located physically different.

In order to introduce the reader to babble noise, an example of a babble noise signal constituted of a group of speakers talking simultaneously is depicted in Figure 5.2. One should note that the modified spectrogram used throughout the report enhances speech signals by truncating the dynamic range, see Section 1.4. Compared to a clean-speech signal, the babble noise is seen

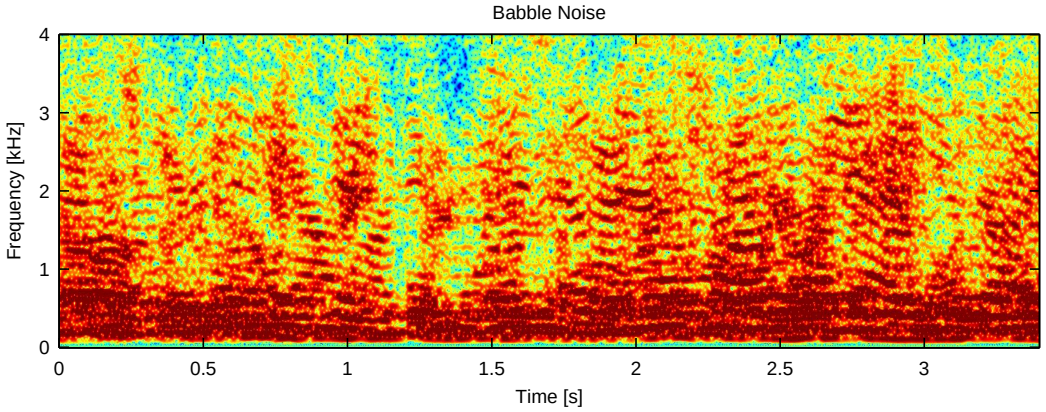


Figure 5.2: Modified spectrogram of a 3.5 s babble noise sequence. The signal is seen to resemble the clean-speech signal. It is compressed in the low frequencies and in mid and high frequencies formant tracks can be recognised.

to share spectral characteristics. This is of course expected, as the babble noise signal consists of multiple simultaneously speaking speakers. In the figure, the low frequencies are represented with dark red areas indicating signal intensity in the lower spectral band. At mid and high frequencies, formant traces are clearly visible, which emphasises the non-stationarity of this type of background noise. The amount of uncertainty makes a priori assumptions about the noise difficult, and, together with the spectral and dynamic characteristics, this amounts to one of the most difficult noise types when considering the problem of noise reduction in speech signals.

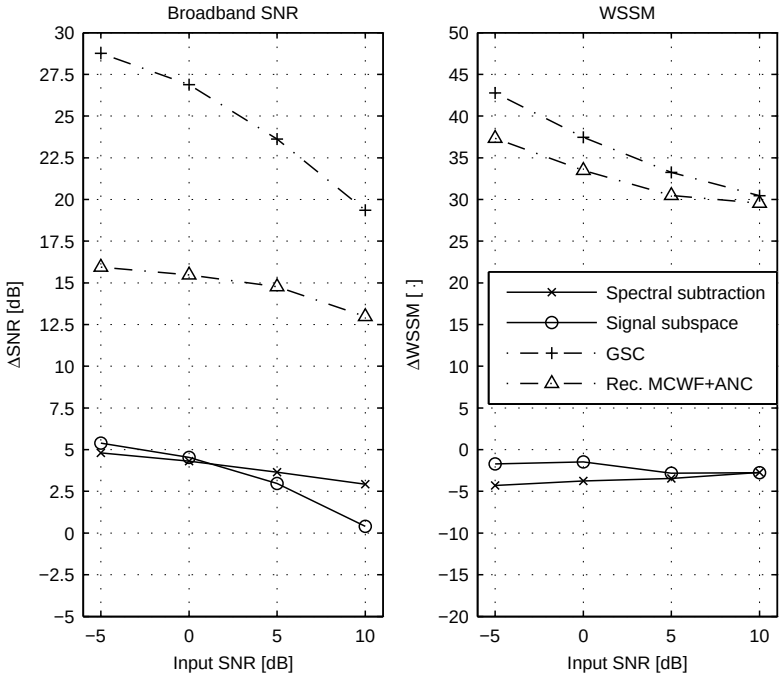


Figure 5.3: Same test case as in Figure 5.1, however, the additive noise is now babble noise. No convolutional noise.

In this second test, the observation signal is degraded by babble noise. The results are seen in Figure 5.3. Two major observations can be made; the GSC obtains very good results in both assessment scores, and spectral subtraction and signal subspace obtain more similar results than for the white-noise case. The GSC is seen to the remove more noise than in the case of white noise.

The MCWF method obtains almost similar results as in the white noise case. Thus, babble noise has little effect on this method. As the babble noise adds additional formant tracks and

high energy in the low-frequency area, the MCWF is expected to improve the speech distortion measure better than compared to the white-noise case. This is indeed seen to correspond to the observation by comparing Figure 5.1 and Figure 5.2.

Spectral subtraction is seen to fail in the babble-noise test. This is not due to the spectral subtraction procedure, rather the noise estimation method is the problem. Martin's method of minimum statistics used in the spectral subtraction was based on the assumption that the signal often decay to the level of the background noise. The gist of the method is, of course, to determine how often and whether the background noise is stationary enough to be tracked successfully over long high-energy utterances. When the method is applied to babble noise, it fails. Even for shorter search windows (tested at approx. $350\,\mathrm{ms}$), the method fails to obtain positive WSSM improvement. The signal subspace approach is able to remove the noise because the ideal VAD marks the noise and thus the method uses a reliable noise estimate. The WSSM improvement is however still negative due to large peak-to-valley differences in the output. As mentioned, a spectral floor might be the right remedy for this problem, i.e. drown the musical noise by a noise floor, however, the noise reduction is not likely to increase by this approach.

In Figure 5.4 the spectrograms of all four methods are plotted in the babble noise test with $\mathrm{SNR} = -5\,\mathrm{dB}$. From the top, the observed signal is clearly masked by babble noise and it can be observed that babble noise and speech, indeed, share the same spectral content. Second, the output of spectral subtraction using Martin's noise estimation method. It is from the spectrogram clear that the method is not capable of tracking babble noise. Most of the noise and speech in the lower frequencies remain. However, musical noise is not very distinct compared to the spectrograms presented in Chapter 2 for the case of white noise. Musical noise can, however, be observed around $1.2\,\mathrm{s}$ and $3.1\,\mathrm{s}$. The small amount of observable musical noise might be attributed to the noise estimation method. For observation signals of high dynamic behaviour, the optimal smoothing is reduced and the minima attained is more likely to be an underestimate of the instantaneous noise signal power. This results in reduced noise reduction, but also in reduced probability for introducing musical noise, logically. Another explanation might be found in the spectrogram of the babble noise signal. It is seen, that little noise energy is located in the high-frequency area. Although the noise estimate is wrong, compared to the instantaneous noise, the musical noise introduced will be small enough to go unnoticed – especially in our modified spectrogram.

In the third figure, we see the signal subspace approach. The method is able to remove the noise where the VAD marks noise frames (the blue areas), but having removed the noise entirely, musical noise also gets more distinct, decreasing the WSSM score. The GSC, in the fourth figure is seen to retrieve the clean-speech signal. This is consistent with the large noise reduction observed in Figure 5.3. Recall that the fixed beamformer left a large noise residual in the low-frequency area, see Chapter 3. This can be observed in the lower part of the spectrogram. Comparing the spectrogram of the GSC with the fifth spectrogram of the MCWF we clearly see the difference, $12\,\mathrm{dB}$ in noise reduction, according to Figure 5.3. The performance of the GSC is clearly superior. An explanation should be found in the estimated noise signal, fed as noise reference to the succeeding ANC stage in both methods. The ideal case of direct-path clean speech signal renders the noise reference signal from the fixed beamformer signal-leakage free. This is due to the subtraction of the observation signals, which effectively cancels the clean-speech signal in the perfectly non-reverberated case. The filtering in the MCWF is slightly more complicated and the noise reference signal, although very good, still suffers from signal leakage. Inspection of the spectrograms of the ANC-stage noise references in the case of GSC and MCWF confirms this (not shown). By increasing the number of filter taps in the ANC stage of the MCWF to 256 taps, as in the case of the GSC, so that they are on equal terms, no additional performance gain is observed. Thus, it is very likely that the ideal circumstances renders the noise estimates for the GSC almost-perfect.

The crucial observation for both the case of white and babble noise is, that multi-channel methods obtain better noise reduction and less speech distortion than single-channel methods. In the temporal domain, babble noise resembles speech, and the single-channel methods are not

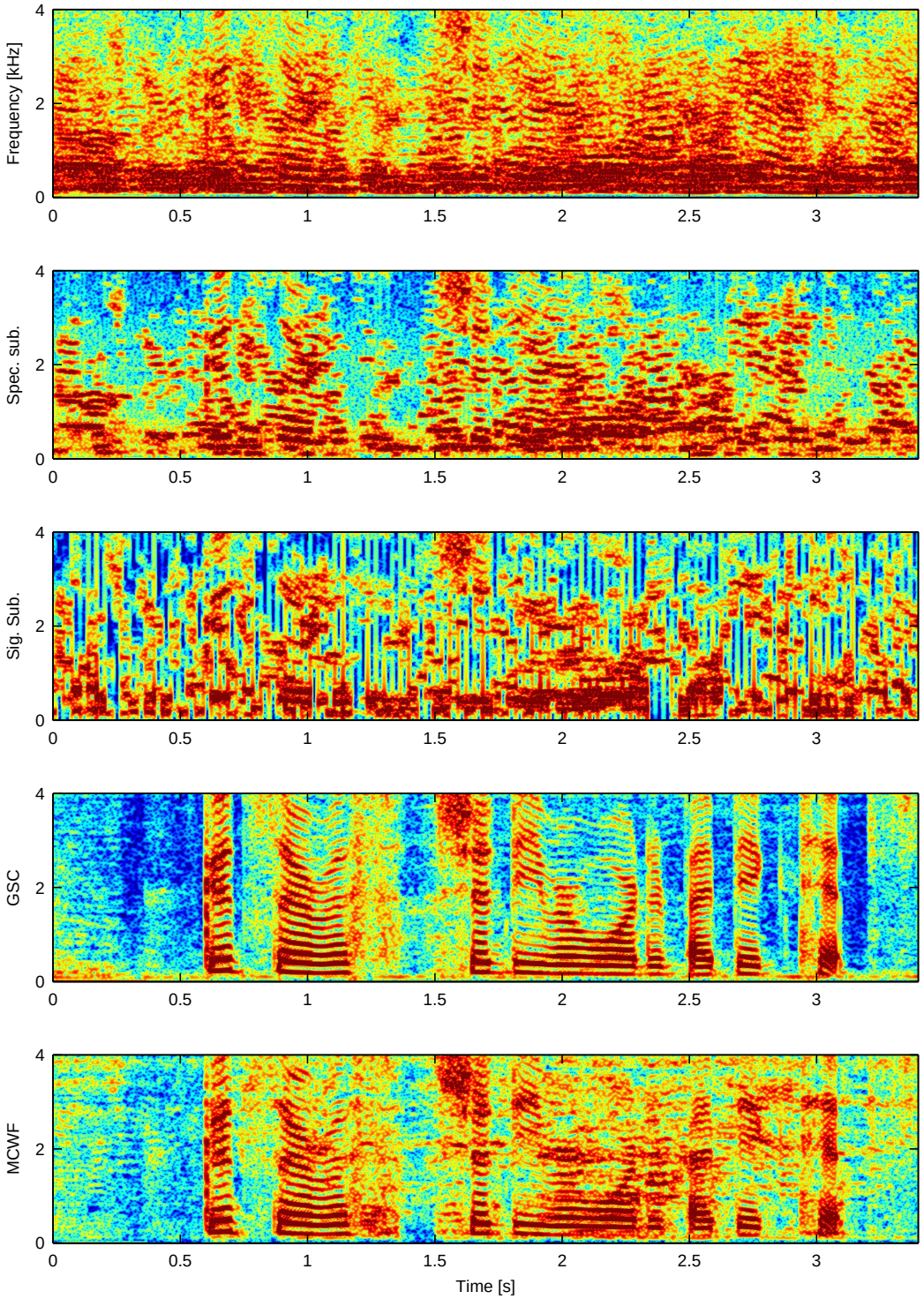


Figure 5.4: Modified spectrogram of all methods at $\mathrm{SNR} = -5$ dB with an observation signal degraded by babble noise in a non-reverberated environment. From the top, the observation, spectral subtraction, signal subspace, GSC, and MCWF.

able to remove much noise. Using spatial information, however, circumvents the problem of noise reduction in a babble-noise case by adding extra information to the speech processing mechanism.

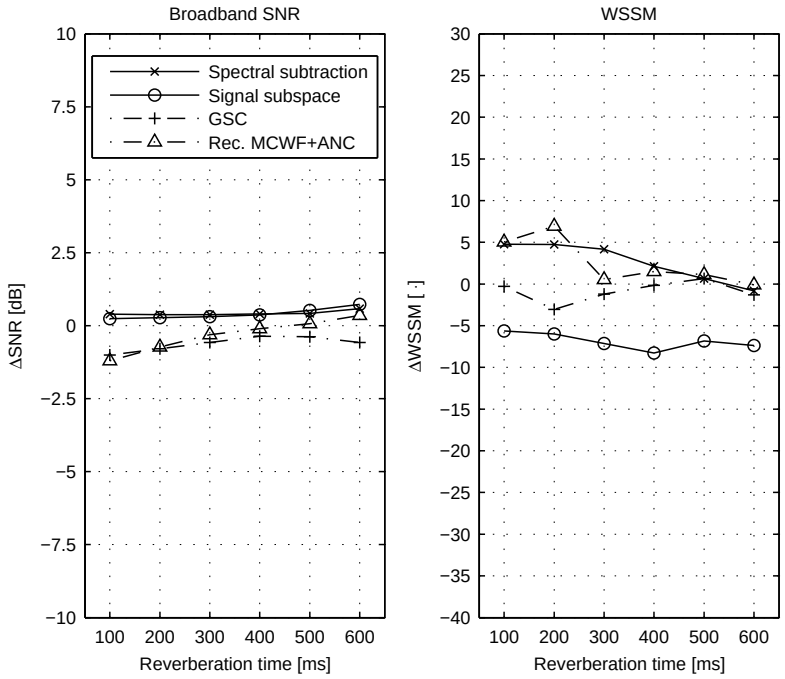


Figure 5.5: Comparison of the four chosen methods in the case of an observation degraded by white noise and increasing reverberation.

In order to test for a more realistic scenario, we compare the four methods in a reverberated environment. The results are shown in Figure 5.5. As observed in the simulation results in preceding chapters, none of the investigated methods are able to remove substantial amounts of noise in a reverberated environment. However, care should be taken in these conclusions. The definition of broadband SNR, as presented in Section 1.3, is

$$\mathrm{SNR} = 10\log_{10}\left[\frac{\sum\limits_{k} s^2[k]}{\sum\limits_{k}(s[k]-\hat{s}[k])^2}\right] \tag{5.1}$$

and the observation model

$$y[k] = g[k] * s[k] + b[k] = x[k] + b[k] \tag{5.2}$$

from which it is seen that the noise reduction measured by the expression in Eq. (5.1) is somewhat misleading. The denominator term, $\big(s[k]-\hat{s}[k]\big)$, becomes, $\big(s[k]-(x[k]+b[k])\big)$, from which it is seen, that though the noise should be completely removed, the noise (denominator) term would become $\big(s[k]-\hat{x[k]}\big) \neq 0$. Some researchers circumvent the problem by measuring $x[k]$ compared to $\hat{x}[k]$, which is also somewhat misleading, as this recasts the problem of speech enhancement (retrieving the signal $s[k]$) to a noise reduction problem (to retrieve the signal $x[k]$). Inspecting the performance of the four methods using $x[k]$ as the reference signal instead of $s[k]$ yields results with clear noise reduction performance, however, we have chosen to use $s[k]$, despite its downsides, as it is in better accordance with our initial goal of the project – to retrieve the speech signal as observed $1\,\mathrm{m}$ from the speaker in an anechoic room.

The results in Figure 5.5 show, surprisingly, that spectral subtraction obtains better WSSM scores than the GSC at low levels of input SNR, and that spectral subtraction obtains similar performance as the multi-channel Wiener filter. The GSC does not obtain any noise reduction

or decrease of speech distortion. This is in accordance with the expected sensitivity of the GSC when applied in a reverberated environment. The results stems from substantial signal leakage due to the fixed beamformer. The MCWF obtains comparable noise reduction performance, but opposed the GSC, the WSSM measure is positive for the MCWF. The MCWF is shown a little more robust to reverberation, but the filter length of 40 lacks to obtain a good beam pattern when the reverberation time increases. Using longer filters in the MCWF might obtain better results, as is indicated by simulations shown in Chapter 3. Another suggestion is to increase the long-term estimate. As indicated in Section 3.2.2, the MCWF relies on long-term estimates of the underlying stochastic processes. Therefore using approximately $1.5\,\mathrm{s}$ as basis for the estimation ($\lambda = 0.99975$, $f_{\mathrm{WF}} = 8\,\mathrm{kHz}/3$), as done in these tests, might be too short.

In this discussion we have discussed the speech enhancement performance of four noise reduction techniques in terms of noise reduction, broadband SNR, and speech distortion, WSSM. For spectral subtraction, the noise reduction performance was dependent on the noise estimation procedure applied. When applying babble noise, the estimation method of minimum statistics failed, and both noise reduction and speech distortion fell. For the other methods an ideal VAD was applied, as it was presumed that the underlying assumptions for the log-energy-based VAD would render the determination of noise and speech-dominated samples unsuccessful. Although an ideal VAD was used, the performance of the GSVD-based signal subspace method indicated, that the added complexity was not worthwhile compared to the performance of the spectral subtraction. The multi-channel Wiener filter showed promising results in the non-reverberated environment, both for white noise and babble noise. The speech distortion measure was drastically improved in both cases. However, for babble noise, the MCWF was outperformed by the GSC. When applying reverberation, the GSC, however, showed limited performance, likely due to the signal-leakage problems known to be attributed the fixed beamformer.

To summarise, it seems that under ideal circumstances, i.e. no reverberation and using an ideal VAD, the multi-channel methods outperform the single-channel methods. The multi-channel methods managed to reduce large amounts of noise, while reducing the speech distortion by decades of WSSM points. It seems that reverberation is an impeding factor for both single-channel and multi-channel noise reduction methods. This indicates that either one has to define convolutional noise explicitly in the formulation of the speech enhancement, or at least juxtapose the problems of additive and convolutional noise.

CHAPTER 6

Conclusions and Further Work

We have in this project addressed the problem of noise reduction and speech quality in speech enhancement techniques. We have investigated both single- and multi-channel methods, and finally we have chosen one method, where the initial steps towards an implementation was taken.

In the first chapter we outlined the field of speech enhancement, leading to the aim of this thesis, to improve the speech quality for a hearing impaired, with the assumption that by improving the quality for a normal hearing person, this will also improve the speech quality for a hearing impaired. In order to specify the problem, an observation model was formulated, from which it was evident that the speech quality was degraded by additive and convolutional noise. Improving the speech quality therefore consisted of primarily removing the additive noise and secondarily the convolutional noise, without introducing additional speech distortion. To measure the noise reduction and speech distortion different assessment techniques were investigated. We chose the broadband SNR, the segmental SNR, and the weighted spectral slope measure (WSSM). The SNR was chosen in order to have a well-known measure regarding noise reduction. WSSM was chosen to measure speech distortion and during this project it proved itself a robust, qualitative speech distortion measure with scores matching both informal listening tests among the group members and the information observable in our modified spectrograms.

Based on the goal of the project, to retain the speech signal as if is was recorded $1\ \mathrm{m}$ from the speaker in an anechoic chamber, we have examined both single- and multi-channel speech enhancement techniques, and a single dereverberation technique.

The single-channel methods gained a noise reduction performance at the expense of introduced speech distortion, as e.g. musical noise, in all noise types, being white, pink, and babble noise. Signal subspace approaches were seen to obtain worse results than spectral subtraction even though the subspace have higher computational complexity.

An additional observation was introduced and different beamforming techniques gained noise reduction and improved the WSSM measure. This was not seen in any of the single-channel methods. A delay-and-sum (DS), a generalised sidelobe canceller (GSC), and a multi-channel Wiener filter (MCWF) were presented and the MCWF was tested to be most robust with respect to different noise types, even though the GSC obtained better performance in a babble noise setup.

It quickly became evident that all methods depended on an estimate of the underlying stochastic process of the background noise. An ideal VAD was used at first in order to examine the *functionality* of the different methods. However, since noise estimation was identified as one the main problems in speech enhancement, two methods for estimating the background noise were presented; the well-established state-of-the-art method of minimum statistics proposed by Martin [59], and a voice activity detector (VAD) based on different log-energy distributions of speech and noise proposed by Gerven and Xie [29]. These methods obtained similar performance as with an ideal VAD using the spectral subtraction technique in white and pink noise. The methods however failed as expected in babble noise. It was shown that multi-channel methods using an ideal VAD was robust against babble noise, likely due to spatial information. As the MCWF is sensitive

in respect of a correct-working VAD, it is important to either develop robust VADs or to develop advanced methods for noise estimation of multi-channel observation, as Martin [59] and Cohen and Berdugo [11] have done for single-channel observations. Whereas, the log-energy-based VAD is seen to fail in the case of babble noise, Maj et al. [56] have proposed to use the output of the MCWF to operate the VAD. The initial frames are assumed noise dominated. Other interesting work, proposed by Herbordt et al. [42, 41, 40] concerns using a fixed low-performance beamformer for spatial separation of the noise and desired speech signal. In his method, the estimation of the background noise is shifted to the estimation of the coherence function of the room, which safely can be assumed stationary over several seconds in many applications. In order to integrate this method with the MCWF, a GEVD-based MCWF must be used.

In reverberated environments, all the speech enhancement methods were unable to obtain noise reduction, but the MCWF, however, obtained a small improvement in speech distortion.

Motivated by the lacking performance of speech enhancement in reverberated rooms, we investigated a method for dereverberation of the observed signal proposed by Affes and Grenier [1]. This method, however, relied on a priori information of a frequency-dependent ambiguity factor related to the impulse response of the room. We therefore developed a method, *spectral addition*, to estimate this ambiguity factor. Unfortunately, we did not find any improvement compared to the observation, by using either, the ambiguity factor estimated by spectral addition, nor by using the true impulse response for the room. Finally, we combined the decolouring method with the MCWF. However, the conclusion was that the method did not fulfil the assumptions taken by Affes and Grenier and thus noise reduction alone was shown to obtain better results.

It was identified in the previous chapter that additive and convolutional noise was two juxtaposed problems. Future work, could be regarded deconvolution techniques in order to improve speech quality. A topic which only has been touched upon in this thesis.

Considerations regarding an implementation of the time-domain MCWF was carried out and showed that a $32\,\mathrm{bit}$ floating-point recursion-based GSVD updating procedure was running in real-time on a Pentium-based workstation. The hardware modelling language SystemC was used to simulate a fixed point implementation, but fixed-point quickly revealed itself inferior to $32\,\mathrm{bit}$ floating-point due to limited dynamic in the matrix elements. Future work could investigate the dynamic range for each matrix element and thereby maybe obtain a realisable fixed-point implementation.

Bibliography

[1] AFFES, S., AND GRENIER, Y.
A signal subspace tracking algorithm for microphone array processing of speech.
IEEE Trans. on Speech and Audio Processing 5, 5 (September 1997), 425–437.

[2] ANDERSON, E., BAI, Z., BISCHOF, C., BLACKFORD, L. S., DEMMEL, J., DONGARRA, J. J., CROZ, J. D., HAMMARLING, S., GREENBAUM, A., MCKENNEY, A., AND SORENSEN, D.
LAPACK Users' Guide, third ed.
Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, Aug. 1999.

[3] ATAL, B.
Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification.
The Journal of the Acoustical Society of America 55, 6 (June 1974), 1304–1312.

[4] BAI, Z., AND DEMMEL, J. W.
Computing the generalized singular value decomposition.
SIAM Journal on Scientific Computing 14, 6 (1993), 1464–1486.

[5] BENESTY, J.
Adaptive eigenvalue decomposition algorithm for passive acoustic source localization.
The Journal of the Acoustical Society of America 107, 1 (Jan. 2000), 384–391.

[6] BEROUTI, M., SCHWARTZ, R., AND MAKHOUL, J.
Enhancement of speech corrupted by acoustic noise.
In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP '79.)* (Apr 1979), vol. 4, pp. 208–211.

[7] BOLL, S.
Suppression of acoustic noise in speech using spectral subtraction.
IEEE Trans. on Acoustics, Speech, and Signal Processing 27, 2 (Apr 1979), 113–120.

[8] CHARLIER, J., VANBEGIN, V., AND VAN DOOREN, P.
On efficient implementations of kogbetliantz's algorithm for computing the singular value decomposition.
Numerische Mathematik 52, 3 (May 1987), 279–300.

[9] CHEN, J., BENESTY, J., HUANG, Y. A., AND DOCLO, S.
New insights into the noise reduction wiener filter.
IEEE Trans.on Audio, Speech and Lang. Proc. 14, 4 (July 2006), 1218–1234.

[10] COHEN, I.
Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging.
IEEE Trans. on Speech and Audio Processing 11, 5 (Sept. 2003), 466– 475.

[11] COHEN, I., AND BERDUGO, B.
Speech enhancement for non-stationary noise environments.

Signal Processing 81, 11 (Nov. 2001), 2403–2418.

[12] COHEN, I., AND BERDUGO, B.
Noise estimation by minima controlled recursive averaging for robust speech enhancement.
IEEE Signal Processing Letters 9, 1 (Jan. 2002), 12–15.

[13] COMPERNOLLE, D. V.
DSP techniques or speech enhancement.
In *Proc. ESCA Workshop on Speech Processing in Adverse Conditions* (November 1992), pp. 21–30.

[14] COOLEY, J., TOOLAN, T., AND TUFTS, D.
A subspace tracking algorithm using the fast fourier transform.
Signal Processing Letters, IEEE 11, 1 (Jan. 2004), 30–32.

[15] DELLER, J. R., PROAKIS, J. G., AND HANSEN, J. H. L.
Discrete-Time Processing of Speech Signals.
Macmillan Publishing Company, Englewood Cliffs, New Jersey, 1993.

[16] DOCLO, S.
Multi-Microphone Noise Reduction and Dereverberation Techniques for Speech Applications.
PhD thesis, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 2001 Leuven (Heverlee), Belgium, March 2003.

[17] DOCLO, S., AND MOONEN, M.
Combined frequency-domain dereverberation and noise reduction technique for multi-microphone speech enhancement.
In *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)* (Darmstadt, Germany, September 2001), pp. 31–34.

[18] DOCLO, S., AND MOONEN, M.
Microphone Arrays - Signal Processing and Applications.
Springer-Verlag, 2001, ch. GSVD-Based Optimal Filtering for Multi-Microphone Speech Enhancement, pp. 111–133.

[19] DOCLO, S., AND MOONEN, M.
GSVD-based optimal filtering for single and multimicrophone speech enhancement.
IEEE Trans. on Signal Processing 50, 9 (September 2002), 2230– 2244.

[20] DOCLO, S., AND MOONEN, M.
Multimicrophone noise reduction using recursive GSVD-based optimal filtering with ANC postprocessing stage.
IEEE Trans. on Speech and Audio Processing 13, 1 (January 2005), 53– 69.

[21] DOLOGLOU, I., AND CARAYANNIS, G.
Physical interpretation of signal reconstruction from reduced rank matrice.
IEEE Trans. on Signal Processing 31, 7 (July 1991), 1681 – 1682.

[22] EPHRAIM, Y., AND MALAH, D.
Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator.
IEEE Trans. on Acoustics, Speech, and Signal Processing 32, 6 (Dec 1984), 1109–1121.

[23] EPHRAIM, Y., AND MALAH, D.
Speech enhancement using a minimum mean-square error log-spectral amplitude estimator.
IEEE Trans. on Acoustics, Speech, and Signal Processing 33, 2 (Apr 1985), 443–445.

[24] EPHRAIM, Y., AND VAN TREES, H.
A signal subspace approach for speech enhancement.
IEEE Trans. on Speech and Audio Processing 3, 4 (July 1995), 251–266.

[25] EVEREST, F. A.
Master Handbook of Acoustics, third ed.

McGraw-Hill, 1994.

[26] GARDNER, W. G.
The virtual acoustic room.
Master's thesis, Massachusetts Institute of Technology, 1982.

[27] GAROFOLO, J. S., LAMEL, L. F., FISHER, W. M., FISCUS, J. G., PALLETT, D. S., AND DAHLGREN, N. L.
"The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CDROM".

[28] GENTLEMAN, W. M.
Least Squares computations by Givens transformations without square roots.
IMA Journal of Applied Mathematics 12, 3 (1973), 329–336.

[29] GERVEN, S. V., AND XIE, F.
A comparative study of speech detection methods.
EUROSPEECH '97 (Sept. 1997), 1095–1098.

[30] GOH, Z., TAN, K.-C., AND TAN, T.
Postprocessing method for suppressing musical noise generated by spectral subtraction.
IEEE Transactions on Speech and Audio Processing 6, 3 (May 1998), 287–292.

[31] GOLUB, G. H., AND LOAN, C. F. V.
Matrix Computations, second ed.
Johns Hopkins University Press, 1989.
Fourth printing, 1993.

[32] GOLUB, G. H., AND LOAN, C. F. V.
Matrix Computations, third ed.
Johns Hopkins University Press, 1996.

[33] GRAY, R., BUZO, A., GRAY, A., AND MATSUYAMA, Y. J.
Distortion measures for speech processing.
IEEE Trans. on Acoustics, Speech and Signal Processing 28, 4 (August 1980), 367– 376.

[34] GRIFFITHS, L., AND JIM, C.
An alternative approach to linearly constrained adaptive beamforming.
IEEE Trans. on Antennas and Propagation 30, 1 (Jan 1982), 27–34.

[35] GÖTZE, J., AND SCHWIEGELSHOHN, U.
A square root and division free givens rotation for solving least squares problems on systolic arrays.
SIAM Journal on Scientific and Statistical Computing 12, 4 (1991), 800–807.

[36] HAMACHER, V., CHALUPPER, J., EGGERS, J., FISCHER, E., KORNAGEL, U., PUDER, H., AND RASS, U.
Signal processing in high-end hearing aids: State of the art, challenges, and future trends.
EURASIP Journal on Applied Signal Processing 18 (2005), 2915–2929.

[37] HANSEN, P. S. K.
Signal Subspace Methods for Speech Enhancement.
PhD thesis, Technical University of Denmark, 1997.

[38] HASLEV, P., BONNICHSEN, B., AND FREDRIKSEN, N.
Low-rank adaptive beamforming for speech processing.
Master's thesis, Institute of Electronic Systems Aalborg University, 2002.

[39] HAYKIN, S.
Adaptive Filter Theory, 4th ed.
Prentice Hall, Englewood Cliffs, New Jersey, 2001.

[40] HERBORDT, W., BUCHNER, H., NAKAMURA, S., AND KELLERMANN, W.
Outlier-robust dft-domain adaptive filtering for bin-wise stepsize controls, and its application to a generalized sidelobe canceller.

Conf. Rec. Intl. Workshop on Acoustic Echo and Noise Control (IWAENC) (Sept. 2005), 113–116.
Eindhoven, The Netherlands.

[41] HERBORDT, W., NAKAMURA, S., AND KELLERMANN, W.
Multichannel estimation of the power spectral density of noise for mixtures of nonstationary signals.
IPSJ SIG Technical Reports 131 (Dec. 2004), 211 – 216.
Kyoto, Japan.

[42] HERBORDT, W., TRINI, T., AND KELLERMANN, W.
Robust spatial estimation of the signal-to-interference ratio (SIR) for non-stationary mixtures.
Proc. of Int. Conf. on Acoustic Echo and Noise Control (IWAENC2003) (Sept. 2003), 247–250.
Kyoto, Japan.

[43] HIGHAM, N. J.
Accuracy and Stability of Numerical Algorithms, 2nd ed. ed.
SIAM, 2002.

[44] HOSHUYAMA, O., AND SUGIYAMA, A.
Microphone Arrays - Signal Processing Techniques and Applications.
Springer-Verlag, 2001, ch. Robust Adaptive Beamforming, pp. 87–111.

[45] HSIEH, S., LIU, K., AND YAO, K.
A unified square-root-free approach for QRD-based recursive-least-squares estimation.
IEEE Trans. on Acoustics, Speech, and Signal Processing 41, 3 (March 1993), 1405–1409.

[46] INGLE, V. K., AND MANOLAKIS, D. G.
Statistical and adaptive signal processing - spectral estimation, signal modeling, adaptive filtering, and array processing.
Artech House, 2005.

[47] JABLOUN, F., AND CHAMPAGNE, B.
Signal Subspace Techniques for Speech Enhancement.
Springer-Verlag, 2005, ch. 7, pp. 135–161.

[48] JANTSCH, A., KUMAR, S., AND HEMANI, A.
The rugby model: A conceptual frame for the study of modelling, analysis and synthesis concepts of electronic systems.
In *DATE '99: Proceedings of the conference on design, automation, and test in Europe* (Feb. 1999), ACM Press.

[49] JENSEN, S. H., HANSEN, P. C., HANSEN, S. D., AND SØRENSEN, J. A.
Reduction of broad-band noise in speech by truncated QSVD.
IEEE Trans. on Speech and Audio Processing 3, 6 (Nov. 1995), 439–448.

[50] KLATT, D. H.
Prediction of perceived phonetic distance from critical-band spectra: A first step.
IEEE International Conference on ICASSP '82 in Acoustics, Speech, and Signal Processing. 7 (May 1982), 1278–1281.

[51] LI, C., AND ANDERSEN, S.
Inter-frequency dependency in MMSE speech enhancement.
In *Signal Processing Symposium, 2004. NORSIG 2004. Proceedings of the 6th Nordic* (2004), pp. 200–203.

[52] LIM, J., AND OPPENHEIM, A.
Enhancement and bandwidth compression of noisy speech.
IEEE Proceedings 67, 12 (Dec. 1979), 1586–1604.

[53] LIU, W. M., JELLYMAN, K. A., MASON, J. S. D., AND EVANS, N. W. D.
Assessment of objective quality measures for speech intelligibility estimation.
In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 2006 (ICASSP 2006)* (May 2006), vol. 1, pp. I–1225 – I–1228.

[54] LUK, F. T.
A parallel method for computing the generalized singular value decomposition.
Journal of Parallel and Distributed Computing 2, 3 (August 1985), 250–260.

[55] LYONS, R. G.
Understanding Digital Signal Processing.
Prentice Hall, 2004.

[56] MAJ, J.-B., MOONEN, M., AND WOUTERS, J.
A robust voice activity detetor for SVD-based noise reduction in hearing aids.
In *3rd IEEE Benelux Signal Processing Symposium (SPS-2002)* (Leuven, Belgium, March 2002).

[57] MARTIN, R.
An efficient algorithm to estimate the instantaneous SNR of speech signals.
In *Proceedings Eurospeech-93* (Berlin, Sep. 1993), pp. 1093–1096.

[58] MARTIN, R.
Spectral subtraction based on minimum statistics.
In *Proc. Eur. Signal Processing conf.* (1994), pp. 1182–1185.

[59] MARTIN, R.
Noise power spectral density estimation based on optimal smoothing and minimum statistics.
IEEE Trans. on Speech and Audio Processing 9, 5 (July 2001), 504–512.

[60] MCGOVERN, S. G.
A model for room acoustics.
Internet [Matlab Exchange], 2003.
http://2pi.us/rir.html and http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=5116.

[61] MERHAV, N.
The estimation of the model order in exponential families.
IEEE Transactions on Information Theory 35, 5 (Sept. 1989), 1109–1114.

[62] MEYER, J., SIMMER, K. U., AND KAMMEYER, K. D.
Comparison of one and two-channel noise estimation techniques.
In *Workshop Acoustic Echo Control Noise Reduction* (1997), pp. 17–20.

[63] MOONEN, M., , VAN DOOREN, P., AND VANDEWALLE, J.
A systolic algortihm for QSVD updating.
Signal Processing 25 (1991), 203–213.

[64] MOONEN, M., DOOREN, P. V., AND VANDEWALLE, J.
A systolic array for SVD updating.
SIAM Journal on Matrix Analysis and Applications 14, 2 (1991), 353–371.

[65] MOONEN, M., DOOREN, P. V., AND VANDEWALLE, J.
A singular value decomposition updating algorithm for subspace tracking.
SIAM Journal on Matrix Analysis and Applications 13, 4 (1992), 1015–1038.

[66] MOOR, B. D.
The singular value decomposition and long and short spaces of noisy matrices.
IEEE Trans. on Signal Processing 41, 9 (September 1993), 2826 – 2838.

[67] OPPENHEIM, A. V., AND SCHAFER, R. W.
Discrete-Time Signal Processing, second ed.
Prentice-Hall, Upper Saddler River, New Jersey 07458, 1999.

[68] OSCI.
v. 2.0.1 User's Guide.
Open SystemC Initiative (OSCI), 2002.
http://systemc.org/.

[69] PAIGE, C. C.
Computing the generalized singular value decomposition.
SIAM Journal on Scientific and Statistical Computing 7, 4 (1986), 1126–1146.

[70] PANGO, P., AND CHAMPAGNE, B.
On the efficient use of Givens rotations in SVD-based subspace tracking algorithms.
Signal Processing 74, 3 (May 1999), 253–277.

[71] PESQ, I.-R. P.
Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.
Tech. rep., ITU, February 2001.

[72] POULSEN, T.
Taleforståelighed, 3 ed.
Laboratoriet for Akustik, Bygning 352, DTH, 2800 Lyngby, 1990.
Note nr. 3111.

[73] PROAKIS, J. G., AND MANOLAKIS, D. G.
Digital Signal Processing: Principles, Algorithms, and Applications, third ed.
Prentice-Hall, Upper Saddler River, New Jersey 07458, 1996.

[74] QUACKENBUSH, S. R., BARNWELL III, T. P., AND CLEMENTS, M. A.
Objective Measures of Speech Quality.
Prentice-Hall, 1988.

[75] QUATIERI, T. F.
Discrete-Time Speech Signal Processing.
Prentice Hall, Upper Saddle River, NJ 07458, 2001.

[76] RABIDEAU, D.
Fast, rank adaptive subspace tracking and applications.
IEEE Trans. on Acoustics, Speech,and Signal Processing 44, 9 (Sept. 1996), 2229–2244.

[77] RABINER, L., AND B.H., J.
Fundamentals of Speech Recognition.
Prentice Hall PTR, Englewood Cliffs, NJ, April 1993.
ISBN:0130151572.

[78] REAL, E., TUFTS, D., AND COOLEY, J.
Two algorithms for fast approximate subspace tracking.
IEEE Trans. on Acoustics, Speech,and Signal Processing 47, 7 (July 1999), 1936–1945.

[79] SCHARF, L., AND TUFTS, D.
Rank reduction for modeling stationary signals.
IEEE Trans. Acoustics, Speech, and Signal Processing 35, 3 (Mar 1987), 350–355.

[80] SCHARF, L. L.
Statistical signal processing - detection, estimation, and time series analysis.
Addison-Wesley, Reading, Mass, 1991.

[81] SHANMUGAN, K. S., AND BREIPOHL, A. M.
Random Signals - Detection, Estimation, and Data Analysis.
John Wiley & Sons, 1988.

[82] SHYNK, J. J.
Frequency-domain and multirate adaptive filtering.
IEEE Signal Processing Magazine 9, 1 (January 1992), 14–37.

[83] SPRIET, A., MOONEN, M., AND WOUTERS, J.
More than 2 microphones in digital hearing aids: a clear benefit for speech intelligibility in noise !
Internal report 00-143, K.U.Leuven, Heverlee, Leuven, Belgium, 2000.

[84] SPRIET, A., MOONEN, M., AND WOUTERS, J.
Stochastic gradient implementation of spatially pre-processed multi-channel wiener filtering for noise reduciton in hearing aids.
IEEE Proc. on Acoustics, Speech and Signal Processing 4 (May 2004), 57–60.

[85] STEWART, G. W.
A jacobi-like algorithm for computing the schur decomposition of a nonhermitian matrix.
SIAM Journal on Scientific and Statistical Computing 6, 4 (1985), 853–864.

[86] STRANG, G., AND BORRE, K.
Linear Algebra, Geodesy, and GPS, first ed.
Wellesley-Cambridge Press, 1997.

[87] SUNG, W., AND KUM, K.-I.
Simulation-based word-length optimization method for fixed-point signal processing systems.
IEEE Trans. on Signal Processing 43, 12 (Dec. 1995), 3087–3090.

[88] TUFTS, D., REAL, E., AND COOLEY, J.
Fast approximate subspace tracking (fast).
In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on* (21-24 April 1997), vol. 1, pp. 547–550vol.1.

[89] VEEN, B. D. V., AND BUCKLEY, K. M.
Beamforming: A versatile approach to spatial filtering.
IEEE Acoustics, Speech and Signal Processing Magazine 5, 2 (April 1988), 4–24.

[90] WALKER, R. A., AND THOMAS, D. E.
A model of design representation and synthesis.
In *DAC '85: Proceedings of the 22nd ACM/IEEE conference on design automation* (1985), ACM Press, pp. 453–459.

[91] WANG, S., SEKEY, A., AND GERSHO, A.
An objective measure for predicting subjective quality of speech coders.
IEEE Journal on Selected Areas in Communications 10, 5 (June 1992), 819–829.

[92] YANG, B.
Projection approximation subspace tracking.
IEEE Trans. on Acoustics, Speech and Signal Processing 43, 1 (Jan. 1995), 95 – 107.

APPENDIX A

Definitions from Linear Algebra

In this appendix we aim at reviewing a number of, for this project, important linear algebra definitions. It should introduce the reader to our notation and for further reference we refer to a text book by Strang and Borre [86] or the classical reference on linear algebra with emphasis on numerical aspects and implementation by Golub and Van Loan [32, 31].

We firstly introduce a number of properties and structures, that is special matrices, followed by a review of a number of important decompositions. We have in this review confined ourselves to real matrices and vectors.

A.1 Structured Matrices

Definition A.1 (Toeplitz) A $p \times m$ rectangular matrix $\mathbf{T}$ is called Toeplitz iff it has constant elements along the diagonals

$$\mathbf{T} = \begin{bmatrix} r_0 & r_1 & r_2 & \dots & r_{m-1} \\ r_{-1} & r_0 & r_1 & \dots & r_{m-2} \\ r_{-2} & r_{-1} & r_0 & \dots & r_{m-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{1-p} & r_{2-p} & r_{3-p} & \dots & r_{m-p} \end{bmatrix} \tag{A.1}$$

a special case is the auto-correlation amtrix, which we have depicted in eq. (A.1).

Another special matrix is one holding the Hankel structure, which is a column-permuted version of the Toeplitz matrix. The permutation is done with the (column) exchange matrix

$$\mathbf{J} = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & & \vdots & \vdots \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{A.2}$$

and this column permutation of $\mathbf{T}$ leads us to the Hankel structure by $\mathbf{TJ} = \mathbf{H}$.

Definition A.2 (Hankel) A $p \times m$ rectangular matrix $\mathbf{H}$ is called Hankel iff it has constant elements along the anti-diagonals

$$\mathbf{H} = \begin{bmatrix} x[m-1] & \dots & x[2] & x[1] & x[0] \\ x[m-2] & \dots & x[1] & x[0] & x[-1] \\ x[m-3] & \dots & x[0] & x[-1] & x[-2] \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x[m-p] & \dots & x[3-p] & x[2-p] & x[1-p] \end{bmatrix} \tag{A.3}$$

By these two Definitions A.1 and A.2 we have introduced two applications of Toeplitz and Hankel systems for signal processing, viz. correlation matrices and data matrices for filtering.

Definition A.3 (Triangular) A triangular matrix $\mathbf{A}$ is a matrix which has zeros either below or above its main diagonal.

$\mathbf{A}$ is upper triangular if $a_{ij} = 0$ whenever $i > j$.

$\mathbf{A}$ is lower triangular if $a_{ij} = 0$ whenever $i < j$.

$\mathbf{A}$ is triangular iff it is either upper or lower triangular.

Definition A.4 (Symmetric) A matrix $\mathbf{A}$ is symmetric iff $\mathbf{A}^T = \mathbf{A}$. This requires the matrix to be square, of course.

Definition A.5 (Positive Definite Symmetric) An $m \times m$ matrix $\mathbf{A}$ is positive symmetric definite iff $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ for all nonzero $x \in \mathbb{R}^m$. A positive definite symmetric matrix will have strictly positive ($\lambda_i > 0$) eigenvalues and positive pivots [86].

Definition A.6 (Hessenberg) A Hessenberg matrix, $\mathbf{H}$, is like a triangular matrix, except that the elements adjacent to the diagonal can be non-zero. A matrix is called upper Hessenberg if $\mathbf{H}_{ij} = 0$ for $i > j + 1$, and lower Hessenberg if $\mathbf{H}_{ij} = 0$ for $i < j - 1$.

A symmetric (hermitian for the complex-valued case) Hessenberg matrix has zero-valued elements except non-zero elements on or immediately adjacent to the main diagonal. This matrix is also called tridiagonal or Jacobi.

A.2 Matrix Decompositions

Definition A.7 (EVD) Assume $\mathbf{A}$ is a square $m \times m$ matrix. The eigenvalues of $\mathbf{A}$ is the m roots of the characteristic polynomial $p(z) = \det(z\mathbf{I} - \mathbf{A})$, denoted by λ_k. The non-zero vectors $\mathbf{v}_k$ that satisfy

$$\mathbf{A}\mathbf{v}_k = \lambda_k \mathbf{v}_k \tag{A.4}$$

are called eigenvectors. Define a diagonalising transformation of $\mathbf{A}$ as [32, 86]

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \quad \text{where} \quad \Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m) \tag{A.5}$$

where the columns of $\mathbf{V}$ comprises the eigenvectors $\mathbf{v}_k$ which forms an orthogonal basis for $\mathbf{A}$. Traditionally the eigenvalues in the diagonal matrix $\mathbf{\Lambda}$ are ordered in a decreasing manner $1 \geq \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0$.

Definition A.8 (GEVD) Assume $\mathbf{A}$ and $\mathbf{B}$ linearly independent, square $n \times m$ matrices. Then the generalised eigenvalue decomposition of $\mathbf{A}$ and $\mathbf{B}$ is

$$\mathbf{A}\mathbf{X} = \mathbf{B}\mathbf{X}\mathbf{\Lambda} \tag{A.6}$$

with $\mathbf{X}$ an invertible, but not necessarily orthogonal matrix, containing the generalised eigenvectors column-wise. Alternatively, this can be reformulated as

$$\begin{cases} \mathbf{A} = & \mathbf{Q}\mathbf{\Lambda}_A\mathbf{X}^{-1} \\ \mathbf{B} = & \mathbf{Q}\mathbf{\Lambda}_B\mathbf{X}^{-1} \end{cases} \tag{A.7}$$

with $\mathbf{Q}$ an $n \times m$ matrix and $\Lambda = \Lambda_A\Lambda_B^{-1}$.

Definition A.9 (SVD) The singular value decomposition (SVD) is a natural extension of the eigenvalue decomposition (EVD). In the EVD we diagonalise, say, $\mathbf{A}$, using one orthonomal basis. In the SVD we do not constrain ourselves to square matrices, thus we need two orthogonal bases to diagonalise a rectangular $\mathbf{A}$.

Let $\mathbf{A}$ be an $p \times m$ rectangular matrix where $\text{rank}(\mathbf{A}) = k \leq \min(p, m)$, then the singular value decomposition $\mathbf{A}$ is given by

$$\begin{aligned} \mathbf{U}^T\mathbf{A}\mathbf{V} &= \mathbf{\Sigma}_A = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_m) \\ \sigma_1 &\geq \sigma_2 \geq \ldots \geq \sigma_m \geq 0 \end{aligned} \tag{A.8}$$

where $\mathbf{U} \in \mathbb{R}^{p\times m}$ and $\mathbf{V} \in \mathbb{R}^{m\times m}$. The diagonal matrix, $\mathbf{\Sigma}_A$ contains the singular values sorted in decreasing order. The vectors of

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \tag{A.9}$$

GSVD

are called left singular vectors. Suppose the matrix $\mathbf{A}$ is of rank k, then the columns of $\mathbf{U}_1 \in \mathbb{R}^{p\times k}$ and $\mathbf{U}_2 \in \mathbb{R}^{p\times m-k}$ corresponds to the column space and left null space of $\mathbf{A}$, respectively. The vectors of

$$\mathbf{V}^T = \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \tag{A.10}$$

are called right singular vectors, and the columns of $\mathbf{V}_1^T \in \mathbb{R}^{m\times k}$ and $\mathbf{V}_2^T \in \mathbb{R}^{m\times m-k}$ corresponds to the row space and null space of $\mathbf{A}$, respectively. This is the Fundamental Theorem of linear algebra.

Sometimes an economy or thin SVD is applied. For the $p \times m$ matrix $\mathbf{A}$, the SVD can be correctly represented by

$$\begin{aligned} \mathbf{A} &= \widetilde{\mathbf{U}}_1\mathbf{\Sigma}_1\mathbf{V}^T \quad \text{where} \\ \widetilde{\mathbf{U}}_1 &= \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 & \ldots & \mathbf{U}_m \end{bmatrix} \\ \mathbf{\Sigma}_1 &= \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_m) \end{aligned} \tag{A.11}$$

The SVD finds use in principle component analysis, numerical weather prediction, study of linear inverse problems, and in signal processing. The SVD relates to least squares. The generalised singular value decomposition (GSVD) relates to constrained least squares.

Definition A.10 (GSVD) Defining rectangular $p \times m$ matrix $\mathbf{A}$ and rectangular $q \times m$ matrix $\mathbf{B}$, the GSVD of $\mathbf{A}$ and $\mathbf{B}$ is defined as [32]

$$\begin{cases} \mathbf{U}_A^T\mathbf{A}\mathbf{Q} = & \mathbf{\Sigma}_A\mathbf{R} \\ \mathbf{V}_B^T\mathbf{B}\mathbf{Q} = & \mathbf{\Sigma}_B\mathbf{R} \end{cases} \tag{A.12}$$

where $\mathbf{\Sigma}_A = \text{diag}(\alpha_1, \alpha_2, \ldots, \alpha_m)$ and $\mathbf{\Sigma}_B = \text{diag}(\beta_1, \beta_2, \ldots, \beta_m)$. By reordering (A.12) and equating $\mathbf{R}$, we obtain following

$$\mathbf{U}_A^T\mathbf{A}\mathbf{Q} = \left(\mathbf{\Sigma}_A\mathbf{\Sigma}_B^{-1}\right)\mathbf{V}_B^T\mathbf{B}\mathbf{Q} \tag{A.13}$$

which shows, that $\mathbf{U}_A^T\mathbf{A}\mathbf{Q}$ and $\mathbf{V}_B^T\mathbf{B}\mathbf{Q}$ have parallel rows with row-scaling factor $\mathbf{\Sigma} = \mathbf{\Sigma}_A\mathbf{\Sigma}_B^{-1} = \text{diag}(\alpha_1/\beta_1, \alpha_2/\beta_2, \ldots, \alpha_m/\beta_m)$. These quotients are called the *generalised singular values*, $\sigma_i = \alpha_i/\beta_i$.
The elements of $\mathbf{\Sigma}_A$ and $\mathbf{\Sigma}_B$ are normally ordered as in decreasing, $1 \leq \alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_m \leq 0$, increasing order, $0 \geq \beta_1 \geq \beta_2 \geq \cdots \geq \beta_m \geq 1$, respectively.

By reordering (A.13) we obtain

$$\mathbf{U}_A^T\left(\mathbf{A}\mathbf{B}^{-1}\right)\mathbf{V}_B = \mathbf{\Sigma} \tag{A.14}$$

which emphasises, that the GSVD of $\mathbf{A}$ and $\mathbf{B}$ corresponds to the SVD of $\mathbf{A}\mathbf{B}^{-1}$ [54]. Lastly, one can rewrite equation (A.12) to a widely used form, which resembles two separate SVDs, as

$$\begin{cases} \mathbf{A} = & \mathbf{U}_A\mathbf{\Sigma}_A\mathbf{X}^T \\ \mathbf{B} = & \mathbf{V}_B\mathbf{\Sigma}_B\mathbf{X}^T \end{cases} \quad \text{where} \quad \mathbf{X}^T = \mathbf{R}\mathbf{Q}^T \tag{A.15}$$

This is the form used throughout the report. From (A.14) it can be seen how to compute the GSVD of $\mathbf{A}$ and $\mathbf{B}$ by forming $\mathbf{A}\mathbf{B}^{-1}$, however, Paige [54, 69] has shown how to compute the GSVD with implicit inversion of $\mathbf{B}$, which is numerically preferable.

Definition A.11 (Givens rotation) A Givens rotation (or plane rotation) rotates the vector $\mathbf{x} = \begin{bmatrix} x_i & x_j \end{bmatrix}^T$ through the (i,j)th plane to $\mathbf{y} = \begin{bmatrix} y_i & y_j \end{bmatrix}^T$ by multiplication [43]

$$\mathbf{y} = \mathbf{\Theta}(i,j)\mathbf{x}$$

where the Givens matrix, $\mathbf{\Theta}(i,j)$, is given by

$$\mathbf{\Theta}(i,j) = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{A.16}$$

The rotation offered by Givens rotation is usually to introduce zeros in a vector (or matrix) one at a time. The (i,j) is sometimes omitted whenever it is clear from the context. Defining

$$y_k = \begin{cases} x_k, & k \neq i,j \\ cx_i + sx_j, & k = i \\ -sx_i + cx_j, & k = j \end{cases} \tag{A.17}$$

can be used to introduce a zero, $y_j = 0$, when

$$s = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}, \quad c = \frac{x_i}{\sqrt{x_i^2 + x_j^2}} \tag{A.18}$$

As can be seen, the rotation angle, θ, never needs to be explicitly computed, since c and s are all that are needed to do the planar rotation.

Definition A.12 (QR) The QR decomposition of a rectangular $p \times m$ matrix $\mathbf{A}$ is given by [32]

$$\mathbf{A} = \mathbf{QR} \tag{A.19}$$

where $\mathbf{Q} \in \mathbb{R}^{p \times p}$ is an orthonormal (orthogonal with unit-length column vectors) matrix, and $\mathbf{R} \in \mathbb{R}^{p \times m}$ is upper triangular. If $\mathbf{A}$ has full rank, the first m columns of $\mathbf{Q}$ forms an orthonomal basis for $\mathrm{rank}(\mathbf{A}) = m$.

The QR decomposition of $\mathbf{A}$ relates to the Cholesky of $\mathbf{A}^T\mathbf{A}$ as in

$$\mathbf{A}^T\mathbf{A} = \mathbf{R}^T\mathbf{R} \tag{A.20}$$

where $\mathbf{R}$ is the Cholesky factor of $\mathbf{A}^T\mathbf{A}$, but also the triangular matrix, $\mathbf{R}$, in (A.19).

The computations involved in forming $\mathbf{Q}$ and $\mathbf{R}$ from $\mathbf{A}$ can be based on Householder, Givens, or fast Givens transoformations. In this report we focus on Givens transformations because of the local (2 by 2) operations in order to re-orthogonalise an almost triangular $\mathbf{R}$. Whilst $\mathbf{R}$ is an $p \times m$ upper triangular, rectangular matrix, only the upper square part contains information. The $m - p$ lower rows contains zeros only, and can be omitted in an economy QR-decompostion (also known as thin QR)

$$\mathbf{Q}^{-1}\mathbf{A} = \mathbf{Q}^T\mathbf{A} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \tag{A.21}$$

A.3 Matrix and Vector Norms

Definition A.13 (p-norm) We introduce the p-norm of a length-K vector, $\mathbf{x}$ as [32]

$$\|\mathbf{x}\|_p = \sqrt{\sum_{i=1}^{K} x_i^p} = \left(\mathbf{x}^T\mathbf{x}\right)^{1/p} \tag{A.22}$$

where choices of p traditionally include 1, 2, and ∞. The 1-norm is the sum of absolute values, the 2-norm (or $\mathcal{L}-2$) is often used to measure the euclidean distance. The ∞-norm equals $\max(x_i), 1 \leq i \leq K$.

Definition A.14 (off-diagonal norm) The off-diagonal norm is interesting as a measure of approximate diagonal matrix decompositions. It is a measure of the energy which resides on the off diagonal. A matrix diagonalisation can be approximated and to quantify the quality of the approximation, the off-norm of $\mathbf{A}$, say, is a useful measure. The off-norm of an $p \times m$ matrix $\mathbf{A}$ is defined as [32]

$$\text{off}(\mathbf{A}) = \sum_{i=1}^{p} \sum_{\substack{j=1 \\ i \neq j}}^{m} \mathbf{A}_{ij}^2 \tag{A.23}$$

where $\mathbf{A}_{ij}$ denotes the scalar element on the (i, j)th position of matrix $\mathbf{A}$.

APPENDIX B

Techniques for Noise Estimation

In a stochastic approach to optimal filtering, for example Wiener filtering or matched filtering, estimates of the signal second-order statistics are a prevalent problem. Speech enhancement is in its simplest form concerned with retrieving a speech signal which is observable embedded in noise. In order to obtain second-order statistics of noise and speech, estimators holding adequate characteristics regarding bias and variance are needed. However, this only solves the estimation problem if the noise and speech signal can be separately observed, which is not realistic in speech enhancement applications. Data used in the estimator needs to be qualified by some updating policy. Frequently, voice activity detection, histogram, or thresholding techniques have been employed [62]. Assuming the noise to be stationary during speech periods is the main drawback of VAD-based estimation. It is more desirable to update the estimators continuously, as for example by the celebrated method by Martin [59, 57], which has proven very efficient for spectral subtraction [58], even for non-stationary noise.

The efficiency/usefulness of optimal filters are related to whether speech signal and interference signal are residing in different frequency bands, or, that we safely can assume different dynamics, that is, stochastic characteristics. The most advanced/recent algorithms for noise estimation are not based on either assumption, but on different probabilities of interference and speech signal [62, 59, 10]. The power spectral density (PSD) of the speech-interference mixture frequently decays to a spectral floor which is tracked and used as an estimate of the noise. This type of estimation fails whenever the interference PSDs are rapidly time-varying, as for example those of noise or music sources, since the tracked minima are not representative for the interference PSD [41].

In speech enhancement, the speech signal of interest is estimated based on the statistics of the observed signal, noise and speech, and an estimate of the noise statistics. For multi-channel optimal filtering, such as the multi-channel Wiener filter, the second-order statistics contains both auto-correlations, one for each channel, and cross-correlations, one between each channel. Algorithms for multi-channel noise estimation are traditionally based on the assumption, that the speech signal is highly correlated at each sensor, while the interference(s) are either diffuse or spatially separated from the speech signal source [62]. One recent method based on a DS beamformer in combination with thresholding based on minimum/maximum estimates is presented by Herbordt [42, 41]. This method exploits the quasi-stationarity in the coherence function rather than in the auto-correlation quasi-stationarity.

The following sections will deal with fundamental issues regarding estimation of correlation sequences and periodograms.

B.1 Estimation of the Correlation Sequence

For a stochastic approach to optimal filtering, statistical quantities of the stochastic process need to be known, or to be estimated. In the latter case, the definitions of various statistics are determined by means of the ensemble operator, which means we need to do estimation using multiple realisations of the process in a "time-frozen" environment. This is of course very impractical in speech applications where we have access to only one realisation of the speech or noise process. Therefore we rely on time averages, which allow us to infer the statistical characteristics from a single realisation.

For ergodic, and most stationary and quasi-stationary, processes, the time-averages are sufficient to estimate the time-invariant statistics. This means for an ergodic process, the statistics can be inferred from either ensemble or time averaging. For a non-stationary process, the statistics are generally time-variant, and cannot be estimated by means of time averaging, that is, of course, we cannot assume quasi-stationarity and do estimation, which is valid within the nearly-stationary time scope.

We will in the following sections introduce estimators of the measure of statistical relation between one sequence and itself, *auto-correlation*, or between two sequences, *cross-correlation*. The frequency counterpart, the *power spectral density* (PSD), will also be treated and we will touch upon normalised measures, such as *coherence*.

The auto-correlation of a discrete-time signal, $x[n]$, is, defined using the ensemble operator, $E\{\}$, given by

$$r_{xx}(n_1, n_2) = E\left\{x[n_1]x^*[n_2]\right\} \tag{B.1a}$$

where * denotes the complex conjugate (even though we assume time signals real-values). In a similar manner, the cross-correlation, to measure statistical relationship between two signals, $x[n]$, and, $y[n]$, is defined as

$$r_{xy}(n_1, n_2) = E\left\{x[n_1]y^*[n_2]\right\} \tag{B.1b}$$

Those definitions, (B.1a) and (B.1b), are ensemble averages averaging multiple products of realisations at time indices n_1 and n_2. Using the ergodic property we can redefine the correlation measures by time averages

$$r_{xx}(l) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n]x^*[n-l] \tag{B.2a}$$

$$r_{xy}(l) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x[n]y^*[n-l] \tag{B.2b}$$

In a real application we only have finite-length data segments available. We then use the $(2N+1)$-length data segment to make an estimate, $\hat{r}_{xx}(l)$, of the true quantity, $r_{xx}(l)$. An example of an efficient empirical auto-correlation estimator, is presented in [24], and defined as

$$\hat{r}_{xx}(l)[k] = \begin{cases} \frac{1}{2N} \sum\limits_{n=-N+1}^{N-l} x[k+n]x[k+n+l] & 0 \le l \le M-1, \\ 0 & \text{otherwise} \end{cases} \tag{B.3}$$

which can be seen as a convolution (without the folding, or time-reversing) of ever shorter signals. The time lag, l, is the difference between n_1 and n_2, which is constant for wide-sense stationary processes (ergodicity is a stronger assumption, than stationarity). The larger the time lag, l, the shorter data sequences are used in computing, $\hat{r}_{xx}(l)[k]$, (for time $k = N$, the indices used are $k+n = 1 \ldots 2N-l$ and $k+n+l = 1+l \ldots 2N$, clearly shorter segments for larger l). This type of windowing is known as the auto-correlation method in the speech processing community [39],

and maximises the number of samples used for each l. This method is generally preferred over pre- and post-windowing, and the less structured auto-covariance method. The latter forms the basis for a data matrix which is the basis for may SVD-based algorithms. We will return to the data matrix in (B.4).

It is worth noting that in a practical application at least $N = 50$ samples should be used, and that one should respect $|l| < N/4$ to create reliable results [46]. If it is desired to compute (B.3) in a realisable manner, we can shift the sum to only compute over past (causal) samples. This does not change the characteristics of (B.3). Assuming stationary and ergodic conditions, the empirical correlation sequences converge in probability to the true correlation sequence, i.e.

$$r_{xx}(l) = \lim_{N\to\infty} \hat{r}_{xx}(l)[k]$$

which is, however, a biased estimate of the true auto-correlation. This is generally preferred over the unbiased estimator, normalised by $1/(2N-l)$, because (B.3) guarantees a symmetric, positive definite auto-correlation matrix [46]. Positive definiteness is also (almost always) guaranteed due to observation data corrupted by additive noise.

Another frequently used empirical auto-correlation estimator is given by

$$\hat{r}_{xx}(l)[k] = \begin{cases} \frac{1}{2N} \sum_{n=-N}^{N} x[k+n]x[k+n+l] & 0 \leq l \leq M-1, \\ 0 & \text{otherwise} \end{cases} \tag{B.4}$$

mainly because this particular estimator forms the basis for a Toeplitz data-matrix product estimator. It is also known as the auto-covariance windowing method in the speech processing community [39]. This is useful in the context of optimal filtering using singular value decomposition. First let us introduce correlation matrices, then we will return to the properties of this estimator, and why it is useful when considering the singular value decomposition.

For the one-channel case, assuming wide-sense stationarity, the auto-correlation matrix is defined as

$$\mathbf{R}_{xx} = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \dots & r_{xx}(M-1) \\ r_{xx}(1) & r_{xx}(0) & \dots & r_{xx}(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(M-1) & r_{xx}(M-2) & \dots & r_{xx}(0) \end{bmatrix} \tag{B.5}$$

which can be formed by using the first M samples of the empirical auto-correlation sequence, (B.2a), in a Toeplitz structure [47]. For a 3-channel observation, this means, that the cross-correlation matrix, $\mathbf{R}_{xx}$, will equal

$$\mathbf{R}_{xx} = \begin{bmatrix} \mathbf{R}_{x_1x_1} & \mathbf{R}_{x_1x_2} & \mathbf{R}_{x_1x_3} \\ \mathbf{R}_{x_2x_1} & \mathbf{R}_{x_2x_2} & \mathbf{R}_{x_2x_3} \\ \mathbf{R}_{x_3x_1} & \mathbf{R}_{x_3x_2} & \mathbf{R}_{x_3x_3} \end{bmatrix} \tag{B.6}$$

which is seen to hold a block-Toeplitz structure. The main diagonal holds the auto-correlation (temporal-only) information, while the other entries holds spatial information as well (cross-terms). In a similar manner, this matrix can be formed using M samples (from each channel) using (B.3).

In order to form the auto-correlation martrix of (B.5), we can use either of (B.3) or (B.4), but the results will be quite different. The method that seems most popular in the adaptive filtering community [39], is the auto-covariance method. This method relates to the singular value decomposition, in that, by forming a data vector

$$\mathbf{x}[k] = \begin{bmatrix} x[k] & x[k-1] & \dots & x[k-M+1] \end{bmatrix}^T \tag{B.7}$$

of length M, and insert these segmented observation vectors into an $L \times M$ *Toeplitz data matrix*

structure, where $N = L + M - 1$ and usually $L > M$

$$\mathbf{X}[k] = \mathcal{T}(\mathbf{x}[k]) = \begin{bmatrix} \mathbf{x}^T[k-L+1] \\ \mathbf{x}^T[k-L+2] \\ \vdots \\ \mathbf{x}^T[k-1] \\ \mathbf{x}^T[k] \end{bmatrix} = \begin{bmatrix} x[k-L+1] & x[k-L] & \dots & x[k-N+1] \\ x[k-L+2] & x[k-L+1] & \dots & x[k-N+2] \\ \vdots & \vdots & \ddots & \vdots \\ x[k-1] & x[k-2] & \dots & x[k-M] \\ x[k] & x[k-1] & \dots & x[k-M+1] \end{bmatrix} \tag{B.8}$$

we form the auto-correlation matrix estimate by pre-multiplication the Toeplitz data matrix by itself transposed

$$\hat{\mathbf{R}}_{xx} = \mathbf{X}[k]^T \mathbf{X}[k]/L \in \mathbb{R}^{M \times M} \tag{B.9}$$

The resulting auto-correlation matrix will have M *different* entries of $r_{xx}(l = 0)$, due to the different samples used in each column of (B.8). It can be seen from the data matrix (B.8), that each entry of (B.5) is a product-sum over M samples. The resulting matrix is a symmetric, non-Toeplitz correlation matrix

$$\hat{\mathbf{R}}_{yy} = \begin{bmatrix} \hat{r}_{yy}(0,k) & \hat{r}_{yy}(1,k) & \dots & \hat{r}_{yy}(M-1,k) \\ \hat{r}_{yy}(1,k) & \hat{r}_{yy}(0,k+1) & \dots & \hat{r}_{yy}(M-2,k+1) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{r}_{yy}(M-1,k) & \hat{r}_{yy}(M-2,k+1) & \dots & \hat{r}_{yy}(0,k+M-1) \end{bmatrix} \tag{B.10}$$

with non-constant diagonals. The advantage of the covariance windowing method used in (B.8) is, that an SVD of symmetric $\mathbf{X}[k]$ is equivalent to an EVD of $\hat{\mathbf{R}}_{xx} = \mathbf{X}[k]^T \mathbf{X}[k]$, which is advantageous from a numerical perspective.

We could summarise, by saying we use only length-M samples for each entry in the auto-correlation matrix using the covariance method, although we have N samples available. One remedy is to use a more structured windowing, such as the auto-correlation method, as suggested by Ephraim and Van Trees [24], and seen in Eq. (B.3). Using this estimator much better, and more data efficient estimators can be achieved. Using (B.3), we redefine the data matrix as

$$\mathbf{X}'[k] = \left[\begin{array}{ccc|ccc} x[k] & x[k-1] & \dots & x[k-L+1] & x[k-L] & \dots & x[k-N+1] \\ 0 & x[k] & \dots & x[k-L+2] & x[k-L+1] & \dots & x[k-N+2] \\ \vdots & \vdots & \ddots & \multicolumn{3}{c}{\vdots} & \vdots \\ 0 & 0 & \dots & x[k] & x[k-1] & \dots & x[k-M+1] \end{array}\right. \left.\begin{array}{ccc} & & \\ & & \\ & & \\ 0 & \dots & 0 \\ x[k-N+1] & \dots & 0 \\ \vdots & \ddots & \vdots \\ x[k-M] & \dots & x[k-N+1] \end{array}\right] \tag{B.11}$$

and use (B.9) to form (B.5). The resulting auto-correlation matrix is symmetric and Toeplitz, as opposed to the one formed using (B.8). In order to see the relation, we have used vertical lines to emphasise the truncation performed using the covariance windowing method, (B.8), over the auto-correlation windowing method, (B.11), [39]. For speech processing the auto-correlation windowing is generally preferred for stability.

Having introduced the problem of estimating the auto- and cross-correlation of signals $x[k]$ and $y[k]$, we will introduce the concepts of power spectral density, i.e. the frequency-domain counter parts to the time-domain correlation.

B.2 Estimation of the Power Spectral Density

The auto power spectral density (PSD or auto-PSD) of a process is the Fourier transform of its autocorrelation sequence, $r_{xx}(l)$. Using the notation $\mathcal{F}\{\}$ for the discrete-time Fourier transform (DTFT), we can define the PSD of the process x as

$$s_{xx}(\omega) = \mathcal{F}\{r_{xx}(l)\} = \sum_{l=-\infty}^{\infty} r_{xx}(l)e^{j\omega l} \tag{B.12}$$

which is known as the Wiener-Khinchine relation [81]. It is also possible to define a cross power spectral density (cross-PSD or CPSD), which is the Fourier transform of the cross-correlation sequence, $r_{xy}(l)$. We formally define the CPSD as

$$s_{xy}(\omega) = \mathcal{F}\{r_{xy}(l)\} = \sum_{l=-\infty}^{\infty} r_{xy}(l)e^{j\omega l} \tag{B.13}$$

Considering the discussion of the ensemble operator vs. time averages, we will alway use a time- and frequency-discrete version of the above formulae.

The coherence function is a normalised version of the cross-power spectrum and is normalised with respect to the two auto spectra.

$$C_{xy}(ej\omega) \triangleq \frac{s_{xy}(e^{j\omega})}{\sqrt{s_{xx}(e^{j\omega})s_{yy}(e^{j\omega})}} \tag{B.14}$$

with $0 \leq C_{xy}(ej\omega) \leq 1$.

B.2.1 Instantaneous Noise Power Spectrum

It is sometimes useful to have a definition of an instantaneous power spectrum, which is segmental computed using tappering/time-windowing and usually an overlap. The PSD can be obtained by Fourier transforming the tappered signal, and carrying out the correlation entirely in the frequency domain. We denote the generalised instantaneous power spectrum (*periodogram*)[1]

$$|X(r,n)|^2 = \frac{\frac{1}{N}\left|\sum_{k=0}^{N-1} w[k]x[k+rR]\exp(-2\pi jkn/N)\right|^2}{\frac{1}{N}\sum_{k=0}^{N-1}\left|w[k]\right|^2} \tag{B.15}$$

where $w[k]$ are the tappers and $x[k+rR]$ a length-N segment with an increment of R samples, i.e. $N-R=L$, where L is the amount of overlap. The PSD estimate is denoted by a block-time index, r, and a discrete-frequency index, n.

The tappers can be any of the windows traditionally employed in signal processing, e.g. rectangular, Hann, Hamming, Barlett, Blackman, or Kaiser, to name a few. For a rectangular window, $w[k]$, the normalisation comes down to $\frac{N}{N}$, and can of course be omitted

$$|X_{\text{rect.}}(r,n)|^2 = \frac{1}{N}\left|\sum_{k=0}^{N-1} x[k+rR]\exp(-2\pi jkn/N)\right|^2 \tag{B.16}$$

In our implementations, we often employ a Hann (Hanning) window with an overlap of 50% as this does not require any normalisation.

[1] The generalised instantaneous power spectrum is sometimes denoted modified periodogram, where modified refers to the effect of tappering.

The instantaneous power spectral density is an asymptotically unbiased estimator of $s_{xx}(\omega)$, that is

$$\lim_{N\to\infty} E\left\{|X(l,k)|^2\right\} = s_{xx}(\omega) \tag{B.17}$$

the variance, however, can for large N be approximated by [46]

$$\text{var}\left\{|X(l,k)|^2\right\} = \begin{cases} s_{xx}^2(\omega) & 0 < \omega < \pi \\ 2s_{xx}^2(\omega) & \omega = 0\,, \pi \end{cases} \tag{B.18}$$

This result indicates, that the variance of the periodogram remains at a level of $s_{xx}^2(\omega)$, which is the quantity we are trying to estimate, and that the variance is independent of the length of the sequence, $x[k]$, [46].

B.2.2 PSD Variance Reduction by Averaging Multiple Periodograms

By splitting a data sequence into multiple, possibly overlapping, segments, the instantaneous PSD of each segment is averaged in order to reduce the variance. The method is known as Bartlett-Welch averaging, or just *Welch averaging*. The process reduces the estimator variance by the number of segments, K. One useful alternative formulation is the recursively averaged (modified) periodogram

$$P(r,n) = \alpha(r,n)P(r-1,n) + (1-\alpha(r,n))|Y(r,n)|^2 \tag{B.19}$$

where the parameter $0 < \alpha < 1$ determines the responsiveness of the smoothing. The larger α, the more smoothing, or the longer moving-average time constant.

Other known methods used to reducing the PSD estimator variance are Black-Tuckey, Burgs methods and Thomsons Multitaper method. Blackman-Tuckey performs frequency-domain smoothing by windowing the auto-correlation sequence before transforming it to the frequency-domain. For practical implementation, the Welch averaging method is often preferred over the Blackman-Tuckey, as Welch averagning can be formulated in a recursive manner. Burg uses LPC analysis, i.e. a parametric spectrum analysis, to reduce the variance, by assuming that the underlying model of the signal can be well approximated by a finite-order AR model. Enhanced PSD estimation using Thomson Multitaper method is not based on segmenting the signal, rather the signal is transformed into multiple orthogonal, full-length segments, which are averaged [46]. The result is variance reduction comparable to that of the Welch averaging, but without the resolution loss in frequency which stems from splitting the sequence into multiple segments.

APPENDIX C

On the Weighted Spectral Slope Measure (WSSM)

The problem of assessing speech quality is a prevalent problem in the field of speech processing. Traditionally the problem is divided into measures of noise reduction and measures of speech distortion. This stem largely from the observation model generally applied and the functionality of the speech processing. More recent, advanced methods, such as the PESQ [71], aim at measuring speech quality in one objective assessment metric. In this project, the speech distortion measure *weighted spectral slope measure* (WSSM), also known as the *Klatt measure* after its inventor [50], has been chosen. This choice is motivated by the methods emphasis on perceptual features and simplicity.

The purpose of this appendix is to quantify the WSSM measure. Whereas a measure such as the SNR is well-known, the WSSM is a less-used tool. Most engineers and researchers have a fairly common, intuitive feeling of the meaning and interpretation of SNR. For example noise reduction improvement of $0.5\,\mathrm{dB}$ is rather inaudible, while an improvement of $3\,\mathrm{dB}$ is clearly audible.

This appendix contains a summary of the computations involved in the WSSM measure followed by a figure to establish a relationship between input SNR and the WSSM measure. Both white and pink noise is applied, as well as reverberation.

The signal at hand is divided into time segments and each segment separated into subbands by critical band analysis. In Klatts original paper [50], 36 critical bands are used, however, the number of bands are less important than the fact that the analysis approximates the human perception. We have used second-order butterworth bandpass filters to approximate Bark bands. The WSSM values for each segment are averaged to arrive at a single metric for the entire signal.

For each segmental spectrum separately, a weighting function is computed. These are averaged and enter the computation of the final slope metric. The weighting function depends on the maximum subband (global maximum), the nearest peak (local maximum), and a number of empirical constants. Klatt noted, that the actual form of the weighting computation was not critical as long as the largest peak weighted more than lesser peaks, and that spectral peaks weighted more than spectral valleys [50, 74].

As discussed, engineers and researched familiar with the field of speech processing have developed an intuitive feeling of the SNR as a metric of noise reduction. By establishing a relationship between input SNR and noise type with the WSSM metric, we intend to quantify the measure and contribute towards a similar intuitive feeling of the speech distortion measure, which is extensively used throughout this report.

The standard signal of a female voice uttering: "*Good service should be rewarded with big tips*"; has been degraded with various noise types at a number of SNRs. The result is seen in Figure C.1, where the WSSM measure is plotted as a function of input SNR of an observation according to

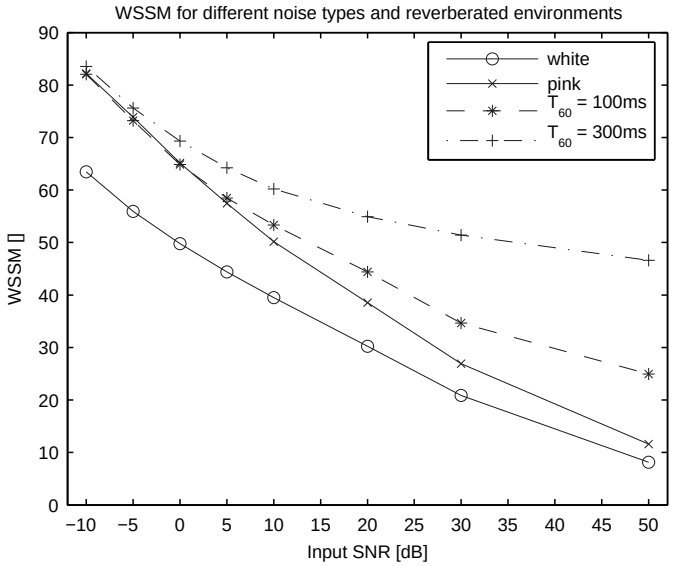


Figure C.1: WSSM metric as a function of input SNR. Both white and pink additive noise is shown. Reverberated observations added white noise are also shown. As expected, the WSSM measure is lower for higher SNRs. Notice that pink noise has a more profound effect than white noise. As expected reverberation has a profound effect on high-SNR observations.

the following single-channel observation model

$$y[k] = g * s[k] + b[k] \tag{C.1}$$

where $s[k]$ is the clean speech signal, $b[k]$, additive noise, and g, convolutional noise. The WSSM measures are computed based on the observation signal and the clean-speech signal. Values for signals with white and pink noise added, and reverberation of $100\,\mathrm{ms}$ and $300\,\mathrm{ms}$, both added white noise, are shown in the figure.

The WSSM measure is best for low values, thus an increase in WSSM corresponds to an increase in speech distortion. It can from the figure be observed, that low-SNR as expected gives rise to the most speech distortion, and that pink noise has a more profound effect on WSSM than white noise. This corresponds very well with our expectations, as the pink noise, at a given input SNR, adds considerably more noise into the lower frequency spectrum than white noise. This affects the formants, which finally results in reduced intelligibility. When the observation is degraded with white noise and furthermore convolved with an acoustic room impulse response, the WSSM values are further increased. As expected this is most notable for high-SNR observations. For observation with low SNR, the effect of the additive noise drown out the intelligibility-degrading effect of the convolutional noise. We can conclude that the WSSM measure reflects our expected perception of intelligibility; that the measure seems to take perceptual features into account; and, that the measure is sensitive to additive and convolutional noise. Figure C.1 furthermore provides insight into the range and scale of the WSSM measure.

In the previous paragraph we made observations on the WSSM measure based on computations on the un-reverberated, clean-speech signal, $s[k]$, and the observation, $y[k]$. If we see these observations in the perspective of speech enhancement functionality, the measure is useful for combined noise reduction and dereverberation. If we, however, consider the functionality of noise reduction alone, it might be more interesting to use the WSSM measure to measure the amount of speech distortion introduced, rather than overall speech distortion. Therefore, computation of WSSM could be made on a basis of the reverberated signal, $x[k] = g * s[k]$, and the observed signal, $y[k]$. By recomputing the values presented in Figure C.1 in this manner, we would expect the measure to be insensitive to convolutional noise. In Figure C.2 the WSSM measure as a function of input SNR is made based on the observed signal and the reverberated signal (without additive noise).

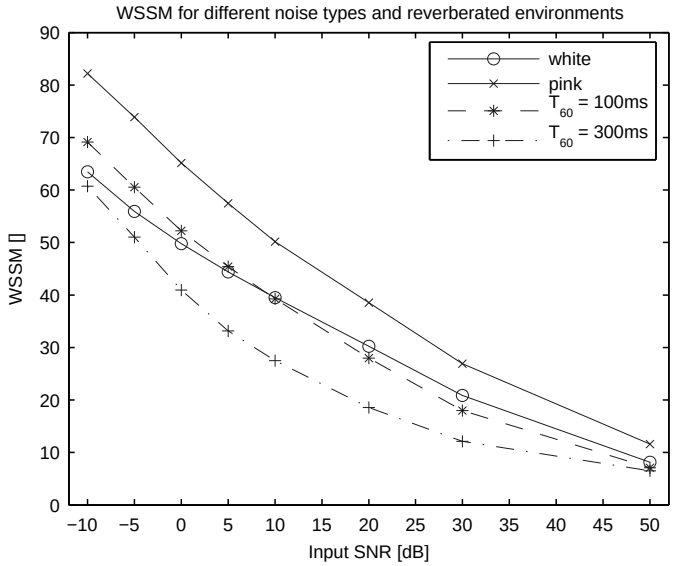


Figure C.2: WSSM metric as a function of input SNR. The measure is computed based on the reverberated, clean-speech signal, $x[k] = g * s[k]$, (without additive noise), and the observed signal, $y[k]$. Compared to Figure C.1 we surprisingly observe that reverberation has a positive effect on the WSSM measure (towards lower values).

As found in Figure C.2, it can be observed that pink noise has a more profound effect on WSSM than white noise. It might be surprising that increasing reverberation has a positive effect on the WSSM values (towards lower values) compared to the white noise case. The explanation should be found in the characteristics of the impulse response of g in (C.1). First we describe the acoustic room impulse response, then we return to the actual explanation.

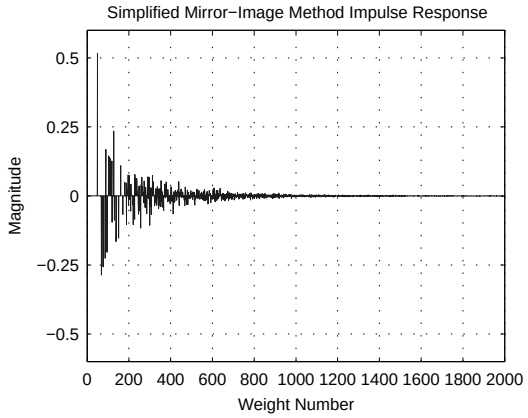


Figure C.3: Acoustic room impulse response creating using a modified mirror-image method [60] for reverberation time of $t_{60} = 300\,\mathrm{ms}$. The impulse response is characterised by a direct path and early echoes followed by far-field echoes. The latter is resemble a white noise sequence.

Acoustic room impulse responses consists of a direct path followed by what is known as early echoes. Those normally arrives within the first $200 - 400\,\mathrm{ms}$ (notice the difference between reverberation time and length of impulse response). After the early echoes, what is know as far-field echoes, arrives. These echoes somewhat resembles a white noise sequence. An example, showing these effects for a simulated acoustic room impulse response, is shown in Figure C.3. The convolution of the speech signal with the impulse response adds these different types of echoes to the observation signal, so to say.

By peeking into the segmental (non-averaged) WSSM, we find rather large values in speech pauses for the case of white noise only. In Figure C.4 the segmental WSSM for using the rever-

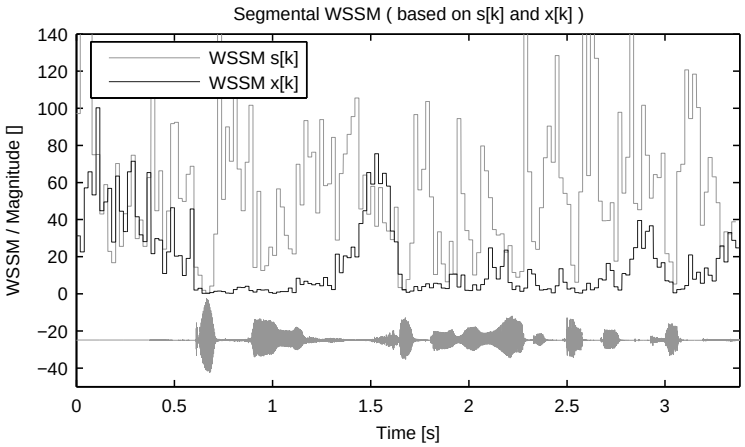


Figure C.4: Segmental WSSM for using $s[k]$ (black) or $x[k]$ (grey) in the computation of the metric. The clean speech signal is seen beneath the WSSM plot. For speech pauses, the segmental WSSM values are notably larger for of using the clean-speech signal than for using the reverberated signal, $x[k]$. This stem from the from the effect of zero noise vs. convolutional noise in speech pauses of the reference signal, $s[k]$ or $x[k]$, compared to the observation signal, $y[k]$.

berated signal, $x[k]$, or the un-reverberated signal, $s[k]$, is shown. The input SNR is $5\,\mathrm{dB}$ in both cases, and the noise type is white. It can be observed, that the large WSSM values observed for the case of using $s[k]$ (grey), are not present for the case of using $x[k]$ (black). The explanation should indeed be found in the speech pauses, by observing the two signals fed into the WSSM computation. We compute the WSSM based on two signals. One signal is the observation, $y[k]$, the other is either the clean-speech signal, or the reverberated speech signal. In in speech pauses, $s[k]$ contains zeros only, while the reverberated speech signal, $x[k]$, contains the tails of the convolution $x[k] = g * s[k]$. This latter sequence is a noise sequence which in a random fashion fits better, spectrally, to the additive noise, than the zeros do. This results in distinct peaks in the black line in the duration $2.3\,\mathrm{s}$ to $3.3\,\mathrm{s}$ in Figure C.2.

Remember the starting point was to investigate the WSSM measures ability to measure speech distortion for noise reduction functionality of speech enhancement algorithms. It was expected, that when applying the WSSM measure to $x[k]$ and $y[k]$, the measure would be insensitive to reverberation. It is observed from Figure C.2 that this is not the case. Increasing reverberation has the effect of lowering the WSSM values (which indicates less distortion). This counterintuitive results makes us confident, that using the WSSM measure on the reverberated signal instead of the clean-speech signal is not justifiable. Lowered WSSM values due to reverberation might be faulty interpreted as speech intelligibility improvement as a side-effect of the noise reduction algorithm under consideration.

An often followed path, when it is desired to circumvent abnormal behaviour of computations in noise-dominated segments, is to make use of a voice activity detector (VAD). This mechanism separates speech-dominated time segments from noise-dominated segments. However, VADs are known to perform worse in reverberated environments, with convolutional noise, than in additive-noise only environments. We do not recomend to compute the WSSM over speech-dominated time segments by employing a VAD. This view is strengthened when we compare the $x[k]$-based (black) and the $s[k]$-based (grey) WSSM over speech-dominated segments in Figure C.2. Notable differences are seen. Furthermore, using the $s[k]$-based WSSM computation as in Figure C.1 is seen to attain values expected when evaluating speech quality.

We have summarised the computations underlying the WSSM metric and presented a figure to contribute towards the readers intuitive feeling of the measure. It was shown that the WSSM values corresponded in an expected manner to the expected intelligibility of signals degraded by additive and convolutional noise. By examining the segmental WSSM it was concluded to use the observation signal, $y[k]$, and the clean-speech signal, $s[k]$, as a basis for the WSSM computations.

APPENDIX D

Code Portfolio

In this section we provide an overview of the most important MATLAB files writing during this project period. Not all files have been included, we have, however, found space to mention some C-files, too. The files are arranged corresponding to their use in context of the project. We refer back to the introduction, where our work flow is presented in Figure 1.2 on page 2. The files are divided into the following categories *data acquisition*, this being files that generate test signals, a section that contains the different *speech enhancement techniques*, and, finally, a section that contains *evaluation methods*. Of the latter most notable are the broadband SNR, segmental SNR, and WSSM. Finally we have an extra section for files in none of the above categories, *miscellaneous*. In this category a few C-files are also listed.

Data Acquisition

Function	Description
`generate_scenario.m`	Generate scenario, i.e. create reference signal(s) of certain gender, sentence, reverberation and reverberation time, and noise type at given SNR.
`signalgen.m`	Generate an observation signal (see `generate_scenario.m`).

Speech Enhancement

Function	Description
`Berouti79.m`	The spectral subtraction method proposed by Berouti et al. [6].
`SSBoll79.m`	The spectral subtraction method proposed by Boll [7].
`Ephraim95.m`	The signal subspace method proposed by Ephraim and Van Trees [24].
`jensen1995.m`	The signal subspace method proposed by Jensen et al. [49].
`gsc_with_steering.m`	The GSC beamformer with a fixed steering vector.
`gsc.m`	The GSC beamformer with a delay and sum and with a delay and subtract.
`gsc_fdaf.m`	The GSC with frequency-domain adaptive filter, see Shynk [82],in the ANC.
`gjbf.m`	The beamformer originally proposed by Griffiths and Jim.
`mcwf_time_batch.m`	Time-domain GSVD-based multi-channel Wiener filtering.
`mcwf_freq_batch.m`	Frequency-domain GSVD-based multi-channel Wiener filter.
`mcwf_freq_batch_gevd.m`	Frequency-domain GEVD-based multi-channel Wiener filter.
`mcwf_recursive_gsvd_mat.m`	The recursive MCWF.
`mcwf_time_batch_subsampling.m`	The MCWF using sub-sampling.
`spectral_addition.m`	Method for calculating the ambiguity factors used for the method proposed by Affes and Grenier [1].

Speech Enhancement (cont'd)

`svd2.m`	SVD using two-sided cyclic-Jacobi (Givens rotation).
`givens_rot.m`	c and s parameters for a Givens rotation.
`givens_row_update.m`	Row update for a Givens rotation.
`givens_col_update.m`	Column update for a Givens rotation.

Speech Quality Evaluation

Function	Description
`specgram2.m`	Modified spectrogram with limited dynamic range.
`eval_speech.m`	Evaluate a speech signal using broadband SNR, segmental SNR, and WSSM. Includes time aligning.
`snr_seg.m`	Computes the segmental SNR and/or broadband SNR.
`wslope.m`	Computes the weighted spectral slope measure (WSSM).
`sqnr.m`	Computes signal-to-quantisation noise ratio.
`bsd.m`	Bark spectral distortion.
`snr_fw_seg.m`	Computes the frequency-weighted, segmental signal-to-noise ratio.
`lpccd.m`	Computes the cepstral distortion in the LPC coefficients.
`itakura_saito.m`	Computes the Itakura-Saito spectral distortion.
`spec_deviation.m`	Computes the logarithmic spectral deviation.

Miscellaneous

Function	Description
`Martin94.m`	Noise power spectral density estimation based on the paper by Martin [58] (minimum statistics).
`martin2001`	Noise power spectral density estimation based on the paper by Martin [59] (optimal smoothing).
`herbordt2003.m`	Multi-channel power spectral density estimation based on the paper(s) by Herbordt et al. [41, 42].
`stft.m`	Segmented Fourier transform of multi-channel signals with optionally windowing.
`fdaf2.m`	Multi-dimensional frequency-domain adaptive filtering using overlap-save sectioning, see Shynk [82].
`filterzmatched.m`	Computes a matched FIR filter output in the frequency domain using the overlap-save method.
`eta.m`	Estimate "arrival" time of a MATLAB program based on a number of iterations.
`gerven1997algo3.m`	VAD based on log-energy distribution of noise, see Gerven and Xie [29].
`pwelch_ext.m`	Computes the PSD of the using Welch averaging.
`gsvd_pwelch.m`	Computes the CPSD in a manner similar to using data matrices (as in the GSVD-based MCWF).
`pdirect.m`	Computes the segmented, instantaneous PSD (using FFT for fast computation of long sequences).
`overlapadd.m`	Computes a linear convolution on a block basis using overlap-add sectioning.
`filterz.m`	Frequency-domain FIR filtering using overlap-save sectioning.
`offnorm.m`	Computes the off-diagonal norm of the matrix.
`avg_adiag.m`	Does arithmetic averaging along the anti-diagonals of a matrix.
`fxpt/mcwfgsvd.cxx`	Fixed-point recursive GSVD-based MCWF.
`flpt32/mcwfgsvd.cxx`	32 bit floating-point recursive GSVD-based MCWF.
`flpt64/mcwfgsvd.cxx`	64 bit floating-point recursive GSVD-based MCWF.