Aalborg University

Master Thesis - Electronics & IT
- Signal Processing and Computing

Distant Speech Recognition

Author:
Nicolai B. Thomsen

Supervisors:
Zheng-Hua Tan (AAU)
Søren Holdt Jensen (AAU)
John H.L. Hansen (UTD)

June 6, 2013

Department of Electronic Systems
Electronics & IT
Fredrik Bajers Vej 7 B
9220 Aalborg Ø
Phone 9940 8600
http://es.aau.dk

Title: Distant Speech Recognition

Subject: Signal Processing

Project period:
P9/P10, Fall 2012 / Spring 2013

Project group:
976 / 1076

Participant:
Nicolai B. Thomsen

Supervisors:
Zheng-Hua Tan (AAU)
Søren Holdt Jensen (AAU)
John H.L. Hansen (UTD)

Number of copies: 6

Pagenumber: 67

Attachments: 1 CD

Appendices: 6

Ended the 6/6 2013

Abstract:

This project concerns the investigations of using a microphone array to suppress reverberation and noise such that the Phone Error Rate (PER) for Automatic Speech Recognition (ASR) system is reduced, when the distance between speaker and microphone is relatively large. The general theory of array processing is presented along with the classical Generalised Sidelobe Canceller (GSC) beamforming algorithm, which uses the Mean Square Error (MSE) as optimization criteria. This algorithm is extended to adapt the filter block-wise instead of sample-wise and further adapt them using a kurtosis criteria, where it is sought to maximise the kurtosis of the output. Histograms of reverberant speech and clean speech are plotted to confirm that clean speech has a higher kurtosis and is more super-gaussian than reverberant speech. A simple cosine-modulated filter bank and Zelinski postfiltering is implemented and verified to further extend the system. The fundamental theory of Hidden Markov Model (HMM) ASR along with two popular adaptation methods, Vocal Tract Length Normalisation (VTLN) and Maximum Likelihood Linear Regression (MLLR), is stated. The beamforming algorithm is benchmarked against the classical and well-known delay-and-sum beamformer (DSB), both with and without Zelinski postfiltering. The benchmarks were done using two data sets each consisting of 610 phonemes, but where one has synthetic generated reverberation and the other is collected from a real speaker recorded in a classroom and an auditorium. The speech recognition software, Kaldi, is used the generate PER. The reults show that the DSB without postfiltering performs better than maximum kurtosis GSC in all case. The reasons for this are discussed in the end.

Institut for Elektroniske Systemer
Elektronik og IT
Fredrik Bajers Vej 7 B
9220 Aalborg Ø
Telefon 9940 8600
http://es.aau.dk

Titel:

Talegenkendelse på afstand

Tema:

Signalbehandling

Projektperiode:
P9/P10, efterår 2012/ forår 2013

Projektgruppe:
976/1076

Deltager:
Nicolai B. Thomsen

Vejleder:
Zheng-Hua Tan (AAU)
Søren Holdt Jensen (AAU)
John H.L. Hansen (UTD)

Oplagstal: 6

Sidetal: 67

Bilag: 1 CD

Appendikser: 6

Afsluttet den 6/6 2013

Synopsis:

I dette projekt undersøges en måde, hvor flere mikrofoner i et array kan bruges til at undertrykke efterklang og støj således at automatisk talegendekelsessystemer opnår bedre resultater i tilfælde, hvor afstanden mellem taler og mikrofon er relativ stor. Den fundamentale array signalbehandlingsteori er kort beskrevet sammen med udledning af den klassiske GSC array algoritme, som anvender MSE som optimeringskriterie. Denne algoritme er udvidet således, at det adaptive filter estimeres i forhold til at maksimere kurtosis af outputtet. Ydermere opdateres filteret kun blok vist. Histogrammer af ren tale og tale med efterklang er plottet, hvilket bekræfter at ren tale er mere super-gaussisk og har en højere kurtosis værdi end tale med efterklang. En simpel filter bank og Zelinski postfiltrering implementeres og verficeres gennem test. Den fundamentale teori bag HMM ASR præsenteres sammen med to metoder, hvor taleren og de akustiske omgivelser kan tilpasses til den eksisterende model. Algoritmen testes mod den velkendte DSB med og uden postfiltrering. Der anvendes to typer datasæt, hver bestående af 610 phonemer. En type datasæt, hvor efterklangen er genereret syntetisk vha. MATLAB og en type, hvor data er optaget i et klasseværelse og et auditorie. Som talegenkendelsessystem anvendes Kaldi. Resultaterne viser, at DSB uden postfiltrering opnår bedre resultater end maksimum kurtosis GSC i alle tilfælde. Årsagerne hertil diskuteres til sidst.

Preface

This report has been made by Nicolai Bæk Thomsen in the period September 2012 to June 2013 as documentation of Master Thesis in Signal Processing and Computing at the Department of Electronic Systems, Aalborg University. From ultimo January to medio April I was a visiting student at Center for Robust Speech Systems (CRSS) at UT Dallas, Texas, under the supervision of Professor Dr. John H.L. Hansen. This stay was among other spent on setting up an Automatic Speech Recognition (ASR) system and collecting real-world data. I would like to thank everybody at CRSS for making the stay a succes through fruitful debates and discussions within the field of signal processing. A special thanks to Professor Dr. Hansen for letting me visit and for helping me collect data. Thanks to Dr. Seong-Jun Hahm the CRSS for valuable help on setting up the ASR system. All code is written in MATLAB and can be found on the supplied CD. The Kaldi software used to do speech recognition is not supplied on the CD but can be found at http://kaldi.sourceforge.net/index.html.

Reading guide

Matrices are written in bold with capital letters (**A**), and vectors are just written in bold (**a**). Notation, which is not standardized, is explained at first encounter. All relevant equations are numbered. The first time acronyms are used the full word/sentence is stated, and furthermore a list of acronyms is provided. The content of the report is organised in the following way: Chapter 1 gives a soft introduction to the application of speech recognition and the motivation for improving the performance when the distance between speaker and microphone is increased. Chapter 2 states the reverberant signal model and the statistic properties of the signals involved. Chapter 3 gives an overview of array processing and derives the classic Generalised Sidelobe Canceller (GSC) and extends the algorithm using a kurtosis criteria. Chapter 4 gives a brief overview of the theory behind ASR and chapter 5 states and discuss the results achieved. Finally, chapter 6 concludes on the thesis and discusses how to proceed. Appendices are found at the back of the report.

Nicolai B. Thomsen - Aalborg 6/6, 2013

Indhold

Acronyms

AIR Acoustic Impulse Response. 3, 4

ASR Automatic Speech Recognition. iii, v, vii, 2, 3, 32, 33, 35–39, 44

AWGN Additive White Gaussian Noise. 3, 25, 26

CLT Central Limit Theorem. 19, 45

cMLLR constrained Maximum Likelihood Linear Regression. 36

DFT Discrete Fourier Transform. 28, 35

DOA Direction-of-Arrival. 14

DOI Direction-Of-Interest. 12, 31

DSB delay-and-sum beamformer. iii, v, 38, 39, 42–45

DSR Distant Speech Recognition. 1

EVD Eigenvalue Decomposition. 25

GMM Gaussian Mixture Model. 33, 35, 37

GSC Generalised Sidelobe Canceller. iii, v, vii, 1, 7, 12–14, 16, 19, 21, 26, 30, 32, 37–39, 41–45

HMM Hidden Markov Model. iii, v, 33–38, 45

iDFT Inverse Discrete Fourier Transform. 35

MFCC Mel-Frequency Cepstrum Coefficient. 35–37

MLLR Maximum Likelihood Linear Regression. iii, 37, 39, 42, 45

MSE Mean Square Error. iii, v, 18, 27, 32

NLMS Normalised Least-Mean-Square. 9

PDF Probability Density Function. 19, 34, 54

PER Phone Error Rate. iii, 1, 3, 33, 38, 39, 41, 43, 45

PSD Power Spectral Density. 27–29

RIR Room Impulse Response. 38

SNIR signal-to-noise plus interference ratio. 29

SNR signal-to-noise ratio. 29, 30, 32, 39

SOI Signal-Of-Interest. 12–14

ULA Uniform Linear Array. 5, 14, 32, 44

VTLN Vocal Tract Length Normalisation. iii, 37, 39, 42, 45

WER Word Error Rate. 3

WSS Wide Sense Stationary. 26, 27

Kapitel 1

Introduction

It is becoming more and more popular for people to use some kind of computer/device (smartphone, tablet, PC etc.) on a daily basis. The interaction is primarily done using some kind of touch input, which is not very practical since it ties the user's hands to the device or perhaps the user is not able to use his/her hands. A typical scenario of the first case could be when driving a car, in which case the user has to use his/her hands to operate the steering wheel and the gear stick [1]. An example of the second case is disabled people who simply cannot operate their hands at the required level of precision. In such cases it is desirable to be able to interact with the device without the use of hands or physical contact with the device. One method which is becoming more and more popular is the use of voice and speech, where the device is able to understand simple commands or whole sentences. Under ideal situations where the user is close to the microphone talking directly into it in a low-noise environment, performance is acceptable. This can be achieved by using a user-mounted microphone, but at the price of inconvenience, which is acceptable in some applications and situations, but as an example thi is not acceptable in multi-user settings. When the distance between the user and device/microphone is increased (Distant Speech Recognition (DSR)), the performance is seriously degraded due to background noise and echo or reverberation [1]. These problems have to be overcome in order for speech interaction between human and computer to become popular and effective, thus a lot of research has been done within the field of DSR. One particular and interesting method of combating these problems is through the use of multiple microphones, also known as microphone array processing or beamforming. This introduces the possibility to direct the gain towards the user and thereby supressing other sources. The scope of this thesis is to investigate one recent proposed method [2] and evaluate it in terms PER. The outline is as follows: first the problem is described along with a signal model, next a brief overview of basic array processing theory is given along with the derivation and implementation of a classic beamformer called GSC. After this the algorithm is extended according to [2] and evaluated in terms of recognition performance. At last a conclusion on the results is made.

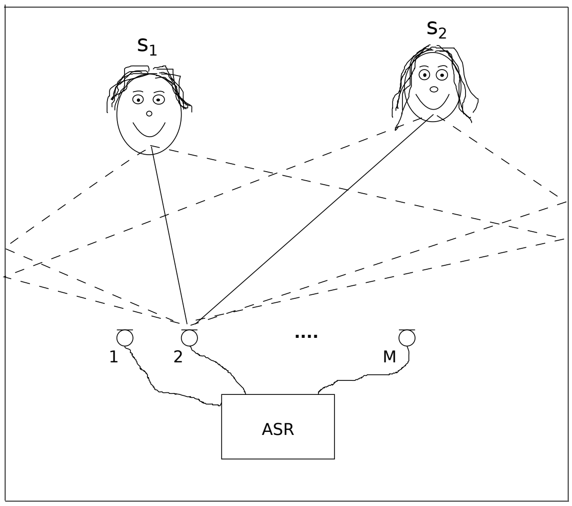
Kapitel 2

Problem Description

The aim of this section is to describe the phenomenon of reverberation and why this poses a problem. Based on this a reverberant signal model will be given and mainly the statistical properties of these signals will be stated. This will set the stage for all further investigation in this report. This section will also explain how the enhancement/dereverberation is assessed in this report, since there are many different ways of measuring this.

2.1 Signal model in acoustic environment

Figure 2.1 shows a simplified version of a ASR system using a linear microphone array with M elements to aquire speech in a reverberant environment, where two sources, s_1 and s_2, are present. We see that the speech from both sources has a direct path to microphone 2 (solid line) and some delayed versions due to reflections on the walls (dashed lines), the latter is called reverberation. Only two reflection for each source is shown due to simplicity, but in reality the number is much greater. The same will off course be the case for all the microphones but for simplicity only the signals going to microphone 2 are indicated. The level or severity is typically described by the reverberation time or $T60$, which describes how long it takes the energy of the reverberation (not included the energy of the direct path) to get below 60dB [3, p. 6]. For low reverberation times, reverberation will not pose as severe a problem to human listeners, but in the case where the speech is picked up by an ASR this has a great influence on the performance of the system and will certainly degrade this [1, p. 8].



Figur 2.1: *Figure showing the situation of doing ASR in a reverberant environment using a linear microphone array. There are two sources, s_1 and s_2 and M microphones connected to an ASR system. Solid lines indicate LoS and dashed lines indicate reflection on walls (reverberation).*

We are now able to state a signal model for the signal received at the mth microphone [4, p. 68]

$$y_m(n) = \sum_{k=1}^{K} g_{m,k}(n) * s_k(n) + v_m(n) \tag{2.1}$$

where:
$y_m(n)$ is the output signal from the mth microphone at time index n
$g_{m,k}(n)$ is the acoustic impulse response between the kth source and the mth microphone at time index n
$s_k(n)$ is the clean signal from the kth source at time index n
$v_m(n)$ is additive white noise at the mth microphone
K is the number of sources

Normally one is interested in only one of the sources and consider this as the signal of interest and then regard all other sources as interference, but for convenience this is not explicitly stated in the signal model here. To get a better understanding of what is going on in equation 2.1 we will list the known and assumed properties of the signals.

Source signals, $s_k(n)$
These are the unknown clean speech signals from the sources, and therefore broadband signals. Each speech signal is assumed to be a non-stationary and zero-mean stochastic process. We further have that the source signals are uncorrellated, e.g. $\mathbb{E}[s_{k1}(n1)s_{k2}(n2)] = 0$ for $k1, k2 = 1{,}2{,}...K$, $k1 \neq k2$ and for all $n1$ and $n2$.

Acoustic Impulse Response, $g_{m,k}(n)$
These are unknown and time-variant. Because the reverberation time is between 0.1s and 1s for normally sized rooms, the length of the Acoustic Impulse Response (AIR)'s is in the order of thousands [3, p. 8].

Additive noise, $v_m(n)$
We assume that the noise is Additive White Gaussian Noise (AWGN) both temporally and spatially (across microphones), e.g. $\mathbb{E}[v_m(n1)v_m(n2)] = 0$ for all $n1$,$n2$ and $n1 \neq n2$ and $\mathbb{E}[v_{m1}(n)v_{m2}(n)] = 0$ for $m1$,$m2 = 1{,}2{,}...M$, $m1 \neq m2$ and for all n.

Microphone signals, $y_m(n)$
We will assume that all microphone signals are zero-mean. Because every microphone will receive signals from all sources (with different delays) the microphone signals are correlated with each other, e.g. $\mathbb{E}[y_{m1}(n1)y_{m2}(n2)] \neq 0$ for all $m1, m2 = 1{,}2{,}...M$ and for all $n1$ and $n2$.

2.2 Objective of speech enhancement

As mentioned earlier there are mainly two reasons to do speech enhancement, where the first is the case when a human listener is perceiving the signal, and the second case is when enhancement is needed in order for an ASR to achieve satisfying performance in terms of Word Error Rate (WER) or PER. This thesis will focus on the last objective.

2.2.1 Suppression vs. Cancellation

Many different methods have been employed trying to eliminate the reverberation of speech and thereby achieve optimum performance of an ASR. All these methods can roughly be divided into two main categories as done in [5]. Here the methods are divided in *reverberation cancellation* and *reverberation suppression*. The basic idea of the two categories and the differences is now explained.

Cancellation

When trying to cancel out the reverberation effect one aims at estimating the true AIR's and then perform an inverse filtering or deconvolution. This is also refered to as *blind deconvolution* due to the fact that the AIR's are estimated blindly. In theory this will yield a perfect reconstruction of the true speech signal [4, p. 152], $s_k(n)$, but the method has some drawbacks. In order for this method to be useful first of all the AIR's must be estimated. Since the lengths of these are typically in the order of hundreds or thousands these can be very difficult to estimate in practice. Also the AIR's cannot share any common zeros when looking at these in the z-domain as this will result in a rank-deficient filter matrix, thus making it non-invertible [4, p. 152].

Suppression

These methods primarily relies on optimum filtering by exploiting the statistical properties of the desired speech source. One example of a suppression method is fixed/adaptive beamforming, where knowledge of the direction of the desired signal is used to suppress signals impinging from other direction. These types of method are generally more robust then cancellation methods because nothing needs to be estimated, but as a consequence the potential is not as great [5, p. 74].

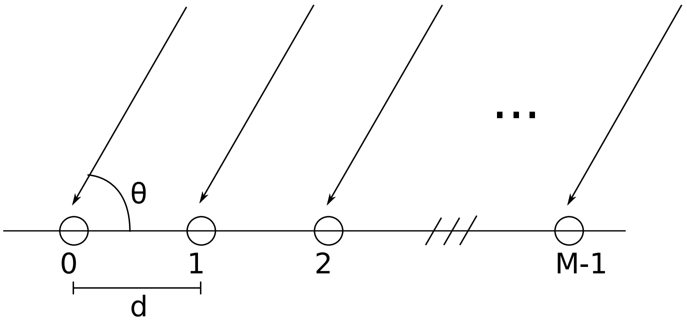
In this thesis focus will be on suppression methods using multiple microphones.

Kapitel 3

Array Signal Processing

3.1 Array response and signal model

This section will define the signal model for a Uniform Linear Array (ULA), which is used throughout the report. Figure 3.1 shows a linear array of M microphones, where linear referes to the microphones being equally spaced by the distance d. We also make the assumption that the source of the signal is located in the far-field, such that the incident wave is plane [6, p. 117].



Figur 3.1: *Linear array of M microphones and an impinging signal from the direction of angle given by Θ.*

First we define the response of the microphone array at the direction, θ by

$$\mathbf{a}(\theta) = [g_0(\theta)\ g_1(\theta)e^{(-j2\pi\cos(\theta)\frac{d}{\lambda})}\ g_2(\theta)e^{(-j2\pi\cos(\theta)\frac{2d}{\lambda})}\ ...\ g_{M-1}(\theta)e^{(-j2\pi\cos(\theta)\frac{(M-1)d}{\lambda})}]^T \tag{3.1}$$

where:
θ is angle
d is the spacing between microphones
$\lambda = \frac{c}{f}$ is the wavelength
$g_m(\theta)$ denotes the directivity pattern for the mth microphone

In array processing equation 3.1 is called the steering vector. It is important to note, that the response is dependent on the spacing of the microphones, d, and the frequency of the signal, f. For now we assume isotropic microphones, thus we have $g_m(\theta) = 1$ for $m = 0,1,...,M-1$ and $\theta \in [0; 2\pi[$. We thus get [7]

$$\mathbf{a}(\theta) = [1\ e^{(-j2\pi\cos(\theta)\frac{d}{\lambda})}\ e^{(-j2\pi\cos(\theta)\frac{2d}{\lambda})}\ ...\ e^{(-j2\pi\cos(\theta)\frac{(M-1)d}{\lambda})}]^T \tag{3.2}$$

For a single wave, $s(t)$, impinging from the constant direction θ and without noise we have

$$\mathbf{x}(t) = \mathbf{a}(\theta)s(t) \tag{3.3}$$

We see from equation 3.3 that the signals are continuous in time. After sampling is done we get the following discrete-time signal model

$$\mathbf{x}(n) = \mathbf{a}(\theta)s(n) \tag{3.4}$$

where:
n is the sample index

We are now able to define the discrete-time output of the array when K waves are impinging and additive noise is present [7]

$$\mathbf{x}(n) = \mathbf{A}(\theta)\mathbf{s}(n) + \mathbf{v}(n) \tag{3.5}$$

where:
$\mathbf{A}(\theta) \in \mathbb{C}^{M\times K}$ is a matrix, whose columns are the steering vectors corresponding to the impinging signals
$\mathbf{s}(n) \in \mathbb{R}^{K\times 1}$ is a vector containing the K signals at time n
$\mathbf{v}(n) \sim \mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})$ is additive noise

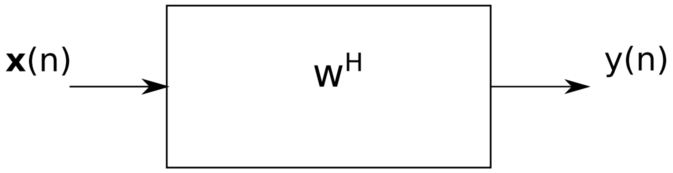
A very important observation is that when no noise is present $\mathbf{x}(n)$ is contained in the K-dimensional subspace of the M-dimensional signal-subspace, assuming that $K < M$ [7].

3.2 Generalised Sidelobe Canceller (GSC)

This section will explain and derive a classical adaptive beamformer called the Generalised Sidelobe Canceller. We start by defining the signal model and scenario. Afterwards the solution is derived and a practical implementation based on this is explained. At last some simulations are conducted by implementing the beamformer in Matlab.

3.2.1 Problem description

The problem at hand is illustrated by the block diagram in figure 3.2. Given the input $\mathbf{x}(n)$, which is a response of a uniform linear array as described in section 3.1, we are interested in finding a filter or a vector $\mathbf{w}$ such that the output obeys some constraints. In other words we are seeking a spatial filter with certain properties according to the direction.



Figur 3.2: *Block diagram showing the input, output and the optimum filter.*

The input signal $\mathbf{x}(n)$ consists of the desired signal, interfering signals and some additive noise at each microphone by

$$\mathbf{x}(n) = \underbrace{\mathbf{a}(\theta_u)u(n)}_{\text{desired}} + \sum_{k=1}^{K} \underbrace{\mathbf{a}(\phi_k)d_k(n)}_{\text{interference}} + \underbrace{\mathbf{v}(n)}_{\text{noise}} \tag{3.6}$$

where:
$\mathbf{a}(\theta)$ is a steering vector, see equation 3.2
$u(n)$ is the desired signal
θ_u is the direction of the desired signal
K is the number of interfering signals

$d_k(n)$ is the kth interfering signal
ϕ_k is the direction of the kth interfering signal signal
$\mathbf{v}(n) \sim \mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})$

3.2.2 Derivation

The GSC is an implementation of a the Linear Constrained Minimum-Variance (LCMV) beamformer [6, p. 120]. Some assumptions are neccessary in order for the GSC to be valid

- The direction of the desired signal is known and does not change over time
- The desired signal is narrowband

The problem of finding the LCMV optimum filter can be stated as an optimization problem, where it is sought to find the filter coefficients $\mathbf{w}$, which yields a minimum output power and at the same time obey some linear constraints.

$$\begin{aligned}&\min \mathbb{E}[|y(n)|^2] = \mathbb{E}[y(n)y(n)^*] = \mathbb{E}[\mathbf{w}^H\mathbf{x}(n)(\mathbf{w}^H\mathbf{x}(n))^*] = \mathbf{w}^H\mathbf{R}_{xx}\mathbf{w}\\ &\text{subject to } \mathbf{C}^H\mathbf{w} = \mathbf{g}\end{aligned} \tag{3.7}$$

where:
$\mathbb{E}$ is the expectation operator
$\mathbf{R}_{xx}$ is the correlation matrix of the input $\mathbf{x}(n)$
$\mathbf{C}$ is a constraint matrix

The solution to equation 3.7 is found by using the method of Lagrange multipliers and is given by

$$\mathbf{w}_o = \mathbf{R}_{xx}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}_{xx}^{-1}\mathbf{C})^{-1}\mathbf{g} \tag{3.8}$$

The full derivation of the solution is given in appendix A. There are many ways of constraining the problem and thereby choosing $\mathbf{C}$ and $\mathbf{g}$ [8, p. 514-525]. We see from equation 3.8 that the solution requires that the covariance matrix of the input signal is known in beforehand. This is not the case in real-world problems, thus we need to do something else. The next subsection will explain how using the covariance matrix is avoided.

3.2.3 Implementation

The idea behind the GSC is to divide the M-dimensional signal space into a subspace given by the constraints and a subspace which is orthogonal to the constraint subspace [8]. We assume the constraints to be linearly independent and that the number of constraints is lower than the number of microphones, $L < M$. The constraint subspace therefor has the dimension L and the dimension of the orthogonal space is $M - L$. The range of the constraint subspace is thus given by the span of the columns of $\mathbf{C}$ and we define the matrix $\mathbf{B}$, which column space span the orthogonal space. In the literature the matrix $\mathbf{B}$ is called the blocking matrix, so we adopt this. The orthogonality requirement can be stated as

$$\mathbf{C}^H\mathbf{B} = \mathbf{0} \tag{3.9}$$

where:
$\mathbf{0}$ is a matrix of zeros

We see from 3.9 that the column of $\mathbf{B}$ span the null space of $\mathbf{C}^H$. The optimum filter is split into a contribution from the constraint subspace and a contribution from the orthogonal subspace [8]

$$\mathbf{w}_o = \mathbf{w}_q - \mathbf{w}_p \tag{3.10}$$

where:
$\mathbf{w}_q$ is the part from the constraint subspace
$\mathbf{w}_p$ is the part from the orthogonal subspace

$\mathbf{w}_q$ and $\mathbf{w}_p$ are found by projecting $\mathbf{w}_o$ onto $\mathbf{C}$ and $\mathbf{B}$, respectively. The projection matrix onto the constraint space is given by

$$\mathbf{P}_C = \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H \tag{3.11}$$

We can now find an expression for $\mathbf{w}_q$

$$\begin{aligned}\mathbf{w}_q &= \mathbf{P}_C\mathbf{w}_o && (3.12)\\ &= \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{g} && (3.13)\\ &= \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{g} && (3.14)\end{aligned}$$

An important thing to notice here is that $\mathbf{w}_q$ does not depend on the statistics of the input signal, but only the constraints. Another important thing is in the case where we constrain to have unit gain in the desired direction, θ_u, we thus have the following constraint

$$\mathbf{C}^H\mathbf{w} = \mathbf{a}(\theta_u)^H\mathbf{w} = 1 \tag{3.15}$$

This is a special case of the LCMV and is called Minimum-Variance Distortionless Response (MVDR) beamformer [6, p. 119]. We note that the single linear constraint in equation 3.15 is equal to the steering vector in equation 3.2, e.g $\mathbf{C} = \mathbf{a}(\theta_u)$. By replacing the constraint matrix, $\mathbf{C}$, in the last expression in equation 3.14 with the single constraint from equation 3.15 and using the fact that $\mathbf{C} = \mathbf{a}(\theta_u)$, we get

$$\mathbf{w}_q = \mathbf{a}(\theta_u)\left(\mathbf{a}(\theta_u)^H\mathbf{a}(\theta_u)\right)^{-1}1 = \frac{\mathbf{a}(\theta_u)}{||\mathbf{a}(\theta_u)||_2^2} \tag{3.16}$$

where:
$||\cdot||_2$ denotes the euclidian norm.

From comparing equation 3.16 with equation 3.4 we see that $\mathbf{w}_q$ turns out to be a matched filter to the desired signal.

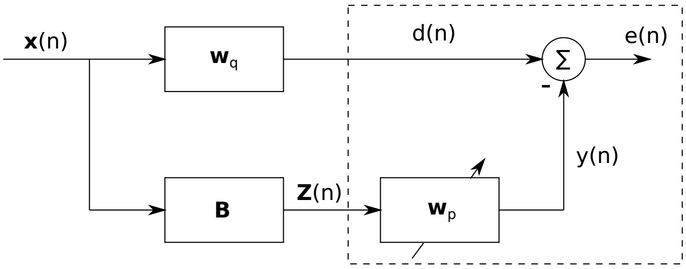
Equation 3.11 can also be used to create a matrix $\mathbf{B}$, which comply with equation 3.9, in the following way

$$\mathbf{B} = \mathbf{I} - \mathbf{P}_C \tag{3.17}$$

We now take the first $M - L$ columns of $\mathbf{B}$ [8, p. 532].

It is now possible to find $\mathbf{w}_p$ in the same way as $\mathbf{w}_q$ was found. This is however not satisfying and a better solution exists. We can reformulate the problem into an optimum filtering problem. This is illustrated in figure 3.3.

Figure 3.3 shows how the input signal is split into an upper and lower path. The upper path makes sure that unit gain is achieved in the desired direction, and the lower path takes care of interference. The lower path is thus implemented as an adaptive filter, since the interference and noise is not known before hand. In this way the filter can adapt to changing environments. To ensure that the lower path do not conflict with the upper path, the input to the lower path is first projected on to the orthogonal space of the constraint space by multiplying with the blocking matrix, $\mathbf{B}$, hence the name.



Figur 3.3: *Block diagram of the GSC. The dashed line frames the part, which can be considered as an optimum filter [6, p. 123].*

3.2.4 Simulation

A MATLAB implementation of the GSC has been made, where the adaptive filter in the lower path on figure 3.3 is a Normalised Least-Mean-Square (NLMS) adaptive filter [6, p. 320-324]. The equation for updating the filter weight is given by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta}{\epsilon + ||\mathbf{z}(n)||_2^2}\mathbf{z}(n)e^*(n) \tag{3.18}$$

where:
β is the step-size. Should obey $0 < \beta \leq 2$
ϵ is a small positive constant to ensure numerical stability when $||\mathbf{z}(n)||_2$ is small

It is not the scope of this report to investigate the theory behind adaptive filtering. Three scenarios are chosen to illustrate the effect of the GSC. To keep focus on its ability to suppress interference and not noise, the simulations were run without adding noise. We construct the signal using a narrowband signal-of-interest and narrowband interference. The signal received by microphone m is described by

$$x_m(n) = \underbrace{A\cos(2\pi Fn)}_{u(n)} \cdot e^{-j2\pi m\frac{\cos(\theta)}{\lambda_u}} + \sum_{k=1}^{K} \underbrace{B_k\cos(2\pi f_k n + \psi_k)}_{s_k(n)} \cdot e^{-j2\pi m\frac{\cos(\phi_k)}{\lambda_k}} \tag{3.19}$$

where:
A is the amplitude of the desired signal
F is the frequency of the desired signal
θ is the direction of arrival of the desired signal
K is the number of interfering signals
B_k is the amplitude of the kth interfering signal
f_k is the frequency of the kth interfering signal
ψ_k is the phase of the kth interfering signal
ϕ_k is the direction of the kth interfering signal

In both simulation we use the MVDR beamformer given by equation 3.15.

Simulation 1 - Single interfering source

Table 3.1 shows the settings for this simulation, where only one interfering source is present.

Figure 3.4 shows how the mean-squared error (MSE) develops over time in frames of 128 samples for $e(n)$, $d(n)$ and in the case of the raw input from a single microphone $x(n)$. The MSE for the error signal is estimated by

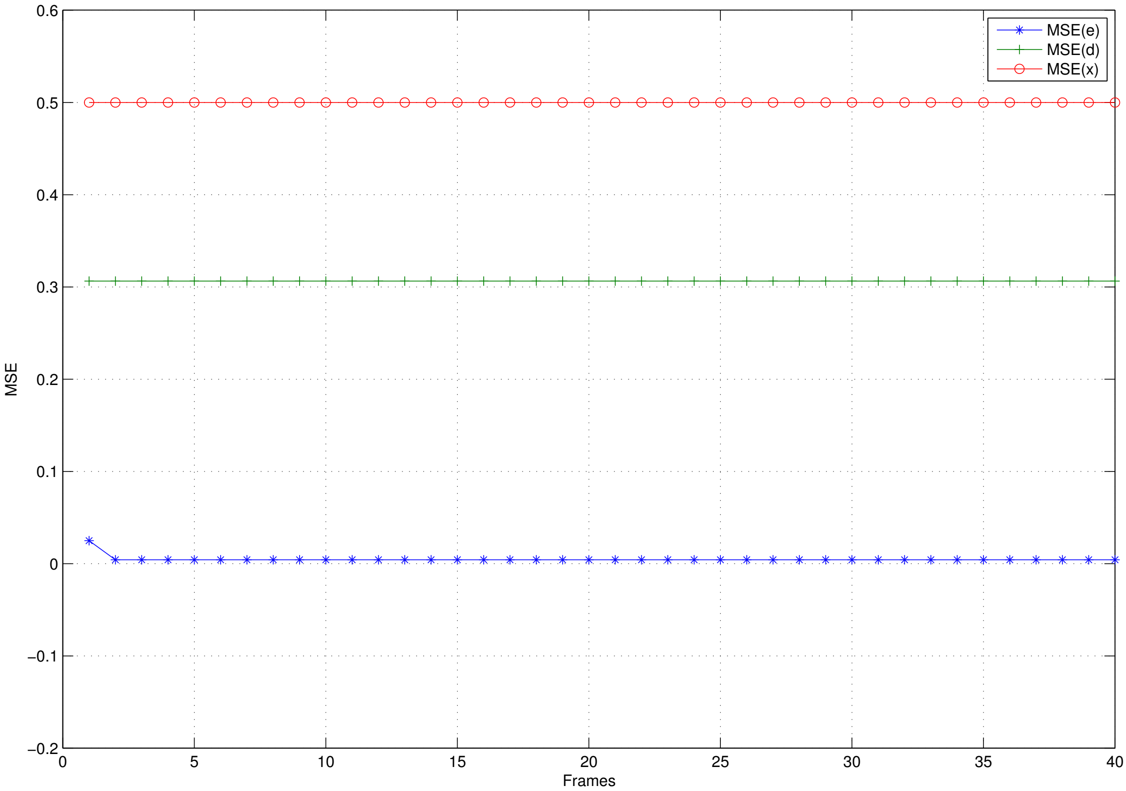
$$\text{MSE}(\mathbf{e}) = \frac{1}{N}\sum_{k=1}^{N}(u(k) - e(k))^2 \tag{3.20}$$

Parameter	Value(s)
ϵ	0.1
β	0.1
d	$\frac{\lambda}{2} = 5.7$ m
M	4
A	1
F	30 Hz
θ	80°
K	1
B	1
f	5 Hz
ψ	0 rad
ϕ	70 °

Tabel 3.1: *Parameter values for simulation 1.*

where:
$N = 128$
e is the error signal
u is the true signal of interest
k denotes the kth sample of the block

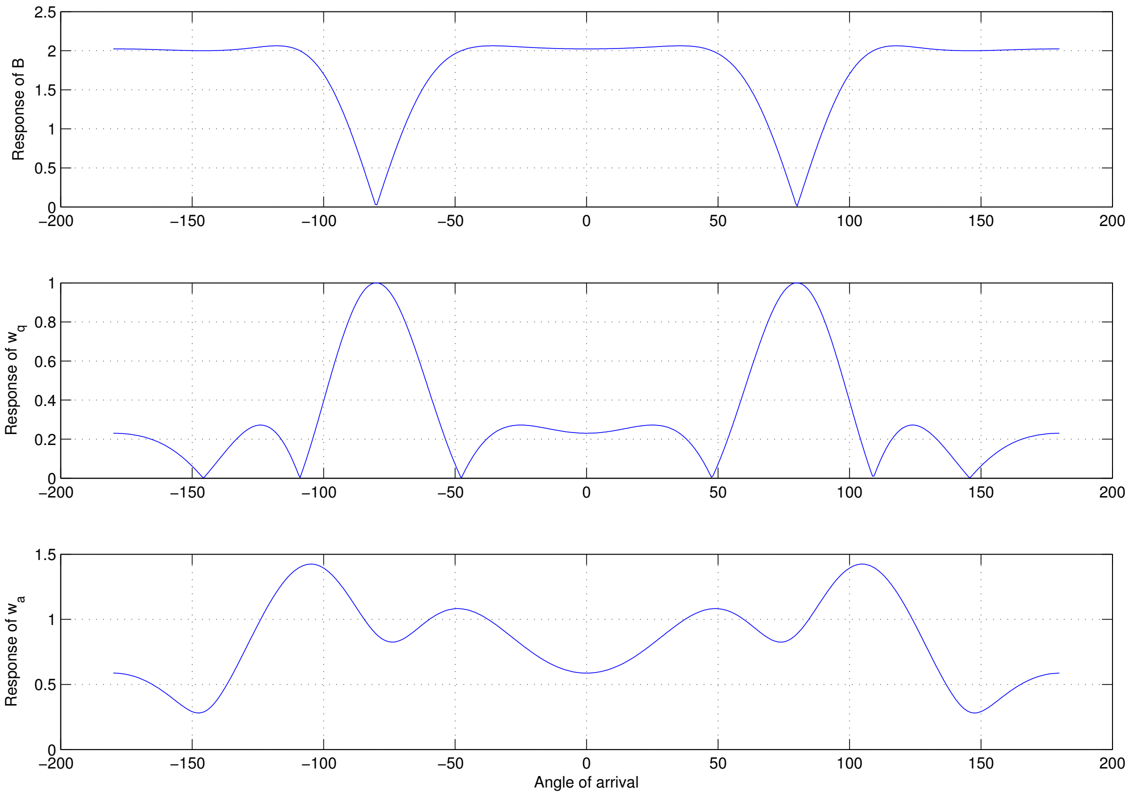
The MSE for $x(n)$ and $d(n)$ is calculated in the same way by replacing $e(k)$ in equation 3.20.



Figur 3.4: *Simulation 1: Plot of how the MSE develops over time.*

It is seen from 3.4 that the MSE for the error signal converges to approximately 0. This is compared to the case when only a single microphone is used and no enhancement is done, where the MSE oscillates around approximately 0.5. The last case is when only the matched filter, $\mathbf{w}_q$, is used. In this case the MSE is 0.3 and we see that we get an improvement compared to the single microphone case, but still not as good as the whole GSC. The GSC clearly outperforms the matched filter in this case, because the interfering signal has an impinging angle close to the desired signal together with the fact that the beam of matched filter improves proportionally with the number of microphones.

Figure 3.5 shows the response of the blocking matrix (top), the matched filter, $\mathbf{w}_q$ (middle) and the adaptive filter, $\mathbf{w}_p$ (bottom). As mentioned in section 3.1 the response is dependent on frequency. In the following plots the responses are measured at the frequency of the desired signal, thus the response of the adaptive filter $\mathbf{w}_p$ cannot be used directly to determine from which directions interfering signal are coming, unless the frequency of these are close to the frequency of the desired signal. We first note that the blocking matrix can be interpreted as a filter-bank, where each column acts as a band-rejection filter [6, p. 126]. We clearly see that the blocking matrix has 0 gain at the desired angle whereas the matched filter has unit gain, which was also expected. Due to the limited number of microphones the matched filter has a very slow varying response. This is due to that fact that $\mathbf{w}_q$ only contains $M - L$ coefficients, where L is the number of constraints. In this case $\mathbf{w}_q$ contains 3 coefficients which does not yield a very good fit.



Figur 3.5: *Simulation 1: Plot of the response of the blocking matrix,* $\mathbf{B}$*, (top), matched filter,* $\mathbf{w}_q$*, (middle) and adaptive filter,* $\mathbf{w}_p$ *at the last iteration (bottom).*

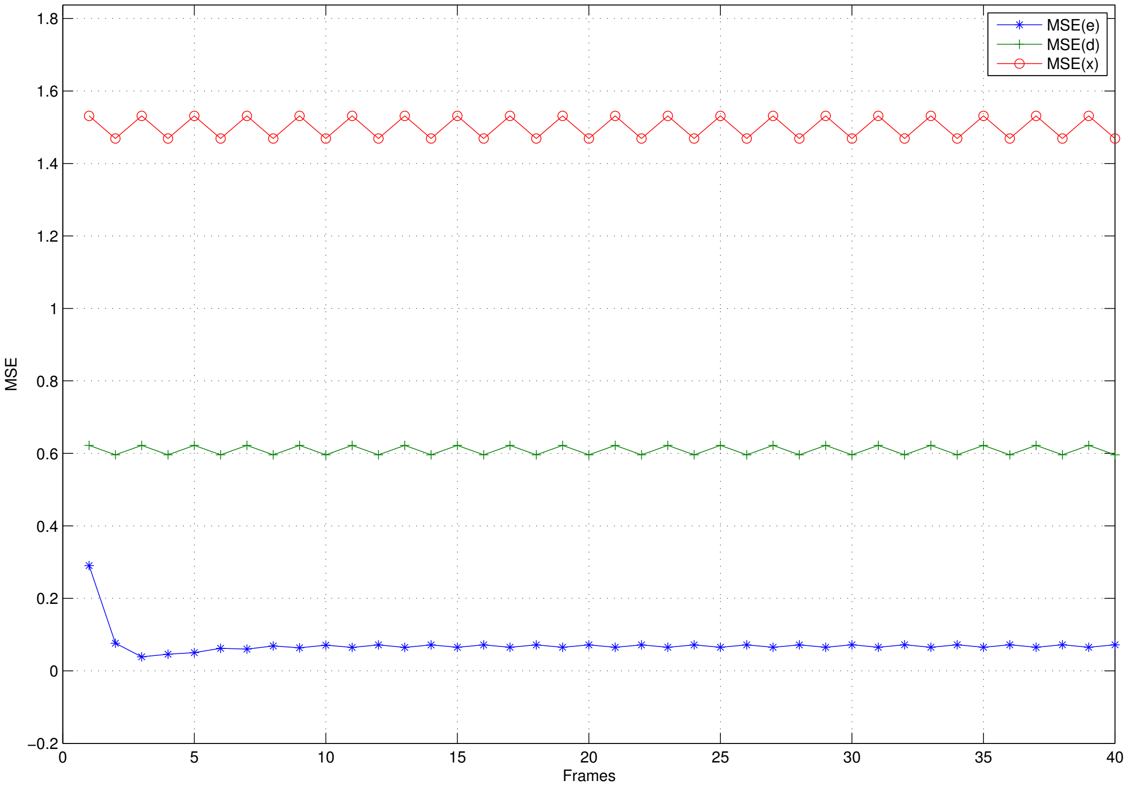
Simulation 2 - Multiple interfering sources

Table 3.2 shows the settings for this simulation.

Similar to simulation 1 the MSE has been calculated in frames of 128 samples and the result is seen on figure 3.6. We again see that there is a great improvement when using the GSC compared to the single-microphone case (red).

Parameter	Value(s)
ϵ	0.1
β	0.1
d	$\frac{\lambda}{2} = 5.7$ m
M	4
A	1
F	30 Hz
θ	80°
K	3
B	[1,1,1]
f	[5, 10, 15] Hz
ψ	[0, 0, 0] rad
ϕ	[78°, 82°, 40°]

Tabel 3.2: *Parameter values for simulation 2.*

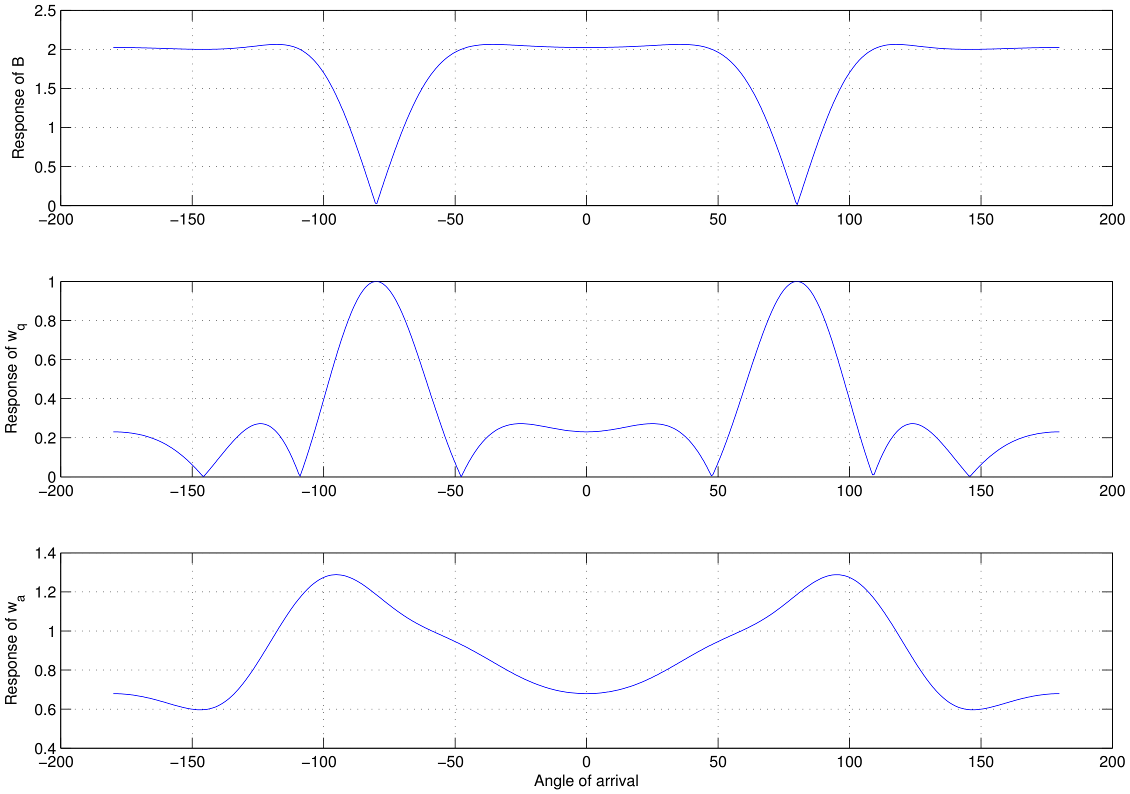


Figur 3.6: *Simulation 2: Plot of how the MSE develops over time.*

Figure 3.7 shows the response of the blocking matrix (top), the matched filter, $\mathbf{w}_q$ (middle) and the adaptive filter, $\mathbf{w}_p$ (bottom). We again see that the blocking matrix and the matched filters are orthogonal to each other.

Simulation 3 - Correlated interference

As stated in section 2 the Signal-Of-Interest (SOI) is reflected on walls and other objects, which will result in delayed and phase-shifted versions of SOI impinging from different angles other than the Direction-Of-Interest (DOI). This corresponds to $u(n)$ and $s_k(n)$ for $k = 1,2,...K$ being correlated in equation 3.19. To see how the GSC handles correlated noise, the same settings as in simulation



Figur 3.7: *Simulation 2: Plot of the response of the blocking matrix,* $\mathbf{B}$*, (top), matched filter,* $\mathbf{w}_q$*, (middle) and adaptive filter,* $\mathbf{w}_p$ *at the last iteration.*

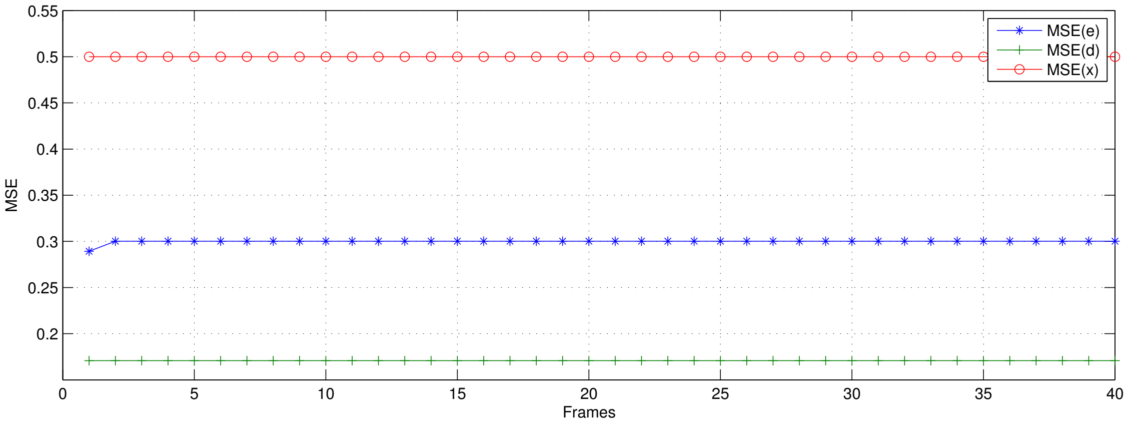
1 is chosen except for the phase and frequency of the interfering signal. The simulation is done by averaging over 100 different realisations each with different phase of the interfering signal.

Table 3.3 shows the settings for this simulation.

Parameter	Value(s)
ϵ	0.3
β	0.1
d	$\frac{\lambda}{2} = 5.7$ m
M	4
A	1
F	30 Hz
θ	80°
K	1
B	1
f	30 Hz
ϕ	70°

Tabel 3.3: *Parameter values for simulation 3.*

Figure 3.8 shows the same types of plot as for the first simulation. We clearly see, that the GSC performs very poor when the interference is correlated with the SOI. This phenomenon is called signal cancellation [9]. In this case the matched filter performs better. Because of this the GSC is not suitable for dereverberation, where the interfering signals can be considered to be delayed and phase-shifted versions of the SOI.



Figur 3.8: *Simulation 3: Plot of how the MSE develops over time. The plot has been made by averaging over 100 simulation with random phase of the interfering signal.*

3.2.5 Summary

In this section we have derived and investigated a simple narrowband beamformer called the Generalised Sidelobe Canceller. A MATLAB implementation has been made and simulations have showed its ability to attenuate interfering signals coming from different directions. We have seen that the GSC is able to filter out the interfering signals when these are not correlated with the SOI. In the case of correlated interfering signals the GSC is unable to suppress the interfering signals and thus performs poorly. Another significant drawback of the GSC is that it is intended for narrowband signal and not broadband signals which is the case when we are dealing with speech signals.

3.3 Maximum Kurtosis Subband GSC

This section will describe an improved version of the standard GSC, which was described in 3.2. The improved version is described and tested in [10, 2], where it achieves good performance. It is however important to note that the ULA consists of 64 microphones with a spacing of 2 cm, which results in a large aperture and a very narrow beam in the desired direction.

The subband structure and the improved GSC are shown in figure 3.9. In the subband structure on figure 3.9(a) there is also a block for estimating the Direction-of-Arrival (DOA), however this is only shown for a conceptual purpose and will not be implemented or described.

The four improvements are

Subband structure Compensates for the array response being frequency dependent.

Maximising block kurtosis Avoids the signal cancellation problem.

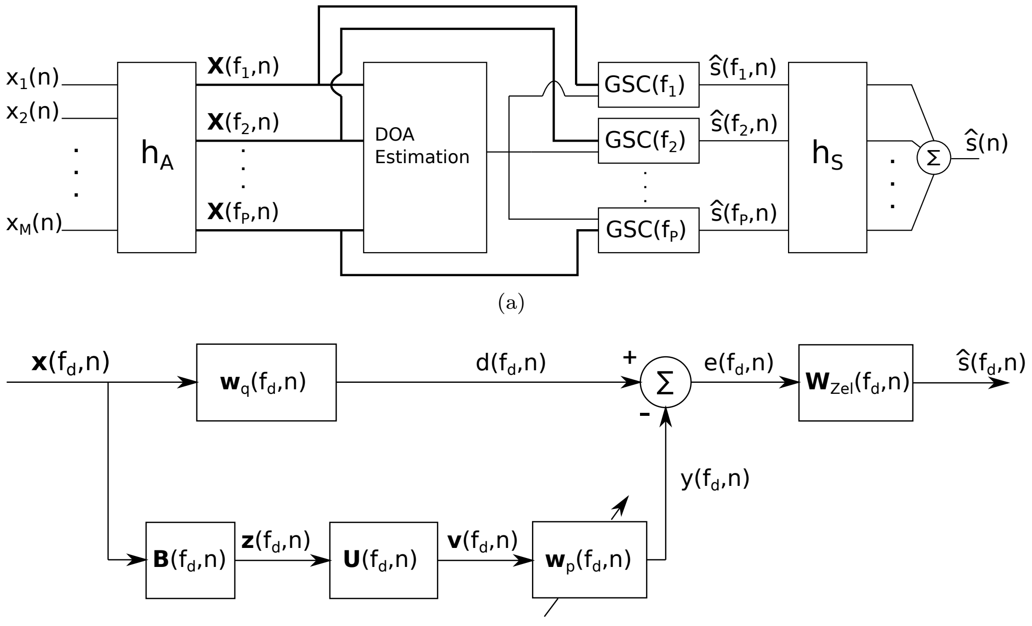
Subspace filtering Makes the kurtosis estimate more robust.

Postfiltering Noise reduction on the output from the beamformer.

The motivation for making these improvements and further details are described in the following sections, where each improvement is described, implemented and verified.

To get an overview of when things are updated and calculated, pseudo code of the improved GSC [2] is stated in algortihm 1.

It is important to note here, that some elements are updated for every input snapshot sample, while other elements are only updated for every block of input snapshot samples.



(a)

(b)

Figur 3.9: *Structure for the improved GSC. (a) the subband structure, (b) the GSC for the dth subband including postfilter.* $\mathbf{h}_A$ *and* $\mathbf{h}_S$ *are the analysis and synthesis filter banks respectively and* $\mathbf{w}_{Zel}$ *denotes the Zelinski postfilter.*

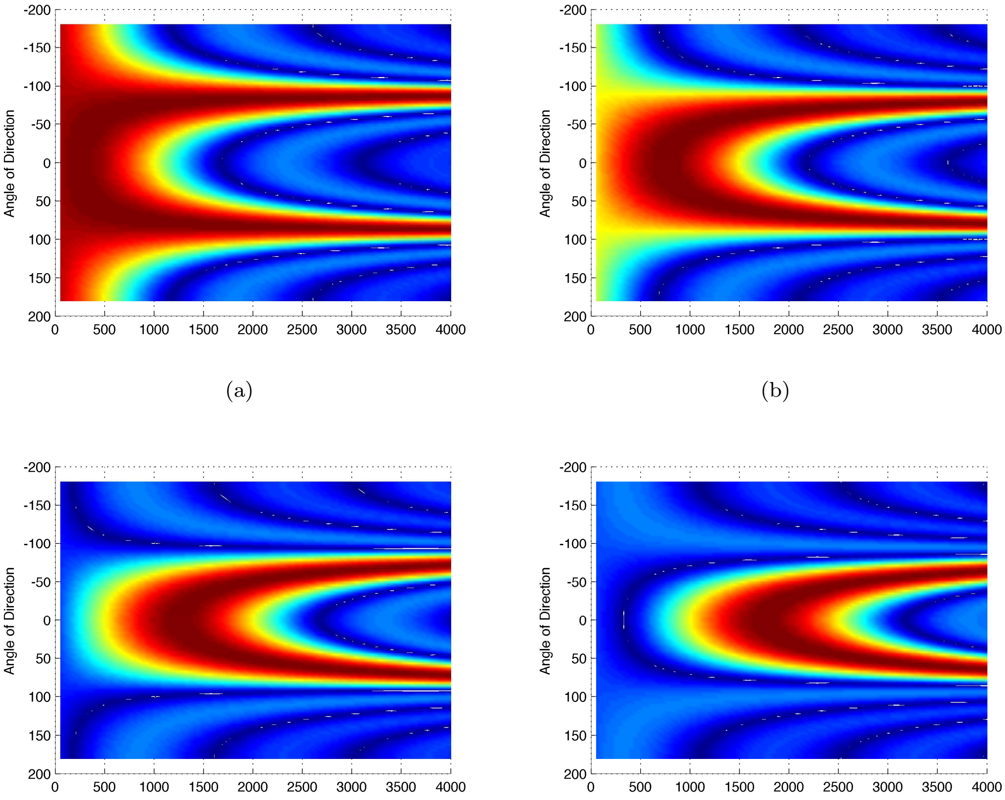
Algorithm 1 Maximum Kurtosis GSC

```
$\mathbf{w}_p \leftarrow [0, 0, \ ... \ , 0, 1]$
for every snapshot sample do
    Update $\mathbf{B}$ and $\mathbf{w}_q$ (Not done in this project)
    if Block of samples received then
        Update covariance matrix $\mathbf{\Sigma}(b) \leftarrow \mu\mathbf{\Sigma}(b-1) + (1-\mu)\hat{\mathbf{R}}_{zz}(b)$
        Generate subspace filter $\mathbf{U}$
        Update filter $\mathbf{w}_p$
    end if
end for
```

where:
$\hat{\mathbf{R}}_{zz}(b)$ is the sample covariance matrix for the current block
$\mathbf{\Sigma}(b)$ is the iterated covariance matrix used to generate $\mathbf{U}$

3.3.1 Filterbank

As mentioned and showed in section 3.1 the response of a sensor array is frequency dependent. The problem is now how to choose the frequency to generate the filter $\mathbf{w}_q$ in the GSC, when speech is broadband. The problem is illustrated on figure 3.10, which shows the array response for a uniform linear array with fixed interspacing in terms of frequency and direction for different choices of $\mathbf{w}_q$. There are two things to notice from these plot. The first thing is that the maximum gain (dark red) is not in the same direction across all frequencies. This will result in some undesirable coloration of the signal. The second thing to notice is that at low frequencies there is a lot of coloration, which is highly undesirable. This can be solved by using a high number of microphones or by increasing the spacing between them. Both methods are not very practical. We will not look into the last problem.



(a)

(b)

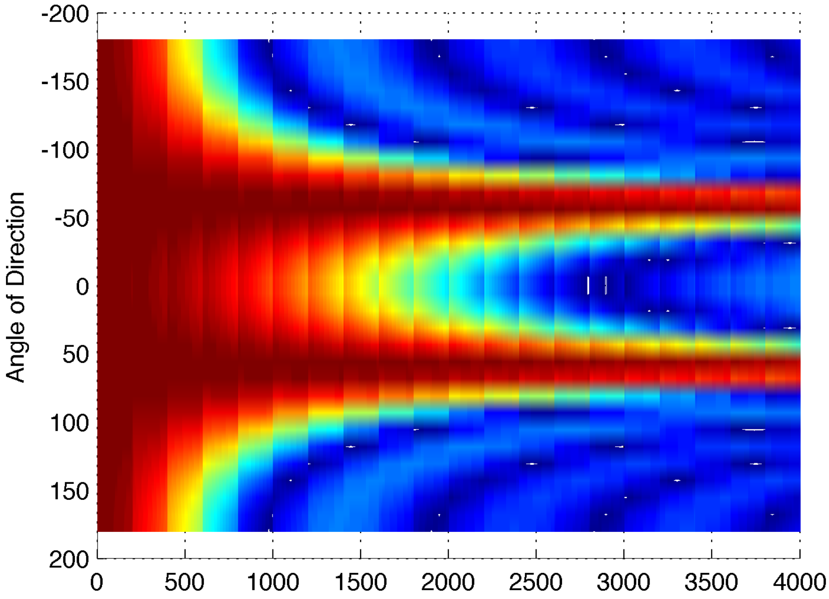
(c)

(d)

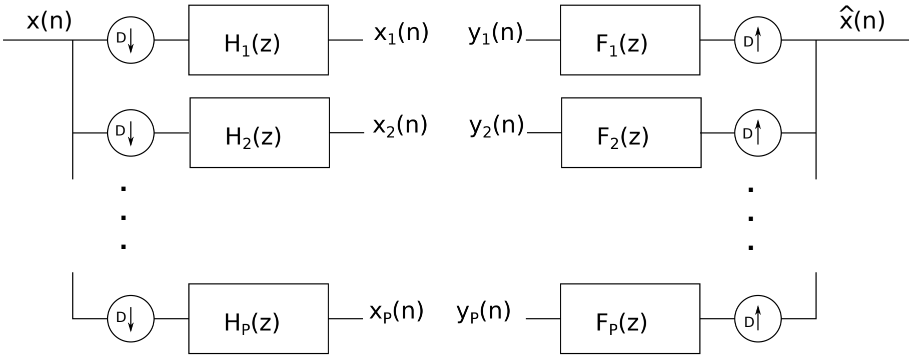
Figur 3.10: *Joint angle and frequency response for a microphone array for different frequencies of incomming signal with $M = 6$, $d = 0.04$ and $\theta = 60°$. (a) $f = 500$ Hz, (b) $f = 1500$ Hz, (c) $f = 2500$ Hz and (d) $f = 3500$ Hz.*

The first problem however can be solved by employing a subband structure where the spectrum is divided into P subbands and then assume the output from each subband to be a narrowband signal. Figure 3.11 shows the response of the same array in figure 3.10, but now with a subband structure, such that the beamformer $\mathbf{w}_q$ is created with the center frequency of each subband. We clearly see that maximum gain is attained at 60° across all frequencies as opposed to previous.

For the narrowband assumption to be valid infinitely many subbands must be used, thus making it infeasible in a real-world application. Because of this a finite number of subbands is used. A general subband system with P subbands is shown in figure 3.12.



Figur 3.11: *Joint angle and frequency response for microphone array when using 30 subbands.*



Figur 3.12: *Block diagram of a general filter bank system consisting of analysis bank (left) and synthesis bank (right) [11, p. 114]. The decimation is D.*

Lower complexity

Lower complexity is achieved by decimating the signal after subband filtering it. There is however only something to gain if the signal processing to be done has a higher complexity than doing the analysis filtering and synthesis filtering.

The potentially lower complexity do not come for free. The introduction of a filter bank will result in a time delay which is not desirable and makes it difficult implement in application requiring real-time performance. In this thesis this is however not the case, thus the time delay is not a problem.

Implementation

To implement a subband structure without introducing distortion or spectral coloration of the signal there are some properties, which are desirable. The first one is the perfect reconstruction property [11, p. 133], which is given by

$$\hat{x}(n) = c \cdot x(n - n_0) \tag{3.21}$$

where:
c is a non-zero constant scalar
n_0 is some integer

In words equation 3.21 states that in order for perfect reconstruction the output of the filter bank must be a constant scaled and fixed time-delayed version of the input signal. Another design rule is that the decimation factor, D, is chosen to be at maximum equal to the number of subbands, e.g. $D \leq P$. In this project it is chosen to use a cosine modulated filter bank, where the analysis- and synthesis filters are given by

$$h_k(n) = 2p_0(n) \cdot \cos\left(\left(k + \frac{1}{2}\right)\left(n + \frac{N}{2}\right)\frac{\pi}{P} + (-1)^k \frac{\pi}{4}\right) \tag{3.22}$$

$$f_k(n) = 2p_0(n) \cdot \cos\left(\left(k + \frac{1}{2}\right)\left(n + \frac{N}{2}\right)\frac{\pi}{P} - (-1)^k \frac{\pi}{4}\right) \tag{3.23}$$

where:
$k = 0,1,...,P-1$ is the subband index
$n = 0,1,...,N$ is the sample index
$p_0(n)$ is the prototype filter

It has the advantage of being simple to implement. From equation 3.23 we see that the filter bank is realised by finding a low-pass prototype filter and then multiplying by a modulating cosine to get the desired bandpass-filter. It is therefore of importance to chosse the right prototype filter.

Verification

This section will verify the implementation of a filter bank implementation by applying it to a speech signal and then comparing to the original signal using spectrograms and MSE. Table 3.4 shows the parameter values for the verification.

Parameter	Value(s)
P	8
D	8
N	2048
F_s	8kHz

Tabel 3.4: *Parameter values for filter bank verification.*

The prototype filter is chosen to be a FIR-filter designed using the window method, furthermore a Hanning window is used. The response of this filter is seen on figure 3.13.

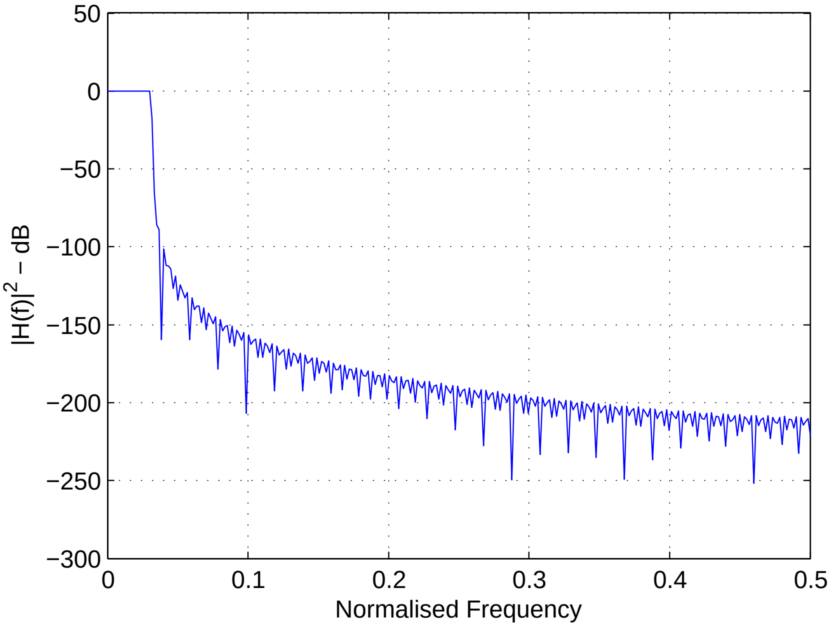
Figure 3.14 shows the magnitude response of the analysis bank and synthesis bank.

Figure 3.15 shows the time series and the spectrogram of the signal before and after the filter bank. We see that these are almost identical.

The MSE between $x(n)$ and $\hat{x}(n)$ was found to be $4.35 \cdot 10^{-7}$. Based on comparison of time series and spectrograms and the MSE, we conclude that the filter bank is implemented correct.

3.3.2 Kurtosis Adaptive filter

This section will explain and derive the adaptive filter problem when it is desired to maximize the kurtosis of the output.



Figur 3.13: *Magnitude frequency response of prototype filter.*

Motivation

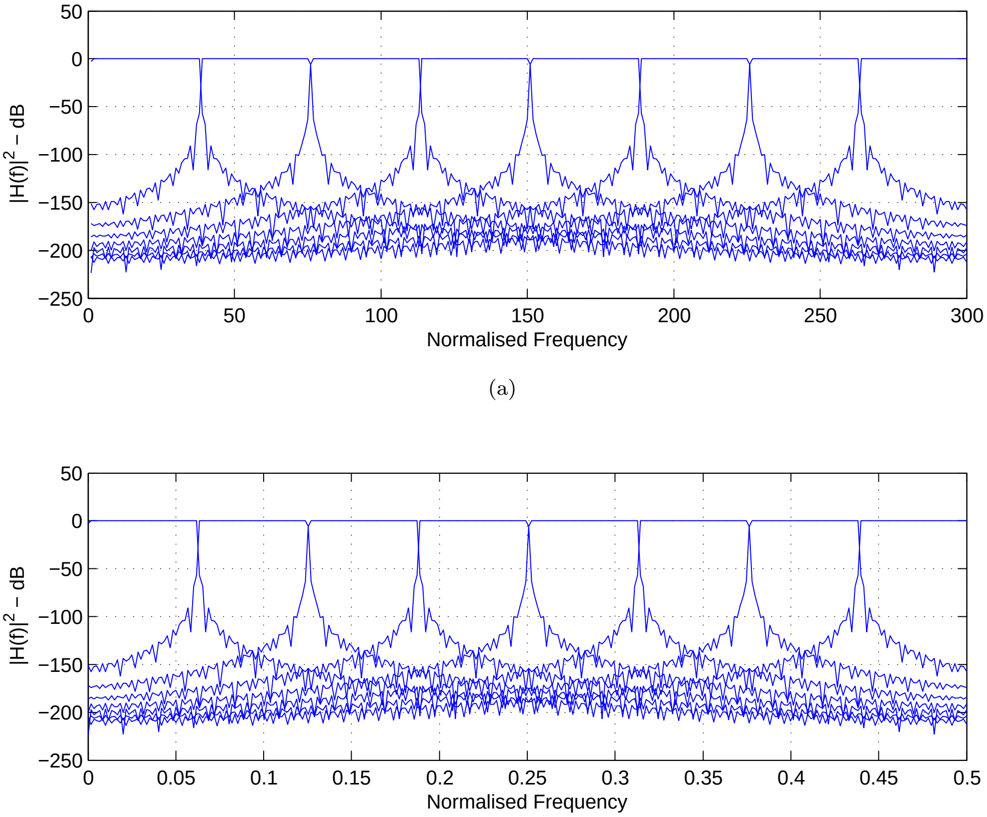
The adaptive filter problem in the conventional GSC, described in section 3.2, aims at minimizing the mean-squared error between $d(n)$ and $y(n)$, e.g. $\mathbb{E}[(d(n) - y(n))^2]$. The reason for this was to minimise the power in all other directions than the desired one. In this section the approach given in [2, 10] is investigated. Here it is sought to maximize the kurtosis of the output, $e(n)$. The Kurtosis of a random variable e is given by [12]

$$\text{Kurt}(e) = \mathbb{E}[|e|^4] - \beta\mathbb{E}[|e|^2]^2 \tag{3.24}$$

The kurtosis quantifies the shape of a Probability Density Function (PDF) as being high if the PDF is narrow and has long and heavy tails and vice versa. Setting $\beta = 3$ gives the following interpretation of the kurtosis for a given PDF of a random variable, e, [12]

- Super-gaussian, $\text{Kurt}(e) > 0$
- Gaussian, $\text{Kurt}(e) = 0$
- Sub-gaussian, $\text{Kurt}(e) < 0$

The proof that the kurtosis of a Gaussian random variable with zero mean and unit variance is zero, is given in appendix C. It has been observed that the PDF of clean speech is super-gaussian [13], thus this can be used as a measure to distuinguish clean speech from other sources. By looking at the signal model for the mth microphone in equation 2.1 and assuming that the sources, noise and reverberation are independent samples, we can employ the Central Limit Theorem (CLT), which states that the sum of an infinite number of independent random variables is distributed according to a gaussian distribution. We recall from equation 2.1 that all the reflections are *not* independent, however as the reverberation time is increased, the reflections and direct-path signal becomes almost independent. This claim is supported by empirical results found in [13], where it is found that the distribution of reverberant speech (not considering noise) tends toward a gaussian distribution as the reverberation time increases. The idea is then to adjust the filter coefficients in such a way that the output has a super-gaussian distribution (e.g. maximize the kurtosis) and thus will resemble the speech signal from the desired source.



(a)

(b)

Figur 3.14: *Frequency magnitude reponse of (a) the analysis bank (b) the synthesis bank, for a filter length of $N = 2048$ and $P = 8$ subbands.*

Estimating the Kurtosis

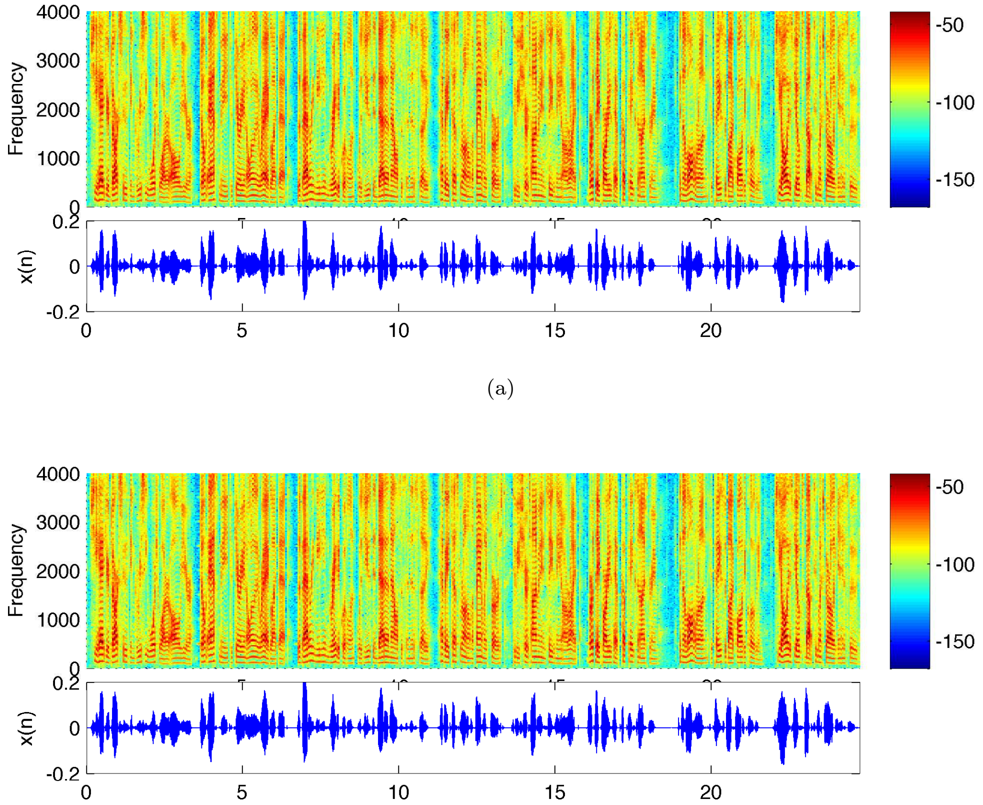
In practice the kurtosis is not known and therefore needs to be estimated. This is done by using the sample kurtosis, which for a data set $\mathbf{e} = [e(1)\ e(2)\ ...\ e(M)]^T$ is given by [2]

$$\widehat{\text{Kurt}}(\mathbf{e}) = \frac{1}{M}\sum_{n=1}^{M}|e(n)|^4 - \beta\left(\frac{1}{M}\sum_{n=1}^{M}|e(n)|^2\right)^2 \tag{3.25}$$

where:

M is the block/segment size

To support the claim that clean speech is super-gaussian and that reverberant speech has a more gaussian-like distribution some empirical investigations are carried out. Figure 3.16 shows the time series, histogram along with fitted distributions and the kurtosis, for a speech signal of 4s recorded close to the speaker (left) and recorded using a distant microphone (right). We denote these signals as clean speech and reverberant speech, respectively. Figure shows the histogram for the two signals together with fitted Gaussian and Laplace distributions. We see that for clean speech the histogram is very peaky and has relatively much weight or mass in the tails, thus it is very super-gaussian. The reverberated speech is also super-gaussian but not as much as clean speech. We see that it seems to be very well approximated by a Laplace distribution. Figure shows the kurtosis calculated with three different block sizes using equation 3.25. First we note that the kurtosis is generally higher for the clean speech than the reverberant speech across all block sizes.



(a)

(b)

Figur 3.15: *Spectrogram of (a) $x(n)$ (b) $\hat{x}(n)$, for a filter length of $N = 2048$ and $P = 8$ subbands.*

Second it is interesting to see how much the kurtosis varies depending on the block size. This indicates, that the block size can have a great influence on the estimation of the kurtosis.

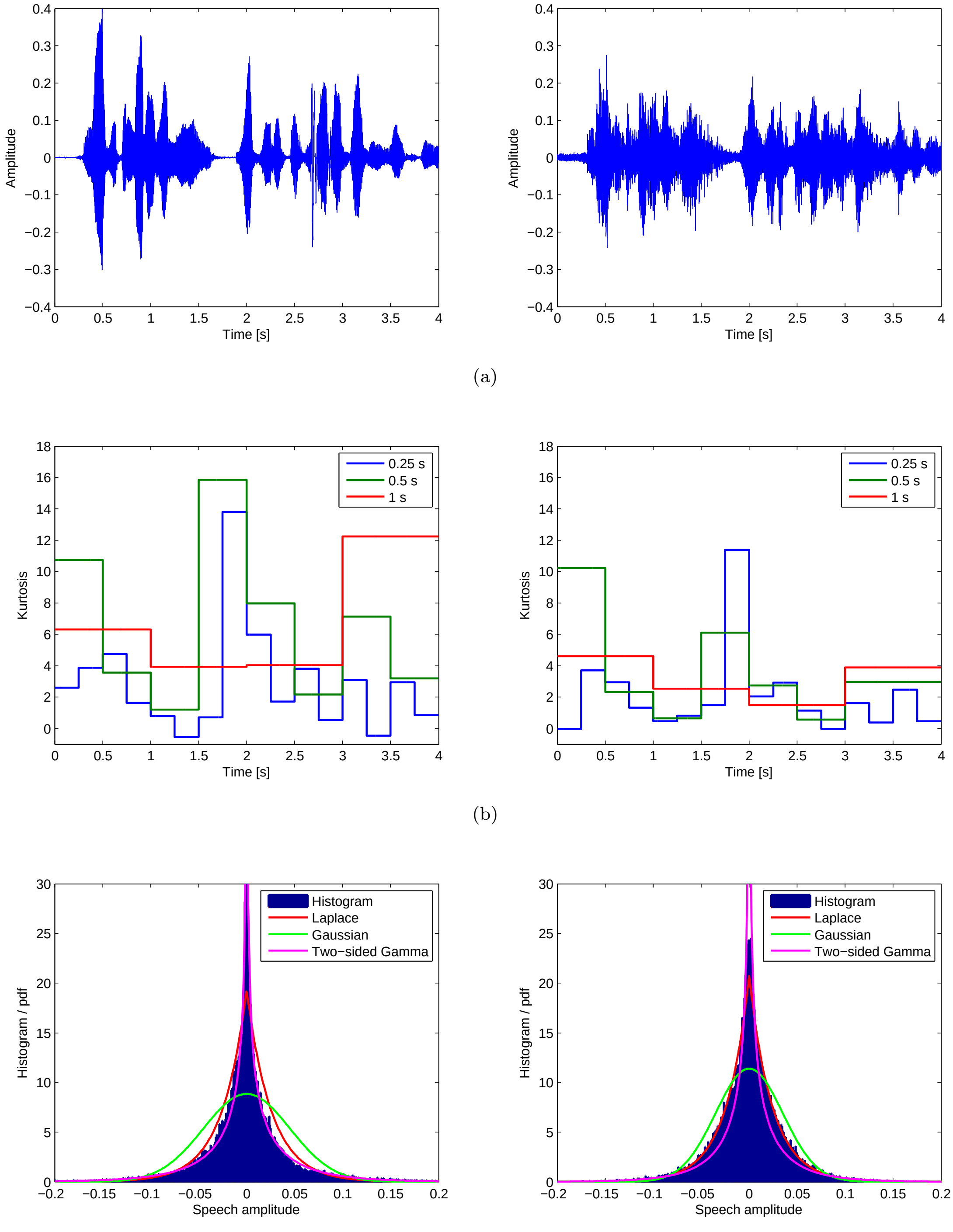
Last, we note that for the clean speech and block size of 0.25s the kurtosis is low for parts where speech is present, which may indicate that some parts of speech do not have a super-gaussian distribution. To investigate this further the a subset of the TIMIT database was used to find the average kurtosis of each phoneme group and each phoneme. The average kurtosis of the phoneme classes is seen in figure 3.17. It is interesting to see how much the kurtosis varies across phoneme classes and that some classes actually have a very low kurtosis. This shows that some parts of speech do not have a super-gaussian distribution.

The sample kurtosis calculated for the entire time series is 8.8 and 3.6 for clean speech and reverberant speech, respectively. Based on these plots, we thus confirm that reverberant speech is less super-gaussian than clean speech.

There are however drawbacks of using the kurtosis as a measure of non-gaussianity, because this is sensitive to outliers, which is not ideal [12, p. 182] and can lead to false estimates of the filter weights. This issue will be addressed later.

Updating the filter coefficients

As mentioned in the introduction to the improved GSC the adaptive filter is only updated for every block of samples and we are interested in finding the filter which maximizes the sample kurtosis for the current block of samples. We can define the cost function as the sample kurtosis and add a term which penalizes large filter coefficents. If this term is not added, it is easily seen that equation 3.25 is maximized by making the coefficients of $\mathbf{w}$ infinitely big.

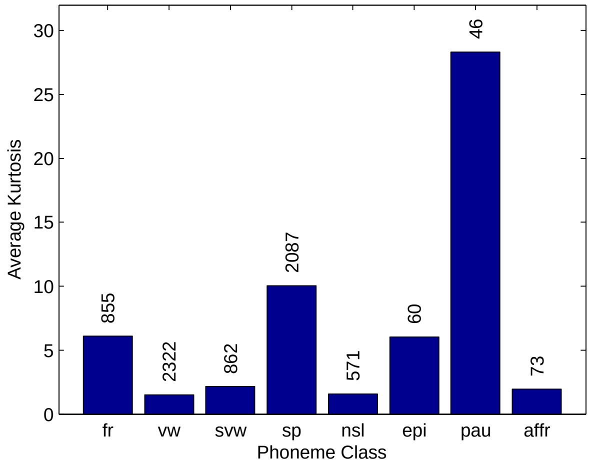


(a)

(b)

(c)

Figur 3.16: *(a) Time series, (b) sample kurtosis and (c) histogram and fitted distribution for close microphone recording (left) and distant microphone recording (right). The length of the signal is 4s sampled at* 16000 *kHz. Histograms are generated with 1000 bins.*



Figur 3.17: *Average kurtosis for each phoneme class. The number over each bar is the number of phonemes used to find the average.*

$$\mathcal{J}(\mathbf{w}) = \frac{1}{M}\sum_{n=1}^{M}|e(n)|^4 - \beta\left(\frac{1}{M}\sum_{n=1}^{M}|e(n)|^2\right)^2 - \alpha\,||\mathbf{w}||_2^2 \tag{3.26}$$

The strategy is now to find the gradient and use this to find the optimum filter. The gradient is derived in appendix B and is given by

$$\begin{aligned}\mathbf{g}(\mathbf{w}(k)) = &-\frac{2}{M}\sum_{n=b_k}^{b_k+M-1}|e(n)|^2\cdot\mathbf{v}(n)e^*(n)\\ &+\left(\frac{2\beta}{M^2}\sum_{n=b_k}^{b_k+M-1}|e(n)|^2\right)\cdot\sum_{n=b_k}^{b_k+M-1}\mathbf{v}(n)e^*(n) - \alpha\mathbf{w}(k)\end{aligned} \tag{3.27}$$

where:
$k = 1{,}2{,}....P$ is the block-index
M is the block size given in samples
b_k is the index of the first sample in the kth block
$\mathbf{v}(n) = \mathbf{U}^H\mathbf{B}^H\mathbf{x}(n)$ is given by figure 3.9(b) on page 15

For each block of samples we use the gradient ascent method along with backtracking line search to find the optimum filter to apply to the current block of samples, which is given by [14, p. 464]. Pseudo code for this algorithm is shown in 2.

A typical stopping criteria is when the norm of the gradient becomes smaller than some predefined threshold, i.e. $||\mathbf{g}(\mathbf{w})||_2 < \epsilon$. Note that according to [2], there is a need for projecting the filter onto the unit circle if the norm of the filter exceeds 1. The advantage of using the gradient method is the simplicity, however we are only guaranteed a local optimum and the convergance rate depends much on the condition number of the Hessian [14, p. 475]. This means that the algorithm may become very slow in some cases.

Verification

In this section the implementation of the gradient ascent method with backtracking line search is verified. To simplify the verification the kurtosis cost function in equation 3.26 is replaced by an analytical function of the form

Algorithm 2 Gradient ascent with backtracking line search

```
$t = 1$, $\alpha \in ]0{,}0.5]$, $\beta \in ]0{,}1]$ and starting point $\mathbf{w}$
while Stopping criteria not satisfied do
  while $\mathcal{J}(\mathbf{w} + t\mathbf{g}(\mathbf{w})) < \mathcal{J}(\mathbf{w}) + \alpha t\,||\mathbf{g}(\mathbf{w})||_2^2$ do
    $t \leftarrow \beta t$
  end while
  $\mathbf{w} \leftarrow \mathbf{w} + t\mathbf{g}(\mathbf{w})$
  if $||\mathbf{w}||_2 > 1$ then
    $\mathbf{w} = \frac{\mathbf{w}}{||\mathbf{w}||_2}$
  end if
end while
```

$$\mathcal{J}(\mathbf{w}) = \mathbf{w}^T\mathbf{R}\mathbf{w} + \mu\mathbf{w}^T\mathbf{w} \tag{3.28}$$

and the gradient is thus given as

$$\mathbf{g}(\mathbf{w}) = \mathbf{R}\mathbf{w} + \mu\mathbf{w} \tag{3.29}$$

To simplify even further and to be able to visualize the cost function, we constrain the problem to 2 dimenions, i.e. $\mathbf{w} \in \mathbb{R}^{2\times1}$. Based on the gradient we know that the optimum point is a vector of zeros, i.e. $\mathbf{w}_{opt} = [0\ 0]^T$. Table 3.5 shows how the paramteres are chosen for the verification.

Parameter	Value(s)
t	1
α	0.1
β	0.4
μ	0.3
ϵ	0.0001
$\mathbf{R}$	$\begin{bmatrix} -0.5 & 0 \\ 0 & -1.5 \end{bmatrix}$

Tabel 3.5: *Parameter values for gradient verification.*

Figure 3.18 shows a 3D plot of the cost function and a contour plot with the results for gradient ascent method.

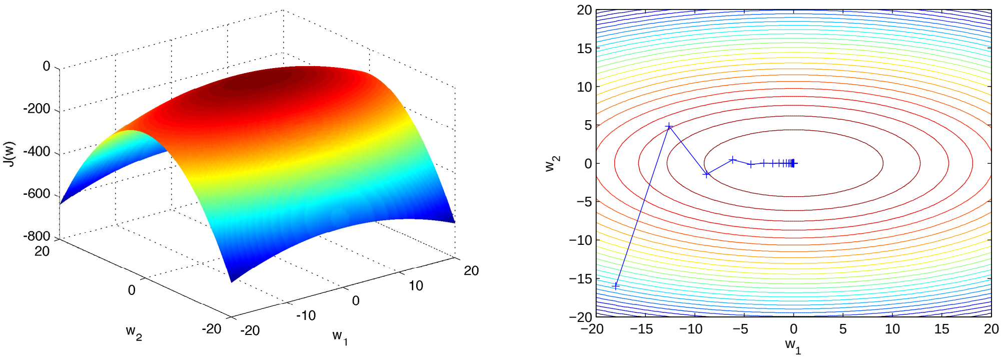
The output of the algorithm after is seen in table 3.6 and we see that it reaches the optimum as expected. We thus conclude that the implementation is correct.

Parameter	Value(s)
Number of iterations	49
$\mathbf{w}$	$[-4\cdot10^{-4}\ -4.5\cdot10^{-33}]^T$
$\mathcal{J}(\mathbf{w})$	$-3.2\cdot10^{-8}$

Tabel 3.6: *Result for gradient verification.*

3.3.3 Subspace filtering

As mentioned in section 3.3.2 the sample kurtosis is sensitive to outliers, thus outliers can cause incorrect updates of the filter, $\mathbf{w}_q$. To avoid this the noise subspace is estimated as an average over all noise-vectors making it more robust and one-dimensional.



(a)

(b)

Figur 3.18: *(a) 3D plot of the cost function $\mathcal{J}(\mathbf{w})$ (b) Contour plot of the cost function $\mathcal{J}(\mathbf{w})$ together the result of the gradient ascent algorithm for each iteration.*

Method

Consider the $(M-1)\times 1$ output, $\mathbf{z}(n)$, from the blocking matrix $\mathbf{B}$ shown in figure 3.9(b) on page 15. We ommit the frequency index, f for convenience. Due to the orthogonality between the blocking matrix and $\mathbf{w}_q$ and assuming perfect steering, $\mathbf{z}(n)$ will *not* contain any contribution from the desired signal but only contributions from interfering signals and additive white gaussian noise, both spatially and in time. Here we use the same signal model as in 3.5 on page 6

$$\mathbf{z}(n) = \mathbf{A}\mathbf{s}(n) + \mathbf{v}(n) \tag{3.30}$$

where:
$\mathbf{s}(n)$ contains the signal from D interferers
$\mathbf{v}(n)$ is AWGN

We assume that there are fewer interfering signals than there are microphones, e.g. $D < M$. In the case of a highly reverberant room, there will be reflections impinging from many different angles, thus there will be a very high number of "interferers", which will probably exceed the number of microphones. Furthermore these reflections are not independent, which makes the task more difficult. This will be mentioned in the end. First we consider the case of independent interferers and spatially uncorrelated white noise. Taking the covariance matrix of $\mathbf{z}$ and exploiting that the interfering signals and the noise are uncorrelated yields

$$\mathbf{R}_{zz} = \mathbb{E}[\mathbf{z}\mathbf{z}^H] = \mathbf{A}\mathbf{R}_{zS}\mathbf{A}^H + \mathbf{R}_{zV} = \mathbf{A}\mathbf{R}_{zS}\mathbf{A}^H + \sigma_V^2\mathbf{I} \tag{3.31}$$

where:
$\mathbf{R}_{zS} = \mathbb{E}[\mathbf{s}(n)\mathbf{s}(n)^H]$
$\mathbf{R}_{zV} = \mathbb{E}[\mathbf{v}(n)\mathbf{v}(n)^H]$
$\mathbf{I}$ is the identity matrix
σ_V^2 is the noise-variance

We now want to find a basis for the D-dimensional subspace spanned by the interfering signals. This can be achieved by first taking the Eigenvalue Decomposition (EVD) of the covariance matrix given in equation 3.31 and then picking the eigenvectors corresponding to the D largest eigenvalues [15, p. 166]. Since $\mathbf{R}_{zz}$ is hermitian the EVD is given by [15, p. 348]

$$\mathbf{R}_{zz} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^H \tag{3.32}$$

where:
$\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{M-1}]$ are the eigenvectors
$\mathbf{\Lambda} = \text{diag}\ [\lambda_1, \lambda_2, ..., \lambda_{M-1}]$ contains the eigenvalues

When not taking reflections into consideration, the eigenvalues attain the following values when they are sorted in descending order [15, p. 166]

$$\lambda_k = \begin{cases} \sigma_S^2 + \sigma_V^2 & \text{for } 1 \leq k \leq D \\ \sigma_V^2 & \text{for } D+1 \leq k \leq M \end{cases}$$

Based on this we can now define our signal subspace as $\mathcal{S}_S = \mathcal{R}\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_D\}$ and our noise subspace as $\mathcal{S}_V = \mathcal{R}\{\mathbf{e}_{D+1}, \mathbf{e}_{D+2}, ..., \mathbf{e}_{M-1}\}$, where $\mathcal{R}\{\cdot\}$ denotes the range operator [15]. The subspace filter is now constructed in the following way

$$\mathbf{U} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_D, \mathbf{e}_{\tilde{V}}] \tag{3.33}$$

where:
$\mathbf{e}_{\tilde{V}} = \sum_{k=1}^{M-1-D} \mathbf{e}_{D+k}$

We see that we have seperated the signal and noise subspaces and reduced the noise subspace to be of one dimension instead of $M - D - 1$ by making an average noise vector. This makes the estimation of the noise much more robust and reduces the dimensionality in the case where many microphones are used.

As mentioned earlier, when many reflections are present the number of signals will exceed the number of microphones, e.g. $D > M$, which makes this method useless. However some reflections may have a very small amplitude compared to the noise-variance and can therefore be neglected. Another problem arise if the signals are perfectly correlated, then it is impossible to divide the range of the covariance matrix into a signal- and noise subspace [16, p. 378].

Choosing the size of signal subspace and noise subspace

It is necessary to find a robust and automatic way of estimating how many eigenvectors the signal subspace and noise subspace comprises of. In [2] it is suggested to use a measure called contribution ratio and then threshold on this. The contribution ratio for the ith eigenvector is given by

$$C_i = \frac{\lambda_i}{\sum_{k=1}^{M-1} \lambda_k} \tag{3.34}$$

We then decide if an eigenvector belongs to either the signal subspace or the noise subspace by thresholding on C_i, if $C_i \geq \mathit{threshold}$ then eigenvector $\mathbf{e}_i$ belongs to the signal subspace and if not, then it belongs to the noise subspace.

3.3.4 Postfiltering

So far attention has been given to suppress interfering signals and not reducing the noise in equation 3.5. This section describes how to reduce noise after beamforming has been applied, hence the name *post*filtering. We assume that the true signal has been corrupted by AWGN, thus the signal model for the output of the GSC can be described in the following way:

$$e(n) = s(n) + w(n) \tag{3.35}$$

where:
$e(n)$ is the output from the GSC at time-index n
$s(n)$ is the true signal at time-index n
$w(n)$ is AWGN at time-index n

To reduce the noise, we can apply the well-known Wiener-filter [17, p. 612]. In order for the use of this filter to be valid, $s(n)$ and $w(n)$ must be Wide Sense Stationary (WSS) processes and uncorrelated, $\mathbb{E}[s(n_1)w(n_2)] = 0$ for all n_1 and n_2. We assume that $w(n)$ obey the assumptions, but as mentioned in section 2 the source signals are non-stationary, hence $s(n)$ is also non-stationary, which violates the WSS assumption. This can however be overcome by considering frames of 20−30 ms seperately. The Wiener-filter seeks to find a linear filter, h, which minimizes the MSE given by

$$\mathbb{E}[(s(n) - \hat{s}(n))^2] \tag{3.36}$$

where:

$$\hat{s}(n) = \sum_{k=-\infty}^{\infty} h(k)e(n-k)$$

The solution is given by

$$H(f) = \frac{P_s(f)}{P_s(f) + P_w(f)} = \frac{P_s(f)}{P_e(f)} \tag{3.37}$$

where:

$H(f)$ is the frequency-domain Wiener-filter
$P_s(f)$ and $P_w(f)$ are the Power Spectral Density (PSD) of $s(n)$ and $w(n)$, respectively
$P_e(f)$ is the PSD of $e(n) = s(n) + w(n)$

The time-domain filter can then be obtained by applying the inverse Fourier Transform on $H(f)$. Since we do not know $P_s(f)$ and $P_w(f)$, these must be estimated in some way, which will be described next.

Zelinski postfiltering

Since the signal, $s(n)$, can only be considered WSS in frames of $20 - 30$ ms the PSD's cannot be estimated by averaging over a long time series, in other words we need to estimate the PSD's using only data from the current frame. One possibility is to assume ergodicity to split the data into smaller sets and then do ensemble averaging. However this results in a degradation of resolution in the frequency domain, which is not desirable. This problem can be tackled by using Zelinski postfiltering [18], where the method refers to estimating the PSD's and not the actual filter. This method uses the fact that multiple microphone signals are present. We assume the following signal model (same as in equation 2.1) for the signal at the mth microphone

$$y_m(n) = \sum_{k=1}^{K} g_{m,k}(n) * s_k(n) + v_m(n) \tag{3.38}$$

and also that each microphone signal, $m = 1,...,M$, has been compensated for delay such that they are aligned according to the desired direction. This compensation method will not be described in this report. Using the signal model we can now find $P_s(f)$ and $P_e(f)$.

Estimating $P_e(f)$

Zelinski postfiltering estimates $P_e(f)$ by estimating the PSD for each of the microphone signals and then average over them. The PSD of $y_m(n)$ is given as [17, p. 569]

$$\mathbb{E}[Y_m^*(f)Y_m(f)] = \mathbb{E}\left[\left(\sum_{k=1}^{K} G_{m,k}(f)S_k(f) + V_m(f)\right)^* \left(\sum_{k=1}^{K} G_{m,k}(f)S_k(f) + V_m(f)\right)\right] \tag{3.39}$$

where:

$S(f)$ is the Discrete Fourier Transform of $s(n)$

For simplicity we assume that $K = 2$, which yields

$$\begin{aligned}\mathbb{E}[Y_m^*(f)Y_m(f)] =& \mathbb{E}[(G_{m,1}^*(f)S_1^*(f) + G_{m,2}^*(f)S_2^*(f) + V_m^*(f))(G_{m,1}(f)S_1(f) \qquad (3.40)\\ & + G_{m,2}(f)S_2(f) + V_m(f))]\\ =& \mathbb{E}[G_{m,1}^*(f)S_1^*(f)G_{m,1}(f)S_1(f) + G_{m,1}^*(f)S_1^*(f)G_{m,2}(f)S_2(f) + \qquad (3.41)\\ & G_{m,1}^*(f)S_1^*(f)V_m(f) + G_{m,2}^*(f)S_2^*(f)G_{m,1}(f)S_1(f) +\\ & G_{m,2}^*(f)S_2^*(f)G_{m,2}(f)S_2(f) + G_{m,2}^*(f)S_2^*(f)V_m(f) +\\ & V_m^*(f)G_{m,1}(f)S_1(f) + V_m^*(f)G_{m,2}(f)S_2(f) + V_m^*(f)V_m(f)]\end{aligned}$$

All the cross-terms equal zero due to the assumtions that all sources are uncorrelated and zero-mean [17, p. 651] resulting in

$$\mathbb{E}[Y_m^*(f)Y_m(f)] = |G_{m,1}(f)|^2 \underbrace{\mathbb{E}[|S_1(f)|^2]}_{P_{s_1}(f)} + |G_{m,2}(f)|^2 \underbrace{\mathbb{E}[|S_2(f)|^2]}_{P_{s_2}(f)} + \underbrace{\mathbb{E}[|V_m(f)|^2]}_{P_{v_m}(f)} \tag{3.42}$$

We can thus estimate $P_e(f)$ by taking the power of the Discrete Fourier Transform (DFT) of each of the microphone signals and then average over them, which can be stated as

$$\hat{P}_e(f) = \frac{1}{M}\sum_{m=1}^{M} |\mathcal{F}(y_m(n))|^2 \tag{3.43}$$

where:
$\hat{P}_e(f)$ denotes the estimate of $P_e(f)$
M is the number of microphones
$\mathcal{F}()$ denotes the Fourier Transform

There are two things to notice from equation 3.42. The first thing is that assuming our source of interest is $s_1(n)$ and that the beamformer perfectly removes all other $(K-1)$ sources, then equation 3.35 can be written as

$$e(n) = s_1(n) + w(n) \tag{3.44}$$

$$\tag{3.45}$$

and the PSD of $e(n)$ is given by

$$P_e(f) = P_{s_1}(f) + P_w(f) \tag{3.46}$$

Comparing equation 3.42 and equation 3.46 it is seen that $P_e(f)$ is overestimated by the sum of the PSD of each of the interfering signals. Another thing that is also seen by comparing equation 3.42 and equation 3.46 is that unless $P_w(f) = P_{v_m}(f)$ the noise is also overestimated. It is thus not taken into consideration that the beamformer itself will remove some of the noise making $P_w(f) \le P_{v_m}(f)$ for all f.

Estimating $P_s(f)$
$P_e(f)$ can be estimated by taking the cross-spectrum of the microphone signals and assuming that the noise for two different microphones are uncorrelated, e.g. $\mathbb{E}[v_m(k)v_p(k)]$ for $m,p = 1,...,M$ and $m \neq p$. The cross-spectrum is given by

$$\mathbb{E}[y_m^*(f)y_p(f)] = \mathbb{E}\left[\left(\sum_{k=1}^{K} G_{m,k}(f)S_k(f) + V_m(f)\right)^* \left(\sum_{k=1}^{K} G_{p,k}(f)S_k(f) + V_p(f)\right)\right] \tag{3.47}$$

For simplicity we again assume $K = 2$, which yields

$$\begin{aligned}\mathbb{E}[Y_m^*(f)Y_p(f)] =& \mathbb{E}[(G_{m,1}^*(f)S_1^*(f) + G_{m,2}^*(f)S_2^*(f) + V_m^*(f))(G_{p,1}(f)S_1(f)+ & (3.48)\\ & G_{p,2}(f)S_2(f) + V_p(f))] \\ =& \mathbb{E}[G_{m,1}^*(f)S_1^*(f)G_{p,1}(f)S_1(f) + G_{m,1}^*(f)S_1^*(f)G_{p,2}(f)S_2(f)+ & (3.49)\\ & G_{m,1}^*(f)S_1^*(f)V_p(f) + G_{m,2}^*(f)S_2^*(f)G_{p,1}(f)S_1(f)+ \\ & G_{m,2}^*(f)S_2^*(f)G_{p,2}(f)S_2(f) + G_{m,2}^*(f)S_2^*(f)V_p(f)+ \\ & V_m^*(f)G_{p,1}(f)S_1(f) + V_m^*(f)G_{p,2}(f)S_2(f) + V_m^*(f)V_p(f)]\end{aligned}$$

Again all the cross-terms are equal to zero due to the same assumption as before, and we thus get

$$\mathbb{E}[Y_m^*(f)Y_p(f)] = G_{m,1}^*(f)G_{p,1}(f)\underbrace{\mathbb{E}[|S_1(f)|^2]}_{P_{s_1}(f)} + G_{m,2}^*(f)G_{p,2}(f)\underbrace{\mathbb{E}[|S_2(f)|^2]}_{P_{s_2}(f)} \tag{3.50}$$

$P_s(f)$ can now be estimated by first estimating all possible cross-spectra and then average over them. This can be stated by

$$\hat{P}_s(f) = \frac{2}{M(M-1)} \mathfrak{Re}\left[\sum_{m=1}^{M-1} \sum_{q=m+1}^{M} \mathcal{F}(y_m(n))^* \mathcal{F}(y_q(n))\right] \tag{3.51}$$

where:
$\mathfrak{Re}[\cdot]$ denotes the Real-operator

Taking only the real part of the estimate is justified by the fact that the true PSD of $s(n)$ is real-valued [17, p. 573].

From equation 3.50 we again see that in the case where all interfering sources are removed, the PSD of $s(n) = s_1(n)$ is overestimated.

Combining the two estimates of the PSD's we get the following

$$\hat{H}(f) = \frac{\hat{P}_s(f)}{\hat{P}_e(f)} = \frac{\frac{2}{M(M-1)} \mathfrak{Re}\left[\sum_{m=1}^{M-1} \sum_{q=m+1}^{M} \mathcal{F}(y_m(n))^* \mathcal{F}(y_q(n))\right]}{\frac{1}{M}\sum_{m=1}^{M} |\mathcal{F}(y_m(n))|^2} \tag{3.52}$$

Verification

In this section the implementation of Zelinski postfiltering is verified by running a small numerical example as in section 3.2.4 on page 9. We use the signal-to-noise plus interference ratio (SNIR) as a measure of quality, which is defined as

$$\text{SNIR}_{dB} = 10 \cdot \log_{10}\left(\frac{P_S}{P_I + P_N}\right) \tag{3.53}$$

where:
P_S is the power of the desired signal
P_I is the power of the interfering signal
P_N is the power of the noise

When no interference is present SNIR corresponds to the well-known signal-to-noise ratio (SNR). The verification is done by sweeping over a range of input SNIR and then calculate the output SNIR in case 1: narrowband where no interference is present, case 2: narrowband when a single interferer is present, and case 3: real speech from TIMIT database. In all cases we use the same signal model as in equation 3.19 in section 3.2.4 on page 9 and the same settings unless stated otherwise. Furthermore the postfiltering is implemented using the overlap-add method, thus a specific window and overlap has to be chosen. The settings for both narrowband cases (1 and 2) are given in table 3.7

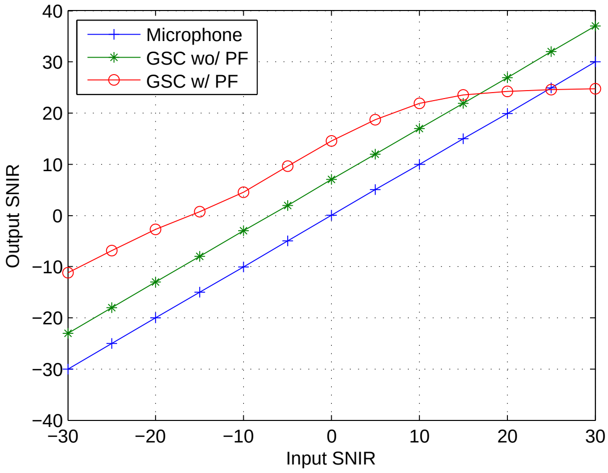
Case 1 and 2

Parameter	Value
Fs	256 Hz
N	8192
Number of simulation pr. SNIR	5
Window	Hanning
Overlap	50%
Postfilter block size	32 samples = 12.5 ms

Tabel 3.7: *Parameter values for postfilter.*

Parameter	Value
d	$\frac{\lambda}{2} = 7.6$ m
M	5
A	1
F	45 Hz
θ	90°

Tabel 3.8: *Parameter values for case 1; without interference.*



Figur 3.19: *Case 1: Plot of output $SNIR_{dB}$ as a function of input $SNIR_{dB}$.*

Parameter	Value
d	$\frac{\lambda}{2} = 7.6$ m
M	5
A	1
F	45 Hz
θ	90°
K	1
B	0.1
f	10 Hz
ϕ	70 °

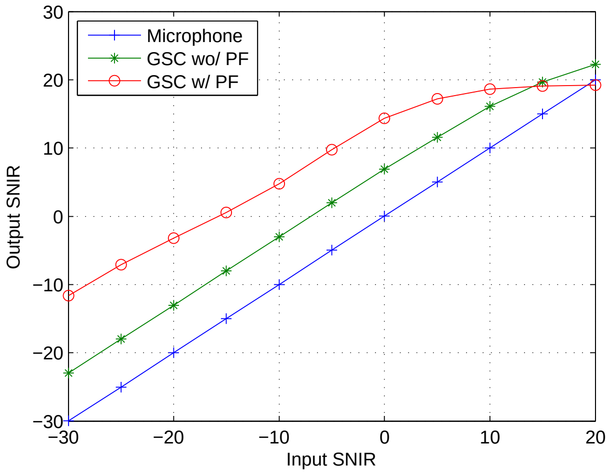
Tabel 3.9: *Parameter values for case 2; with interference.*

Real speech

Table 3.10 shows the settings for the verfication using real speech.

Figure 3.21 shows the output SNR as a function of the input SNR. We see that for low values of SNR the postfiltering enhances the signal by approximately 12dB. As the SNR increases we see that the SNR output of the postfilter converges towards the SNR of the GSC, which is to be expected because the Wiener filter in equation 3.37 can be written as

$$H(f) = \frac{P_s(f)}{P_s(f) + P_w(f)} = \frac{\frac{P_s(f)}{P_w(f)}}{\frac{P_s(f)}{P_w(f)} + \frac{P_w(f)}{P_w(f)}} = \frac{\mathrm{SNR}(f)}{\mathrm{SNR}(f) + 1} \approx 1, \text{ for SNR} \gg 1 \tag{3.54}$$

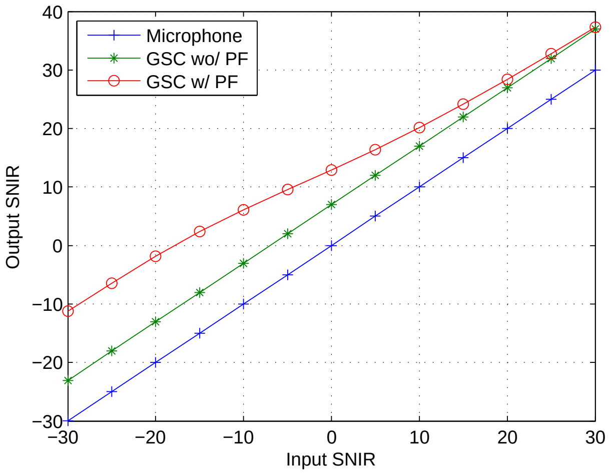


Figur 3.20: *Case 2: Plot of output $SNIR_{dB}$ as a function of input $SNIR_{dB}$.*

Parameter	Value
TIMIT-sentence	Region 1, Speaker FAKS0, file SA1.wav
Fs	8000 Hz
N	63488 samples
M	5
Number of simulation pr. SNIR	3
Window	Hanning
Overlap	50%
Postfilter block size	2048 samples = 128 ms
DOI	90°

Tabel 3.10: *Parameter values for postfilter.*

Based on these simulations it is fair to conclude that the Zelinski postfilter is implemented correct.



Figur 3.21: *Case 3: Plot of output SNR_{dB} as a function of input SNR_{dB}.*

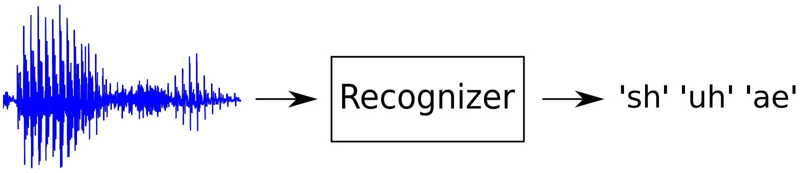
3.4 Summary

In this chapter the response of a Uniform Linear Array (ULA) has been given. Furthermore the a classic adaptive beamforming algorithm, Generalised Sidelobe Canceller (GSC) was derived, implemented and verified in the case of narrowband signals. It showed good performance and was able to suppress interfering signals. The classic GSC was extended according to [2] to maximize the kurtosis of the output instead of minimizing the MSE. The well-known Zelinski Wiener-filter for postfiltering of the output of the beamforming algorithms was derived, implemented and verified through testing in various SNR conditions. The next chapter will give a brief overview of the general theory Automatic Speech Recognition (ASR) along with two widely-used adaptation methods.

Kapitel 4

Speech Recognition

This chapter will give a brief overview of the problem of performing speech recognition and how this is solved. In this chapter we are concerned with doing phoneme recognition as PER is used as performance metric later in the report. The extension to recognizing words and sentences is however very easy. We start by defining the problem. Given an input waveform the recognizer should output a sequence of phonemes, which corresponds to the sequence of phonemes responsible for generating the input waveform. This is shown in figure 4.1



Figur 4.1: *Illustration of the task of phoneme recognition. The waveform is arbitrary speech and does not correspond to the shown phoneme sequence.*

The most fundamental elements of modern ASR systems are the HMM and Gaussian Mixture Model (GMM) topology and the features used, thus these are described next.

4.1 HMM and GMM

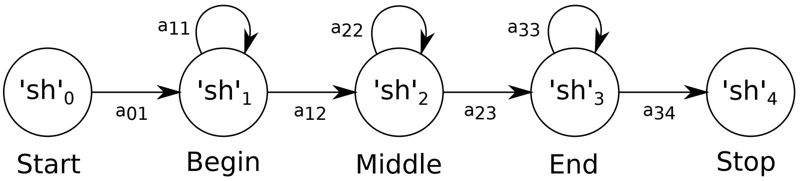
This section will go through the basics of Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs) for speech recognition.

4.1.1 HMM

HMMs have been used in the process of speech recognition for a long time [19] and is the most widely used method. HMMs are used to model the state of things, which can only be observed indirectly via another observation, hence the word hidden. We can describe a HMM using the following elements [19]

- Number of hidden states (phonemes), N.
- Transition probabilities, the probability of being in state i and transitioning into state j, i.e. $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$.
- Observation / Emission probabilities, the probability of observing a specific observation at time t, o_t, when being in state h, i.e. $b_h(o_t) = P(o_t | q_t = S_h)$. These are also refered to as likelihood probabilities.
- Initial state probabilities, the probability of beginning in state h at time $t = 1$, i.e. $\pi_h = P(q_1 = S_h)$.

In the context of speech recognition the obervations are the acoustic features (described later) which are generated from the input waveform in figure 4.1 and the hidden states are the true phonemes responsible for generating the acoustic feature. Because a phoneme can be pronounced differently and at different speeds, they are typically modelled by three emitting states and a start and end state, where it is only possible to stay in the current state or transition to the right state. This is illustrated in figure 4.2.



Figur 4.2: *Illustration of a HMM for the phoneme 'sh'.*

When only one HMM per phoneme is used it is called context-independent recognition. This can naturally be extented to context-dependent recognition, where a phone has several HMMs depending on the phone just before and after [20]. This is due to the observation that a the pronounciation of a phone depends on adjacent phones. Modelling of words (sequences of phonemes) can now be done by concatenating HMMs for different phonemes.

4.1.2 GMM

As stated in the previously subsection we need to find the observation probabilities, which in the context of speech recognition is denoted acoustic modelling. The observation / acoustic feature is a continuous vector, which will be described in more detail later. We need to make a model for each state (phoneme), which can tell how likely this state generated a given observation / acoustic feature. This PDF is typically modelled by a mixture of multivariate Gaussian distributions in the following way

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M} c_{jm} \frac{1}{\sqrt{2\pi|\mathbf{\Sigma}_{jm}|}} \exp\left((\mathbf{o}_t - \mu_{jm})^T \mathbf{\Sigma}^{-1} (\mathbf{o}_t - \mu_{jm})\right) \tag{4.1}$$

where:
M is the number of mixtures
c_{jm} is the mth mixture coefficient for the jth state
μ_{jm} is the mth mean vector of the jth state
$\mathbf{\Sigma}_{jm}$ is the mth covariance matrix for the jth state

For each state we thus need to estimate the M mixing coefficients, covariance matrices and mean vectors. This is done through training.

4.1.3 Putting it together

We have now seen how speech can be modelled using HMMs and how the observation probabilities can be modelled. The problem of recognizing a sequence of phonemes can now be solved by making a HMMs based on all the possible phonemes and then finding the most probable / likely path through it. A more formal way of stating this is: Given a sequence of t observations as $\mathbf{O} = \mathbf{o_1},\mathbf{o_2},...,\mathbf{o_t}$ find a sequence of N states / phones as $\mathbf{V} = v_1,v_2,...,v_N$, that is most probable to have generated the observation sequence. This problem is refered to as decoding and can be written as [20]

$$\hat{\mathbf{V}} = \arg\max_{\mathbf{V}\in\mathcal{L}} P(\mathbf{V}|\mathbf{O}) \tag{4.2}$$

where:
$\mathcal{L}$ is the set of all possible sequences of states / phonemes

Equation 4.2 can be restated in the following way by using Bayes' well-known rule

$$\hat{\mathbf{V}} = \arg\max_{\mathbf{V}\in\mathcal{L}} \frac{P(\mathbf{O}|\mathbf{V})P(\mathbf{V})}{P(\mathbf{O})} = \arg\max_{\mathbf{V}\in\mathcal{L}} P(\mathbf{O}|\mathbf{V})P(\mathbf{V}) \tag{4.3}$$

We see in equation 4.3 that the denominator can be dropped since this is constant for all possible $\mathbf{V}$. We see that $P(\mathbf{V})$ are the transition probabilities mentioned earlier, which is called the language model in the context of speech recognition. The likelihoods, $P(\mathbf{O}|\mathbf{W})$ can be computed using the trained acoustic models in equation 4.1. Since all possible sequences of states/phonemes have to be evaluated it is necessary to do this efficient. This is achieved by using the Viterbi algorithm [19].

4.2 Features

As depicted in figure 4.1 the input to an ASR system is an acoustic waveform. This waveform has to be split into features such that the HMM topology can be applied. The most popular features are called Mel-Frequency Cepstrum Coefficients (MFCCs) and is computed using the following steps [21, 22]

Pre-emphasis A high-pass filter is applied to put emphasis on higher frequencies.

Windowing A window is applied to split the waveform into frames with a typical duration of 25ms and an overlap of 10ms. A non-rectangular window is often chosen to avoid problem when transforming to frequency domain.

DFT Transforms the time frame into frequency domain.

Mel filter bank A non-uniform filter bank is applied and the log-energy in each band is found. The filter bank is non-uniformly spaced due to the fact that human hearing is not equally sensitive to all frequencies. The filters are spaced according to the Mel scale. A frequency response of this filter bank is shown in figure 4.3. Typically, only the first 12 coefficients are used.

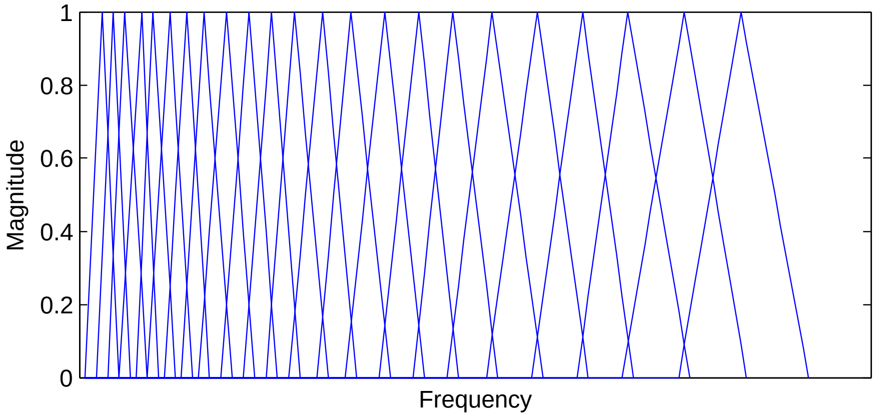
Inverse Discrete Fourier Transform (iDFT) Apply the iDFT to the log-energies mainly to make the coefficients uncorrelated, which has the advantage of making it sufficient to use diagonal matrices as covariance matrices in the GMM in equation 4.1 [21].

Energy Find the energy of the frame.

We now have a vector of 12 MFCCs and the energy adding up to 13 coefficients. To model the change in speech first- and second order differences between coefficients are also computed. The final acoustic feature thus contains $13 \cdot 3 = 39$ coeffients.

4.3 Adaptation

This section briefly describes two popular methods for adapting and normalising data such that the effects of mismatch between gender, age and acoustic environments are reduced.



Figur 4.3: *Frequency response of the Mel filter bank.*

4.3.1 VTLN - Vocal Tract Length Normalisation

The vocal tract of men, women and children all have different lengths making the spectrum af speech different [23]. This has an effect on the MFCC, which is not desirable. VTLN reduces this effect by making a frequency warping of the training data and testing data. We can state a criteria for finding this optimum frequency warping, $\hat{\alpha}$ [23]

$$\hat{\alpha} = \arg\max_{\alpha} P(\mathbf{O}_i^{\alpha}|\lambda,\mathbf{T}_i) \tag{4.4}$$

where:
$\mathbf{O}_i^{\alpha}$ is a sequence of feature vectors generated from a utterances from speaker i warped by α
λ is the parameters for the given HMM
$\mathbf{T}_i$ is the transcription of the utterances

Since a lower and upper bound on α is known due to the minimum and maximum length of the vocal tract, the optimum value is simply found by sweeping over $0.88 \leq \alpha \leq 1.12$.

4.3.2 MLLR - Maximum Likelihood Linear Regression

When there is a mismatch between training and testing data in terms of speaker-variability, acoustic environment noise etc., the performance of ASR systems is degraded [24]. This effect due to mismatch can be reduced using an affine transformation in either the feature-space (in this case the MFCC observation vectors) or in the model-space (in this case the parameters describing the individual multivariate gaussians used to describe the observation probabilities in equation 4.1). The equations given in the model-space are given by [25]

$$\bar{\mu} = \mathbf{A}\mu + \mathbf{b} \tag{4.5}$$
$$\bar{\mathbf{\Sigma}} = \mathbf{H}\mathbf{\Sigma}\mathbf{H}^T \tag{4.6}$$

where:
$\mathbf{A}$, $\mathbf{b}$ and $\mathbf{H}$ are the transformation parameters to be estimated.
$\bar{\mu}$ and $\bar{\mathbf{\Sigma}}$ are the new model parameters after adaptation.

Sometimes a constrain is forced such that $\mathbf{A} = \mathbf{H}$, which is a variant called constrained Maximum Likelihood Linear Regression (cMLLR) [25]. The general method works in the following way:

Given test data from a new speaker, some (small) amount of this is used to determine the transformation parameters such that the likelihood of the observation adaptation data is maximised [26]. This can be stated in the following way [24]

$$(\hat{\mathbf{T}},\hat{\mathbf{A}},\hat{\mathbf{b}},\hat{\mathbf{H}}) = \underset{(\mathbf{T},\mathbf{A},\mathbf{b},\mathbf{H})}{\arg\max}\ P(\mathbf{O}|\mathbf{T},\mathbf{A},\mathbf{b},\mathbf{H},\lambda)P(\mathbf{T}) \tag{4.7}$$

where:
$\hat{\mathbf{A}}$, $\hat{\mathbf{b}}$ and $\hat{\mathbf{H}}$ are the estimated transformation parameters.
$\mathbf{O}$ is the observation sequence from the adaptation data.
$\mathbf{T}$ is the state sequence.
λ is the unadapted trained model.

The adaptation can either be supervised (the true state generating the observation sequence is known) or unsupervised. If a transformation for all gaussian mixtures from all states are to be found this corresponds to a full training problem and thus requires much data. It is assumed that the same transformation can be applied to a several parameters, based on the assumption that the mismatch has effected all parameters in a similar way [25].

4.4 Kaldi

As mentioned earlier a ASR system is used to evaluate the performance of the beamforming algorithms. This section describes the system setup in this project. We use an engine called Kaldi [27]. The engine uses the MFCC as acoustic feature and models each state of a phoneme using a GMM. Some important parameters are listed in table 4.1.

Parameter	Value(s)
Number of MFCC	13
Length of feature vector	$3 \cdot 13 = 39$
Number of states per phoneme	3
Frame length	25 ms
Frame overlap	10 ms
Number of iterations for training	40

Tabel 4.1: *Settings for ASR engine.*

In the paper, where the maximum kurtosis GSC is proposed [2, 10], the recognition error rate is found for different settings of the ASR, thus the same is done here. First an experiment where no adaptation is done is performed. Kaldi can use both a context-independent and context-dependent HMM, where context-dependent means that the phoneme-model depends on the phone just before and after it [20]. The second experiment is where VTLN and MLLR (as described previously) is used.

Kapitel 5

Experimental Results

To show an improvement using the GSC with kurtosis criteria the algorithm is tested in terms of PER for different acoustic environments. This chapter is dedicated to describing the data used, stating the results and finally discuss and compare them with results achieved in other projects.

5.1 Data

The purpose of this section is to describe the data used to benchmark the performance of the array processing algorithms. In both the synthetic case and the real-world case we use data from the well-known TIMIT database. Appendix E lists the 16 sentences used which makes a total of 610 phonemes to be recognised. This is considered enough to show a performance gain if any. The synthetic reverberation is generated using a MATLAB implementation made by [28], which basically uses the image-source model to generate the desired Room Impulse Response (RIR) and then convolves this with the 16 TIMIT sentences in appendix E on page 56. Table 5.1 shows the settings for generating the synthetic data, where x_M refers to the position of the center of the microphone array.

Parameter	Value(s)
Room dimension [x,y,z]	[3, 4, 2.5]
x_M [x,y,z]	[1.5, 1, 1.3]
x_S [x,y,z]	[1.5, 2.5, 1.5]
Incident angle	90°

Tabel 5.1: *Room settings for generating synthetic data.*

The real-world data was captured at in the spring 2013 at UT Dallas, Texas. It was generated by having a speaker read the aforementioned TIMIT sentences in two rooms with different acoustic characteristica, while recording with a microphone array and a single microphone attached close to the speaker's mouth. Images and drawings of the rooms are shown in appendix F along with a table showing dimensions of the rooms, location of the speaker and microphone array. The microphone array used for recording has the same geometry as the one used to generate synthetic data.

5.2 Results

This section states the results achieved by applying the maximum kurtosis GSC on the data described in the previously section, and then comparing with the very simple DSB, as this seems to be the general approach. The PER for the clean speech and raw distant microphone are also stated. Table 5.2 shows how the beamformers are abbreviated in the rest of this chapter.

As mentioned in 4 there are context-independent and -dependent HMM modelling and furthermore different adaptation methods which can be applied in order to increase performance of the ASR system. Because of this three PERs are given for each method. We denote them in the following way: Context-independent recognition is denoted *MONO*, context-dependent recognition

Signal	Abbreviation
Clean data	CLEAN
Single-channel reverberant data from center microphone of microphone array	RAW
Delay-and-sum beamformer	DSB
Delay-and-sum beamformer + Zelinski postfiltering	DSB-PF
GSC with Kurtosis criteria	GSC-K
GSC with Kurtosis criteria + Zelinski postfiltering	GSC-K-PF
GSC with Kurtosis criteria and subspace filtering	GSC-K-SP
GSC with Kurtosis criteria and subspace filtering + Zelinski postfiltering	GSC-K-SP-PF

Tabel 5.2: *Table of abbreviations*

is denoted *TRI* and context-dependent recognition with VTLN and MLLR is denoted *VTLN & MLLR*. Besides using only the PER, histograms and spectrograms are also shown for selected settings. When looking at the gradient of the cost function given in equation 3.27 in section 3.3.2 we see that for a signal with range in amplitude the two first terms might be very small. Because of this a sufficent high step size should be chosen. Also it is therefore very important that *alpha* is not set too high forcing **w** to become all zeros. The parameters are found imperically on data not in the test set. The gradient method used to find the optimum filter weights is terminated when the kurtosis of the output has converged. Through initial experiments so change was seen using the contribution ratio described in equation 3.34, so when testing with subspace filter the dimension of the signal space is fixed to 2.

5.2.1 Synthetic data

Two different types of experiments are conducted; First where the block size used to estimate the filter weights is fixed and the reverberation time is varied along with different SNRs, and second where the reverberation time is fixed and the block size is varied. In the first experiment the two SNRs are chosen to be 20dB and 60dB, and the case of varying block size SNR is set to 60dB.

Different reverberation times

Table 5.3, 5.4 and 5.5 show the ASR results obtained for a reverberation time of 0.1s, 0.3s and 0.5s, respectively. It is first noted that even a low reverberation time of 0.1s degrades performance dramatically and that a reverberation time of 0.5s doubles the PER. As an overall trend, both the DSB and maximum kurtosis GSC increase performance for all three settings of the ASR system significantly. When comparing the DSB and the maximum kurtosis GSC, the first performs the best in all cases when postfiltering is not considered. When comparing the maximum kurtosis GSC without subspace filtering to the one with subspace filtering no difference in performance is seen, however the dimension of the filter is reduced with one dimension but at the cost of calculating the sample covariance matrix.

To see how the maximum kurtosis improves the speech signal, spectrograms and histograms are shown for the case of a reverberation time of 0.5s and a SNR of 60dB. Figure 5.1 shows the histogram and fitted distributions for (a) the clean speech, (b) the raw speech and (c) the output from the maximum kurtosis GSC. We clearly see that the clean speech is peaky and has heavy tails, which is best modelled by a gamma distribution, whereas the raw speech is better modelled as a laplace distribution as we also saw in section 3.3.2. The output of the maximum kurtosis GSC is best fitted by a gamma distribution, which indicates that the algorithm has improved this aspect as expected.

Figure 5.2 shows the spectrogram for (a) the clean speech, (b) the raw speech, (c) the output from maximum kurtosis GSC and (d) the output from the maximum kurtosis GSC with postfiltering. We first note the degradation of the from the clean speech to the raw speech and the effect of the reverberation is clearly seen. When comparing the raw speech with the output from the maximum kurtosis GSC we do see an improvement and that some reverberation has been decreased,

Method	MONO (60 / 20) [%]	TRI (60 / 20) [%]	VTLN & MLLR (60 / 20) [%]
CLEAN	34.59	33.1	29.02
RAW	48.36/59.51	44.59/55.08	39.02/48.52
DSB	46.56/53.11	41.15/45.08	35.74/39.84
DSB-PF	**46.23**/47.38	**41.15**/**43.61**	**33.44**/**36.39**
GSC-K	46.56/53.44	42.46/45.25	36.72/39.34
GSC-K-PF	46.56/**46.89**	41.31/43.93	34.43/36.72
GSC-K-SP	46.56/52.79	42.46/45.25	36.72/40.49
GSC-K-SP-PF	46.56/46.56	41.31/44.10	34.43/36.72

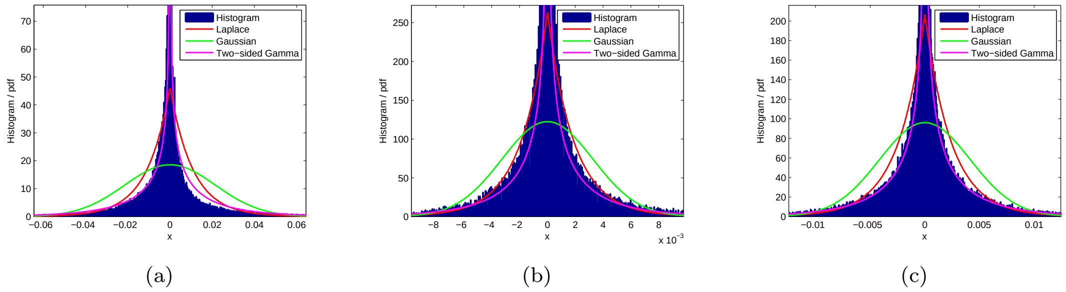
Tabel 5.3: *PER results for running ASR on synthetic data. T60* = 0.1*s, step size* = 10^{11}, $\alpha = 10^{-13}$, *block size* = 0.5*s and size of signal subspace (D)* = *2.*

Method	MONO (60 / 20) [%]	TRI (60 / 20) [%]	VTLN & MLLR (60 / 20) [%]
CLEAN	34.59	33.1	29.02
RAW	62.62/66.72	61.64/69.18	58.20/65.74
DSB	60.66/64.26	54.59/61.48	51.64/57.38
DSB-PF	56.07/**56.39**	**52.79**/**55.25**	48.20/**50.00**
GSC-K	62.62/66.23	57.38/61.97	55.25/59.02
GSC-K-PF	**55.41**/57.54	53.28/**55.25**	50.49/51.48
GSC-K-SP	62.62/66.23	57.38/61.97	55.25/59.02
GSC-K-SP-PF	**55.41**/57.54	53.28/**55.25**	50.49/51.48

Tabel 5.4: *Results for running ASR on synthetic data. T60* = 0.3*s, step size* = 10^{11}, $\alpha = 10^{-13}$, *block size* = 0.5*s and size of signal subspace (D)* = *2.*

Method	MONO (60 / 20) [%]	TRI (60 / 20) [%]	VTLN & MLLR (60 / 20) [%]
CLEAN	34.59	33.1	29.02
RAW	70.98/75.08	68.69/71.64	66.89/70.49
DSB	66.72/69.18	64.59/67.54	61.64/66.07
DSB-PF	64.43/**65.25**	**60.00**/**61.15**	61.48/65.08
GSC-K	67.05/72.30	66.23/68.03	64.10/67.87
GSC-K-PF	**63.77**/66.07	62.13/63.93	**61.15**/**62.46**
GSC-K-SP	67.05/72.30	66.23/68.03	64.10/67.87
GSC-K-SP-PF	**63,77**/66,07	62.13/63,93	**61.15**/**62.46**

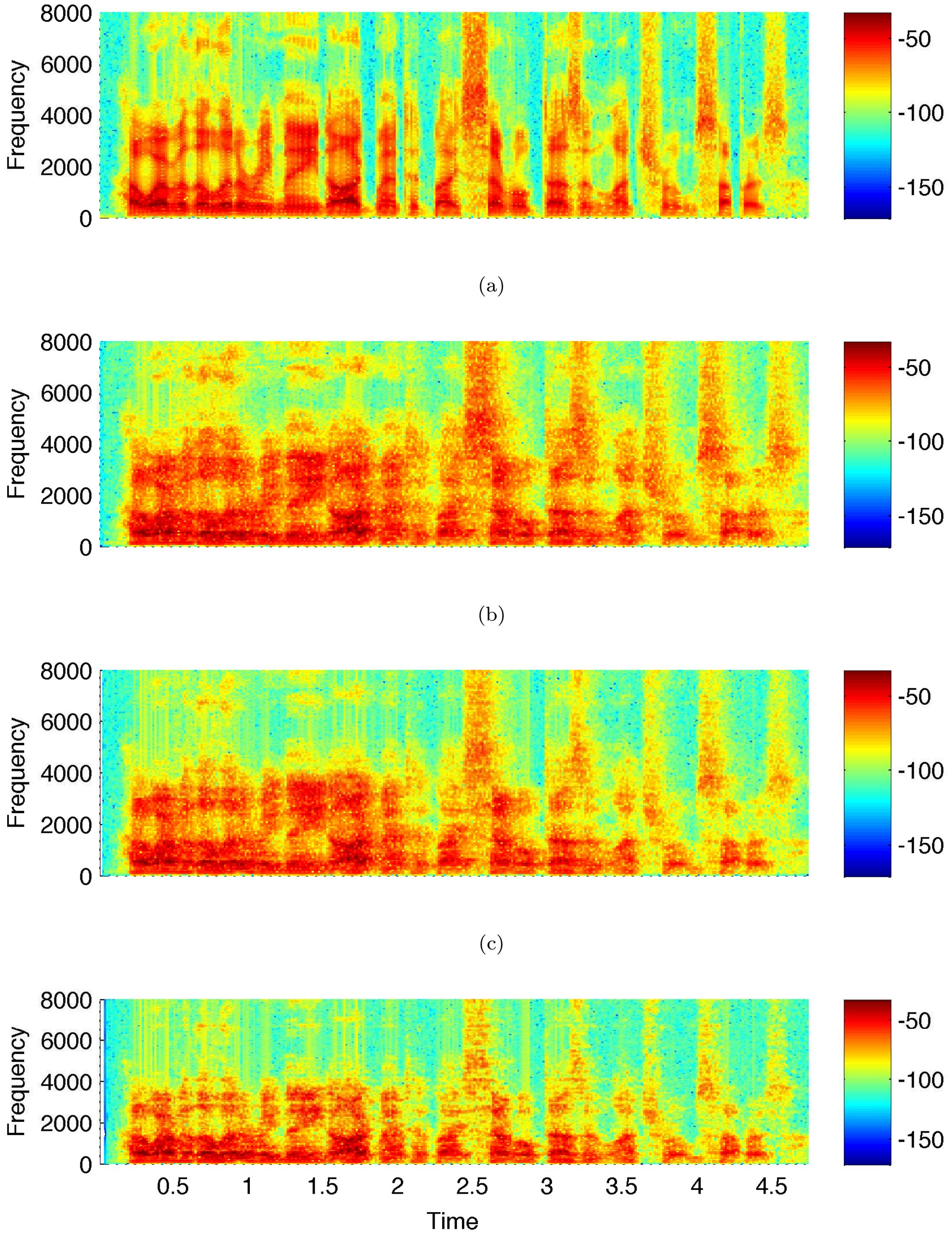
Tabel 5.5: *Results for running ASR on synthetic data. T60* = 0.5*s, step size* = 10^{11}, $\alpha = 10^{-13}$, *block size* = 0.5*s and size of signal subspace (D)* = *2.*



(a) (b) (c)

Figur 5.1: *T60* = 0.5*s: Histogram and fitted distributions for (a) the close microphone, (b) the center array-microphone and (c) the GSCK output.*

which corresponds with the fact that a small improvement is seen in the recognition performance. When looking at figure 5.2(d) we see that the postfiltering removes some noise and also helps on the reverberation, which is also seen in the error rates in table 5.5.



(a)

(b)

(c)

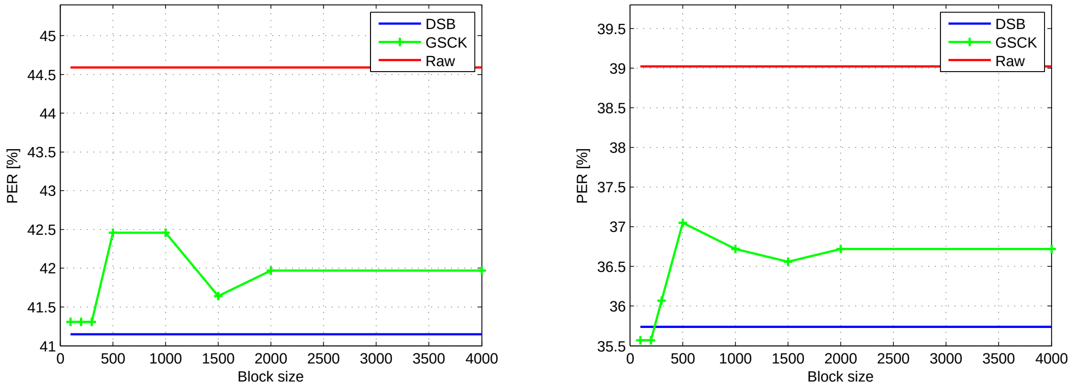
(d)

Figur 5.2: *T60 = 0.5s: Spectrograms for (a) the close microphone, (b) the center array-microphone, (c) the GSCK output and (d) the GSCK output with postfiltering. FFT-length =* 2^8 *samples and 1/8 overlap between frames.*

Different block sizes

To see how the block size for estimating the filter affects the PER, the maximum kurtosis GSC has been run with different block and the results has been evaluated. This is shown in figure 5.3

for triphone modelling and triphone modelling with VTLN and MLLR together with results for the raw speech and the DSB. We see that there does not seem to be a consistent trend in how the algorithm performs as a function of block size.



(a) (b)

Figur 5.3: *$T60 = 0.1s$: PER for different block sizes for (a) triphone modelling and (b) triphone modelling with VTLN and MLLR. The last measurement point is for block size equal to the whole utterance. Since recognition performance for the raw signal and delay-and-sum beamformer do not depend on the block size this is just plotted as a flat line for reference.*

5.2.2 Real Data

This subsection will describe the results achieved when applying the algorithms on real data collected in two rooms, an auditorium and a classroom.

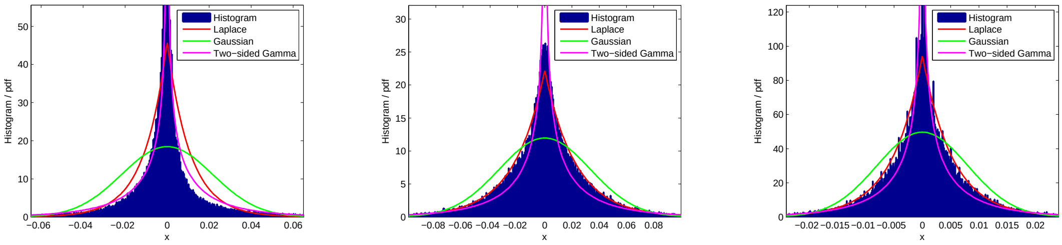
TI-auditorium

The results obtained for real data recorded in an auditorium is stated in table 5.6. In this case we see that the maximum kurtosis GSC without postfiltering almost breaks down and even degrades the performance compared to the raw signal in the case where VTLN and MLLR is used. Again DSB turns out to be best with and without postfiltering.

Method	MONO [%]	TRI [%]	VTLN & MLLR [%]
CLEAN	47.54	46.89	41.15
RAW	70.33	69.51	66.89
DSB	68.03	67.38	64.92
DSB-PF	**67.05**	**64.59**	**63.28**
GSC-K	70.16	67.16	68.85
GSC-K-PF	68.69	65.08	65.08
GSC-K-SP	70.66	67.70	68.69
GSC-K-SP-PF	68.69	65.25	64.43

Tabel 5.6: *ASR results for TI-auditorium. step size $= 10^5$, $\alpha = 10^{-7}$, block size $= 0.5s$ and size of signal subspace (D) = 2.*

Figure 5.4 shows the histograms for the clean speech, raw speech and the output from the maximum kurtosis GSC. As expected the clean signal is approximated very well by a gamme distribution, whereas both the raw speech and GSCK is almost indentical and best approximated by a laplace distribution. This corresponds well the recognition results obtained in table 5.6.



(a) (b) (c)

Figur 5.4: *TI-auditorium: Histogram and fitted distributions for (a) the close microphone, (b) the center array-microphone and (c) the output from the maximum kurtosis GSC.*

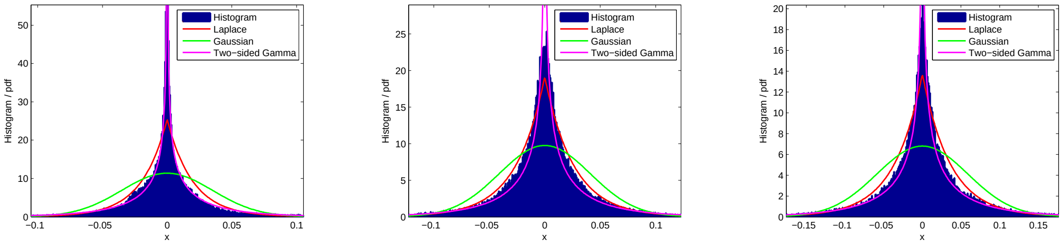
Classroom

Table 5.7 shows the results for the recordings done in a classroom. We again see that DSB performs better than maximum kurtosis GSC and that the maximum kurtosis GSC alone does not improve the PER significantly compared to the raw signal. However the combination of maximum kurtosis GSC and postfiltering performs the best. It is also noted that the subspace filter does not change anything significantly.

Method	MONO [%]	TRI [%]	VTLN & MLLR [%]
CLEAN	50.82	45.90	42.13
RAW	68.69	66.56	61.97
DSB	64.59	61.64	59.02
DSB-PF	**60.82**	61.97	57.21
GSC-K	68.03	64.43	61.80
GSC-K-PF	62.79	**60.49**	**56.72**
GSC-K-SP	68.03	64.10	62.13
GSC-K-SP-PF	63.11	60.66	57.05

Tabel 5.7: *ASR results for classroom. Step size* $= 10^6$, $\alpha = 10^{-6}$, *block size* $= 0.5s$ *and size of signal subspace (D) = 2.*

Figure 5.5 shows the histograms for the clean speech, raw speech and the output from the maximum kurtosis GSC. As expected the clean speech is modelled very well by a gamma distribution and the raw reverberant speech fit well with a laplace distribution. We do however see that no significant change is seen in the distribution by applying the maximum kurtosis GSC, which corresponds very well with obtained results from table 5.7.



(a) (b) (c)

Figur 5.5: *Classroom: Histogram and fitted distributions for (a) the close microphone, (b) the center array-microphone and (c) the output from the maximum kurtosis GSC.*

5.3 Discussion

In the previously section ASR results were obtained for the classical DSB and maximum kurtosis GSC with and without Zelinski postfiltering in the case of synthetic reverberation and recorded data. In both cases the DSB showed better performance, however in some cases the combination of maximum kurtosis GSC and postfiltering turned out to yield the best performance. The results obtained in this report contradicts the results obtained in the three reference papers, [10, 2, 29], where the maximum kurtosis algorithm performs better than DSB in the last paper, and outperforms other beamforming algorithms in the two first papers. Experiments were also conducted to see if the amount of data used to adapt the filter had an influence in the performance. In this project no clear trend was seen as opposed to [29], where the algorithm improves with more data. There are however differences between the two papers and this report. In [10] and [2] a ULA with 64 microphones is used compared to the 5-element ULA used in this report, however it is not believed that the array geometry has any impact on how the maximum kurtosis algorithm compares to DSB. The main difference between this work and [10] is the number of subbands used, where 8 subbands are used in this work, 1024 is used in [10], which is a significant difference, that could explain the difference in results. Another difference is that the ASR systems, training and test data are not the same in the two cases. It is difficult to say whether this has an influence or not. During the testing of the maximum kurtosis GSC relatively big variations $(1-2\%)$ were observed in the error rates just by changing the regularization parameter, α, in equation 3.26 and 3.27. This could indicate that the right value just has not been found, since it has to be set based on empirical results just as in [2].

Kapitel 6

Conclusion

This project has concerned the use of array processing to improve speech recognition in scenarios where reverberation is a significant problem. A reverberant signal model for a microphone array was stated along with some important statistical properties. Focus was naroowed down to investigate the proposed beamforming algorithm method in [10, 2]. The method is an extended version of the well-known GSC beamforming algorithm, where kurtosis is used as an optimization criteria, based on the observation that clean speech has a higher kurtosis than reverberant speech due to the CLT. This observation was confirmed by using histograms of clean and reverberant speech. A similar system as in [10, 2, 29] was implemented and each block was verified. The recognition software Kaldi was set up such that the algorithm could be benchmarked against the classic DSB and the general theory of HMM speech recognition was presented along with two popular adaptation methods, namely VTLN and MLLR. As test data both synthetic data and real recorded data was used. The method improved the recognition performance in almost all cases compared to the raw signal, but did not perform better than DSB. This contradicts with the results stated in [10, 2], where the method achieves good results compared to other beamforming algorithms. The main difference between the work in this project and the reference papers is the number of frequency subbands used. This will be investigated further to determine if this is the cause of the poor performance. The results also showed that Zelinski posfiltering had a positive effect on reducing the PER in almost all cases.

References

[1] J. McDonough and M. Wölfel, *Distant Speech Recognition.* John Wiley & Sons, Inc., 2009.

[2] K. Kumatani, J. McDonough, and B. Raj, "Maximum kurtosis beamforming with a subspace filter for distant speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, dec. 2011, pp. 179 –184.

[3] P. A. Naylor and N. D. Gaubitch, *Speech Dereverberation.* Springer, 2010.

[4] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing.* Springer, 2008.

[5] E. A. P. Habets, "Single- and multi-microphone speech dereverberation using spectral enhancement," Ph.D. dissertation, Eindhoven University of Technology, 2007.

[6] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Prentice Hall, 2002.

[7] H. Krim and M. Viberg, "Two decades of array signal processing research," *IEEE Signal Processing Magazine*, jul 1996.

[8] H. L. V. Trees, *Optimum Array Processing. Part IV of Detection, Estimation and Modulation Theory.* Wiley, 2002.

[9] B. Widrow, K. Duvall, R. Gooch, and W. Newman, "Signal cancellation phenomena in adaptive antennas: Causes and cures," *Antennas and Propagation, IEEE Transactions on*, vol. 30, no. 3, pp. 469 – 478, may 1982.

[10] K. Kumatani, J. McDonough, and B. Raj, "Block-wise incremental adaptation algorithm for maximum kurtosis beamforming," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*, 2011, pp. 229–232.

[11] P. P. Vaidyanathan, *Multirate Systems and Filter Banks.* Prentice Hall, 1993.

[12] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, 1st ed. John Wiley & Sons, Inc., 2001.

[13] T. Petsatodis, C. Boukis, F. Talantzis, Z.-H. Tan, and R. Prasad, "Convex combination of multiple statistical models with application to vad," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 8, pp. 2314 –2327, nov 2011.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[15] P. Stoica and R. Moses, *Introduction to Spectral Analysis.* Prentice Hall, 1997.

[16] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing - Concepts and Techniques*, 1st ed. Prentice Hall, 1993.

[17] S. Kay, *Intuitive Probability and Random Processes using MATLAB*, 1st ed. Springer, 2006.

[18] R. Zelinski, "A microphone array with adaptive post-filtering for noise reduction in reverberant rooms," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, apr 1988, pp. 2578 –2581 vol.5.

[19] L. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[20] K.-F. Lee and H.-W. Hon, “Speaker-independent phone recognition using hidden markov models,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 11, pp. 1641–1648, 1989.

[21] C. Becchetti and L. P. Ricotti, *Speech Recognition - Theory and C++ Implementation.* John Wiley & Sons, Inc., 2009.

[22] J. R. Deller, J. H. L. Hansen, and J. G. Proakis, *Discrete-Time Processing of Speech Signals.* Wiley - Interscience, 2000.

[23] L. Lee and R. Rose, “Speaker normalization using efficient frequency warping procedures,” in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 1, 1996, pp. 353–356 vol. 1.

[24] A. Sankar and C.-H. Lee, “A maximum-likelihood approach to stochastic matching for robust speech recognition,” *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 3, pp. 190–202, 1996.

[25] M. Gales, “Maximum likelihood linear transformations for hmm-based speech recognition,” *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[26] C. Leggetter and P. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models,” *Computer Speech & Language*, vol. 9, no. 2, pp. 171 – 185, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230885700101

[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The kaldi speech recognition toolkit,” in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding.* IEEE Signal Processing Society, dec 2011, IEEE Catalog No.: CFP11SRW-USB.

[28] E. Lehmann and A. Johansson, “Diffuse reverberation model for efficient image-source simulation of room impulse responses,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 6, pp. 1429–1439, 2010.

[29] K. Kumatani, J. Mcdonough, B. Rauch, P. N. Garner, W. Li, and J. Dines, “Maximum kurtosis beamforming with the generalized sidelobe canceller.”

[30] C. H. Edwards and D. E. Penney, *Calculus Early Transcendentals*, 7th ed. Prentice Hall, 2008.

Appendix

Appendix A Deriving the Linear Constrained Minimum-Variance optimum filter

The derivations in this section are primarily from [8]. The optimization problem is given by

$$\begin{aligned} &\min \mathbf{w}^H\mathbf{R}\mathbf{w} \\ &\text{subject to } \mathbf{C}^H\mathbf{w} = \mathbf{g} \end{aligned} \tag{A.1}$$

where:
$\mathbf{w} \in \mathbb{C}^{M\times 1}$
$\mathbf{R} \in \mathbb{R}^{M\times M}$ has full rank
$\mathbf{C} \in \mathbb{C}^{M\times L}$ is the constraint matrix and has full rank
$\mathbf{g} \in \mathbb{C}^{L\times 1}$

This problem is solved using the well-known method of Lagrange multipliers. The Lagrangian is given by

$$\mathcal{L}(\mathbf{w},\lambda) = \mathbf{w}^H\mathbf{R}\mathbf{w} + \lambda^H(\mathbf{C}^H\mathbf{w} - \mathbf{g}) \tag{A.2}$$

where:
λ is a vector of Lagrange multipliers.

Taking the derivative with respect to $\mathbf{w}$, setting equal to 0 and solving for $\mathbf{w}$ gives

$$\nabla\mathcal{L}(\mathbf{w},\lambda) = 2\mathbf{R}\mathbf{w} + \mathbf{C}\lambda = 0 \Rightarrow \tag{A.3}$$

$$\mathbf{w} = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{C}\lambda \tag{A.4}$$

We still need to find an expression for the lagrange multiplier. This is done by inserting equation A.4 into the equality constraint in equation A.2 and solving for λ, which yields

$$\mathbf{g} = -\frac{1}{2}\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C}\lambda \Rightarrow \tag{A.5}$$

$$\lambda = -2(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{g} \tag{A.6}$$

It is noted, that we are guaranteed that the inverse of $\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C}$ exist since both $\mathbf{C}$ and $\mathbf{R}_{xx}$ have full rank. By inserting the last expression in A.6 into the last expression of A.4 we arrive at the solution

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{C}(\mathbf{C}^H\mathbf{R}^{-1}\mathbf{C})^{-1}\mathbf{g} \tag{A.7}$$

Appendix B Derivation of the sample kurtosis gradient

First we define our cost function by

$$\mathcal{J}(\mathbf{w}) = \frac{1}{M}\sum_{k=0}^{M-1} |e(k)|^4 - \beta\left(\frac{1}{M}\sum_{k=0}^{M-1} |e(k)|^2\right)^2 - \alpha\,||\mathbf{w}||_2^2 \tag{B.1}$$

where $e(k) = d(k) - \mathbf{w}^H\mathbf{v} = d(k) - \mathbf{w}^H\mathbf{U}^H\mathbf{B}^H\mathbf{x}$ according to figure 3.9(b) on page 15.
We start by splitting the expression for convenience in the following way

$$\mathcal{J}(\mathbf{w}) = \underbrace{\frac{1}{M}\sum_{k=0}^{M-1} |e(k)|^4}_{\mathcal{J}_1(\mathbf{w})} - \underbrace{\beta\left(\frac{1}{M}\sum_{k=0}^{M-1} |e(k)|^2\right)^2}_{\mathcal{J}_2(\mathbf{w})} - \underbrace{\alpha\,||\mathbf{w}||_2^2}_{\mathcal{J}_3(\mathbf{w})} \tag{B.2}$$

and then find the derivative with respect to the filter, $\mathbf{w}$, for both terms. We ommit the time-dependency for convenience, but it is re-inserted in the final expression.

<u>$\mathcal{J}_1(\mathbf{w})$ **:**</u>

We see that this expression can be rewritten in the following way

$$\mathcal{J}_1(\mathbf{w}) = \frac{1}{M}\sum_{k=0}^{M-1} |e|^4 = \frac{1}{M}\sum_{k=0}^{M-1} (|e|^2)^2 \tag{B.3}$$

By using the well-known chain-rule the derivative is easily found

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{w}^*}\mathcal{J}_1(\mathbf{w}) &= \frac{2}{M}\sum_{k=0}^{M-1} |e|^2 \cdot \frac{\partial}{\partial \mathbf{w}^*}|e|^2 && \text{(B.4)}\\
&= \frac{2}{M}\sum_{k=0}^{M-1} |e|^2 \cdot \frac{\partial}{\partial \mathbf{w}^*}(dd^* - d\mathbf{v}^H\mathbf{w} - d^*\mathbf{w}^H\mathbf{v} + \mathbf{w}^H\mathbf{v}\mathbf{v}^H\mathbf{w}) && \text{(B.5)}\\
&= \frac{2}{M}\sum_{k=0}^{M-1} |e|^2 \cdot (-d^*\mathbf{v} + \mathbf{v}\mathbf{v}^H\mathbf{w}) && \text{(B.6)}\\
&= -\frac{2}{M}\sum_{k=0}^{M-1} |e|^2 \cdot \mathbf{v}(d^* - \mathbf{v}^H\mathbf{w}) && \text{(B.7)}\\
&= -\frac{2}{M}\sum_{k=0}^{M-1} |e|^2 \cdot \mathbf{v}e^* && \text{(B.8)}
\end{aligned}$$

<u>$\mathcal{J}_2(\mathbf{w})$ **:**</u>

Again in this term it is suitable to use the chain-rule

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{w}^*}\mathcal{J}_2(\mathbf{w}) &= 2\beta\left(\frac{1}{M}\sum_{k=0}^{M-1} |e|^2\right) \cdot \frac{\partial}{\partial \mathbf{w}^*}\left(\frac{1}{M}\sum_{k=0}^{M-1} |e|^2\right) && \text{(B.9)}\\
&= 2\beta\left(\frac{1}{M}\sum_{k=0}^{M-1} |e|^2\right) \cdot \frac{1}{M}\sum_{k=0}^{M-1} \frac{\partial}{\partial \mathbf{w}^*}|e|^2 && \text{(B.10)}
\end{aligned}$$

The last term in equation B.10 has also already been derived thus we get

$$\frac{\partial}{\partial \mathbf{w}^*}\mathcal{J}_2(\mathbf{w}) = -2\beta\left(\frac{1}{M}\sum_{k=0}^{M-1}|e|^2\right)\cdot\frac{1}{M}\sum_{k=0}^{M-1}\mathbf{v}e^* \tag{B.11}$$

$$= -2\beta\left(\frac{1}{M^2}\sum_{k=0}^{M-1}|e|^2\right)\cdot\sum_{k=0}^{M-1}\mathbf{v}e^* \tag{B.12}$$

$\underline{\mathcal{J}_3(\mathbf{w})\ \mathbf{:}}$

$$\frac{\partial}{\partial \mathbf{w}^*}\mathcal{J}_3(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}^*}\alpha\mathbf{w}^H\mathbf{w} = \alpha\mathbf{w} \tag{B.13}$$

Finally, putting the three terms back together and inserting the time-dependancy yields

$$\frac{\partial}{\partial \mathbf{w}^*}\mathcal{J}(\mathbf{w}) = -\frac{2}{M}\sum_{k=0}^{M-1}|e(k)|^2\cdot\mathbf{v}(k)e^*(k) + 2\beta\left(\frac{1}{M^2}\sum_{k=0}^{M-1}|e(k)|^2\right)\cdot\sum_{k=0}^{M-1}\mathbf{v}(k)e^*(k) - \alpha\mathbf{w} \tag{B.14}$$

Appendix C Kurtosis of random variable with standard normal distribution

This aims to show that the kurtosis of a random variable with standard normal distribution is zero, i.e.

$$\mathrm{Kurt}(X) = \mathbb{E}[X^4] - 3\mathbb{E}[X^2]^2 = 0, \text{ for } f_X(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} \tag{C.1}$$

where:
$f_X(x)$ is the PDF of the random variable X

Due to the assumption of unit variance, the expression becomes

$$\mathrm{Kurt}(X) = \mathbb{E}[X^4] - 3 = 0; \tag{C.2}$$

We thus need to show that $\mathbb{E}[X^4] = 3$.

$$\mathbb{E}[X^4] = \int_{-\infty}^{\infty} x^4 f_X(x)\,\mathrm{d}x \tag{C.3}$$

$$= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x^4 e^{\frac{-x^2}{2}}\,\mathrm{d}x \tag{C.4}$$

The method of integration by parts, which states that $\int u\mathrm{d}v = uv - \int v\mathrm{d}u$, can now be used [30, p. 521]. Setting

$$u = x^3 \rightarrow \mathrm{d}u = 3x^2\mathrm{d}x \tag{C.5}$$

$$\mathrm{d}v = xe^{\frac{-x^2}{2}} \rightarrow v = -e^{\frac{-x^2}{2}} \tag{C.6}$$

We thus get

$$\mathbb{E}[X^4] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x^4 e^{\frac{-x^2}{2}}\,\mathrm{d}x \tag{C.7}$$

$$= \frac{1}{\sqrt{2\pi}} \left(-x^3 e^{\frac{-x^2}{2}} - \int_{-\infty}^{\infty} -e^{\frac{-x^2}{2}} 3x^2 \mathrm{d}x \right) \tag{C.8}$$

$$= \frac{1}{\sqrt{2\pi}} \left(-x^3 e^{\frac{-x^2}{2}} + 3\int_{-\infty}^{\infty} e^{\frac{-x^2}{2}} x^2 \mathrm{d}x \right) \tag{C.9}$$

$$= \frac{1}{\sqrt{2\pi}} \left(\left[-x^3 e^{\frac{-x^2}{2}} + 3 \cdot \left(\sqrt{\frac{\pi}{2}} \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right) - xe^{\frac{-x^2}{2}} \right) \right]_{-\infty}^{\infty} \right) \tag{C.10}$$

We clearly see that the exponentials evaluate to zero for plus and minus infinity, i.e. $e^{\frac{-x^2}{2}} = 0|_{\pm\infty}$. We are thus left with

$$\mathbb{E}[X^4] = \frac{3}{\sqrt{2\pi}} \left[\sqrt{\frac{\pi}{2}} \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right) \right]_{-\infty}^{\infty} \tag{C.11}$$

$$= \frac{3}{\sqrt{2\pi}} \sqrt{\frac{\pi}{2}} \left(\mathrm{erf}\left(\frac{\infty}{\sqrt{2}}\right) - \mathrm{erf}\left(\frac{-\infty}{\sqrt{2}}\right) \right) \tag{C.12}$$

$$= \frac{3}{\sqrt{2\pi}} \sqrt{\frac{\pi}{2}} (1 - (-1)) \tag{C.13}$$

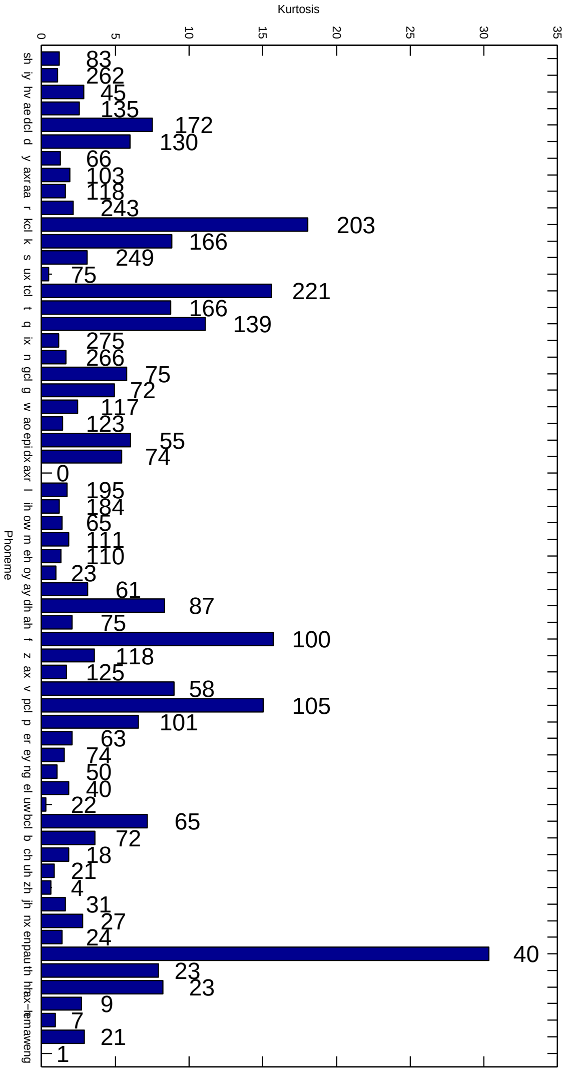
$$= \frac{3}{\sqrt{2}\sqrt{\pi}} \cdot \frac{\sqrt{\pi}}{\sqrt{2}} \cdot 2 \tag{C.14}$$

$$= 3 \tag{C.15}$$

We thus have that $\mathrm{Kurt}(X) = \mathbb{E}[X^4] - 3 = 3 - 3 = 0$.

Appendix D Estimated kurtosis for individual phonemes

Figure D.1 show the average kurtosis estimated for each phoneme in the English language. The bar plot is based on a subset of the TIMIT database. The number over each bar indicates the number of phones used to average over.



Figur D.1: *Bar chart of estimated kurtosis for individual phonemes based on data from the TIMIT database.*

Appendix E TIMIT sentences

DR1 - MDAB0

- He has never, himself, done anything for which to be hated - which of us has?
- Be excited and don't identify yourself.
- Sometimes, he coincided with my father's being at home.
- At twilight on the twelfth day we'll have Chablis.
- The bungalow was pleasantly situated near the shore.
- Are you looking for employment?
- A big goat idly ambled through the farmyard.
- Eating spinach nightly increases strength miraculously.

DR1 - MWBT0

- To many experts, this trend was inevitable.
- However, the litter remained, augmented by several dozen lunchroom suppers.
- Books are for schnooks.
- Those musicians harmonize marvelously.
- A muscular abdomen is good for your back.
- The causeway ended abruptly at the shore.
- Please take this dirty table cloth to the cleaners for me.
- The carpet cleaners shampooed our oriental rug.

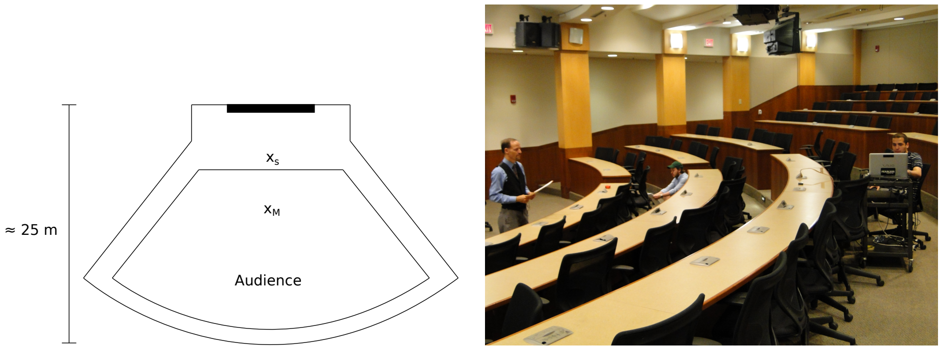
Appendix F Overview of rooms used for recording

This appendix includes picture and sketches of the rooms used to collect reverberant data.

Room	From	To	Distance [m]
TI auditorium	x_M	x_S	4
Classroom	x_M	x_S	1.2

Tabel F.1: *Table of distances between speaker and center of microphone array.*

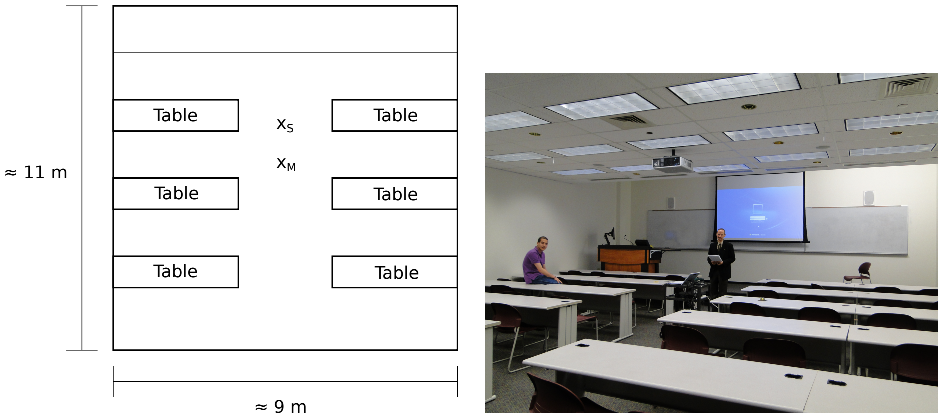
TI auditorium



(a) (b)

Figur F.1: *TI-auditorium, (a) Rough sketch with positions indicated, (b) picture taken during recordings.*

Classroom



(a) (b)

Figur F.2: *Standard class room, (a) sketch with positions indicated, (b) picture taken during recordings.*