Master's Thesis

Distant Speech Recognition Using Multiple Microphones in Noisy and Reverberant Environments

Hanna Runer

Department of Electrical and Information Technology,
Faculty of Engineering, LTH, Lund University, November 2015.

Distant Speech Recognition Using Multiple Microphones in Noisy and Reverberant Environments

Hanna Runer
ael10hru@student.lu.se

Department of Electrical and Information Technology
Lund University

Advisor: Mikael Swartling, LTH

November 27, 2015

Printed in Sweden
E-huset, Lund, 2015

“I’m sorry Dave. I’m afraid I can’t do that.”
A Masters Thesis in
Distant Speech Recognition

Quote from the movie "2001: A Space Odyssey" from 1968 by Stanley Kubrick.

Abstract

Speech is the most natural and primary way of communication for human beings. An increasing number of speech controlled, wireless, and hands-free devices and applications are appearing on the market. As the market becomes more competitive, the demands on the performance is increasing. One property that increases mobility for the user is to be able to use the application in a larger perimeter, without performance being compromised. A large challenge is to distinguish speech from noise.

This thesis addresses the issue of decreasing performance when the user speaks to the application from different distances, in environments with different noise levels and reverberation. The focus of the thesis lies on evaluating whether spatial filtering can increase, or at least keep, the performance when the speaker is located a couple of meters away from the microphones. This problem could be solved by adding multiple microphones and performing spatial filtering used to remove disturbances. Spatial filtering is a well-known technique also known as beamforming and uses the time it takes a sound wave to propagate between microphones placed at different locations. This knowledge, for a set of microphones, can be combined to emphasize a particular signal. In this case, a speech signal. Solutions to this problem was implemented on a DSP and in Matlab.

The main tests were performed in an offline manner, to which many hundreds of test words and four types of noises were recorded. The purposes of the main tests were to analyze different environmental combinations of noise, reverberations, speaker distance, and microphone set-ups. The results show that noise and reverberation severely damage the performance. Results also show that beamforming, in most environments, is a good choice, and that the performance rate gets increasingly better the more microphone utilized. Thus, beamforming is superior to no beamforming. But there is definitely room for improvements. For example, to be able to introduce flexibility in usage environments, one needs to take reverberations into account in the algorithms and perhaps introduce an adaptive beamforming algorithm.

Acknowledgments

I would like to pay a special thanks to my examiner Nedelko Grbic and my advisor Mikael Swartling for giving valuable advice when getting stuck on some problem. For giving encouragement when feeling lost and tired. For reminding me to have fun, and last but not least for believing in me. I would also like to thank my family and friends for support and encouragement, but also for putting up with me being absent and for listening at me talking way too much about my thesis. Thank you! Without all of you, this would not have been possible! ☺

Contents

List of Figures

List of Tables

List of Abbreviations

AD Analog-Digital. 37

ASR Automatic Speech Recognition. 1, 5, 6

dB deciBel. 9, 11, 53, 54

DSP Digital Signal Processor. 1, 27, 28, 34, 35, 41, 51, 55, 57, 83

DSR Distant Speech Recognition. v, 1, 5, 6, 10, 47, 83, 93

FFT Fast Fourier Transform. 37

FIR Finite Impulse Response. 37

HMM Hidden Markov Model. 93

IIR Infinite Impulse Response. 37

IWR Isolated Word Recognition. 5

LPC Linear Prediction Coding. v, 25, 83

LS Least Squares. 23, 54

SNR Signal to Noise Ratio. 11, 37, 53, 54, 57, 59, 63, 66, 71, 74, 81, 93

VAD Voice Activity Detection. 11, 39, 42, 51, 53, 54, 87, 91, 93

WER Word Error Rate. 36, 57, 58, 83, 84

Glossary

collecting Second state of the state machine. vi, 29, 39, 41, 42, 51

deletion The recognizer fail to hear a spoken word. 55, 85, 87, 88, 90

feature vector A set of reflection coefficients produced from one block of samples. 24, 29, 42

identification The recognizer decides which word in the database which is closest to the spoken word. 27, 43, 83

insertion The recognizer hear a word which was not spoken. 25, 39

listening First state of the state machine. vi, 29, 37, 42, 54

Lombard effect The tendency for people to raise their voice in noisy environments. 11

phone The acoustic realization of the basic linguistic unit *phoneme*. 6, 8, 93

processing Third state of the state machine. vi, 29, 41, 42, 55

recognizer A device which employ an ASR algorithm. 1, 2, 5, 10, 11, 29, 35, 36, 41, 45, 51, 53, 57–59, 63, 66, 69, 71, 74, 79, 81, 86, 90, 93

substitution The recognizer mistakes the spoken word for another. 35, 84, 85, 88

validation The recognizer decides if the spoken word is in the library at all. A harsher constraint than identification. 27, 43, 83

Chapter 1
Introduction

This is a master thesis report performed under the institution Electrical and Information Technology (EIT) at Faculty of Engineering (LTH), Lund University, Sweden. This report summarizes and finalizes my studies in Electrical Engineering at Lund University. In this chapter the subject of this thesis is motivated and a description of what will be processed given to the reader. Lastly, an overview of the content of all chapters in this report is presented.

1.1 Motivation and Thesis Topic

Speech is the most natural and primary way of communication for human beings. More wireless and hands-free devices and application which are speech controlled are appearing on the market [1], [2]. As the market becomes more competitive, the demands on the performance is increasing [3]. One property that increases mobility for the user is to be able to use the application in a larger perimeter, without performance being compromised [4]. A large problem is to distinguish speech from noise, which is decreasing the performance. This problem could be solved by adding multiple microphones and performing spatial filtering, which is a well known technique, used to remove disturbances. This technique is known as beamforming, and uses the time it takes for a sound wave to propagate between microphones placed at different locations. This knowledge, for a set of microphones, can be combined to emphasize a particular signal, in this case, a speech signal [5, p. 409].

This thesis addresses the issue of decreasing performance when the user speaks from different distances to the application, in environments with different noise levels and reverberation. The focus of the thesis lies on evaluating if beamforming can increase, or at least keep, the performance when the position of the speaker lies a couple of meters away from the microphones. A comparison between single and multiple microphone set-up will be answered.

As ASR devices, or more closely DSR devices, commonly are wireless, some of the implementations will be done on a DSP. A device which employ an ASR algorithm, in this case a DSP, will henceforth in this report be denoted a recognizer.

The thesis presents the fundamentals of speech, speech recognition, beamforming technique, and evaluations of continuous trial and error implementations. The thesis is concluded by evaluations of the final speech recognition algorithm using

1-4 microphones, with and without beamformer, in environments without noise, with noise, and with both noise and reverberation. In addition to self studies within the topic of speech recognition and enhancement, the course ETIN80 - "Algorithms in Signal Processors - Project Course" is taken as a part of immersing into the said topic and learn DSP programming [6].

1.2 Thesis Disposition

The disposition of the report is as following. This report is best read straight through, as the chapters are based on previous ones.

List of Abbreviations
A list of abbreviations which are used in this report, with page references to the report, can be seen in this chapter.

Glossary
There is also a glossary where a list of the used technical words in this report are explained. Page references to where in the report the words are used can also be found.

Introduction
In this chapter the purpose and disposition of the thesis is presented.

Background
To help the reader to fully comprehend the subsequent chapters, an introduction to acoustics, human perception of speech, noise echo and reverberation, and beamforming is given in this chapter.

Implementation
In this chapter the reader will be introduced to the theory, algorithms, software, hardware and the implementation strategy used in this thesis. The anechoic chamber where recordings and tests are done will also be introduced.

Single Microphone Set-Up
In this chapter the implementation of the single microphone set-up is processed, explained and evaluated.

Multiple Microphone Set-Up
This chapter introduces multiple microphones to the implementation of the recognizer is processed, explained and evaluated. The final evaluation tests are also explained.

Results
The results of the single and multiple microphone tests are presented in this chapter.

Analysis and Conclusion
Analysis of the results displayed in the previous chapter is presented in this chapter, and conclusions thereof drawn.

Recommendations
Subsequently, recommendations and tips for further research and implementation within the acrshortdsr area is given.

Bibliography
In this chapter the references used throughout the thesis are given.

Appendices
Lastly, the two appendices which are referenced to are presented. The two appendices are "The Swedish Alphabet in Graphs" and "Wiener-Hopf Equations".

Chapter 2
Background

To help the reader to fully comprehend the subsequent chapters, an introduction to acoustics, human perception of speech, noise echo and reverberation, and beamforming is given in this chapter.

2.1 Distant Speech Recognition - DSR

Automatic speech recognition is the recognition of spoken words. This is done by digitizing the speech, extracting the pattern of the spoken word and comparing said pattern to a database of stored patterns. Thus matching the spoken unknown word against a library of known words. ASR is a broad concept which includes processing both single words, IWR, and entire sentences.

When people speak of ASR devices it is generally thought of closely positioned products, such as phones or computers. Which means that the physical distance to the recognizer is, at most, at an arms length. The ASR concept also includes DSR, which is ASR applications intended to be used at greater distances than at an arms length. A DSR device enables the user more freely use the application, in therm of being hands-free and moving in a greater perimeter around the device.

2.1.1 History of Speech Signal Processing

In 1968 the Stanley Kubrick movie "2001: A Space Odyssey" premiered. In this movie an intelligent computer "HAL" understood fluently spoken speech and responded in a human sounding voice. Since this movie ASR has been a topic of great interest for the general public. But the journey for speech recognition began much earlier.

In the late 19th century research considering communication techniques using the voice was done, and in 1876 Alexander Graham Bell obtained the patent for the telephone [7]. Then, in the 1930's, a speech synthesizer VODER was invented, a device which could produce artificial human voice, and sat an important milestone in the evolution of speaking machines [8]. At Bell Laboratories in 1952 a recognizer for single-speaker isolated digit speech recognition was created [8] [9]. The research the following decades focused primarily on the applicability of ASR in commercial purposes. There was an emphasis on the systems being speaker independent,

more precisely, the focus was put on the acoustic model being able to handle the variability of different speakers [8].

In the 90's came the first successful commercial applications. Systems with vocabularies larger than the average humans vocabulary started appearing [9] and in the beginning of the 21th century, Nuance, a now world renowned corporation within the ASR field, provides Apple the software to iPhones famous digital assistant Siri [9], [10].

Simply put, the field of speech recognition have been researched for a long period of time, and is still expanding and evolving, with the usage perimeter of applications moving further and further away from the speaker.

2.1.2 Applications of ASR algorithms

Today speech is commonly used in various every day applications, and these applications predicted to grow more diverse as the field of knowledge expands [3]. Product such as phones, cars and computers are common applications for ASR. Also cheaper and smaller devices are starting to appear on the market.

The field is constantly expanding and starting to appear on the market are applications which enable the user to move around more freely without performance of the recognizer being compromised, that is, DSR recognizers. Two examples of DSR applications that are starting to appear on the market are home automation systems and discussions in conference calls being translated to text [2].

2.2 Acoustics

2.2.1 Speech

Speech is created when air flows through the vocal tract making the vocal cords vibrate. These vibrations become the fundamental tone which then resonates through the mouth and nasal cavities. The sound waves originate in the lungs as the speaker exhales. The more air per time unit that is pushed through the vocal tract, the louder the speech [11][12, p. 39]. By placing the mouth and tongue in different position in relation to each other, different sounds are created. These sounds are commonly divided into vowels and consonants. Where vowels creates the volume of the speech the consonants are the information bearers of the speech [11] [12, p. 81-131].

Each vocal tract is physiologically unique which affect the location and prominence of the spectral peaks, the *formants*, during articulation of vowels. Formants are the acoustic resonance of the vocal tract and they are given its name since they "form" the spectral peaks of a sound spectrum. It is sufficient to know the first two formants of a vowel to be able to distinguish between vowels [5, p. 34]. See table 2.1 and figure 2.1.

The basic linguistic unit, *phoneme*, is the smallest building stone of human speech and is characterized by two factors; random noise or impulse train excitation and the shape of the vocal tract. The acoustic realization of a phoneme is called a *phone* [5, p. 34].

Vowel/Formant	F_0	F_1
u	320	800
o	500	1000
å	700	1150
a	1000	1400
ö	500	1500
y	320	1650
ä	700	1800
e	500	2300
i	320	3200

Table 2.1: Table over the frequencies of the first two formants of vowels in the Swedish language.

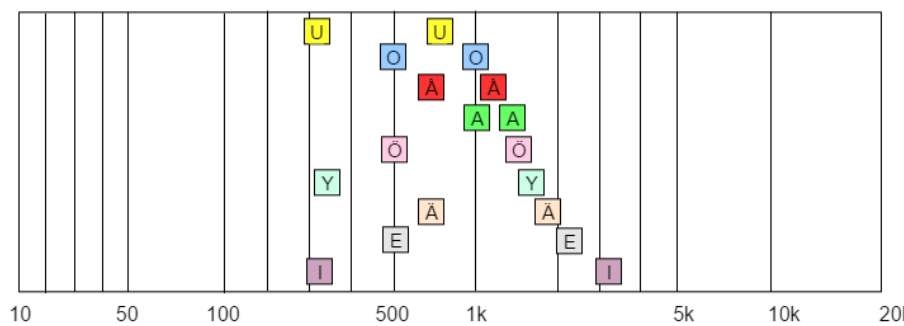


Figure 2.1: Visualization of the first two formants of vowels in the Swedish language.

All human speech can be categorized into two main categories, voiced- and unvoiced speech.

Voiced speech are characterized by its periodicity, which is a result of the vocal cords in the larynx preventing the airflow quasi-periodically. All vowels are voiced and have high energy, this is since the utterance of a vowel is synonymous with the vocal tract being open without any restriction of airflow. There also exist voiced consonants, but they have less energy as the vocal tract is restricted in some sense [5, p. 34-38].

Unvoiced speech are only consonants and it is separated from voiced speech by not causing the vocal cords to vibrate. Instead, the unvoiced speech creates a turbulent airflow through a constriction in the vocal tract, giving the phones noise-like characteristics. The different segments of the vocal tract serve as filters and strengthen and weaken frequencies, see figure 2.2.

Consonants can be divided into *pulmonic* and *non-pulmonic* speech. Pulmonic consonants are sound created by the restriction of airflow through the vocal tract from the lungs. Whereas non-pulmonic consonants are sound created without the

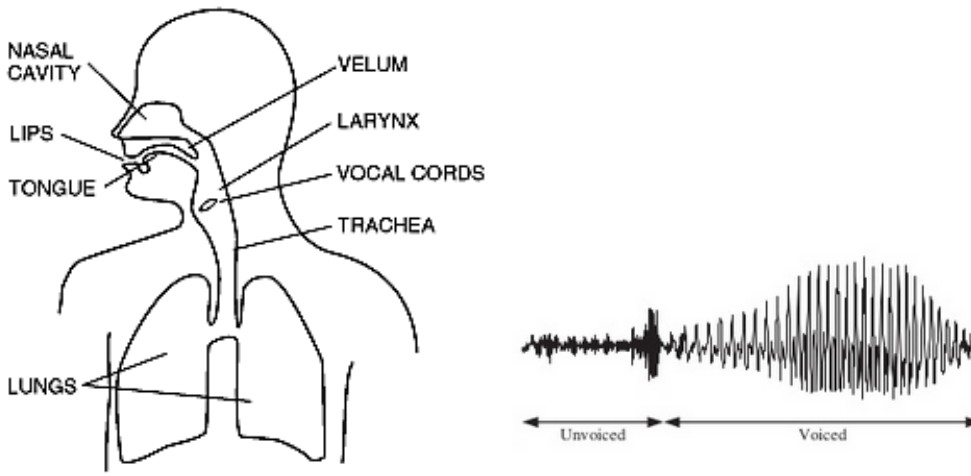


Figure 2.2: (a) Vocal tract, (b) Image illustrating segments of voiced and unvoiced speech.

	Nasals	Plosives	Fricatives	Approximants
Unvoiced		p, t, k	c, f, h, s, w	
Voiced	m, n	b, d, g	v, z	b, d, g, h, j, l, r, v, w

Table 2.2: Table over voiced and unvoiced consonants divided into the most common classifications of the Swedish language.

lungs, for example sounds such as clicks. Western language only have pulmonic consonants. Consonants can also be classified by articulation, where the most commonly occurring ones in different languages are are *nasals, plosives, fricatives* and *approximants*. Nasals are created when there is a restriction in the nasal cavity preventing outwards airflow. Plosives are consonants which are "stop" consonants. They are produced when the outwards airflow is stopped and builds up a pressure in the vocal tract and that pressure is suddenly released. Fricatives are generated when the outwards airflow is pushed through a narrow path in the vocal tract. The approximants are voiced phones which lies in between vocals and consonants. See figure 2.2 and table 2.2 [5, p. 35-38] [12, p. 131-184] [13]. In appendix A the Swedish alphabet is given in graphs.

Apart from the physiology of the vocal tract there are many factors which distinct speakers from each other. See table 2.3 [5, p. 39]. These variations can cause large recognition errors if the database which spoken words are compared to is not prepared to handle these variations [5, p. 39].

Speech is generally a non-stationary signal, but for shorter segments of 5 to 25 ms speech is considered quasi-stationary. Thus, if dividing a speech signal into

Class	Examples
speaking style	read, spontaneous, dictated, hyper articulated
voice quality	breathy, whispery, lax
speaking rate	low, normal, fast
context	conversational, public, man-machine dialogue
stress	emotional, vocal effort, cognitive load
cultural variation	native, dialect, non-native

Table 2.3: Table of different classifications of types of speech.

frames of 16 to 25 ms, frequency analysis of the segment can be performed and the features of the speech frame can be extracted [5, p. 36]. A frame representing 20 ms of speech sampled in 8000 Hz thus consists of 160 samples.

The speech apparatus produce speech in a limited spectral range and the power of the speech across the spectral range is low under 100 Hz, while 80 % of the power lies in the interval from 100 Hz to 1000 Hz. The power over 1000 Hz decides the intelligibility of the speech. This is since many of the consonants are distinguished foremost based on the spectral differences in frequencies over 1000 Hz [5, p. 43].

2.2.2 Human Perception of Speech

For a non-hearing impaired human the hearing frequency lies within the range from 20 Hz to 20000 Hz. Vowels lie within the frequency range from 250 Hz to 2000 Hz. Voiced consonants lie within from 250 Hz to 4000 Hz. The unvoiced consonants lite within the range from 125 Hz to 8000 Hz [11]. The human ear is most perceptive towards the volume of the speech, which is measured in dB. But the ear also has a complex mechanism for perceiving the pitch. *Pitch* is a perceptional impression of sound, who is physically represented by a fundamental frequency. The pitch is an audible feeling which gives a measure of the frequency in sounds and are referred to being higher or lower compared to some other pitch.

The ear apprehends the difference in pitch of two pairs of frequencies to be equal if the ratio between the two pair are equal. That is,

$$\frac{f_{a1}}{f_{a2}} = \frac{f_{b1}}{f_{b2}} \qquad \rightarrow \qquad \text{pitch}(f_{a1} - f_{a2}) = \text{pitch}(f_{b1} - f_{b2}). \tag{2.1}$$

But if the difference in frequency is the same for the two pairs, the pitch in not perceived to be equal. For example, the difference in pitch between the frequencies 100 and 125 Hz are perceived as much greater by the human ear, than the difference between 1000 and 1025 Hz. That is,

$$\frac{f_{a1}}{f_{a2}} \neq \frac{f_{b1}}{f_{b2}} \qquad \rightarrow \qquad \text{pitch}(f_{a1} - f_{a2}) \neq \text{pitch}(f_{b1} - f_{b2}). \tag{2.2}$$

This can be explained by the definition of an octave. An *octave* is the interval between one pitch and another with half or double its fundamental frequency. That is, for a low frequency range from 100 Hz to 125 Hz the interval between two

pitches are shorter than for higher frequency ranges such as the range from 1000 Hz to 1025 Hz and are thus perceived as a greater difference.

The DSR recognizer extracts the features of the speech by mimicking the human ear in the sense that it samples the speech and form a representation of the main characteristics of the speech.

2.2.3 Noise, Echo and Reverberation

The robustness of a Distant Speech Recognition (DSR) recognizer is very dependent on the disturbances in the recording. When speech is traveling through the acoustic environment, the distance between the microphone and the speaker is a vulnerable path on which several types of disturbances can affect the quality of the recording.

During this distance, numerous unwanted transformations of the speech is created and then recorded by the microphone. These transformations include ambient noise, reverberation and echoes [5, p. 47].

Ambient noise, or background noise, is additive unwanted sounds which are either stationary of non-stationary. Noise is a stochastic process, or a random process, which is a signal that cannot be recreated at will as it is purely random. A stochastic process, noise, is divided into two categories, stationary and non-stationary noise. *Stationary* noise has characteristics that do not change over longer periods of time, and the characteristics can therefore be taken into account when having a signal mixed with stationary noise. As for *non-stationary* noise, the characteristics are changing during short periods of time and the characteristics are therefore very hard to model for. The ideal noise is stationary since it is desirable to remove all disturbances from the true desired signal, and to be able to do this the noise must be modeled for. When speaking of noise it is often associated to be Gaussian noise, which is a purely random, normal distributed noise process. White Gaussian noise has the statistical properties of being independent and identically distributed, that is, white Gaussian noise is stationary [5, p. 47-48] [14, p. 48, 58] [15, p. 102].

All stationary noise, like white noise, has the property of non-varying statistics which are nice properties to work with. But most noise cannot be entirely stationary in real applications. Real noise can only resemble being stationary if looking at the noise in a small enough time window. Then, for that short segment of time, the statistics can be considered constant, and the characteristics of the signal can be modeled for. Examples of stationary noise is computer fans and air conditioning. Non-stationary noise can for example be door slams, hard drives, music and printers [5, p. 47-48] [14, p. 48, 58] [15, p. 95-102].
A signal with noise and the desired signal can simply be split, thus removing the noise from the recorded signal, if the noise and the desired signal is not off of the same frequency range. The basic approach to remove white noise is low pass filtering of the mixed signal. As white noise often is of high frequency characteristics, it can be removed. Low pass filtering can also be applied as speech does not go below 100 Hz and disturbances of the electrical outlets of 50 Hz are a common disturbance source. If the desired signal and the noise both lie closely to each other frequency wise, the task of separating the two becomes significantly more difficult.

There exist numerous varieties of noise reduction apart from high, respectively, low pass filtering. Some examples are beamforming, VAD, noise estimations and many others [16].

Echoes and reverberations are closely related to each other. An echo is a single reflection of a sound source, arriving after some delay after the direct sound. If the delay is short enough, the human ear cannot perceive any difference. But if the delay is longer than 0.1 seconds it is noticeable. Reverberations are multiple echoes from one single sound source, joining the direct sound after different, closely separated delays. This makes the reverberations indistinguishable from each other. There are three categories in which sound reaching the ear or a microphone can be divided into: direct wave, early reflections and late reflections [5, p. 47-49] [14, p. 340-342].

Direct wave is the sound wave that reaches the microphone directly, without being reflected off of the surrounding objects before reaching the microphone. *Early reflections* are waves that have been reflected off of surrounding objects and reaching the microphone 50 to 100 ms after the direct wave. *Late reflections* are reflected waves reaching the microphone so closely apart that they become indistinguishable [5, p. 47-49].

The space in which the reverberations are created in determine the number of reflections $\mathcal{N}$. In

$$\mathcal{N} = \frac{V_{sphere}}{V_{room}} = \frac{4\pi}{3}\frac{r^3}{V}, \tag{2.3}$$

where the number of reflections are created from the volume sphere and the volume of the enclosed space in which the source of the sound is described [5, p. 47-49]. The material of the surrounding walls also contribute to the number of reflections as some materials are more absorbent of acoustic waves than others.

The problem with echoes and reverberations is that they are highly correlated with the desired original signal and are therefore hard to remove once added to the desired signal. The results of having these disturbances are a severe performance degradation of the recognizer. There exists methods of removing reverberations, but they require knowledge of the room characteristics, the speaker and microphone location [5, p. 49] [17] [18].

In order to measure the quality of the recording a measure called SNR is commonly used. SNR measures the ratio of the energies of the desired signal, P_{signal}, and the additive and reverberant disturbances, P_{noise}. This ratio is presented in the logarithmic scale dB

$$SNR \triangleq 10 \ \log_{10} \frac{P_{signal}}{P_{noise}}, \tag{2.4}$$

where a high value of SNR indicates that there is more speech than noise in the signal, which is a desired scenario [5, p. 51].

Apart from acoustic environment adding disturbances to the desired signal, speakers tend to raise their voice in environments with high noise levels, which is an effect called the Lombard effect. This reflex cause a variability in speech, see table 2.3, which if not accounted for in algorithms and the database, results in a degradation in recognition rate [5, p. 53-54].

2.3 Beamforming

Beamforming, also known as spatial filtering, is a type of array signal processing which use the signals of several sensors to extract the desired information which is the content of a spatially propagating signal from a certain direction. The technique consists of algorithms which combines signals from multiple sensors and determines the sensor weights to emphasize a desired source and suppress interference from other directions. Using the weights, one can implement a sought after shaping, or steering, of the array directivity pattern. The content of the desired signal may be a message, as in communication applications, or simply the existence of the signal, such as radar or sonar. One creates linear combination of the signals of all sensors using weights so that one can examine the signal arriving from different angles. This technique is called beamforming since the weighing of the signal emphasizes signal of a particular direction while attenuating those from other directions which can be thought of as forming a beam.

Beamforming can be used in both receiving and transmitting signal from multiple sensors. The sensors can be microphones or antennas. In this thesis the receiver case is considered, and is implemented by using four microphones receiving speech signals and outputting one single output signal. Beamforming will in this thesis help to remove noise and increase the intelligibility of speech [5, p. 409] [14, p. 31, 131] [19, p. 631].

Following in this sub chapter the basics of beamforming technique is explained and some examples of conventional beamformers is given. In the explanations the coordinate system which will be used can be seen in figure 2.3. The figure shows the relationship between the spherical coordinates (r, θ, ϕ) and the Cartesian coordinates (x, y, z). The spherical coordinates describes the propagation of sound waves through space. Where $r > 0$ is the *radius/range*, the *polar angle* θ takes values in the range $0 \leq \theta \leq \pi$, and the *azimuth* takes values in the range $0 \leq \phi \leq 2\pi$ [5, p. 412-413]. The plane wave a in the figure is propagating in the direction and can be described as

$$\mathbf{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} -\sin\theta\cos\phi \\ -\sin\theta\sin\phi \\ -\cos\theta \end{bmatrix}. \tag{2.5}$$

2.3.1 Microphone arrays

Consider an arbitrary array consisting of N microphones. If the locations of the microphones are denoted $m_n = 0, 1, ..., N-1$, they produce a set of signal denoted by the vector

$$\mathbf{f}(\mathbf{t}, \mathbf{m}) = \begin{bmatrix} f(t, m_0) \\ f(t, m_1) \\ \vdots \\ f(t, m_{N-1}) \end{bmatrix}, \tag{2.6}$$

where t is the time in the continuous time domain [5, p. 411].

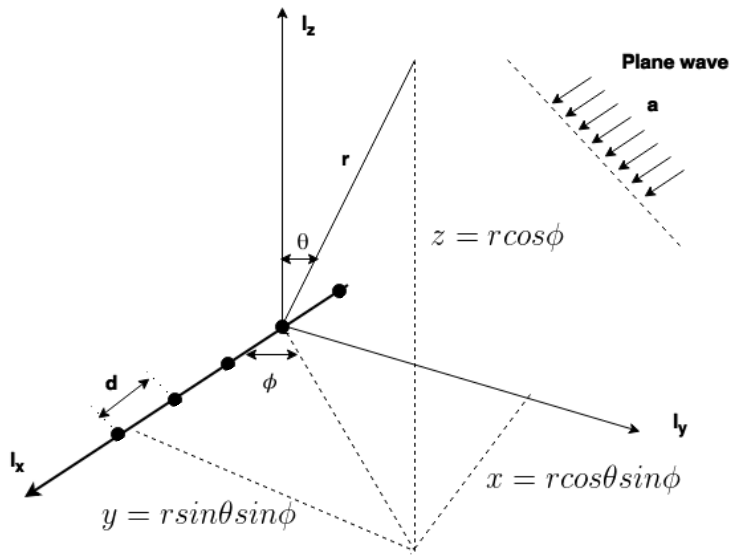


Figure 2.3: The angles in the spherical coordinates and Cartesian coordinates used in the Beamforming sub chapter.

If using more than two microphones, it is possible to arrange the microphones in different formations. In this thesis the sensors are confined to lie in the same plane in a linear formation. The microphones are placed equidistantly to each other

$$m_0 - m_1 = \ldots = m_{N-2} - m_{N-1} = \ d, \tag{2.7}$$

where the interelement spacing d between the sensors is the spatial sampling interval, which is the inverse of the spatial sampling frequency. The distance d can be seen in figure 2.3. To avoid spatial aliasing it is important confine the interelement distant d to

$$d \leq \frac{\lambda}{2}, \tag{2.8}$$

where λ is the length of the shortest sound wave, which corresponds to the highest frequency being sampled

$$\lambda_{min} = \frac{c}{\frac{f_s}{2}} = \frac{343}{4000} \approx 8.6 \text{ cm } \rightarrow \ d \leq 4.3 \text{ cm}, \tag{2.9}$$

where c is the velocity of sound propagating through air and f_s is the sampling frequency, which is this thesis is 8000 Hz. When following the constraint in equation 2.8 one allows the array to be steered over the full plane, $-90° \leq \phi \leq 90°$, which is over the entire half plane [5, p. 424] [19, p. 630].

2.3.2 Sound Wave Propagation

As previously mentioned, speech is created when air flows through the vocal tract making the vocal cords vibrate. These vibrations are periodical perturbations of the pressure in a gas, that is, sound waves traveling through air. If one assumes that the gas is of non-viscous and a homogeneous character the sound waves can be described as

$$\nabla^2 x(t,r) - \frac{1}{c^2} \cdot \frac{\delta^2 x(t,r)}{\delta t^2} = 0, \tag{2.10}$$

where c is the velocity of sound, and $x(t,r)$ is the sound pressure at the coordinates $r = [x\ y\ z]^T$ and the time t. This equation is valid for both planar and spherical waves and is, for planar waves, solved as

$$x(t,r) = Ae^{j(\omega t - k \cdot r)}, \tag{2.11}$$

where A is the amplitude of the wave, $\omega = 2\pi f$ is the angular frequency with the f being the frequency of the wave. The wave number, k, can be defined as

$$k = \frac{2\pi}{\lambda} \cdot a, \tag{2.12}$$

where a is the planar wave seen in figure 2.3. Rewritten the wave number k becomes

$$\mathbf{k}(\phi,\theta) = -\frac{2\pi}{\lambda}[\sin(\theta)\cos(\phi)\ \ \sin(\theta)\sin(\phi)\ \ \cos(\theta)]^T. \tag{2.13}$$

In this thesis only planar waves are considered, as the application is distant speech sources, thus the spherical wave equation solution is omitted. The reason for considering planar waves instead of spherical is that an omni-directional sound source, emitting spherical sound waves at the wave length λ, appear to emit planar waves after some distance. In figure 2.4 this is shown, where the full lines are wave fronts and the dotted lines are subsequent waves, separated by the wavelength λ. One can consider this to be true if the sound source holds the constraint

$$|r| > \frac{2(Nd\cdot)^2}{\lambda}, \tag{2.14}$$

where $|r|$ is the distance between the source and the sensors, d is the interelement spacing on the microphone array, and N is the number of microphones. Thus, in this thesis, $|r|$ is constrained to

$$|r| > \frac{2(4 \cdot 4.3)^2}{4000} \ \rightarrow\ |r| > 14.8 \text{ cm}, \tag{2.15}$$

which holds, as the tests in this thesis are performed on at shortest, at distance of 1 meter [5, p.30, 412-413] [20] [19, p. 624].

Following, in this chapter, narrowband beamforming will be introduced, which then is extended to the wideband beamforming case.

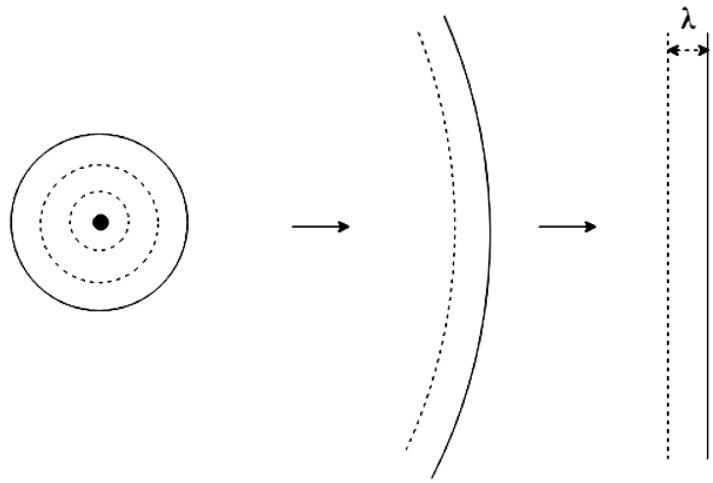


Figure 2.4: An omni-directional sound source emitting spherical sound waves, which appear as planar waves after some distance.

2.3.3 Narrowband Beamforming

Beamforming relies on wave interference to create a directional dependent gain towards the region of interest. Sound waves consist of multiple sinusoids. If two waves have different frequencies they cannot amplify or dampen each other consistently, it is therefore natural to study a narrowband signal, containing only one frequency, being processed by a microphone array [20].

Time delay

When a plane wave arrives at linear microphone array with two microphones, equidistantly spaced by d, at the propagation angle ϕ and the angle θ being fixated at 90°, there is a time delay τ_n between the arrival time of wave reaching the first and second microphone, see figure 2.5. This delay is given by

$$\tau_n = \frac{D}{c}, \tag{2.16}$$

where D is the additional distance the wave travels before reaching the second microphone and c is the velocity of sound. The distance D if θ is fixed is

$$D = d \cdot \cos(\phi). \tag{2.17}$$

The more general case, when θ is not fixed, the distance is given by using scalar projection and defining a unit vector

$$\hat{\mathbf{k}}(\phi, \theta) = [\sin(\theta)\cos(\phi) \;\; \sin(\theta)\sin(\phi) \;\; \cos(\theta)]^T, \tag{2.18}$$

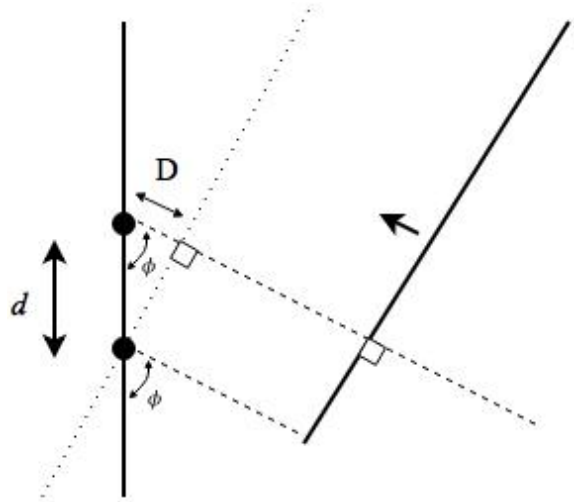


Figure 2.5: Illustration of a plane wave arriving at angle ϕ to a linear array with two microphones.

which points in the direction of the propagating sound wave. One also needs information about the position of the microphones, m_n, then the distance is given by

$$D = m \cdot \hat{\mathbf{k}}(\phi, \theta). \tag{2.19}$$

It should be noted that the delays between microphones are not dependent on rotation of the direction of the incident wave [20].

Directional Gain

Consider a microphone array with N microphones, and a continuous time sinusoid signal $s(t) = e^{iwt}$ with the frequency $f = \frac{\omega}{2\pi}$ and the sound wave propagation direction (ϕ, θ). Then, the directional gain of the microphone array can be analyzed by observing the output of the beamformer, with a complex sinusoid being received as a plane wave. The vector of the received microphone array signals $x(t)$ can then be expressed as

$$\mathbf{x(t)} = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_N(t) \end{bmatrix} = \begin{bmatrix} s(t-\tau_1) \\ s(t-\tau_2) \\ \vdots \\ s(t-\tau_N) \end{bmatrix} = \begin{bmatrix} e^{i\omega(t-\tau_1)} \\ e^{i\omega(t-\tau_2)} \\ \vdots \\ e^{i\omega(t-\tau_N)} \end{bmatrix} = e^{i\omega t} \overbrace{\begin{bmatrix} e^{-i\omega t\tau_1} \\ e^{-i\omega t\tau_2} \\ \vdots \\ e^{-i\omega t\tau_N} \end{bmatrix}}^{\mathbf{d}(\omega,\phi,\theta)}, \tag{2.20}$$

where τ_n is the time delay to the microphone number m_n relative some reference point. Which can be given by

$$\tau_n(\phi,\theta) = \frac{D_n(\phi,\theta)}{c} = \frac{m_n \cdot \hat{\mathbf{k}}(\phi,\theta)}{c}, \tag{2.21}$$

where the vector $\mathbf{d}(\omega,\phi,\theta)$ often is called the steering vector and contains information about frequency dependent delay for a given array. The directional gain of the microphone array is given by weighing the microphone signals with their respective weights

$$\mathbf{w} = [w_1 \; w_2 \; \ldots \; w_N]^T, \tag{2.22}$$

and then being summed. The magnitude of the output of the beamformer is then calculated as

$$|y(t)| = |\sum_{n=1}^{N} w_n^* x_n(t)| = |\mathbf{w}^H \mathbf{x}(\mathbf{t})| = \overbrace{|\mathbf{w}^H \mathbf{d}(\omega,\phi,\theta)|}^{P(\omega,\phi,\theta)} \cdot |s(t)|, \tag{2.23}$$

where $P(\omega,\phi,\theta)$ is the directional gain of the signal [20] [21, p. 4-5].

Beampattern

A plot of the function in equation 2.23 is called beampattern. The beampattern of a linear array with equidistantly spaces microphones, with the interelement spacing $d = 0.04$ m, the angle θ is fixated at $0°$ and the filter weights $\omega = \frac{1}{4}$ can be seen in figure 2.6. Beampatterns are symmetric around the angle $\phi = 180°$ because of symmetry around x-axis, thus, only the region $[0°, 180°]$ needs to be considered. The figure shows that the gain is close to zero around $\pm 33°$, which means that frequencies of a signal arriving from these directions will be next to completely canceled. At the direction of arrival $\phi = 0°$ the signal will be completely let through without being attenuated. The interelement distance is, as previously mentioned, important to keep under $d \leq 4.3$ cm to avoid spatial aliasing. But if choosing the distance to small multiple microphones appear as one microphone in beamforming techniques. In figure 2.7 one can see the differences using $d = 0.01$ m which is too small, $d = 0.04$ m which is good and $d = 0.1$ m which is too large [20].

Delay-and-Sum Beamformer

One common narrowband beamformer is the delay-and-sum beamformer. The technique consists of the alignment of the microphones to compensate for the time delays introduced by the different paths the sound waves take from the source to the microphones, and combining these signals to remove noise. See figure 2.8 which shows an implementation of a delay-and-sum beamformer in time domain.

2.3.4 Wideband Beamforming

If the desired signal contains frequencies in a great range, narrowband beamforming is not suitable. This can be shown as follows.

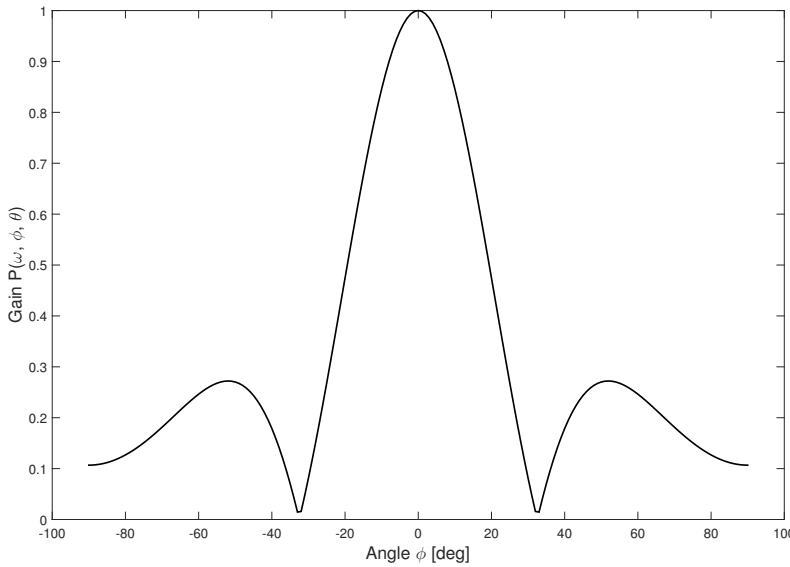


Figure 2.6: Two dimensional beampattern at frequency $f = 4000$ Hz and with the interelement distance $d = 0.04$ meters.

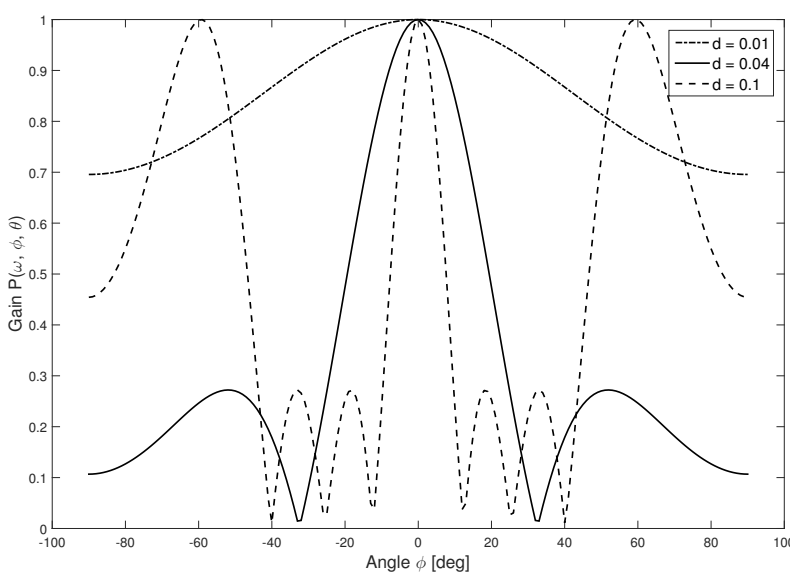


Figure 2.7: Two dimensional beampatterns for three different interelement distances, $d = 0.01, 0.04$ and 0.1 meters, at frequency $f = 4000$ Hz.

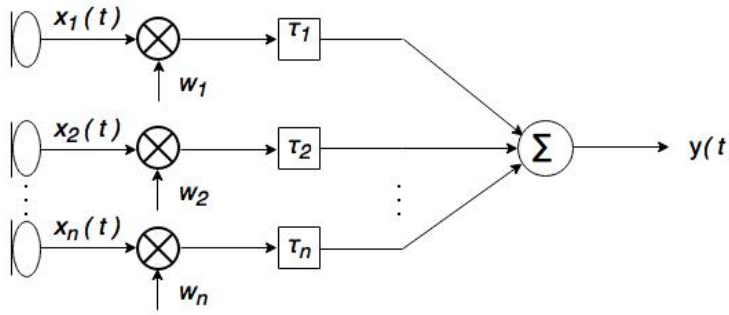


Figure 2.8: Delay-and-sum beamformer in a time domain implementation.

If there are M microphones receiving signals $\mathbf{x_m}(\mathbf{t}), m = 0, 1, ..., M-1$ from the respective directions $\theta_m, m = 0, 1, ..., M-1$. If the first signal, $x_0(t)$, is the desired signal and the others are disturbances. Then the steering vector $\mathbf{d_m}(\omega, \theta)$ is given by

$$\mathbf{d_m}(\omega, \theta) = [1 \; e^{-i\omega\tau_1(\theta_m)} \; \dots \; e^{-i\omega\tau_1(\theta_m)}]^T. \tag{2.24}$$

An ideal beamformer aims to create a fixed response to the desired signal and zero response to disturbing signals. Note that to simplify the following explanation, the effects of noise is omitted. This requirement can be expressed as

$$\overbrace{\begin{bmatrix} 1 & e^{-i\omega\tau_1(\theta_0)} & \dots & e^{-i\omega\tau_{M-1}(\theta_0)} \\ 1 & e^{-i\omega\tau_1(\theta_1)} & \dots & e^{-i\omega\tau_{M-1}(\theta_1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-i\omega\tau_1(\theta_{M-1})} & \dots & e^{-i\omega\tau_{M-1}(\theta_{M-1})} \end{bmatrix}}^{A} \cdot \begin{bmatrix} w_0^* \\ w_1^* \\ \vdots \\ w_{M-1}^* \end{bmatrix} = \begin{bmatrix} constant \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{2.25}$$

As long as matrix A has full rank, a set of weights which cancel the interfering signals can always be found. The exact value of the weights are dependent on the frequency and direction of arrival, θ, of the signal. Signals used in wideband beamforming has, as previously mentioned, a great number of different frequencies. Thus, the values of the weights should be different for different frequencies. That is, the wideband beamformer have a *frequency dependent gain*. The weight vector can be described as

$$\mathbf{w}(\omega) = [w_0(\omega) \; w_1(\omega) \; \dots \; w_{M-1}(\omega)]^T. \tag{2.26}$$

This is the reason why the narrowband beamforming structure with a single constant coefficient for each received sensor signal will not work effectively in a wideband environment [21]. Therefore, in this thesis, wideband beamforming is used.

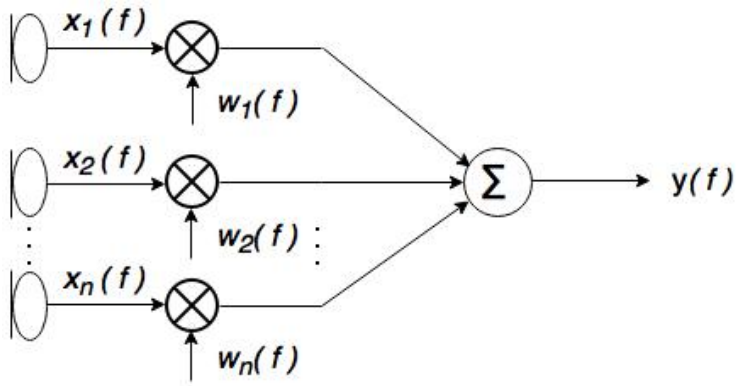


Figure 2.9: Filter-and-sum beamformer in a frequency domain implementation.

Filter-and-Sum Beamformer

The filter-and-sum beamformer is a generalized version of the delay-and-sum beamformer, with the difference that different techniques have been applied to implement the filters. For the commonly used filter-and-sum wideband beamformer, both the amplitude and the phase of the complex weights are frequency dependent. This results in a filtering operation of each array element in the input signal before the filtered microphone input signals are summed. See figure 2.9 for an illustration of the filter-and-sum beamformer. If using the weight vector

$$\mathbf{w}(\omega) = [w_{11}(\omega) \dots w_{N1}\ w_{12}(\omega) \dots w_{N2}\ w_{1M}(\omega) \dots w_{NM}]^T, \tag{2.27}$$

where M is the number of filter taps and N denotes the number of sensors. The M microphone input signals are described as

$$\mathbf{x}(\omega) = [x_{11}(\omega) \dots x_{N1}\ x_{12}(\omega) \dots x_{N2}\ x_{1M}(\omega) \dots x_{NM}]^T. \tag{2.28}$$

The output of the beamformer in frequency domain can be described as

$$y(\omega) = \mathbf{w}(\omega)^H \mathbf{x}(\omega). \tag{2.29}$$

Or with convolution in discrete time domain expressed as

$$y(k) = \sum_{m=1}^{M} \sum_{n=1}^{N} w_{mn} x_{mn}(k). \tag{2.30}$$

The response of the beamformer is given by

$$P(\omega, \theta) = \mathbf{w}(\omega)^H d_m(\omega, \theta) \tag{2.31}$$

where $d_m(\omega, \theta)$ is the steering vector and w is the filter coefficients.

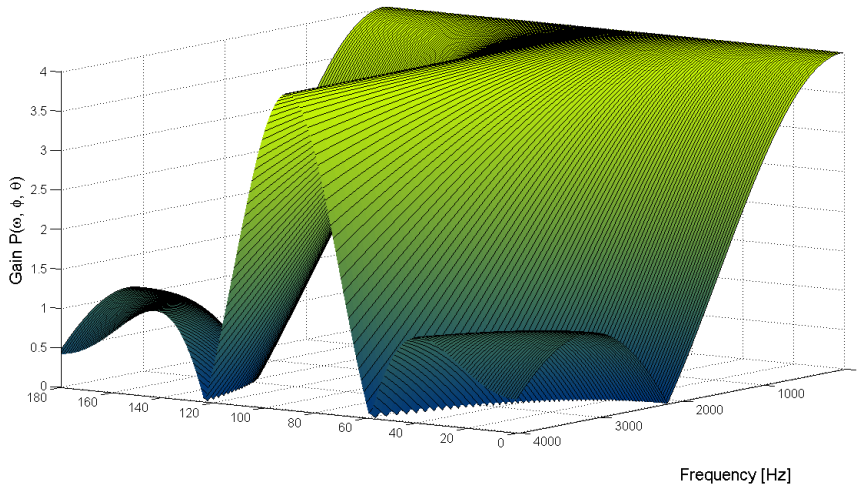


Figure 2.10: A three dimensional beampattern.

Beampattern

As wideband beamforming manipulates multiple frequencies, three dimensional beampattern plots are often used. These types of plots shows how the beampattern changes with frequencies. See figure 2.10, where ϕ is varied from $0°$ to $180°$ and the frequency is varied from 0 to 8000 Hz. The 3D plot can only be done when either ϕ or θ is kept constant [20]. In this plot the interelement distance is $d = 0.04$ m, $\omega = \frac{1}{4}$ and the angle $\theta = 0°$.

Chapter 3

Implementation

In this chapter the reader will be introduced to the theory, algorithms, software, hardware and the implementation strategy used in this thesis. The anechoic chamber where the recordings and tests are done will also be introduced.

3.1 Least Squares Wideband Beamformer

In this thesis a wideband filter-and-sum beamformer is used, and the filter coefficients are calculated by using a Least Squares technique. The LS algorithm aims to minimize the error $e(t)$ by finding a suitable model described by optimal filter coefficients

$$w_{opt} = \arg\min_{w} \overbrace{\mathbb{E}\left[|y(t) - s_n(t)^2|\right]}^{||e(t)||^2}, \tag{3.1}$$

where w_{opt} is the optimal filter which minimize the difference, or error, between the output of the beamformer $y(t)$ and the desired signal $s_n(t), n = 1, ..., N$ by finding a set of filter coefficients w. The microphone which receive the desired signal is denoted by n, the samples i and $\mathbb{E}[\cdot]$ denotes the expectation operator. If the interference signals are denoted $x_n(t), n = 1, ..., N$ the output of the beamformer is given by

$$y(t) = \sum_{n=1}^{N} w_n^H (x_n(t) + s_n(t)). \tag{3.2}$$

The optimal filter coefficients can also be described as

$$\mathbf{w_{opt}} = [\mathbf{R_{ss}} + \mathbf{R_{tt}}]^{-1} r_s, \tag{3.3}$$

where R_{ss} and R_{tt} are auto-covariance matrices and they are defined in a similar way. R_{tt} consists of correlation estimates of the interference signals. R_{ss} consists of correlation estimates of the desired signal and is defined as

$$R_{ss} = \begin{bmatrix} R_{s_1 s_1} & R_{s_1 s_2} & \dots & R_{s_1 s_N} \\ R_{s_2 s_1} & R_{s_2 s_2} & \dots & R_{s_2 s_N} \\ \vdots & \vdots & \ddots & \vdots \\ R_{s_N s_1} & R_{s_N s_2} & \dots & R_{s_N s_N} \end{bmatrix} = \mathbb{E}[ss^H], \tag{3.4}$$

where each element in the R_{ss} matrix is

$$\mathbf{R_{s_n s_j}} = \begin{bmatrix} r_{s_n s_j}(0) & r_{s_n s_j}(1) & \dots & r_{s_n s_j}(L-1) \\ r^*_{s_n s_j}(1) & r^*_{s_n s_j}(0) & \dots & r^*_{s_n s_j}(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*_{s_n s_j}(L-1) & r^*_{s_n s_j}(L-2) & \dots & r^*_{s_n s_j}(0) \end{bmatrix}, \tag{3.5}$$

where L is the filter length, the number of coefficients, and $\mathbf{r_{s_n s_j}(k)}$ is given by

$$\mathbf{r_{s_n s_j}(k)} = \mathbb{E}[s_n(k) s_j(t+k)],\ k = 0, 1, ..., L-1. \tag{3.6}$$

The cross-correlation vector $\mathbf{r_s}$ id defined as

$$\mathbf{r_s} = [r_1\ r_2\ \dots\ r_l], \tag{3.7}$$

where $\mathbf{r_n}$ is

$$\mathbf{r_n} = [r_n(0)\ r_n(1)\ \dots\ r_n(L-1)], \tag{3.8}$$

where each element given by

$$\mathbf{r_n[k]} = \mathbb{E}[s_n[t] s_r[t+k]]\ n, r = 1, 2, ..., N,\ k = 1, 2, ..., L-1. \tag{3.9}$$

This analytic Least Squares solution for the optimal filter is from the Wiener-Hopf equations which solve give the Wiener solution seen in appendix B. One can also use iterative methods, such as Least Mean Square(LMS) or Recursive Least Squares(RLS), which move towards, thus estimating the analytic Least Square solution [20] [21, p. 126-131].

3.2 Speech Feature Extraction

To be able to recognize a recorded spoken word the uttermost important characteristics of the spoken word must be extracted. These characteristics are then matched against a database of words, which creates a decision what, or if, spoken word was deemed spoken. It is therefore crucial that the extracted features in the recorded signal and the database are unique enough to differentiate between different words. But also that they are generic enough as all spoken words are unique in real life. The balance between extracting the features in a unique and generic enough way is a balance act.

To effectively store spoken words in the database it is necessary to minimize the number of bits used to store the signal. This is done extracting unique features which describe the spoken word. These unique features are stored in a vector which will henceforth be referred to as a feature vector. There exists many ways

of extracting and representing the features of a spoken word [19, p. 21-22]. In this thesis the features are extracted and represented using LPC coefficients.

3.2.1 Linear Predictive Coding - LPC

Linear prediction is an important estimation method within the signal processing field. LPC predicts future values of a signal, given previous values. This is useful in algorithms where calculation time is of the essence, such as real time applications as speech recognition [19, p. 21, 286-288].

The vocal tract can be considered a filter whose characteristics change depending of the speech, see figure 3.1 [14, p. 199-200]. This filter have coefficients which LPC identifies

$$H(z) = \frac{G}{1 + \sum_{k=1}^{M} a_k z^{-k}}. \tag{3.10}$$

Thus, LPC coefficients are synonymous to the vocal tract filter coefficients. This filter is excited by the switching between unvoiced and voiced sounds. When extracting features with LPC one decides the length of the vocal tract filter, that is, the number of coefficients one wants, or needs, to depict the spoken word. The number of coefficients decide how fine or oppositely, roughly, the speech characteristics will be stored. Many coefficients will depict the spoken word accurately and give an unique representation of the spoken word. The downside with many coefficients is that it is more difficult to reproduce and match speech features if the resolution of the characteristics is high. Oppositely, with few coefficients representing spoken words, the features are not unique enough. This results in it being increasingly easier to wrongfully match spoken words [14, p. 199-201]. A wrongfully matched word is called an *insertion*.

To calculate the LPC coefficients the *Levinson-Durbin algorithm* is used. This is a recursive algorithm which uses the solution of the Wiener-Hopf equations, see appendix B, for a prediction error filter of order m - *1* to give the solution for a prediction error filter of order m. The Levinson-Durbin algorithm is computationally efficient and also retrieves the *reflection coefficients* as a bi-product [14, p. 162]. The reflection coefficients represent a more robust alternative to the LPC coefficients, and are equally unique and representative of a signal. The reflection coefficients are more robust in the sense that their magnitude does not exceed unity, which in matters of quantization of coefficients, creates a representation of the signal as a stable filter [19, p. 364], [14, p. 44, 166]. A description of the Levinson-Durbin algorithm seen in figure 3.1 [14]. Where the order of the filter, m, is recursively iterated, with the m starting from 1 and ending at an order p. The value of p decides the number of reflection coefficients κ returned from the algorithm.

3.3 Matching Algorithm

A standard matching strategy is to compare the values of the reflection coefficients in the database towards the spoken word. This strategy requires that the spoken

It is initially known that $r(0) = 1 = P_0, a_{0,0} = 1 = \kappa_0$. First off, the auto-correlation of the signal is calculated

$$r(k) = \frac{1}{N} \sum_{n=1+k}^{N} x(n)x(n-k) \quad , \text{k} = 0, 1,.., \text{M}. \tag{3.11}$$

1. The recursion begins at $m = 1$. The scalar Δ_{m-1} is calculated

$$\Delta_{m-1} = \sum_{l=0}^{m-1} r(l-m)a_{m-1,l}. \tag{3.12}$$

2. and reflection coefficient κ_m updated

$$\kappa_m = -\frac{\Delta_{m-1}}{P_{m-1}}. \tag{3.13}$$

3. Then, using either the Yule-Walker equations, or as i this case, repeating the step below, the tap-weights of the filter are calculated

$$a_{m,l} = a_{m-1,l} + \kappa_m \cdot a^*_{m-1,m-l} \quad , \text{l} = 0, 1,..., \text{m}. \tag{3.14}$$

4. Lastly, the prediction-error power P_m is calculated

$$P_m = P_{m-1} \cdot (1 - |\kappa_m|^2). \tag{3.15}$$

5. Then the loop starts over from 1. with m increased by one. The loop quits after the $m = p$ loop has finished.

Table 3.1: Levinson-Durbin recursive algorithm.

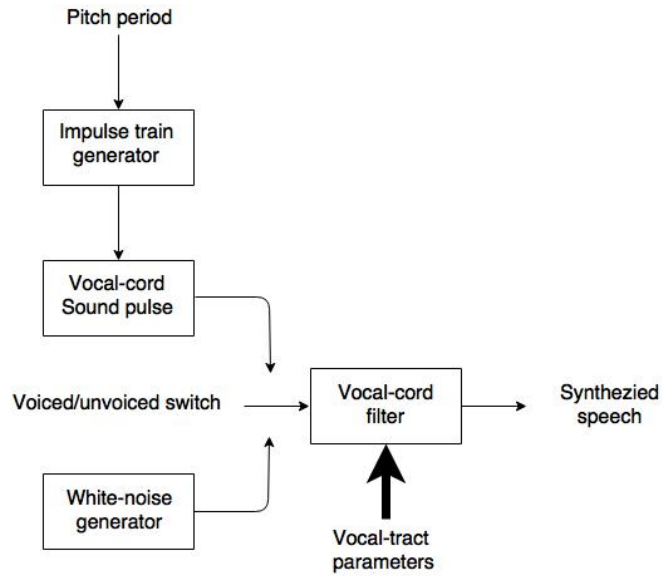


Figure 3.1: Block diagram over the speech production system.

word and the words in the database are represented by the same number of reflection coefficients. The difference between the database and the spoken word is an error named the Euclidean distance. There are different ways of using this error in a matching algorithm, and more precise descriptions of the matching algorithm will be given in chapter 4 and 5.

3.3.1 Identification and Validation

When talking about speech recognition, one has to make difference between identification and validation. Identification is made when it can decided what word, of a multiple word library, was most likely to have been spoken. Validation, on the other hand, is when it can also determined that the spoken word is none of the words in the database. In this thesis validation is considered.

3.4 Software

3.4.1 MATLAB

MATLAB was used to build high level libraries, test algorithms and plot results and other explanatory graphs. The algorithms are easily tested and parameters tweaked with prerecorded signals in an offline manner, before an implementation on the DSP. MATLAB was also used during the DSP implementation as tool for testing if the DSP implemented algorithms produced the same output as MATLAB algorithms.

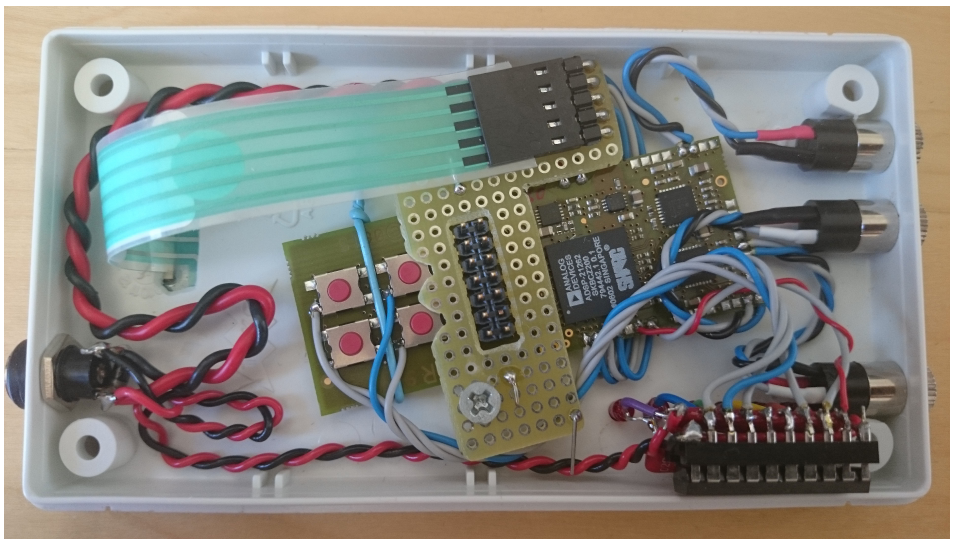


Figure 3.2: The DSP ADSP-21262 used in this thesis.

3.4.2 VisualDSP++ 5.1

The DSP used in this thesis, which will be presented in next sub-chapter, is programmed with the Analog Devices program VisualDSP++ 5.1. This Integrated Development Environment (IDE) gives help with the programming, such as compiler for C and C++ with informative compilation error messages, simulator and emulator, extensive debugging tools and signal processing libraries. VisualDSP++ also has support for displaying graphs which is helpful when evaluating whether the algorithms work according to plan.

3.5 Hardware

3.5.1 ADSP - 21262

The DSP used in this thesis is the 3rd generation low cost 32-Bit floating-point SHARC programmable DSP - ADSP-21262. The core is running at 200 MHz, a 5 ns operation cycle time. The ADSP-21262 has two memory banks which can be read in parallel to each other, Data Memory (DM) and Program Memory (PM). Code is is stored in PM and data in DM by default, but data can be set to be saved in PM as well. It has a 2 Mbit memory and a 4 Mbit non-volatile flash memory. The DSP has a The DSP was given from EIT and was built in a box and added four buttons, six diodes, two $3,5$ mm outputs and one $3,5$ mm input. See image 3.2 for a look inside the DSP box and a block diagram of the DSP.

State Machine

On the DSP, the program runs in a bit different order than offline evaluations in MATLAB, due to the continuous data stream. The recognizer is build as a state machine with three states – listening, collecting and processing, see figure 3.4. The first state, listening, samples the the input source and runs a speech detecting algorithm. If there is a speech, the state changes to the collecting state. In this state the algorithm collects samples in blocks and store the in internal memory. But as the DSP is not able to store a complete signal due to the limited size of the memory, the features of the signal are extracted in this state. Next the processing state becomes active. This state processes the feature vector and matches the spoken word against a database. Lastly, the state machine returns to the listening state.

3.6 Equipment

3.6.1 Microphones

In this thesis one table microphone and four studio microphones was used in the single microphone set-up and the multiple microphone set-up, respectively.

The used **table microphone** was a Deltaco Elecom stand microphone, see picture 3.5. The microphone is a electret type, which is a type of condenser microphone. It is omni-directional and has a frequency range of $[30, 16000]$ Hz and a sensitivity of -38 dB. The microphone has a 3.5 mm connector.

The **studio microphones** was a AKG C417 condenser microphone, see picture 3.6. This microphone has omni directional polar pattern. Its broadband, flat audio reproduction with open and natural sound, makes it ideal for multi-mic set-ups.

3.6.2 Audio Interfaces

Two audio interfaces, or sound cards, was used. **Roland UA-1EX** was used when recording with the table microphone, as the 3.5 mm microphone connector I/O on the laptop is not compatible with the external microphones. This audio interface is connected to the laptop via USB and to the table microphone through a 3.5 mm microphone connector, see picture 3.7.

The second sound card, a **Focusrite Scarlett 18i8 USB 2.0**, seen in picture 3.8 was used to perform the recordings in 4-channels using the AKG C417 condenser microphones. The sound card is connected to the laptop via USB and used via an application running on the laptop. The application is given and written by the advisor.

3.6.3 Speakers

To record noises the Fostex 6301B speaker was used. The context in which these recordings matter, will be explained later on in this report, see picture 3.9.

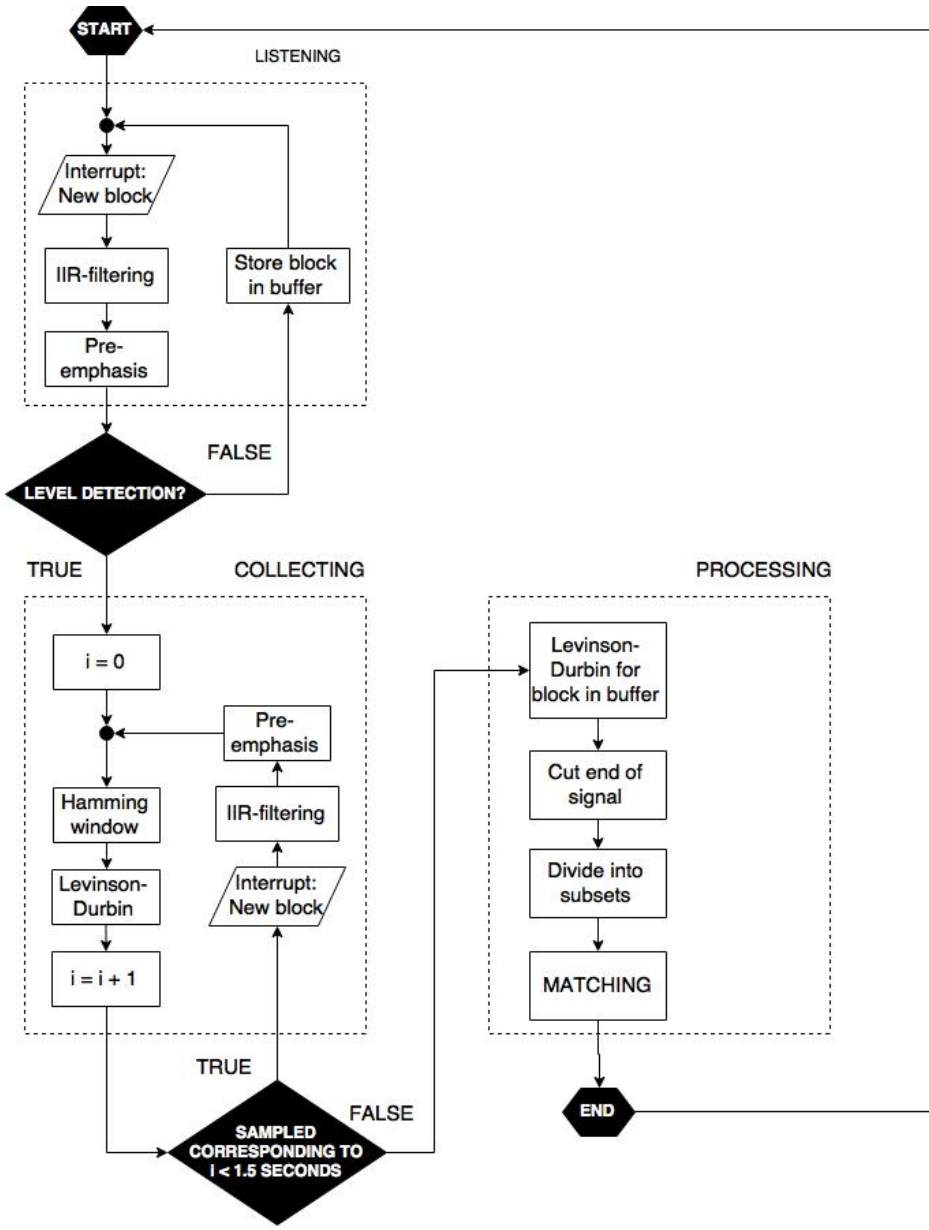


Figure 3.3: Flowchart over the DSR algorithm implemented on the DSP.

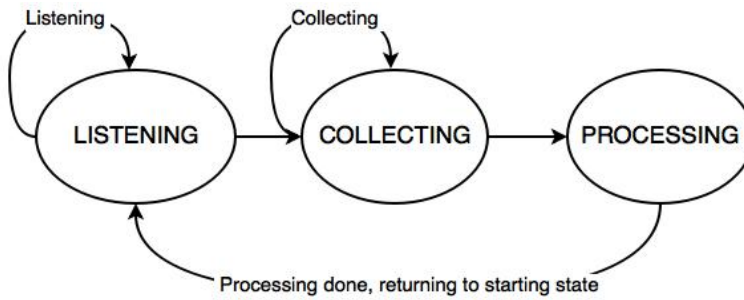


Figure 3.4: The automatic speech recognition algorithm as a state machine.

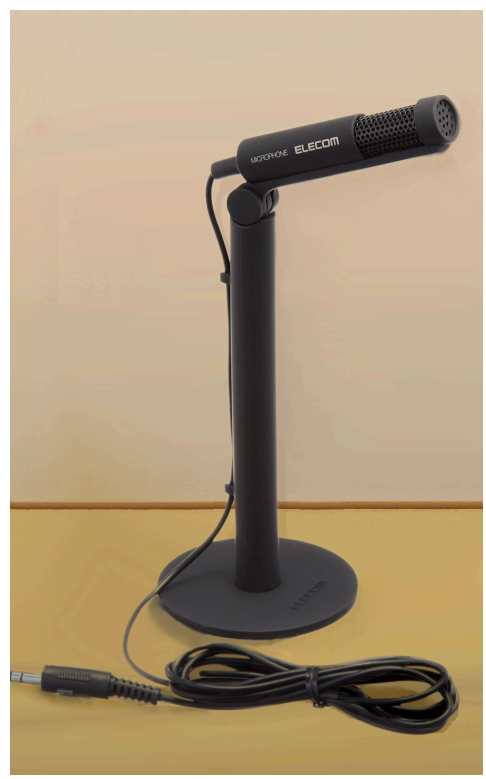


Figure 3.5: Deltaco Elecom stand microphone.

Figure 3.6: AKG C417 condenser microphone with AKG MPA III phantom adapter.

Figure 3.7: Roland UA-1EX audio interface used when recording using the table microphone.

Figure 3.8: Focusrite Scarlett 18i8 USB 2.0 audio interface used when recording in 4-channels using the AKG C417 condenser microphones.

Figure 3.9: Fostex 6301B speaker.

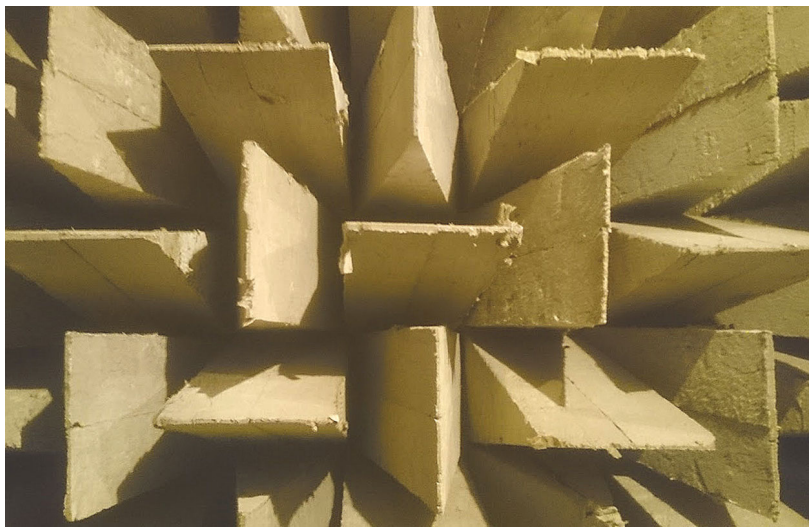


Figure 3.10: Wall in the anechoic chamber in the Perlos antenna laboratory in E-building at Lund University.

3.7 Environment

To make sure that the recording and test environment is not corrupted by noise and reverberation with unknown characteristics, an anechoic chamber was used. An "an-echoic" chamber means a room which is non-reflective, non-echoing or echo-free, as it completely absorb reflections of sound waves. By using this type of environment the results are more independent of reverberations in a specific environment. This particular anechoic chamber is located in the Perlos lab in E-building at LTH, Lund University. In picture 3.10 a piece of a wall in the chamber can be seen. The walls, floor, ceiling and door in the room is covered in the same structural way as in the picture.

3.8 Thesis Execution Strategy

The **single microphone set-up** is implemented on the DSP with the recognizer running solely on the DSP. The single microphone set-up is implemented alongside the course EIT80 [6]. During the implementation performance evaluations are continuously performed, and the lessons learned are introduced to the reader and the implementation. The **multiple microphone set-up** is an extension of the single microphone set-up. It is an extension in the sense that the single microphone implementation has an added part which handles multiple microphones. The added part is implemented in Matlab instead of on the DSP. This is since this simplifies the implementation of the automatized tests. But as the multiple microphone set-up will be tested in an offline manner, the real-time structure of the single microphone set-up is altered to fit an offline implementation. But the algorithms remain the same.

Chapter 4
Single Microphone Set-Up

4.1 Introduction

In this chapter the implementation of the single microphone set-up is processed, explained and evaluated.

4.2 Database

The database consists of 15 versions of each of the words "Vänster" and "Höger". The versions have different pronunciation and are recorded at different distances to the microphone, and are voiced by the same person. The database is stored in the program memory of the DSP, and not the more limited data memory. The database recordings took place inside the anechoic chamber with the table microphone.

Before deciding upon the database stated above, it was considered switching language to English. But as the words "Left" and "Right" both are short words and end with noise-like and silent letters (*f, t, g, h*) the recognizer cut the words "Left" and "Right" to "Le" and "Ri". Cutting database entries into these short segments enable the recognizer to match other words than "Left" and "Right" which contain these segments, thus decreasing robustness of the recognizer. But as this thesis focus lies on testing whether beamforming improves performance it was deemed unnecessary to choose words which are difficult to recognize. Thus the corresponding Swedish words, "Vänster" and "Höger", was chosen.

4.3 Test Set-Up Environment

In the single microphone set-up there was two tests performed on the DSP to test the implementations in this chapter. The tests was performed on the Digital Signal Processor (DSP) speaking on 1.5 meters distance into one microphone. The test set-up environment is illustrated in figure 4.1. The speaker talks directly into the microphone, with varying pronunciations of the words. The tests were performed in the anechoic chamber.

The first test considered the two words "Höger" and "Vänster" being spoken 100 times each and counting the substitutions. That is, for example, if "Vänster"

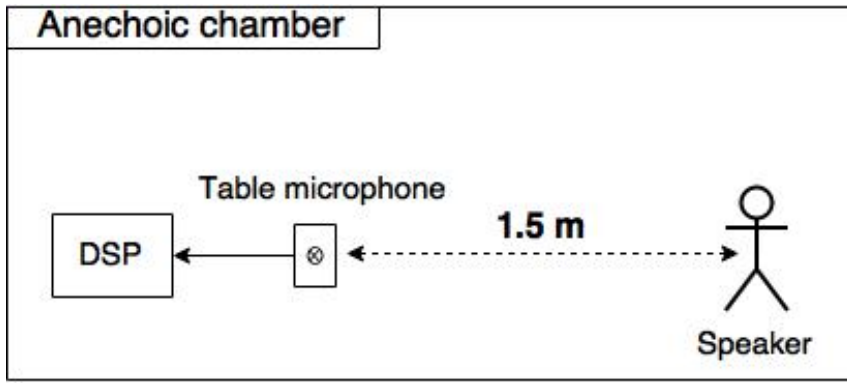


Figure 4.1: Illustration of the test environment of the single microphone set-up.

Listening	Collecting	Processing
Filtering	Recording	Cutting
Level Detection	Feature Extraction	Dividing Into Subsets
		Matching

Table 4.1: Content of the three states in the single microphone set-up.

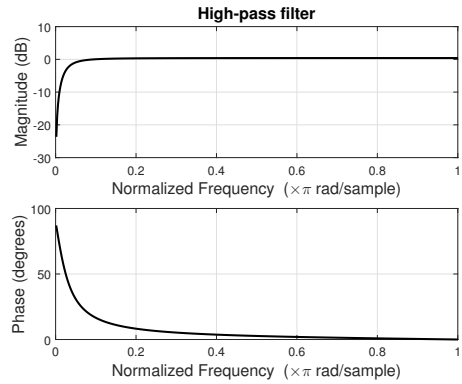
was spoken it was counted how many times "Höger" was recognized and if no match was found, the "no match" results was given. The second test considered random speech without mentioning the words "Höger" and "Vänster" being run through the recognizer 100 times. In the second test a high "no match" results is aimed for.

The WER is a measure of the magnitude of errors presented as a percentage of the quality of the recognizer. Thus, the WER, is sought to be 0%, giving 100% accuracy of the recognizer.

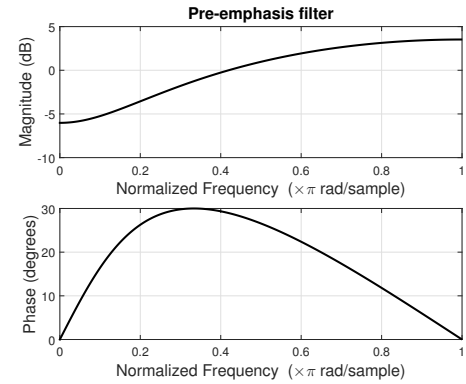
4.4 Implementation

In this implementation of the three states, 3.4, there are a total of seven steps involved - level detection, recording, filtering, feature extraction, cutting, dividing into subsets and matching. Table 4.1 shows which state the steps belong to.

The first two states handle blocks of 160 consecutive samples. After the reflection coefficients are calculated and put into a feature vector, the signal is represented by a vector of feature vectors, that is, a matrix of reflection coefficients.



(a) High-pass filter characteristics.



(b) Pre-emphasis filter characteristics.

Figure 4.2: The filters applied to the signal in the single microphone set-up.

4.4.1 Listening

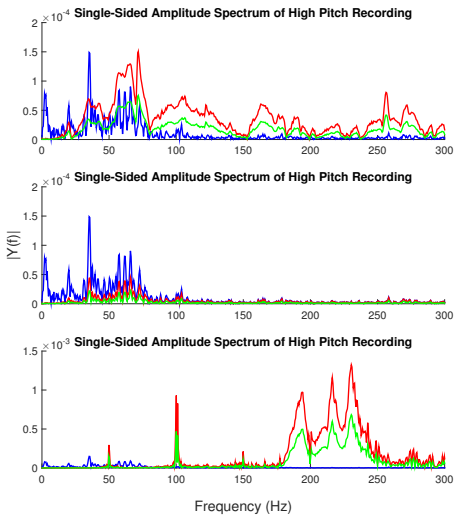
The AD-converter saves samples in a buffer and produces blocks of 80 new samples to be handled, one a time. The buffer stores two blocks of 80 samples each, the latest and the previous block. When a new block of samples is ready it is then filtered, pre-emphasized and windowed. Then the old block is added at the beginning of the new block, thus creating a block of 160 samples. The new block of 80 samples becomes the old one, and the 160 sample block is sent through level detection. The level detection decides if speech was present and a state transition should occur.

Filtering

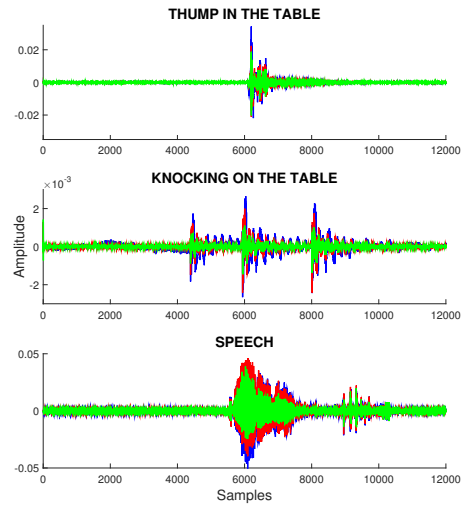
Three types of filters are applied to the buffered signal: two high-pass and one pre-emphasis filter, in that specific order. Since low frequencies normally have a higher energy than the higher ones, the recorded signal is filtered with high-pass filters. The high-pass filter removes low frequency signals such as vibrations from table and floor, and 50 Hz disturbances from the wall socket.

Following after is the pre-emphasis filter, whose purpose is to boost higher frequencies [5]. In this way SNR is boosted. The high-pass filter is a IIR-filter and pre-emphasis an FIR-filter. See figure 4.2 below for the difference in characteristics of one of the high-pass and the pre-emphasis filter.

The reason for using two high-pass filters is that during tests of the performance during implementation, it was noted that the recognizer was sensitive to knocking sounds and thuds on the table. These sounds was picked up as speech, which is not desirable. Figure 4.3 shows FFT's and plots of three types of sounds. Disturbances under 100 Hz was filtered out by adding an extra high-pass filter, see figure 4.4 to see characteristics of the two high-pass filters. The result of this added filter to the three signals can be seen in figure 4.5.

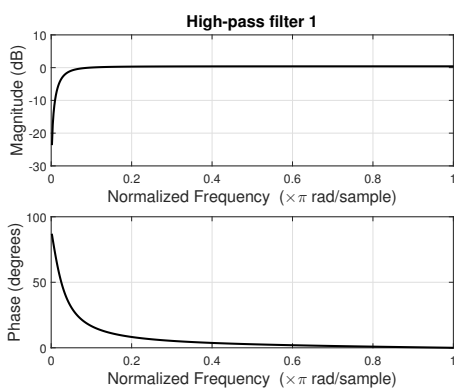


(a) FFT of the signals.

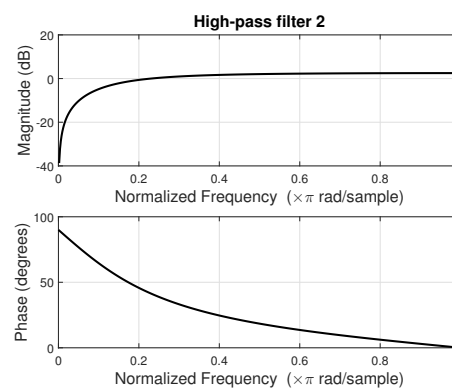


(b) Plot of the signals.

Figure 4.3: The filters of the single microphone set-up applied to three types of signals. Blue is the original signal, red is after the high pass filter and green is after pre-emphasis.

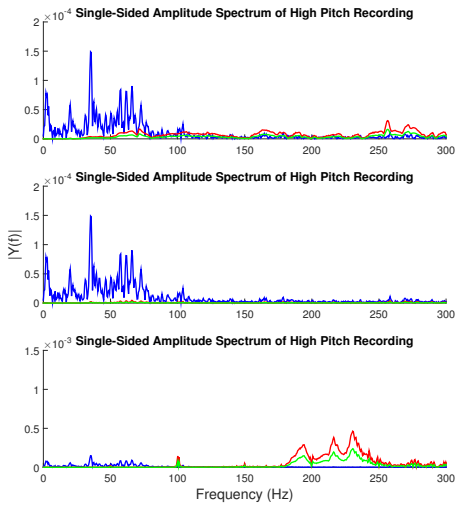


(a) The first high pass filter.

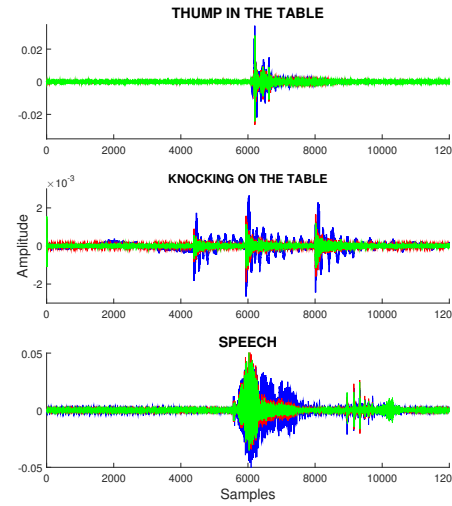


(b) The second high pass filter.

Figure 4.4: The two high pass filter of the single microphone set-up.



(a) FFT of the signals.



(b) Plot of the signals.

Figure 4.5: The filters of the single microphone set-up applied to three types of signals. Blue is the original signal, red is after the high pass filters and green is after pre-emphasis.

Level Detection

To determine if speech is present and recording should commence a VAD algorithm is used. This algorithm is based on a dynamic noise detection which adapts in accordance to its surroundings. That is, in a constantly noisy environment the algorithm will raise the threshold on which speech can be detected, thus minimizing the risk of an insertion.

The VAD takes both slow and fast changes of the energy into consideration. Slow changes in energy is considered speech and fast changes in energy is considered to be noise. These two energies are given by integration of the type seen in

$$threshold_{t+1} = norm_t \cdot \alpha + threshold_t \cdot (1 - \alpha). \tag{4.1}$$

Then the ratio of the two energies are calculated and compared to a constant. This constant gives how much more speech than noise energy is needed to activate the VAD and determine that speech is detected. When speech has been detected the state changes to the collecting state. The values of α, β and T are examples. A large α or β gives slow integration and vice versa. T is the constant which the ratio is compared to. See pseudo code in figure 4.6.

If the VAD-algorithm is activated, a sound signal will be recorded that is, a state transition will take place, and the recognizer enters the collecting state.

```
1   alfa = 0.99;
2   beta = 0.8;
3   T = 5;
4
5   energy = calc_energy(input);
6   Energy_slow = Energy_slow*alfa + energy*(1 - alfa);
7   Energy_fast = Energy_fast*beta + energy*(1 - beta);
8   R = Energy_fast/Energy_slow;
9
10   if (R >= T) {
11       VAD activated --> switch state
12
13   } else {
14       add block of samples to ringbuffer
15   }
```

Figure 4.6: Pseudo code of the level detection algorithm processing one block of samples in the single microphone set-up.

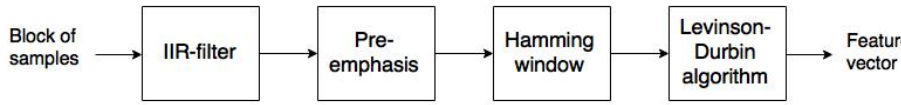


Figure 4.7: Transformation steps of one block of samples during the collecting state in the single microphone set-up.

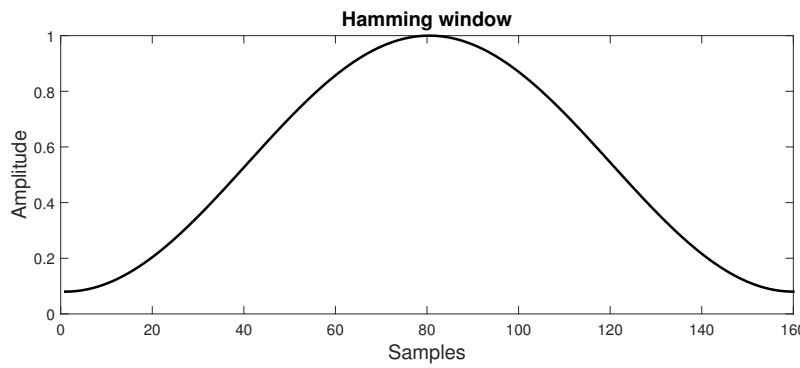


Figure 4.8: Hamming window.

4.4.2 Collecting

When speech is detected the program enters stage two where it collects data. See figure 3.3 of the state machine. The largest problem with the DSP is the limited amount of data that can be stored in the data memory. Therefore it is necessary to reduce the size of the data, which is done by extracting the speech features of the block of samples to a vector of features, a feature vector. This way of collecting data is looped until for the number of blocks corresponding to 1.5 seconds have been sampled. Then the recognizer enters the third state, processing.

Recording

The DSP collects a block of 160 consecutive samples, filters it as described in section 4.4.1 and convolving the signal with the Hamming window seen in figure 4.8 to remove the effect of transients. Then the energy of the signal and the feature vector is extracted. See figure 4.7.

This recording loops until 1.5 seconds have been sampled. A sample rate of 8000 Hz was chosen to keep amount of data down and to prevent disturbance from high frequency components. Speech has usually a maximum frequency of around 4000 Hz and because high frequency consonants, such as k, t, s, f, does not give much information to the reflection coefficients, it is sufficient to sample at this rate.

After the each block have been recorded, an update of the slow and fast energies

`Energy_slow` and `Energy_fast` is done. These energies are updated using the `energy` from each recorded block, as seen in row $6-7$ in figure 4.6. This is added so that after one word has been recorded and processed and the listening state is entered again, the level detection energies `Energy_slow` and `Energy_fast` are up to date on the current speech and noise energies.

Feature Extraction

The Levinson-Durbin algorithm is used to extract the features from a recorded block. See the steps of the Levinson-Durbin algorithm in 3.1 in chapter 3. The algorithm is applied on each block to calculate a set of reflection coefficients. Each set of reflection coefficients is a feature vector. After the reflection coefficients have been calculated the energy of each block is calculated as seen in

$$P_n = \sum^{i} |x_i|^2, \tag{4.2}$$

where x_i are the individual samples in one block. The energy is stored as it is used in the processing state to determine where to cut the recorded signal.

4.4.3 Processing

The first step in the third stage is to process the blocks in the buffer as well. The buffer blocks are convolved with the Hamming window(4.8), reflection coefficients(3.1) and energy(4.2) calculated. Then the entire recorded signal is cut, averaged and matched against the database. When a matching decision have been outputted, the state machine return to the listening state.

Cutting

The recorded signal is 1.5 seconds which is longer than the average spoken word. That is, the recorded signal contains parts where no speech is present. Since the recording of the signal started when speech was detected, there is no need to cut the signal from the start. Thus, the superfluous parts to be removed are in the end of the recording.

The VAD, see figure 4.6, is used when cutting the recorded signal. But `Energy_slow` and `Energy_fast` is not the same variables which are updated for every recorded block in the collecting state. The difference is that they are newly initiated variables for the cutting procedure, and no ringbuffer is used.

As the signal is cut, the recording will start and end with vocal speech, and no unnecessary samples, containing noise, will be saved. See figure 4.9 for an example of a signal being processed by the use of filters and cutting of the signal.

Dividing into Subsets

When every block of samples from the recording have been processed, feature vector from each block have been extracted and unnecessary blocks been removed, a matrix of K feature vectors has been produced. Then, for both memory saving properties and robustness of the characteristics of the speech, the feature vectors

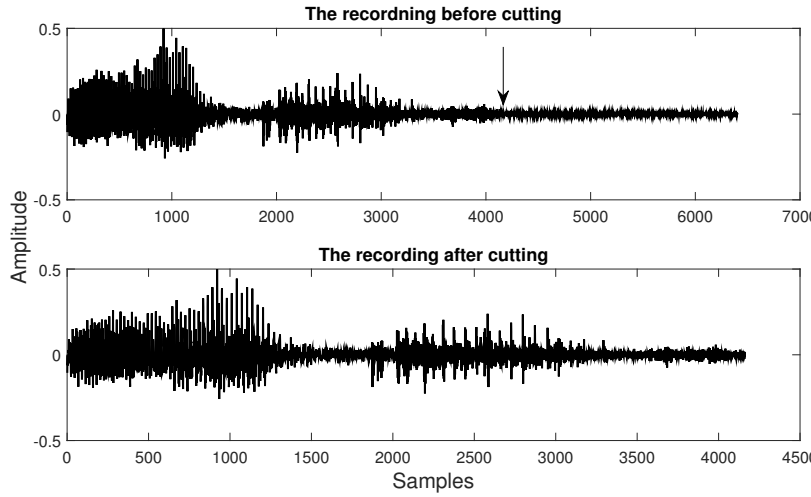


Figure 4.9: Visualization of how the end of the recorded signal is cut in the processing state in the single microphone set-up.

are divided into M subsets by taking a mean value, row wise, along the feature vectors belonging to a subset. After this averaging, the set of M subsets is considered a database containing the characteristics of the recorded spoken word, see figure 4.10.

Matching

To match a recorded word against the database the Euclidean distance is used. Euclidean distance measures the distance between two points in Euclidean space, that is, the two dimensional space in which the points that are to be compared, exist in. See figure 4.11, where the Euclidean distance is the distance between the cross and star, marked out with an arrow. Each dot representing a reflection coefficient.

The recorded speech is represented by M database vectors which contain the features, the reflection coefficients, of the speech. The Euclidean distance is the difference between the recorded reflection coefficients and the database reflection coefficients. This distance is the mismatch error ϵ, of the recording against the database. The recorded word is tested against every word and every version of a word in the database. Two types of error are saved used in the matching decision: ϵ_{min} which is the smallest ϵ of all versions. ϵ_{mean} which is the smallest mean of the total error for all versions for a type of word.

For identification, the word which produced the smallest ϵ_{mean} is the recognized word. But if wanting validation of a word, harsher constraints are needed. To decide on a specific word both ϵ_{min} and ϵ_{mean} have to belong to the same type of word, for example "Vänster", to give a decision that the recognized word is "Vänster". If the two errors do not belong to the same type of word, the decision states that no match was found. For greater accuracy a threshold is also added.

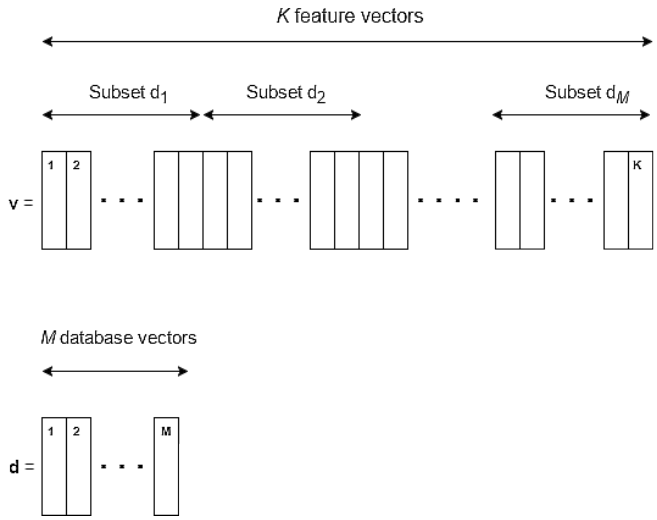


Figure 4.10: Dividing the K feature vectors into M subsets.

That is, alongside the two errors needing to belong to the same type of word, the error ϵ_{min} must lie beneath a certain value to decide upon a certain word.

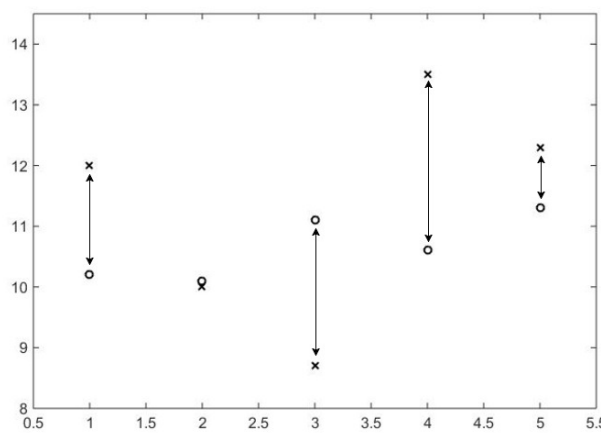


Figure 4.11: Plot of two sets of reflection coefficients, visualizing the Euclidean distance as an arrow.

Chapter 5
Multiple Microphone Set-Up

5.1 Introduction

This chapter introduces multiple microphones to the implementation of the recognizer is processed, explained and evaluated. The final evaluation tests are also explained.

5.2 Database

The database consists of 25 versions of the words "Vänster" and "Höger" recorded in the anechoic chamber on various distances using a condenser microphone. The versions are pronounced slightly different, at different volume, and spoken by one person.

5.3 Recordings for Tests

The evaluation of the performance was executed offline in Matlab. The spoken words that are used when testing the multiple microphone set-up, are pre-recorded in the anechoic chamber. The recordings are 4-channel, performed using the sound card and condenser microphone introduced in a previous chapter. The pre-recorded test library consists of 200 versions of each word at each distance, see table 5.1. Four types of noise were also pre-recorded in the anechoic chamber, which will be added to the speech and put through the beamformer. Each noise recording consists of a left and a right side recording. See figure 5.1 which depict the recording environment.

Spoken word	1 meter	2 meters	4 meters
"Höger"	200	200	200
"Vänster"	200	200	200

Table 5.1: Number of versions of the words for each distance to be used in tests in the multiple microphone set-up.

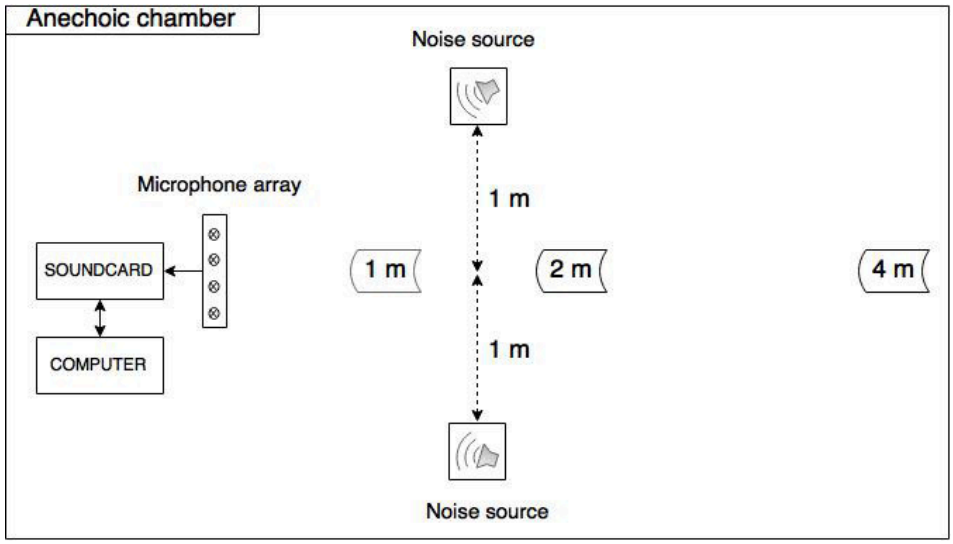


Figure 5.1: An illustration of the recording set-up environment in the anechoic chamber.

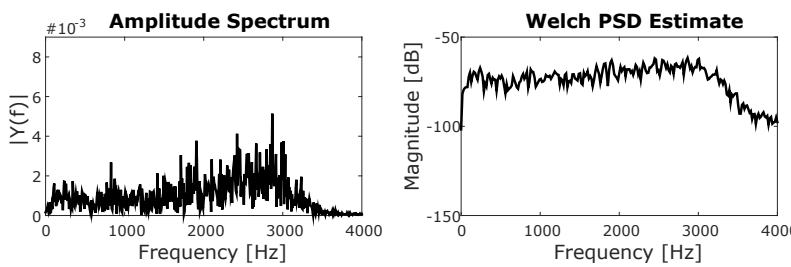


Figure 5.2: White noise characteristics.

5.3.1 White Gaussian Noise

White Gaussian noise is an ideal noise is commonly used in theory but in practice does not exist. White Gaussian noise is a purely random, steady, high frequency noise with constant variance and zero mean. This type of noise is not easy to remove as it cannot be modeled for due to it being a noise with a completely random pattern. In figure 5.2 two plots are presented. One can see from these plots that it is has most of its power in high frequency.

5.3.2 Factory Noise

Factory noise is a pre-recorded noise from a factory environment. It has high variance and contains slams, thuds, and vehicles passing in an large enclosed space. In figure 5.3 one can see that the noise has most of its power in low frequencies where speech also lies.

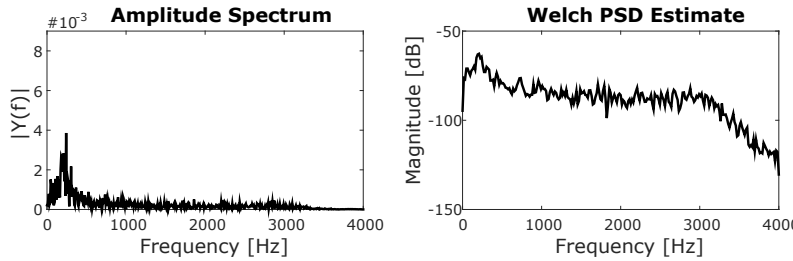


Figure 5.3: Factory noise characteristics.

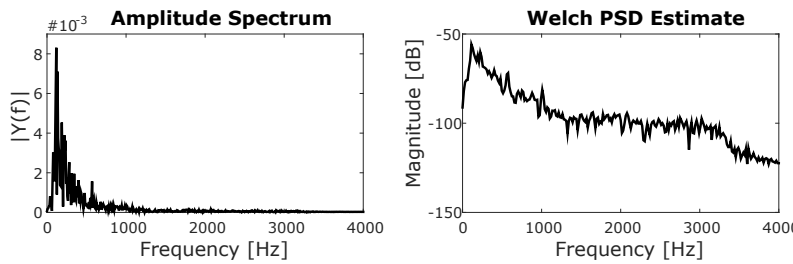


Figure 5.4: Engine noise characteristics.

5.3.3 Engine Noise

Engine noise is a tonal noise, whose frequency range lies in the same frequency range as the first two formants of vowels. This type of noise is constant and of low frequency, and is difficult to remove as it mixes with the speech. As can be seen in figure 5.4 the noise has most power in low frequencies and next to no power in high frequencies.

5.3.4 Babble Noise

Babble noise consists of multiple persons talking low-key in the background, without intelligible words. It is one of the best noises for masking speech for the human ear. In figure 5.5 it can be seen that the noise consists of a wide range of frequencies, with the most power in low frequencies, in the speech frequency range.

5.4 Reverberation

As environments without reverberation, such as the anechoic chamber, is not a realistic environment when using a DSR application. This is since it is realistic to assume that the application is to be used in confined space, reverberation will be added. In this thesis the pre-recorded words for tests are put through a script which simulate and add a room effect, echoes, to each word. This is done with an algorithm called image method.

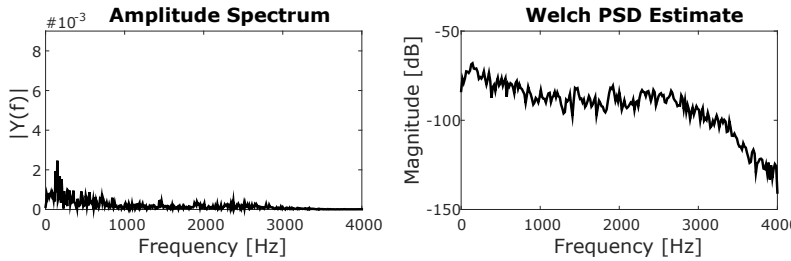


Figure 5.5: Babble noise characteristics.

5.4.1 Image Method

Image methods approximate the room impulse response by transforming multi-path reflections from a single source into direct paths from multiple virtual sources. The virtual sources are determined by mirroring the original source over the surfaces of the room, and the multi-path impulse response is modeled as the sum of the ideal direct paths for a set of virtual sources [22]. The script which perform these calculations and takes input parameters such as size of the room, location of speech source and microphone, number of taps in the impulse response function, sampling frequency, the absorption coefficient of the walls in the room and the number of images in all directions in the room. The chosen absorption coefficient used in this thesis is $RC = 0.5$.

In figure 5.6 the simulated room can be seen where M is the microphone, S1, S2 and S4 are the speech sources from one, two and four meters respectively. In figure 5.7 one can see the reverberation time for the three simulated distances $1, 2$ and 4 meters respectively. RT_{60} is pointed out with an arrow in the respective plot, and the impulse is shown in each graph. RT_{60} gives the time that the reverberation has been attenuated -60 dB and is a common measure of reverberation in rooms.

5.5 Test Set-Up Environment

In the multiple microphone set-up the implementation is tested through many different environments, which is the main test of this thesis. A summary of the five main parameters follows.

1. Speech disturbed by noise only, and by noise and reverberation.
2. Speech located at one, two and four meters distance,
3. Processed with one microphone, one microphone with a Wiener filter, and with two, three and four microphones with a beamformer.
4. The disturbing noise is white Gaussian noise, factory noise, engine noise and babble noise.
5. The speech is played back with an SNR from -10 dB to 20 dB.

A test is made for each combination of these five set-up parameters. In each set-up is 100 words, 50 “Vänster” and 50 “Höger”. In total 732000 words are run through

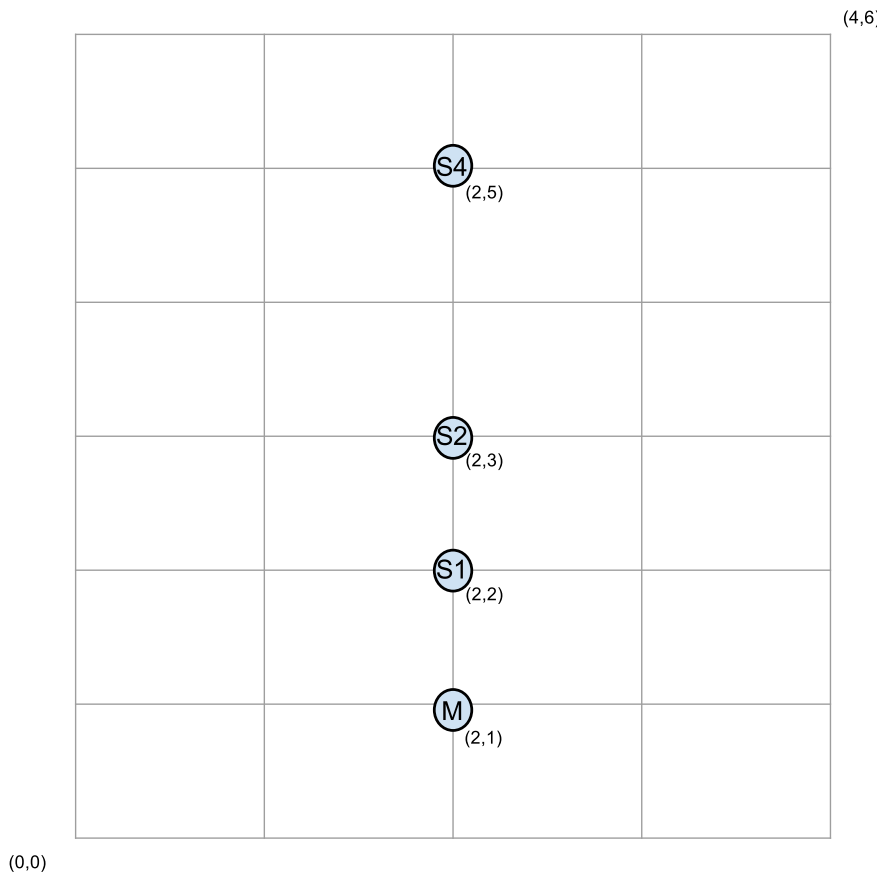


Figure 5.6: The simulated room used in the image method script.

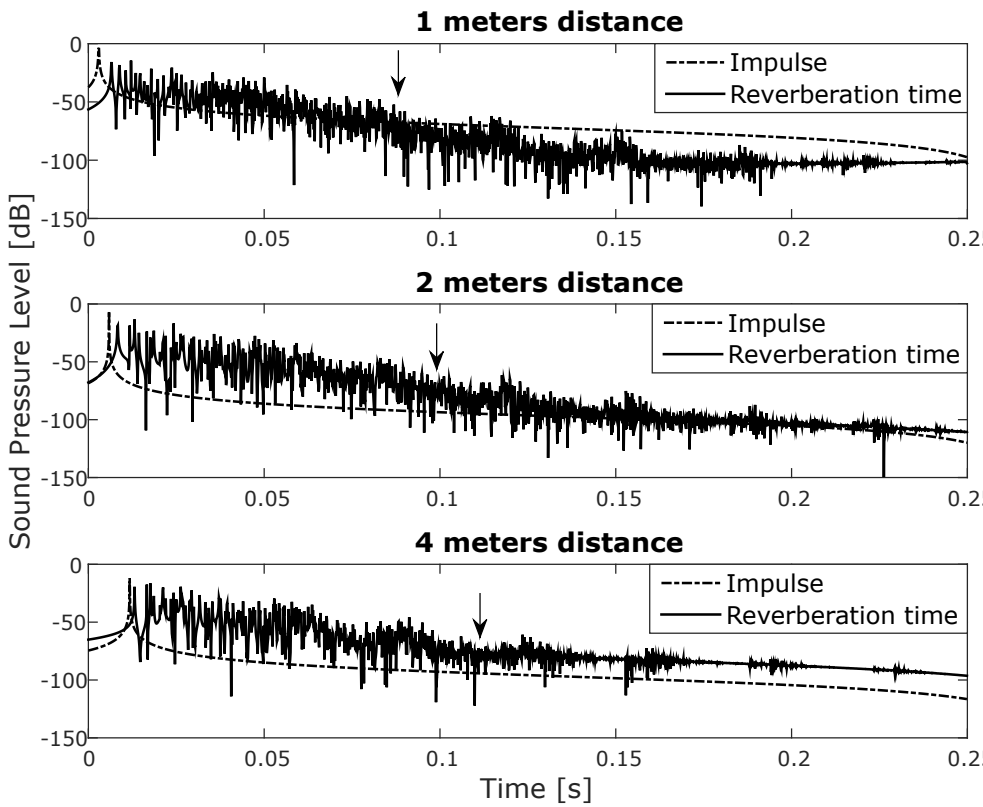


Figure 5.7: The reverberation time for the three distances $1, 2$ and 4 meters, where the arrow shows RT_{60}.

Listening	Collecting	Processing
Set SNR Level		Filtering
Calculate Beamformer Filter		Cutting
Retrieve The Couple of Word + Noise		Divide Into Blocks
Beamforming		Feature Extraction
Level Detection		Dividing Into Subsets
		Matching

Table 5.2: Content of the three states in the multiple microphone set-up.

the tests. This amount of words that are tested is the reason why these tests are not performed on the DSP. This is as these tests would in real-time, if each word takes four seconds in total to pronounce and evaluate, take more than four weeks and five days to perform.

The two previously described main tests of this thesis will test short segments of speech and noise, where the speech begins very short into the segment. The difference from a real-time recognizer implemented on a DSP is that the real-time recognizer will have periods of silence between spoken words. It is therefore reasonable to assume that the VAD, in real-time, will give a slightly different result than the offline VAD as the VAD integrates the speech and noise energies before testing the ratio against a threshold, see the pseudo code 4.6. A test which simulates this real-time scenario with longer periods of only noise will be given.

The performance of the recognizer without noise and reverberation will also be presented, that is, the tests will solely have speech and no disturbances. This test will reveal the efficiency of the recognizer in an optimal environment.

To sum up, for the multiple microphone set-up, four types of tests are performed and presented.

5.6 Implementation

As this part is performed in Matlab, instead of on the DSP, the three previously mentioned states are implemented differently. Before the state machine is entered there is pre-processing of the test words, the database, and the noises. In this implementation the three states contain a different set of steps, see table 5.2. The collecting state contain no steps as this implementation of the recognizer is not tested in real-time, and all words that are tested are pre-recorded and thus a collecting state is not needed. To help the reader understand the test in pseudo code can be seen in figure 5.8.

5.6.1 Pre-processing

Before the automatized tests commences pre-processing of the test words and the noises that are added to the test words must be performed.

```
noise_type = white; % factory, engine, babble
for h = 1:Number of microphones to test
   for i = 1: Number of SNR levels to test
      setSNRLevel(noise_type);
      filter = CalculateBeamformer();
      for j = 1: Number of words to test
         word = GetRandomWord();
         noise = GetRandomNoise(noise_type);
         y_1 = Beamforming(word+noise,filter);
         detect = VAD(y_1);
         if detect = 'no'
            deletions++;
         else
            y_2 = Filter(y_1);
            y_3 = Cut(y_2);
            y_4 = ToBlocks(y_3);
            y_5 = FeatureExtraction(y_4);
            y_6 = CreateSubsets(y_5);
            [match, reason] = Matching(y_6, word);
            if match = 'yes'
               score++;
            else
               substitution++;
               if reason = 'höger'
                  right++;
               elseif reason = 'vänster'
                  left++;
               else % reason = 'no match'
                  no_match++;
               end
            end
         end
      end
   SaveResults(deletions, score, substitutions,
               right, left, no_match);
   deletions, score, substitutions, right,
                                left, no_match = 0;
   end
end
```

Figure 5.8: Pseudo code of the test for the multiple microphone set-up.

Database, Test Word and Noise Recordings

The recorded database and test words consists of multiple 30 second 4 channel recordings with multiple words in each recording. These recordings are initially listened and looked at to discover and remove disturbances in the recordings. The recordings are then separated in an automatized script to single word recordings, in one channel for the database and 4 channels for the test words. The method used to separate the words apply the technique used the VAD in the second implementation of the Single Microphone set-up. The difference is that when speech is detected a starting index is set and from said index a signal of 5000 samples is saved. The number of samples was chosen after examining the 30 second long recordings and finding that words are fully pronounced under 5000 samples, that is, 0.625 seconds.

The test words are all segments of 5000 samples, this is since the noise is to be added to the words later on and a constant length of the segments simplifies the test script.

The noises are also multiple 30 second 4 channel recordings but are not initially split up to segments of 5000 samples as the noise dB level will be altered during the tests. The noise recordings consist of both a left and a right side 4 channel recordings. The left and right side are first balanced SNR wise as there could be differences in the volume settings between the two recordings. After balancing the left and right side the recordings are summed.

Calculating and Balancing SNR

The results of the tests are to be plotted as recognition rate against SNR level, with the noise dB level decreasing, and the speech dB level considered constant. Thus it is necessary to calculate the mean dB of all the words recorded, both "Vänster" and "Höger", at each distance and the dB level of the noises. As the pre-recorded words for the tests contain segments of silence the dB level was calculated by comparing the absolute value of the magnitude of each sample to a constant. If the sample lies above the constant, that sample value is considered speech and accounted for the dB level calculation. The constant value was decided by looking at plots of some of the words. This constant is different depending on the distance the words was recorded on. When the average dB level of the words are determined, the dB level of the noise is adjusted such that the starting SNR value is reached.

Randomly Selected Words and Noise

At each SNR level 100 unique words, 50/50 "Vänster" and "Höger", are put through the recognizer. As previously mentioned, at each distance, there exist 200 recordings each of "Vänster" and "Höger", among these recordings 50 of each word is randomly selected and not being selected more than once. A set of noise segments are also randomly picked. The Matlab functions `randsample` and `setdiff` was used to achieve this.

For all tests, on one distance, the same set of randomly selected words and noises are used. This is to be certain that the only difference is the number of

microphones, which is the main interest of this thesis.

5.6.2 Listening

Set SNR Level

The SNR level starts at -10 dB and increases in steps of 0.5 dB up to 20 dB. The energy of the speech is not altered but that energy of the noise is decreased by calculating

$$Noise = Noise \cdot 10^{-\frac{dB_{step}}{20}}, \tag{5.1}$$

where $dB_{step} = 0.5$ and equation 5.1 being performed on each sample of the noise signal.

Calculate Beamformer Filter

The beamformer filter is calculated by using LS which uses the Wiener-Hopf equations to solve for the Wiener solution, see appendix B, to calculate the optimal filter. The input to the LS calculations is the noisy speech signal and the clean speech signal, also referred to as the *desired signal*, and the number of filter coefficients. The speech which the beamformer was trained on was two randomly picked words of 5000 samples each, one "Vänster" followed by one "Höger". The noise was also two randomly picked 5000 sample long signals. For all beamforming filter calculations, the same speech and noise segments were used, with the noise dB level being altered. In the thesis, the number of filter coefficients was set to 32. This was decided after testing some different filter lengths and realizing that 32 coefficients gave higher recognition rate than 8, 16 and 64 coefficients.

Retrieve The Couple of Word + Noise

As previously mentioned, under the pre-processing, a set of randomly picked words and noise segments were created. This set is being iterated through and couples of speech and noise retrieved. Depending on the number of microphones which the test loop is at, the corresponding number of channels of the word and noise couple is picked out.

Beamforming

Next the word and noise couple, in one to four channels depending on the current loop microphone set-up, are summarized and filtered through the calculated beamforming filter using the Matlab function `filter`. The output from the filtering is a one channel signal with optimally no noise present.

Level Detection

A similar VAD as the single microphone set-up was used in the Multiple Microphone set-up as well, see pseudo code in figure 4.6. The difference is that if the algorithms detects speech the action, instead of adding to a ring-buffer, is to set an

index in the input vector where the speech starts. If this index exceeds half of the input signal length it is decided that a deletion has occurred, as it it known that the speech in the input signal starts before half of the signal length. If a deletion has occurred the test skips the rest of the loop and takes the next word. If no deletion has occurred, next state is entered. Another difference from the single microphone set-up is that this implementation does not take the speech and noise energies into account after level detection has been activated. Thus the VAD in this implementation does not have a continuous update, but starts over for each new word that are put through the VAD.

5.6.3 Processing

Filtering

The same filtering as mentioned in the single microphone implementation is used in the multiple microphone implementation, see figures 4.4 and 4.5.

Cutting

Then the signal is cut, removing only the end of the signal where no speech is present. This is done as it is known that the signal begins directly with speech. The same method as in the single microphone set-up is used in this implementation, with the difference that only end of the signal is cut.

Dividing into Blocks

As the previous implementation was done on the DSP and its confined memory restricted recording the entire signal as a hole. Instead the signal was recorded in blocks of 160 samples with a 50% overlap, thus the signal was from the start divided into blocks before the processing state commenced.

This is not the case in this offline implementation. To divide the signal into blocks the signal was first run through the Matlab function `buffer`. Then the blocks was multiplied with a Hamming window, see figure 4.8, created with the Matlab function `hamming`.

Feature Extraction

The same method as used in the single microphone implementation is used in this implementation as well.

Dividing into Subsets

The same method as used in the single microphone implementation is used in this implementation as well.

Matching

The same method as used in the single microphone implementation is used in this implementation as well. With the minor alteration that the threshold which determine if the error ϵ_{min} is small enough was changed as the quality of the database recordings and test environment has changed.

Chapter 6
Results

In this chapter the results from the mentioned tests from chapters 4 and 5 is presented. The results are presented in tables and graphs.

6.1 Single Microphone Set-Up

Table 6.1 gives the test results from the tests mentioned in 4 can be seen. The tests were performed in real-time on the DSP. In the table, the left column is the spoken word and the top row is the recognized word and the WER.

6.2 Multiple Microphone Set-Up

All tests in this section is performed offline in MATLAB. The figures in this section plot the recognition rate against the SNR level. Recognition rate is a measure of how many words that was correctly recognized, thus a high value is sought after. There are 5 cases plotted in each figure, and they are marked with different types of lines. For each noise type one figure displaying the types of errors which can occur if the correct word is not found, they are plotted as the Error[%] towards the SNR level. To keep down the number of figures it was decided to display two cases - one microphone and no beamformer and four microphones and beamformer, for one meters distance. The tables shows at what SNR level the recognizer reached an 80% for the speech and noise test and 10% recognition rate for the speech, noise, and reverberation test.

Spoken word	"Höger"	"Vänster"	No match	WER (%)
"Höger"	71	7	22	29
"Vänster"	6	92	2	8
Random speech	12	24	64	36

Table 6.1: Results of the two words each spoken 100 times, and random speech without mentioning the two words for the single microphone set-up, tested on the DSP.

Distance [m]	"Höger"	"Vänster"	No match	WER (%)
1	89	0	11	11
2	93	0	7	7
4	98	0	2	2

Table 6.2: Results of "Höger" being spoken 100 times for 1, 2 and 4 meters distance, tested offline with one microphone.

Distance [m]	"Höger"	"Vänster"	No match	WER (%)
1	0	99	1	1
2	0	100	0	0
4	0	100	0	0

Table 6.3: Results of "Vänster" being spoken 100 times for 1, 2 and 4 meters distance, tested offline with one microphone.

6.2.1 Speech

In this sub section the performance of the recognizer without noise and reverberation is given, thus only one microphone and no beamformer is used. The difference from this test and the test of the single microphone set-up, 6.1, is that this test use a different microphone and database and shows the efficiency of the recognizer algorithm for different distances, without potential disturbances from the DSP. This test consists of testing 100 "Höger" and "Vänster" for 1, 2, and 4 meters distance, see tables 6.2 and 6.3 for results.

6.2.2 Speech and Noise

In this sub section the results of tests with speech and noise for three distances, 1, 2, and 4 meter, are presented. Four types of noises are considered - white, factory, engine, and babble. All results from the speech and noises tests are summed up:

White noise Figures: 6.1, 6.2, 6.3 and 6.4 Table: 6.4

Factory noise Figures: 6.5, 6.6, 6.7 and 6.8 Table: 6.5

Engine noise Figures: 6.9, 6.10, 6.11 and 6.12 Table: 6.6

Babble noise Figures: 6.13, 6.14, 6.15 and 6.16 Table: 6.7

The tables shows the SNR level, for each distance and set-up, which the recognizer reached above a 80% recognition rate. The mark "-" means that the recognizer did not reach above an 80% recognition rate and the bold marked numbers indicates at which set-up the smallest SNR level was achieved, for one distance. In figures 6.4, 6.8, 6.12, and 6.16 errors are given. They show the errors on one meters distance, for four microphones and beamformer and one microphone, and

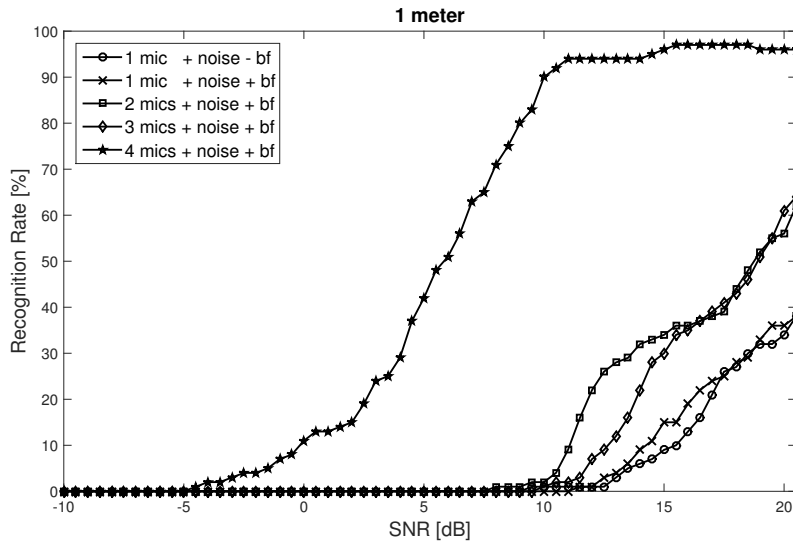


Figure 6.1: Results of the speech plus white noise at 1 meters distance, for the multiple microphone set-up.

no beamformer, is given. The presented errors are the deletion and both types of substitutions, no match found and wrong word chosen.

Distance [m]	1	2	4
1 mic - beamformer	-	-	-
1 mic + beamformer	-	-	-
2 mic + beamformer	-	-	-
3 mic + beamformer	-	-	-
4 mic + beamformer	**9**	**3.5**	**14.5**

Table 6.4: Results of the speech plus white noise test, showing the SNR [dB] which the recognizer reached above a 80% recognition rate.

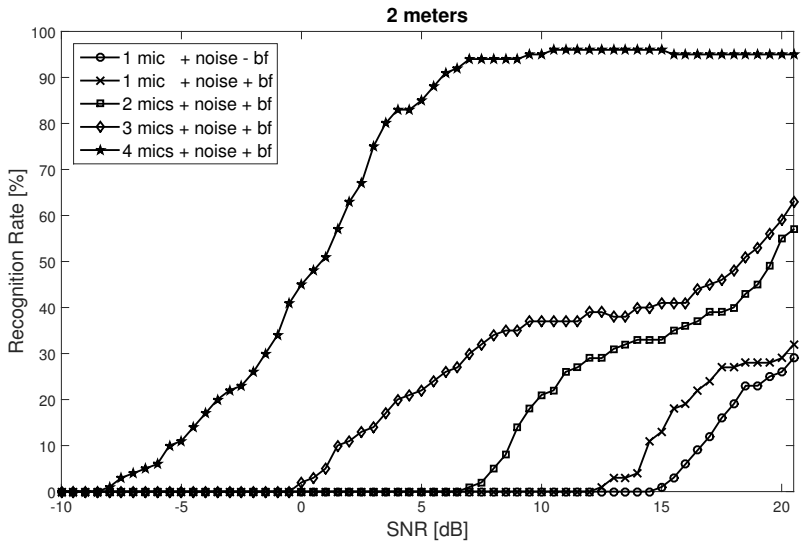


Figure 6.2: Results of the speech plus white noise at 2 meters distance, for the multiple microphone set-up.

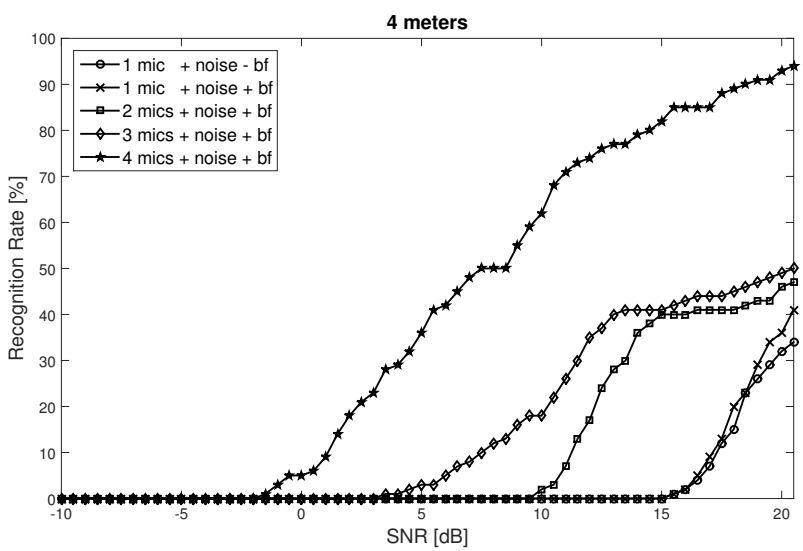


Figure 6.3: Results of the speech plus white noise at 4 meters distance, for the multiple microphone set-up.

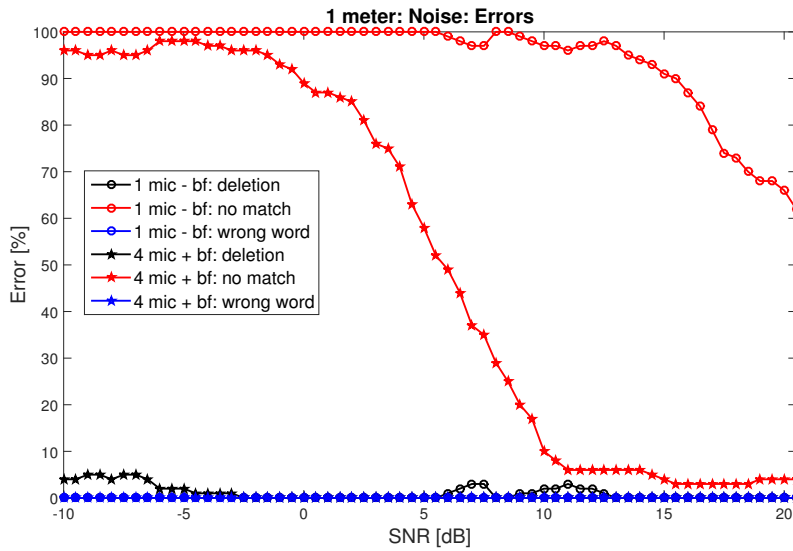


Figure 6.4: The errors (deletion, no match and wrong word) on one meters distance, for four microphones and beamformer and one microphone and no beamformer for the speech and white noise test.

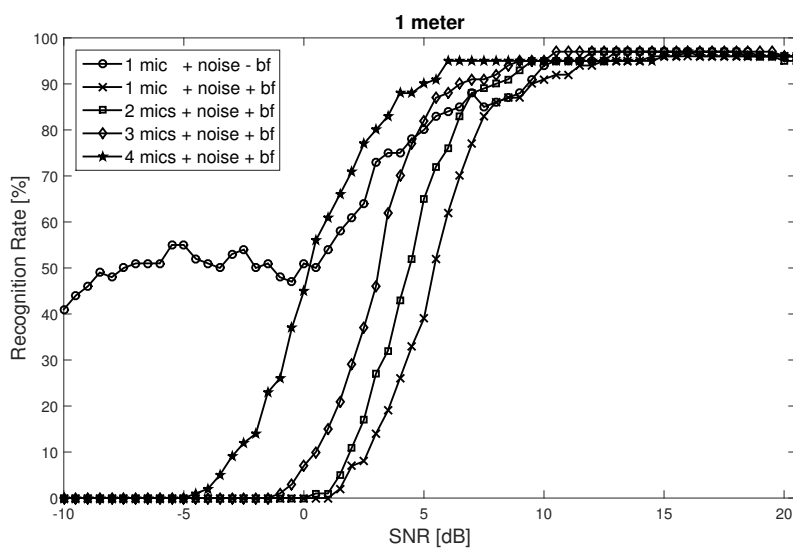


Figure 6.5: Results of the speech plus factory noise at 1 meters distance, for the multiple microphone set-up.

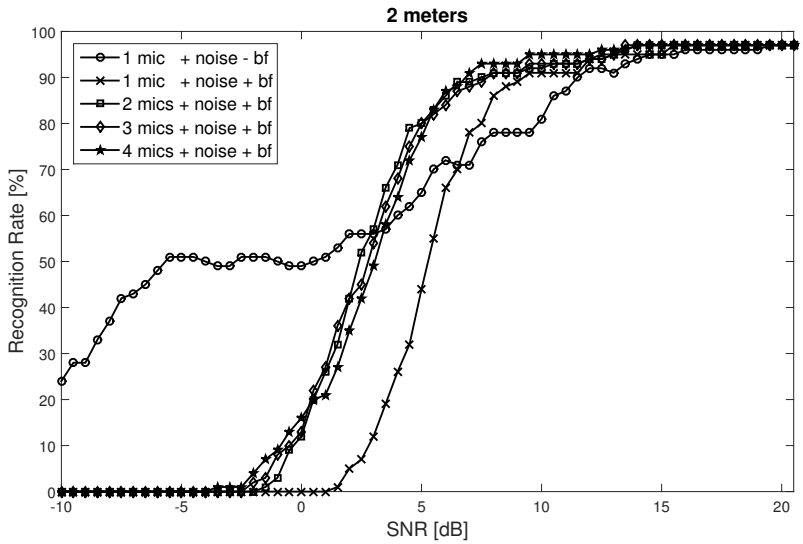


Figure 6.6: Results of the speech plus factory noise at 2 meters distance, for the multiple microphone set-up.

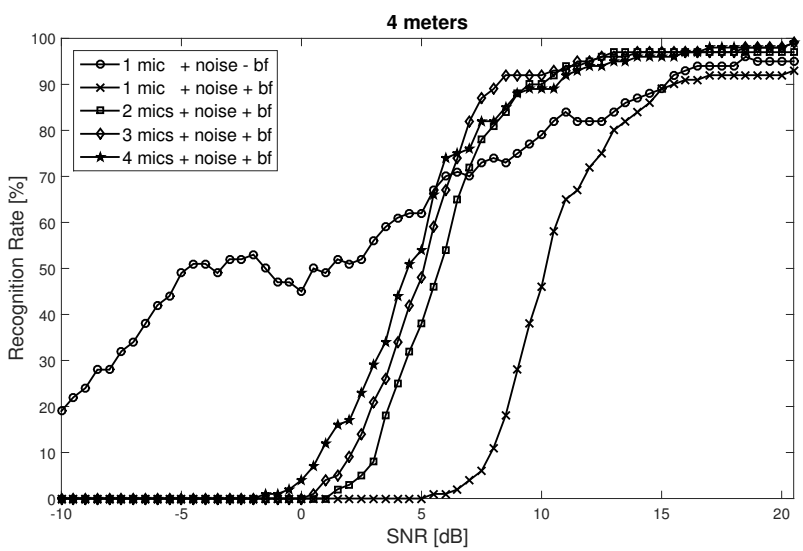


Figure 6.7: Results of the speech plus factory noise at 4 meters distance, for the multiple microphone set-up.

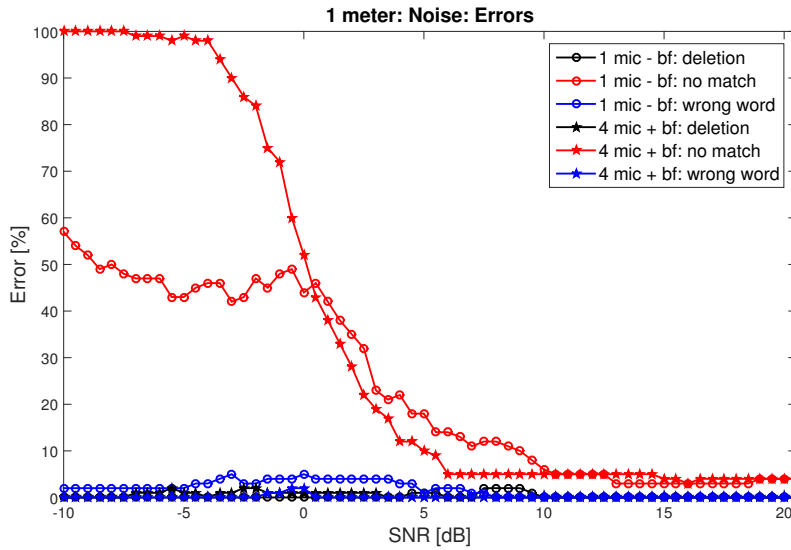


Figure 6.8: The errors (deletion, no match and wrong word) on one meters distance, for four microphones and beamformer and one microphone and no beamformer for the speech and factory noise test.

Distance [m]	1	2	4
1 mic - beamformer	5	10	10.5
1 mic + beamformer	7.5	7.5	13
2 mic + beamformer	6.5	**5**	8
3 mic + beamformer	5	**5**	**7**
4 mic + beamformer	**3**	5.5	7.5

Table 6.5: Results of the speech plus factory noise test, showing the SNR [dB] which the recognizer reached above a 80% recognition rate.

Distance [m]	1	2	4
1 mic - beamformer	9	13	11
1 mic + beamformer	**0.5**	**6.5**	**6**
2 mic + beamformer	10	11	10
3 mic + beamformer	17.5	12	9.5
4 mic + beamformer	12.5	12	9.5

Table 6.6: Results of the speech plus engine noise test, showing the SNR [dB] which the recognizer reached above a 80% recognition rate.

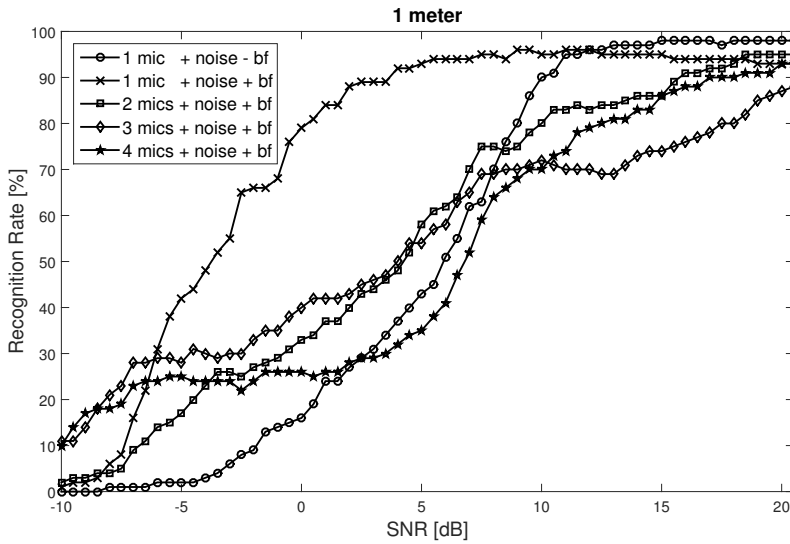


Figure 6.9: Results of the speech plus engine noise at 1 meters distance, for the multiple microphone set-up.

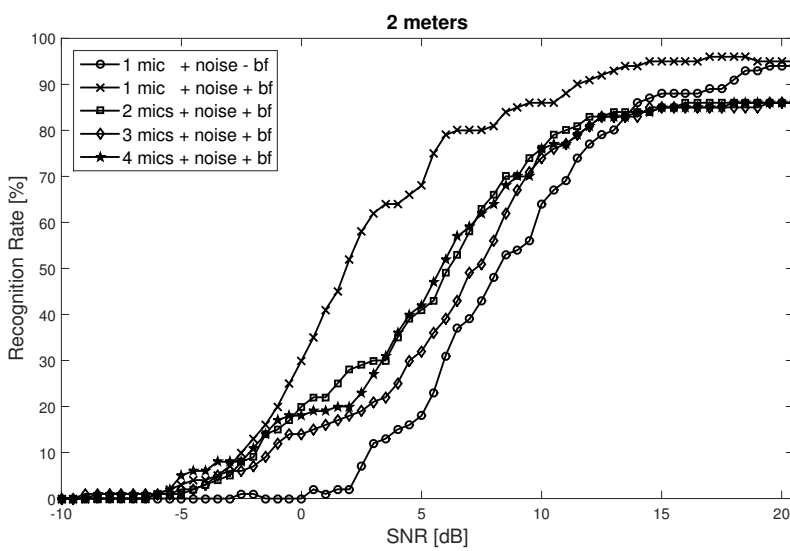


Figure 6.10: Results of the speech plus engine noise at 2 meters distance, for the multiple microphone set-up.

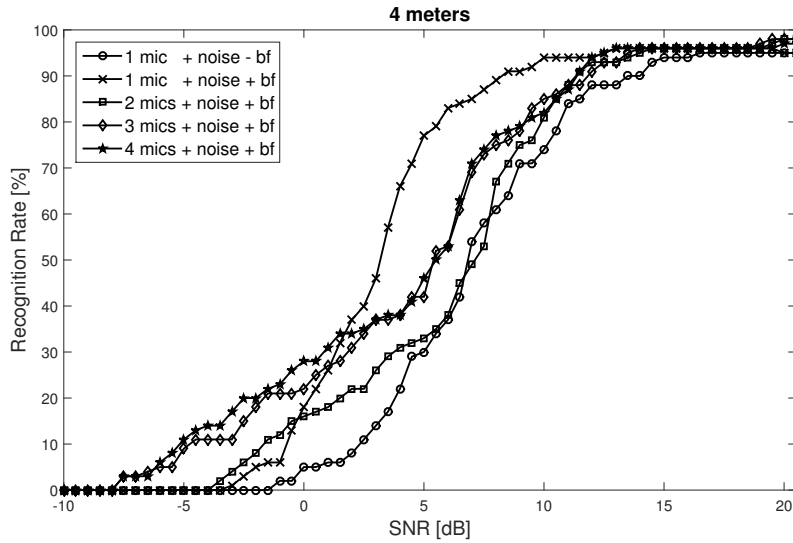


Figure 6.11: Results of the speech plus engine noise at 4 meters distance, for the multiple microphone set-up.

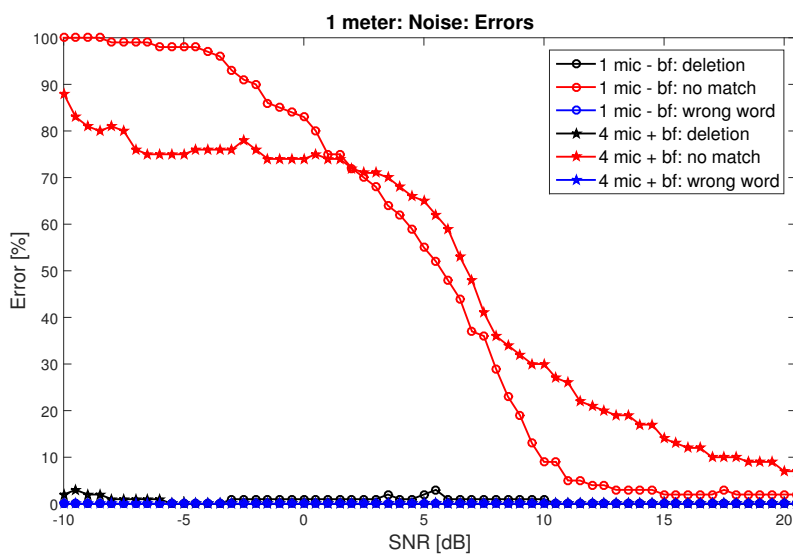


Figure 6.12: The errors (deletion, no match and wrong word) on one meters distance, for four microphones and beamformer and one microphone and no beamformer for the speech and engine noise test.

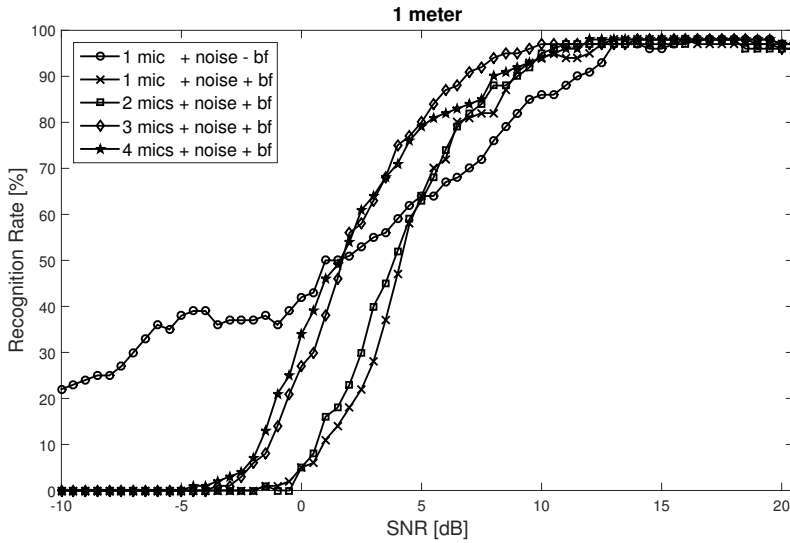


Figure 6.13: Results of the speech plus babble noise at 1 meters distance, for the multiple microphone set-up.

Distance [m]	1	2	4
1 mic - beamformer	9	12.5	10.5
1 mic + beamformer	6.5	14	-
2 mic + beamformer	7	12	15
3 mic + beamformer	**5**	9.5	13.5
4 mic + beamformer	5.5	**7**	**10**

Table 6.7: Results of the speech plus babble noise test, showing the SNR [dB] which the recognizer reached above a 80% recognition rate.

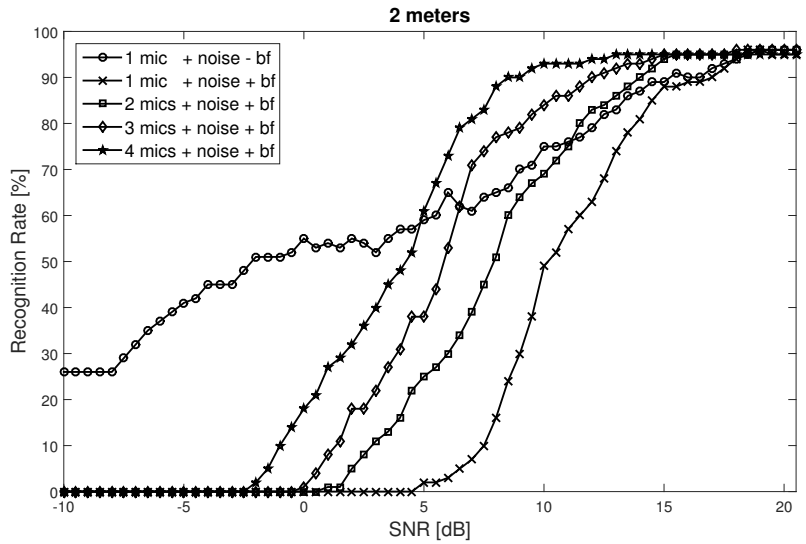


Figure 6.14: Results of the speech plus babble noise at 2 meters distance, for the multiple microphone set-up.

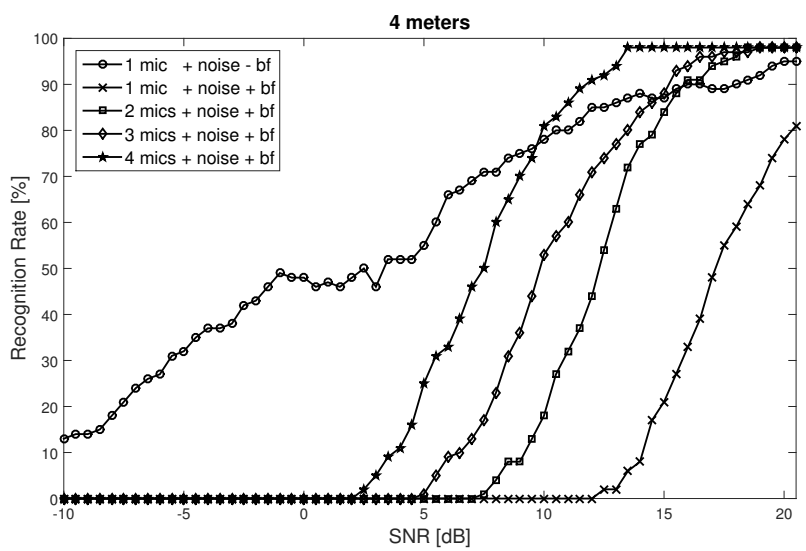


Figure 6.15: Results of the speech plus babble noise at 4 meters distance, for the multiple microphone set-up.

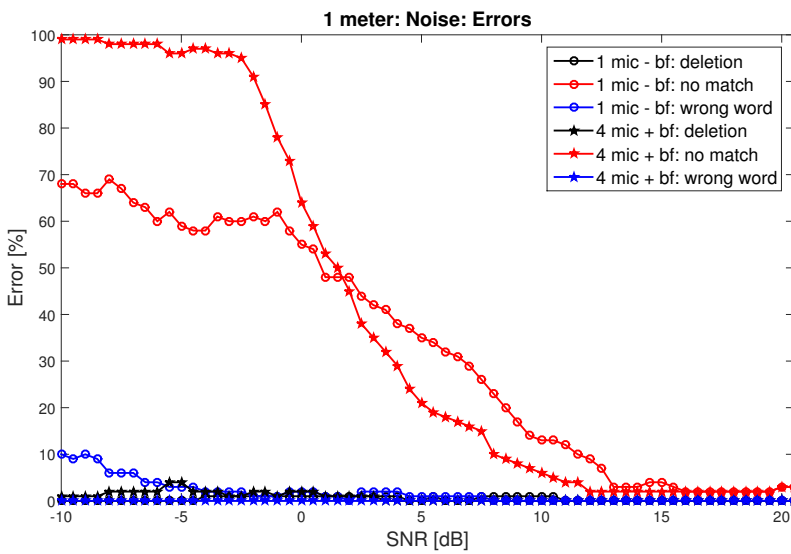


Figure 6.16: The errors (deletion, no match and wrong word) on one meters distance, for four microphones and beamformer and one microphone and no beamformer for the speech and babble noise test.

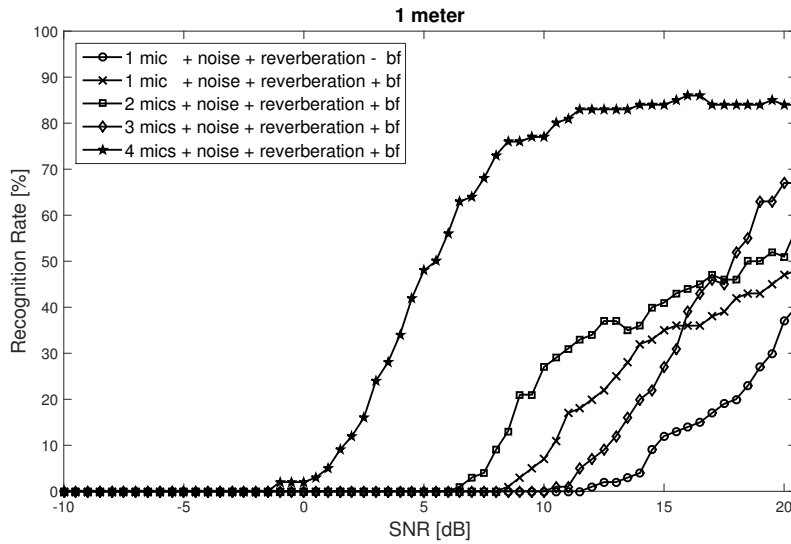


Figure 6.17: Results of the speech plus reverberation and white noise at 1 meters distance, for the multiple microphone set-up.

6.2.3 Speech, Noise and Reverberation

In this sub section the results of tests with speech, reverberation, and noise for three distances, $1, 2$ and 4 meter, is presented. Four types of noises is considered - white, factory, engine, and babble. The tables show the 10% recognition rate crossover instead of an 80% limit as many of the tests did not reach above 80%, and to be able to tell the relationship between the five cases for one distance, the limit had to be lowered. All results from the speech, reverberation, and noises tests are summed as:

White noise Figures: 6.17, 6.18, 6.19 and 6.20 Table: 6.8

Factory noise Figures: 6.21, 6.22, 6.23 and 6.24 Table: 6.9

Engine noise Figures: 6.25, 6.26, 6.27 and 6.28 Table: 6.10

Babble noise Figures: 6.29, 6.30, 6.31 and 6.32 Table: 6.11

The tables shows the SNR level, for each distance and set-up, which the recognizer reached above a 10% recognition rate. The mark "-" means that the recognizer did not reach above a 10% recognition rate and the bold marked numbers indicates at which set-up the smallest SNR level was achieved, for one distance. In figures 6.20, 6.24, 6.28, and 6.32 errors are given. They show the errors on one meters distance, for four microphones and beamformer and one microphone and no beamformer, is given. The presented errors are the deletion and both types of substitutions, no match found and wrong word chosen.

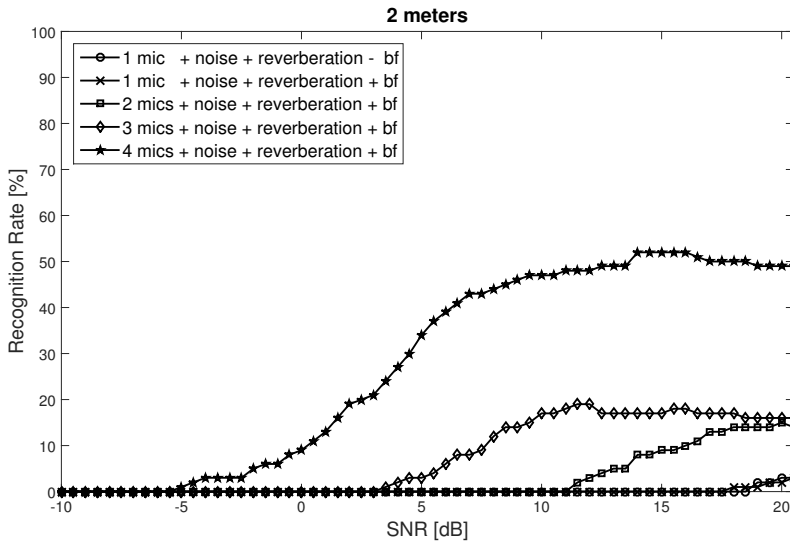


Figure 6.18: Results of the speech plus reverberation and white noise at 2 meters distance, for the multiple microphone set-up.

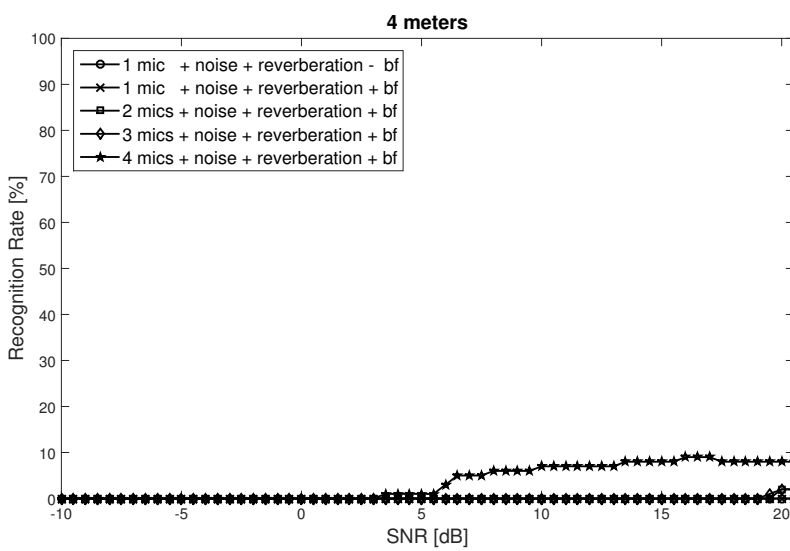


Figure 6.19: Results of the speech plus reverberation and white noise at 4 meters distance, for the multiple microphone set-up.

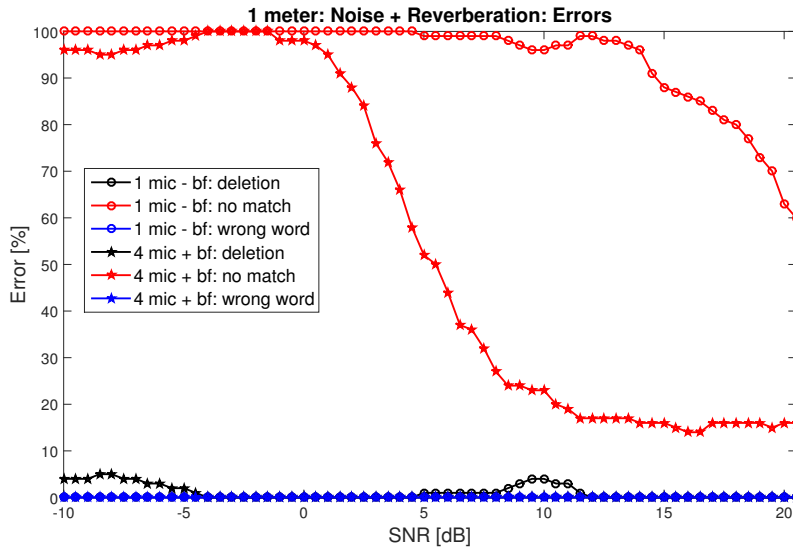


Figure 6.20: The errors (deletion, no match and wrong word) on one meters distance, for four microphones and beamformer and one microphone and no beamformer for the speech, white noise, and reverberation test.

Distance [m]	1	2	4
1 mic - beamformer	15	-	-
1 mic + beamformer	10.5	-	-
2 mic + beamformer	8.5	16	-
3 mic + beamformer	13	8	-
4 mic + beamformer	**2**	**0.5**	-

Table 6.8: Results of the speech plus reverberation and white noise test, showing the SNR [dB] which the recognizer reached above a 10% recognition rate.

Distance [m]	1	2	4
1 mic - beamformer	**-10**	**-10**	**-10**
1 mic + beamformer	4.5	12	-
2 mic + beamformer	6.5	7.5	-
3 mic + beamformer	5	5.5	-
4 mic + beamformer	2.5	5.5	16

Table 6.9: Results of the speech plus reverberation and factory noise test, showing the SNR [dB] which the recognizer reached above a 10% recognition rate.

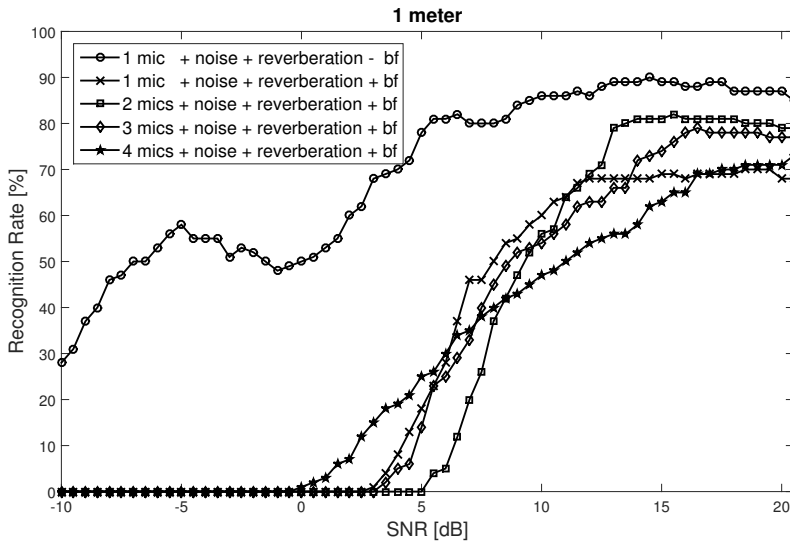


Figure 6.21: Results of the speech plus reverberation and factory noise at 1 meters distance, for the multiple microphone set-up.

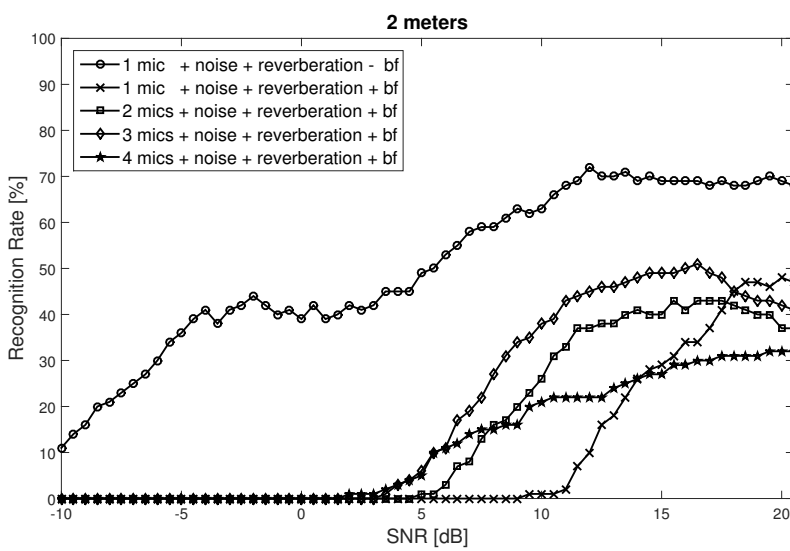


Figure 6.22: Results of the speech plus reverberation and factory noise at 2 meters distance, for the multiple microphone set-up.

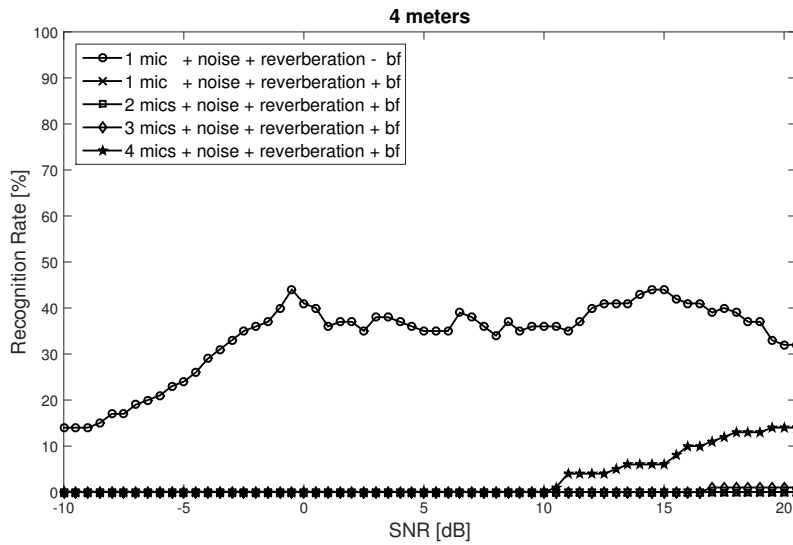


Figure 6.23: Results of the speech plus reverberation and factory noise at 4 meters distance, for the multiple microphone set-up.

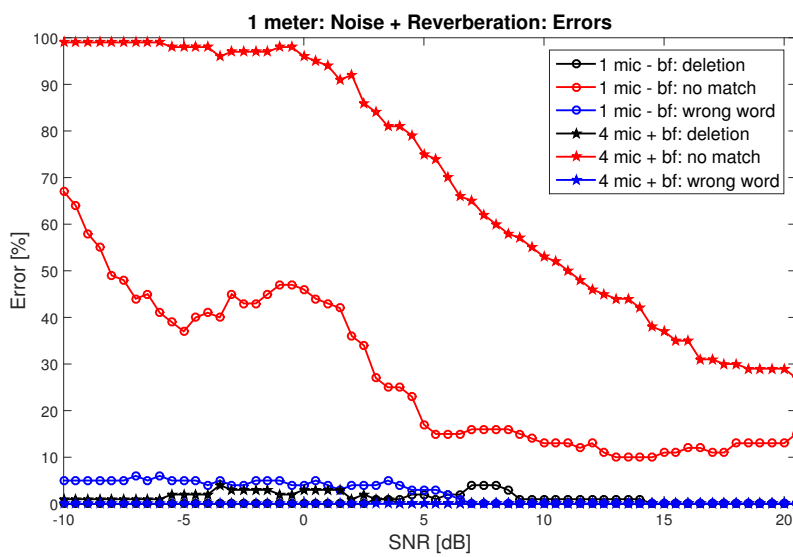


Figure 6.24: The errors (deletion, no match and wrong word) on one meters distance, for four microphones and beamformer and one microphone and no beamformer for the speech, factory noise, and reverberation test.

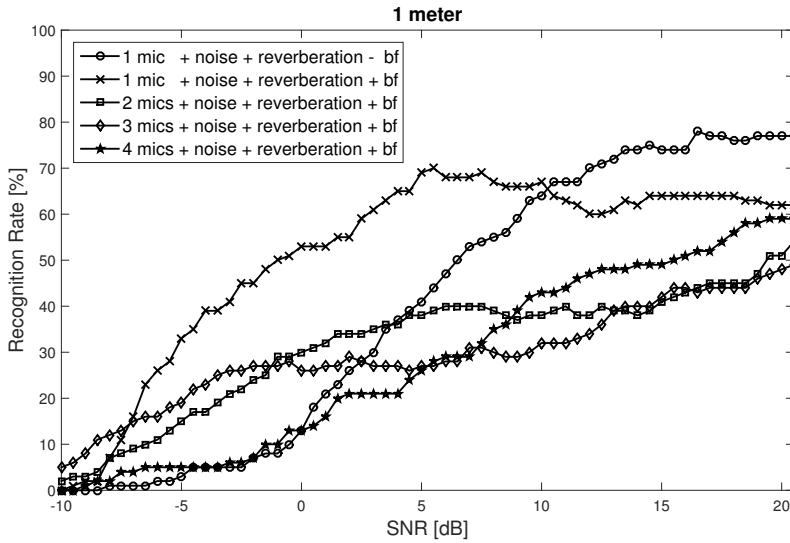


Figure 6.25: Results of the speech plus reverberation and engine noise at 1 meters distance, for the multiple microphone set-up.

Distance [m]	1	2	4
1 mic - beamformer	-0.5	7.5	**13.5**
1 mic + beamformer	-7.5	7	-
2 mic + beamformer	-6.5	7	-
3 mic + beamformer	**-8.5**	**6**	-
4 mic + beamformer	-1.5	6.5	-

Table 6.10: Results of the speech plus reverberation and engine noise test, showing the SNR [dB] which the recognizer reached above a 10% recognition rate.

Distance [m]	1	2	4
1 mic - beamformer	**-10**	**-8**	**-10**
1 mic + beamformer	5	14.5	-
2 mic + beamformer	4	10	-
3 mic + beamformer	2	8.5	-
4 mic + beamformer	1	6.5	-

Table 6.11: Results of the speech plus reverberation and babble noise test, showing the SNR [dB] which the recognizer reached above a 10% recognition rate.

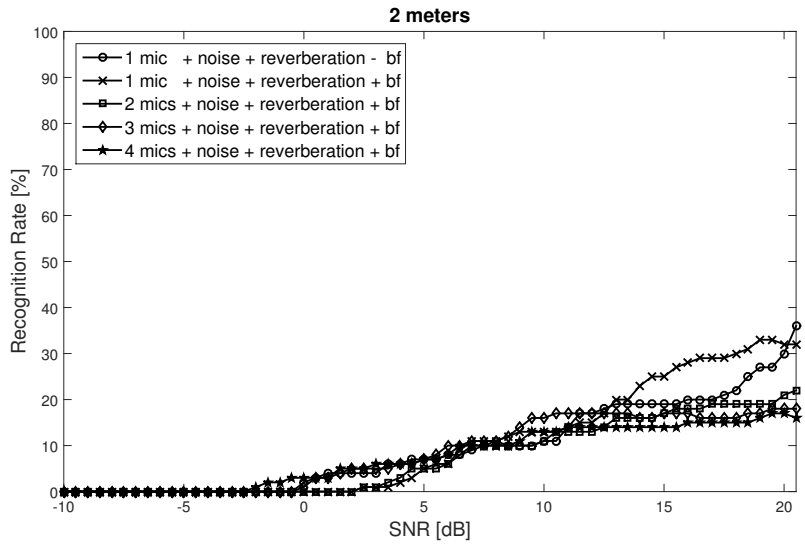


Figure 6.26: Results of the speech plus reverberation and engine noise at 2 meters distance, for the multiple microphone set-up.

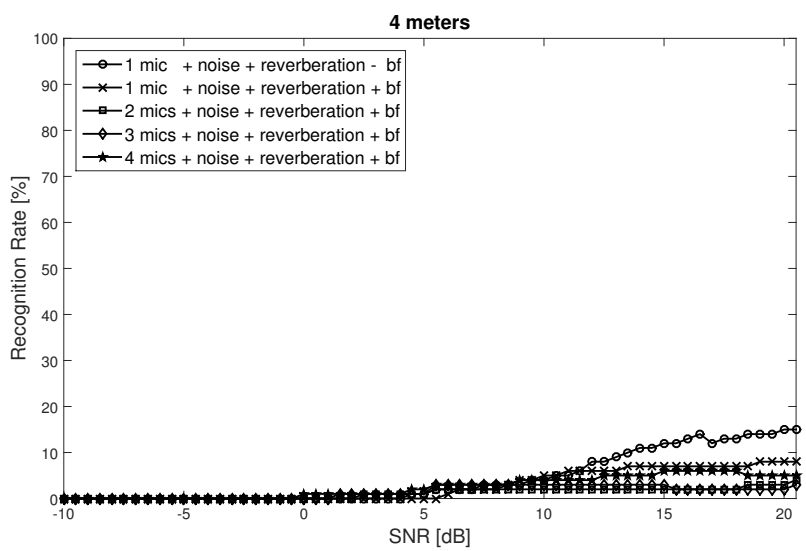


Figure 6.27: Results of the speech plus reverberation and engine noise at 4 meters distance, for the multiple microphone set-up.

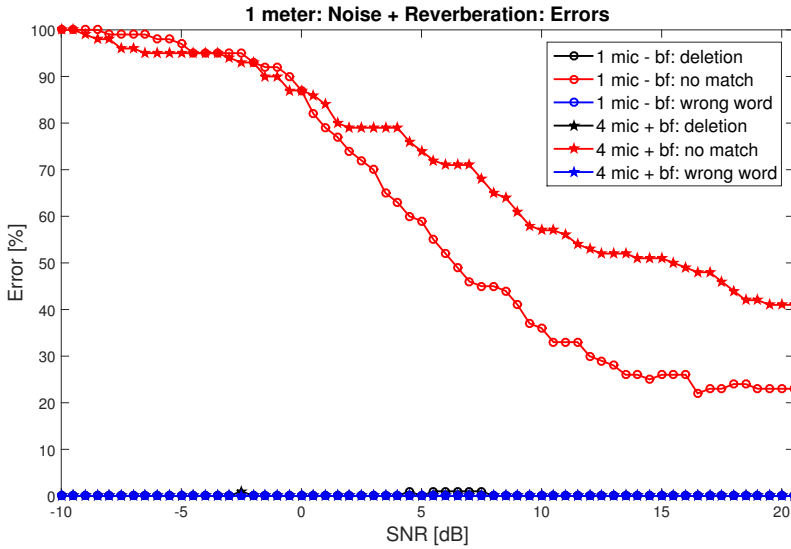


Figure 6.28: The errors (deletion, no match and wrong word) on one meters distance, for four microphones and beamformer and one microphone and no beamformer for the speech, engine noise, and reverberation test.

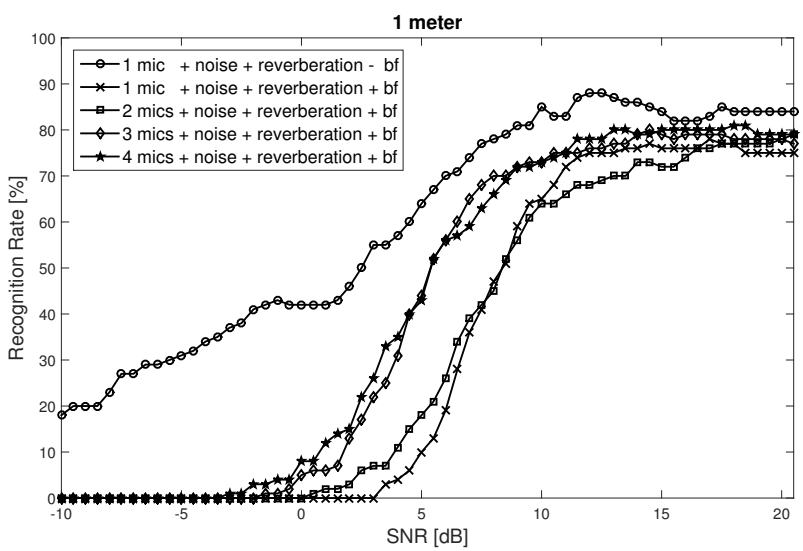


Figure 6.29: Results of the speech plus reverberation and babble noise at 1 meters distance, for the multiple microphone set-up.

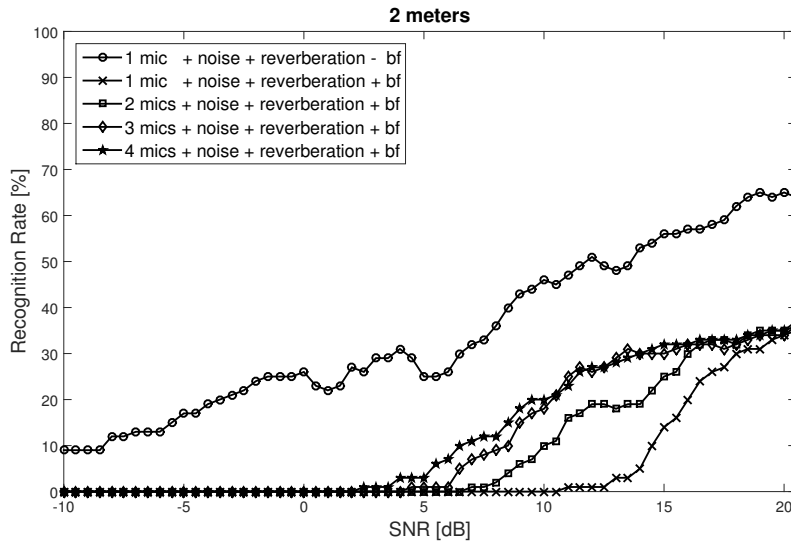


Figure 6.30: Results of the speech plus reverberation and babble noise at 2 meters distance, for the multiple microphone set-up.

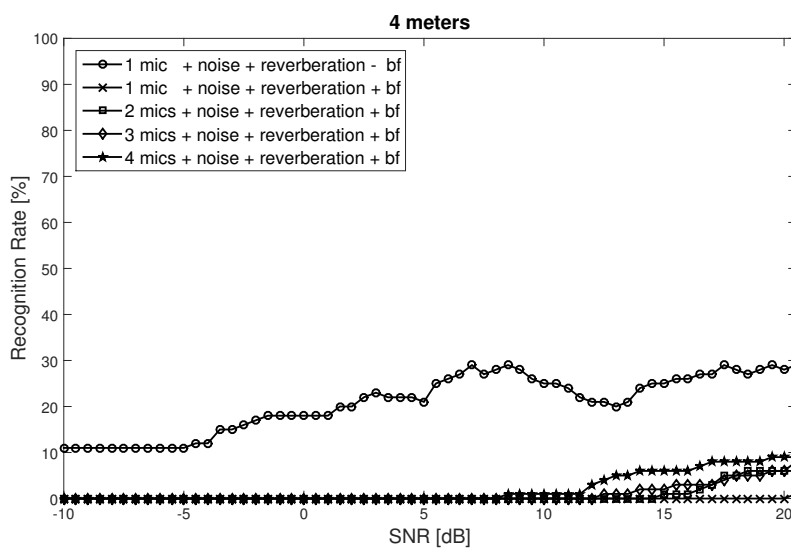


Figure 6.31: Results of the speech plus reverberation and babble noise at 4 meters distance, for the multiple microphone set-up.

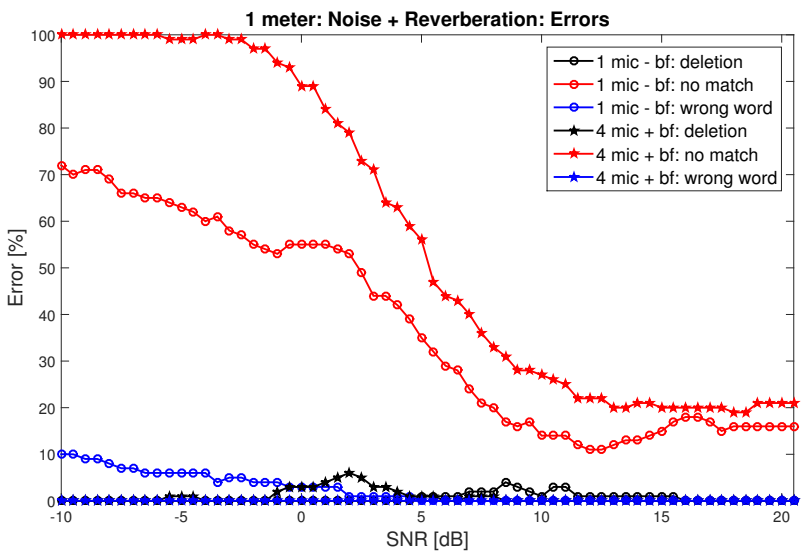


Figure 6.32: The errors (deletion, no match and wrong word) on one meters distance, for four microphones and beamformer and one microphone and no beamformer for the speech, babble noise, and reverberation test.

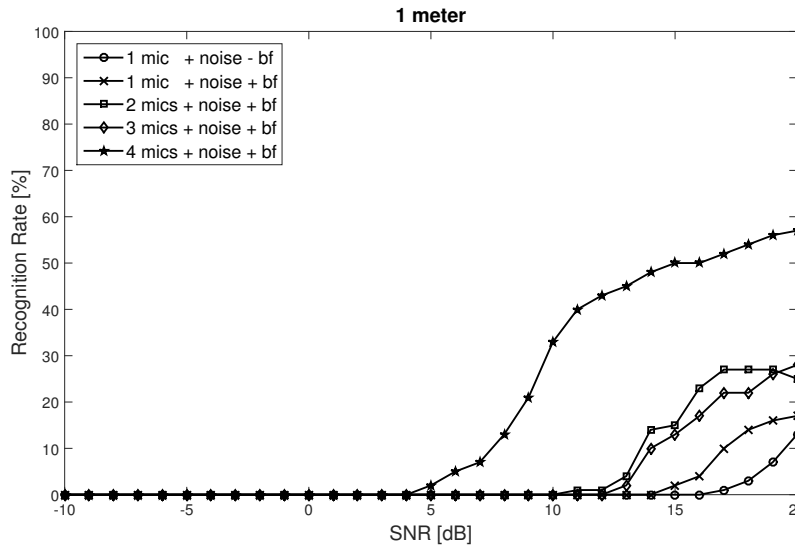


Figure 6.33: Results of real-time simulation considering speech and white noise at 1 meters distance, for the multiple microphone set-up.

6.2.4 Real-Time Simulation

In this subsection the results of the real-time simulation is given. The simulation considers speech embedded in long periods of white noise, which is a more realistic scenario than speech appearing very shortly into the speech + noise segment which is the scenario for the tests performed on the multiple microphone set-up. The results are seen in figures 6.33, 6.34 and 6.35. In table 6.12 the SNR level, for each distance and set-up, which the recognizer reached above a 10% recognition rate. The mark "-" means that the recognizer did not reach above a 10% recognition rate and the bold marked numbers indicates at which set-up the smallest SNR level was achieved, for one distance. In figure 6.36 the errors on one meters distance, for four microphones and beamformer and one microphone and no beamformer, is given. The presented errors are the deletion and both types of substitutions, no match found and wrong word chosen.

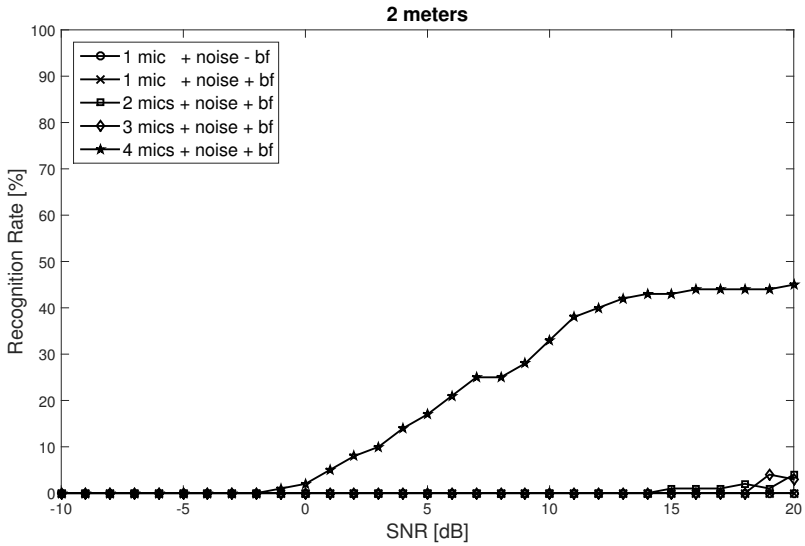


Figure 6.34: Results of real-time simulation considering speech and white noise at 2 meters distance, for the multiple microphone set-up.

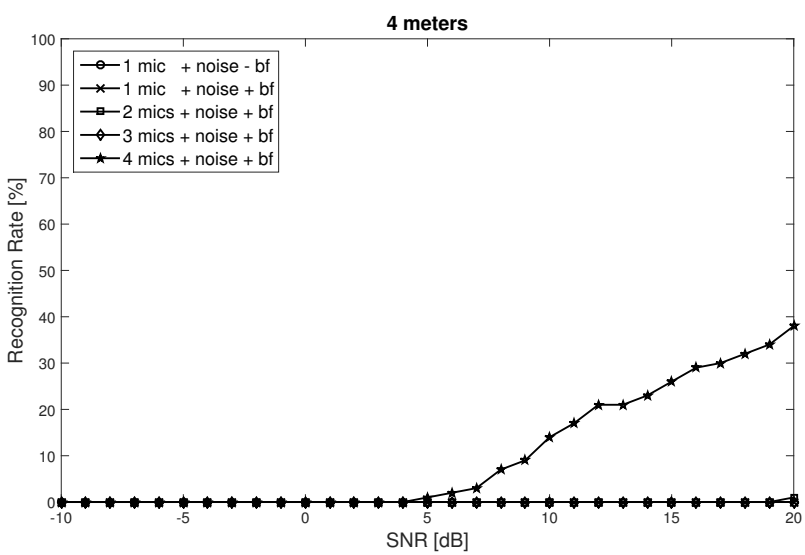


Figure 6.35: Results of real-time simulation considering speech and white noise at 4 meters distance, for the multiple microphone set-up.

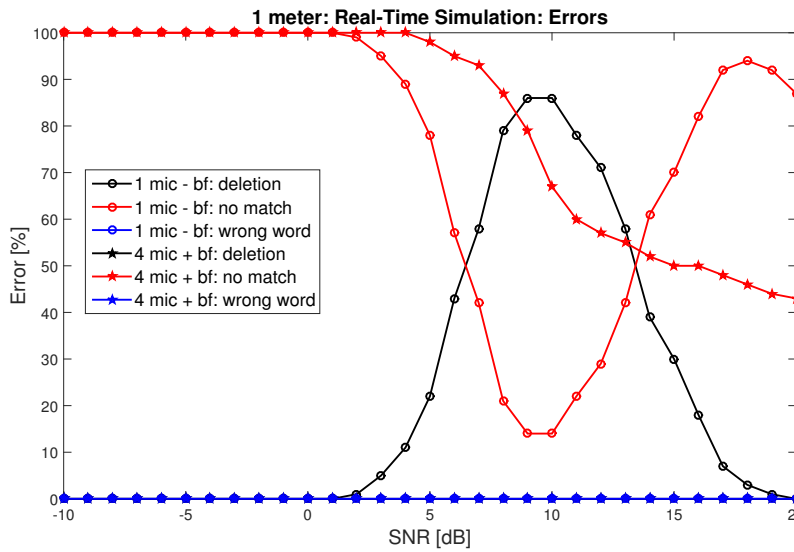


Figure 6.36: The errors (deletion, no match and wrong word) on one meters distance, for four microphones and beamformer and one microphone and no beamformer for the speech and white noise real-time simulation.

Distance [m]	1	2	4
1 mic - beamformer	20	-	-
1 mic + beamformer	18	-	-
2 mic + beamformer	14	-	-
3 mic + beamformer	15	-	-
4 mic + beamformer	**8**	**4**	**10**

Table 6.12: Results of real-time simulation considering speech and white noise test, showing the SNR [db] which the recognizer reached above a 10% recognition rate.

Chapter 7

Analysis and Conclusion

In this chapter the results presented in chapter 6 is analyzed and conclusions thereof drawn. The single microphone set-up and multiple microphone set-up are analyzed separately and an overall conclusion is then drawn.

7.1 Single Microphone Set-Up

The analysis in this section focus on the overall performance of the recognizer, when having poor conditions. The poor conditions are given by using a bad microphone for the real-time tests and for recording the database. The database is also not very large.

The implemented DSR can both conduct identification of two different words and perform validation to see if the spoken word is in the database at all. From the results in 6.1 some interpretations and conclusions can be drawn. It is noticed that "Höger" has a Word Error Rate (WER) 3.6 times larger than "Vänster". One can also notice that for the random speech test "Vänster" is matched twice as often as "Höger". These results could indicate that for example:

- The features of "Höger" are too unique.
- The features of "Vänster" are too generic.
- The features of the words are not extracted properly, thus theLPC is not an optimal method and switching to another feature extracting method could be necessary to reach higher performance.
- The recorded database and the real-time recording may differ in some sense, since the real-time recording is done through the DSP and the database is recorded via an external audio interface through Matlab on a laptop.
- The database may not have enough variation among the entries.
- The table microphone used to record the database and perform the tests is not very good.

7.2 Multiple Microphone Set-Up

In this section the analysis focus on performance of the beamformer being influenced by different noise types and reverberation. An analysis of the real-time analogy will also be presented.

7.2.1 Speech

In this sub section the performance of the recognizer without noise and reverberation is analyzed, thus only one microphone and no beamformer is used. In tables 6.2 and 6.3 the results are presented. One can see that "Vänster" reached better WER than "Höger", and that all of the cases when the words was not recognized correctly was not a substitution but was decided to be "No match". It can also be noted that for both words, the recognition was higher for the distances further away. Conclusions that could be drawn from these results are:

- "Vänster" has either more unique, or more generic, characteristics than "Höger".
- The matching algorithm performs well as it can distinguish the two words from each other.
- Speaking close to the microphones could possibly pick up breathing noise from the speaker. Especially since "Höger" has noise like characteristics in the beginning of the vocalization. See appendix A in figure A.2 for a graph of pronunciation of the letter "H".

7.2.2 Speech and Noise

In this sub section the results of tests with speech and noise for three distances, $1, 2$ and 4 meter, is analyzed. Four types of noises is considered - white, factory, engine, and babble. The focus lies on whether beamforming increases the performance, and what differences there is between different noise types. The results which is considered in this sub section is:

White noise Figures: 6.1, 6.2, 6.3 and 6.4 Table: 6.4

Factory noise Figures: 6.5, 6.6, 6.7 and 6.8 Table: 6.5

Engine noise Figures: 6.9, 6.10, 6.11 and 6.12 Table: 6.6

Babble noise Figures: 6.13, 6.14, 6.15 and 6.16 Table: 6.7

White Noise

From the graphs and the table it is clear that using four microphones and beamforming has better performance than the other cases, as this case is the only one that reached above an 80% recognition rate. The errors consist primarily of the substitution no match found, but there also occurs a small amount of deletions for both of the presented cases. It can also be noted that the performance is significantly increasing with the increasing number of microphones and added beamforming. The highest reached recognition rate is:

1 meter: 4 mics and beamformer, 97%

2 meter: 4 mics and beamformer, 96%

4 meter: 4 mics and beamformer, 94%

Factory Noise

In the test results with speech and factory noise it can be seen that for one meters distance the increasing number of microphones increases the recognition rate. But one microphone and no beamformer performs quite well, especially before 0 dB SNR, and reaches 80% recognition rate at the same SNR level as three microphones and beamformer at one meters distance. But the recognition rate of one microphone and no beamformer increases slower with increasing SNR than the other cases. The errors in the tests consist of primarily of no match found, but also a few wrongly matched words and deletions. These errors occur for both cases. The highest reached recognition rate is:

1 meter: 1 mic and no beamformer and 2, 3, and 4 mics and beamformer, 97%

2 meter: all cases, 97%

4 meter: 3 and 4 mics and beamformer, 99%

Engine Noise

The results from the speech and engine noise tests are somewhat different than the previous two. Here, it can be seen in the figures that one microphone and beamformer is the superior case amongst the five cases presented. Superior in the sense that its recognition rate increases faster with the increasing SNR, than the other cases. As for the errors one can see that, for both cases, the errors are dominated by the substitution case no match found, but one can also find that a couple of deletions occurs. The highest recognition rate is not uniformly achieved by one case, but it is different for the three distances:

1 meter: 1 mics and no beamformer, 98%

2 meter: 1 mics and beamformer, 96%

4 meter: 3 and 4 mics and beamformer, 98%

Babble Noise

Test results from the speech and babble noise tests shows that four microphones and beamformer is the superior one. From the three graphs it can be seen that the further the distance the more significantly better four microphones and beamformer is. It should also be noted that the performance is increasing with the number of microphones used. One microphone and no beamformer shows an almost linearly increasing recognition rate for the three distances. When looking at the types of errors that occur one can see that most errors are no match found for both cases, but there exist some wrongly matched words for the one microphone and no beamformer case. There also occur a few deletions for both cases. The highest recognition rate reached is achieved by different cases for each distance:

1 meter: all cases, 98%

2 meter: 1, 2 and 3 mics and beamformer, 96%

4 meter: 2, 3, and 4 mics and beamformer, 98%

Summing Up

To summarize the discussed results in this sub section one could draw the following conclusions:

- The recognizer performs differently depending on the noise environment.
- For white, factory, and babble noise four microphones and beamformer is the superior case.
- For white, factory, and babble noise the performance increases with the number of microphones and utilizing the beamformer technique.
- For factory and babble noise, one microphone and no beamformer performs significantly better than the other cases for an SNR level below 0 dB. This could be as this case only uses high- and low-pass filtering to remove noise. The cases using beamforming performs Wiener filtering which could alter the characteristics of the speech such that no match can be found. Looking at the characteristics of the factory and babble noise, seen in figures 5.3 and 5.5, one can see that they are quite similar. This could explain the similar behavior.
- The primary error type the substitution case no match found, for all noises and distances. This points to that the characteristics of the speech is not sufficiently thoroughly represented. But it also points to that the matching algorithm works well as it rather than choosing the wrong word decides for no word.

- For all noise types, the deletions error occur in a small amount for both cases, but on different SNR levels. It occurs around -10 dB and -5 dB for four microphones and beamformer and around 5 dB and 10 dB for one microphone and no beamformer. It is interesting that at low SNR levels, for all noise types, the no match error is vastly superior and not deletions. This as one would assume that it would be hardest to find a word in the noisiest SNR levels, and that the no match error would peak in the middle regions of the SNR levels. This hints that the VAD might not perform to complete satisfaction.
- For factory and babble noise errors one can notice that the wrong word matched error is only found for these two noises, primarily for the one microphone and no beamformer case. This error occur only in the lower SNR levels, around -10 dB and 7 dB. This could be explained by that the VAD detects the speech a little wrong, in combination with that the beamformer does not remove enough noise.
- The tonal engine noise shows results which differ from conclusions stated above. Which could be explained by characteristics of the noise, see figure 5.4. Due to the fact that the noise type lies in the same frequency range as speech it mixes well and is hard to remove.

7.2.3 Speech, Noise and Reverberation

In this sub section the results of tests with speech, noise, and reverberation for three distances, $1, 2$, and 4 meter, is analyzed. Four types of noises is considered - white, factory, engine, and babble. The focus lies on whether beamforming increases the performance when reverberations is present, and what differences there is between different noise types. The results which is considered in this sub section is:

White noise Figures: 6.17, 6.18, 6.19 and 6.20 Table: 6.8

Factory noise Figures: 6.21, 6.22, 6.23 and 6.24 Table: 6.9

Engine noise Figures: 6.25, 6.26, 6.27 and 6.28 Table: 6.10

Babble noise Figures: 6.29, 6.30, 6.31 and 6.32 Table: 6.11

White Noise

From the test results from the speech, white, and reverberations tests one can see that four microphones and beamformer is superior. One can also see that the recognition rate increases with number of microphones. At one meters distance one should notice that at start of rise in recognition rate the order is four, two, one, and tree microphones and beamformer and lastly one microphone without beamformer. But when looking at the highest reached recognition rate the order is four, three, two, and one microphone and beamformer and lastly one microphone and beamformer. For two meters start of rise and highest recognition rate has the same order among the cases - four, three, two, and one microphone and beamformer and lastly one microphone and no beamformer. The errors, for both cases, consist mainly of the substitution case no match found, but there also occur a few deletions. The overall highest reached recognition rate values are:

1 meter: 4 mics and beamformer, 86%

2 meter: 4 mics and beamformer, 52%

4 meter: 4 mics and beamformer, 9%

Factory Noise

For the speech, factory noise and reverberation test results one can see that one microphone and no beamformer manages the best. One should notice that the order of start in rise of recognition rate and the highest reached recognition rate are all different for the three distances. But one can see that four microphones and beamformer is the second best case when looking at start in rise time. As for the errors both cases primarily consist of the substitution no match, but they also have a few deletions. One microphone and no beamformer also displays that on several occasions the wrong word was decided upon. The overall highest reached recognition rate values are:

1 meter: 1 mic and no beamformer, 90%

2 meter: 1 mic and no beamformer, 72%

4 meter: 1 mic and no beamformer, 44%

Engine Noise

The results from the speech, engine noise, and reverberation tests shows a rather inconsistent behavior. It seems as one microphone and beamformer is somewhat faster in increasing recognition rate for one and two meters distance, and the highest achieved recognition rate is one microphone without beamformer for both distances. For four meters distance one microphone and no beamformer is both the fastest increasing and highest reached recognition rate of the five cases. There does not seem to be a connection between increasing number of microphones and increasing recognition rate. When looking at the types of errors that occur one can see that all most all errors are the substitution no match found and very fem are deletions, for both displayed cases. The highest reached recognition rate values are:

1 meter: 1 mic and no beamformer, 78%

2 meter: 1 mic and no beamformer, 36%

4 meter: 1 mic and no beamformer, 15%

Babble Noise

From the speech, babble noise and reverberation test results one can see that one microphone and no beamformer is superior as this case has both fastest increasing recognition rate and reaches the highest recognition rate. One should also notice that, for all three distances, the order four, three, two, and one microphone and beamformer follows subsequently after one microphone and no beamformer, in the increase in recognition rate aspect. The errors, for both cases, consist mainly of no match found but also some deletions. One microphone and no beamformer also experience some wrongly matched words. The overall highest reached recognition rate values are:

1 meter: 1 mic and no beamformer, 88%

2 meter: 1 mic and no beamformer, 65%

4 meter: 1 mic and no beamformer, 29%

Summing Up

To summarize the discussion in this sub section, one can conclude that:

- Reverberant speech signals severely damage the performance of the recognizer.
- Utilizing the beamforming technique on signals with reverberation is damaging the performance more than increasing it. The reason for this decline in performance is as no action towards handling to effect of echoes has been taken before the characteristics of speech is extracted. Thus, the quality of the characteristics of the speech could be compromised. This, in combination with the possibilities that the Wiener filtering performed in the beamforming technique changes the characteristics of the speech, creates a poor situation which cannot be resolved with the current implementation.
- As for the types of errors the different noise types experience, the composition is similar to the speech and noise tests in previous sub section.

7.2.4 Real-Time Simulation

In this sub section the results from the real-time simulation test are analyzed. The results which are considered can be seen in figures 6.33, 6.34, 6.35, and 6.36 and in table 6.12. It can be seen that four microphones and beamformer superior over the other cases, for all three distances. From the errors one can see that four microphones and beamformer only experience the substitution error no match. Whereas, rather surprisingly, one microphone and no beamformer experience the no match error up to around -10 dB where there occurs a major increase in deletion errors. The overall highest reached recognition rate values are:

1 meter: 4 mics and beamformer, 57%

2 meter: 4 mics and beamformer, 45%

4 meter: 4 mics and beamformer, 38%

If comparing these results to the speech and white noise results, figures 6.1, 6.2, 6.3, and 6.4 and table 6.4, one can see that for one meters distance they resemble each other in behavior. The only difference is that the general recognition rate for the real-time simulation is a scaled down version of the speech and noise test. As for the difference in errors, the real-time simulation experience significantly more deletion errors in the one microphone and no beamformer case. A reason for this increase could not be thought of, as one would think that it would be harder to detect speech when it is more noisy and not suddenly at 10 dB. To summarize this sub section, one could conclude that:

- Four microphones and beamformer handles the real-time situation the best, and one microphone without beamformer the worst.

- As there are more deletion errors for the real-time simulation than the speech and noise test, it signals that the VAD cannot find the speech very well in segments of speech embedded in long periods of noise. The location of the SNR level is also quite interesting. This as one would assume that it would be harder to find speech at the lower SNR levels and that the number of no match errors would have it maximum around 0 dB. Thus a VAD better suited for a real-time situation is required.

7.3 Conclusion

Most results from speech and noise tests points to that four microphones and using a beamforming technique is preferable, and the results also indicate that the more microphones used, the better the recognition rate one gets. With the exception with engine noise, which gives quite unintelligible results. One can also conclude that reverberation severely impair the recognition rate and that beamforming is not preferable in very reverberant environments, and in those environments one microphone and no beamforming gives the best results. From the speech test, real-time simulation, and error graphs of speech and noise and speech, noise, and reverberation, one could also draw the conclusion that the matching algorithm works nicely as it did not chose the wrong word, but the VAD is does not reach a satisfactory effect. This also signals that the characteristics of speech can not be adequately represented.

Chapter 8
Recommendations

In this chapter recommendations and tips for further research within DSR area is given.

As the analysis concluded reverberation severely impair the recognition rate, thus some kind of action towards getting rid of these unwanted disturbances needs consideration. Echo cancellation is one technique which could lift this restriction in user environment. Also as concluded in previous chapter the VAD is this thesis lacked and the algorithm could benefit from further development.

Switching to an adaptive beamformer which follows the changes in noise and updates the filter as the environment changes, should improve the beamformers ability to cancel out noise.

By letting the speech features in cepstral representation be transformed by the use of HMM, which divides the speech into separate phones one would allow the recognizer to be more agile towards different pronunciations and therefore more accurate in the representation of speech. This would result in a simplification of the introduction of different speakers [5, p.263] [23].

Other ideas which could be considered to be tested is vocal tract normalization, which could increase the robustness if having different speakers [5, p.318]. The recognizer may need adaptive gain control(AGC) which would stabilize the volume thus increasing robustness in environments with varying SNR. Also looking into noise estimation algorithms which estimate and update the noise spectrum continuously which could in improving the attenuation of noise [16, p. 432].

Last tip, in this thesis the recognizer is speaker dependent, which is a rather large restriction. To lift this restriction one would need a much larger database. This would however be impossible to store on the recognizer itself, due to limited memory. But if utilizing a server based database this would be avoided, although this recognizer would be required to be implemented a device which could connect to the internet and not on the DSP used in this thesis.

Chapter 9
Bibliography

[1] Will Knight. Where speech recognition is going. http://www.technologyreview.com/news/427793/where-speech-recognition-is-going/, May 2012.

[2] Megan Wollerton. Voice control comes to the forefront of the smart home. http://www.cnet.com/news/voice-control-roundup/, December 2014.

[3] Global voice recognition market 2015-2019. pages 1–64, January 2015.

[4] Roland Banks. Voice recognition – has it come of age? http://www.mobileindustryreview.com/2014/10/voice-recognition-come-of-age.html, October 2014.

[5] Matthias Wölfel and John McDonough. *Distant Speech Recognition*. Wiley, first edition, 2009.

[6] ETIN80. Course web page. http://www.eit.lth.se/index.php?ciuid=821&coursepage=kursfakta&L=1, January 2015.

[7] Herbert N. Casson. *The History of the Telephone*. Cosimo, originally published in 1910 edition, 2006.

[8] Biing-Hwang Juang and Lawrence R Rabiner. Automatic speech recognition–a brief history of the technology development. *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, 1, 2005.

[9] Wikipedia. Speech recognition. http://en.wikipedia.org/wiki/Speech_recognition#History, April 2015.

[10] Wikipedia. Nuance Communications. http://en.wikipedia.org/wiki/Nuance_Communications, April 2015.

[11] Generating and understanding speech. http://www.ecophon.com/en/resources/acoustic-knowledge-bank/Basic-Acoustics/Acoustics-sound-speech-and-hearing/Generating-and-understanding-speech/ l, April 2015.

[12] Kenneth Crannell. *Voice and Articulation*. Cengage Learning, January 2011.

[13] Picture of vocal tract and larynx. http://www.csi.ucd.ie/staff/fcummins/phon98/vtract.html, April 2015.

[14] Simon Haykin. *Adaptive Filter Theory*. Pearson, (international) fifth edition, 2014.

[15] Sanjit K. Mitra. *Digital Signal Processing - A Computer-Based Approach*. McGraw-Hill, (international) second edition, 2002.

[16] P. C. Loizou. *Speech Enhancement - Theory and Practice*. CRC Press Inc., second edition, 2013.

[17] T. Nakatani, W. Kellermann, P. Naylor, M. Miyoshi, and B. H. Juang. Introduction to the special issue on processing reverberant speech: Methodologies and applications. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(7):1673–1675, Sept 2010.

[18] O. Schwartz, S. Gannot, and E. A. P. Habets. Multi-Microphone Speech Dereverberation and Noise Reduction Using Relative Early Transfer Functions. *Audio, Speech, and Language Processing, IEEE/ACM Transactions*, 23(2):240 – 251, November 2014.

[19] Dimitris G Manolakis, Vinay K Ingle, and Stephen M Kogon. *Statistical and adaptive signal processing: spectral estimation, signal modeling, adaptive filtering, and array processing*, volume 46. Artech House Norwood, 2005.

[20] Johan Nisula and Sebastian Krill. Acoustic solutions for door stations (ms thesis). *Department of Electrical and Information Technology, Lund University*, 2015.

[21] Stephan Weiss and Wei Liu. *Wideband Beamforming: Concepts and Techniques*. Wiley, 2010.

[22] Mikael Swartling. Direction of arrival estimation and localization of multiple speech sources in enclosed environments. 2012.

[23] Mark Gales and Steve Young. The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3):195–304, 2008.

Appendix A

Swedish Alphabet in Graphs

In this appendix the Swedish alphabet is presented in graphs in the figures A.1-A.4. The pronunciation of the letters are as reading the letters, and not as when speaking. For example the letter "s" are spoken as "es", and not "sss". The letters w and z have been omitted.

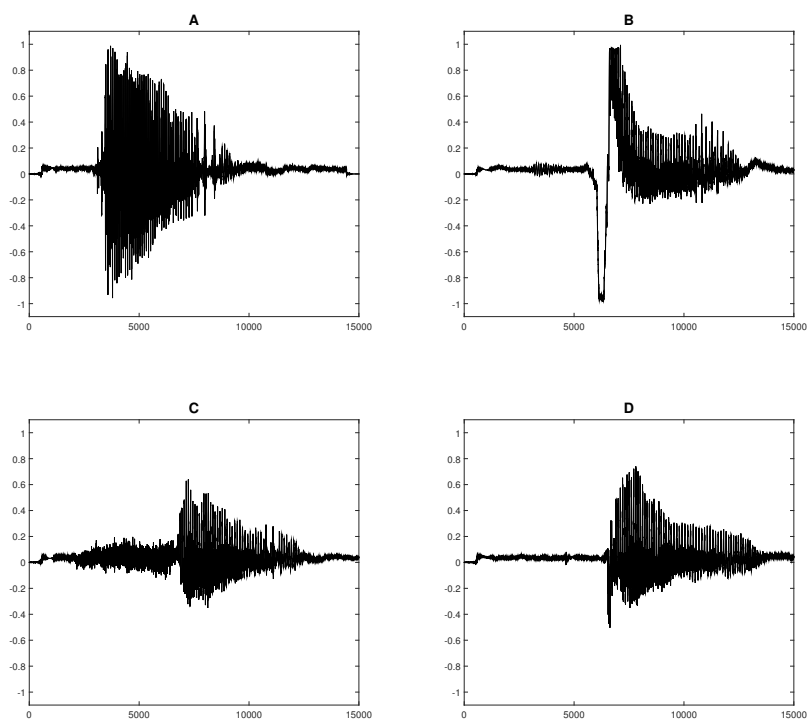


Figure A.1: Swedish alphabet in graphs: A-D

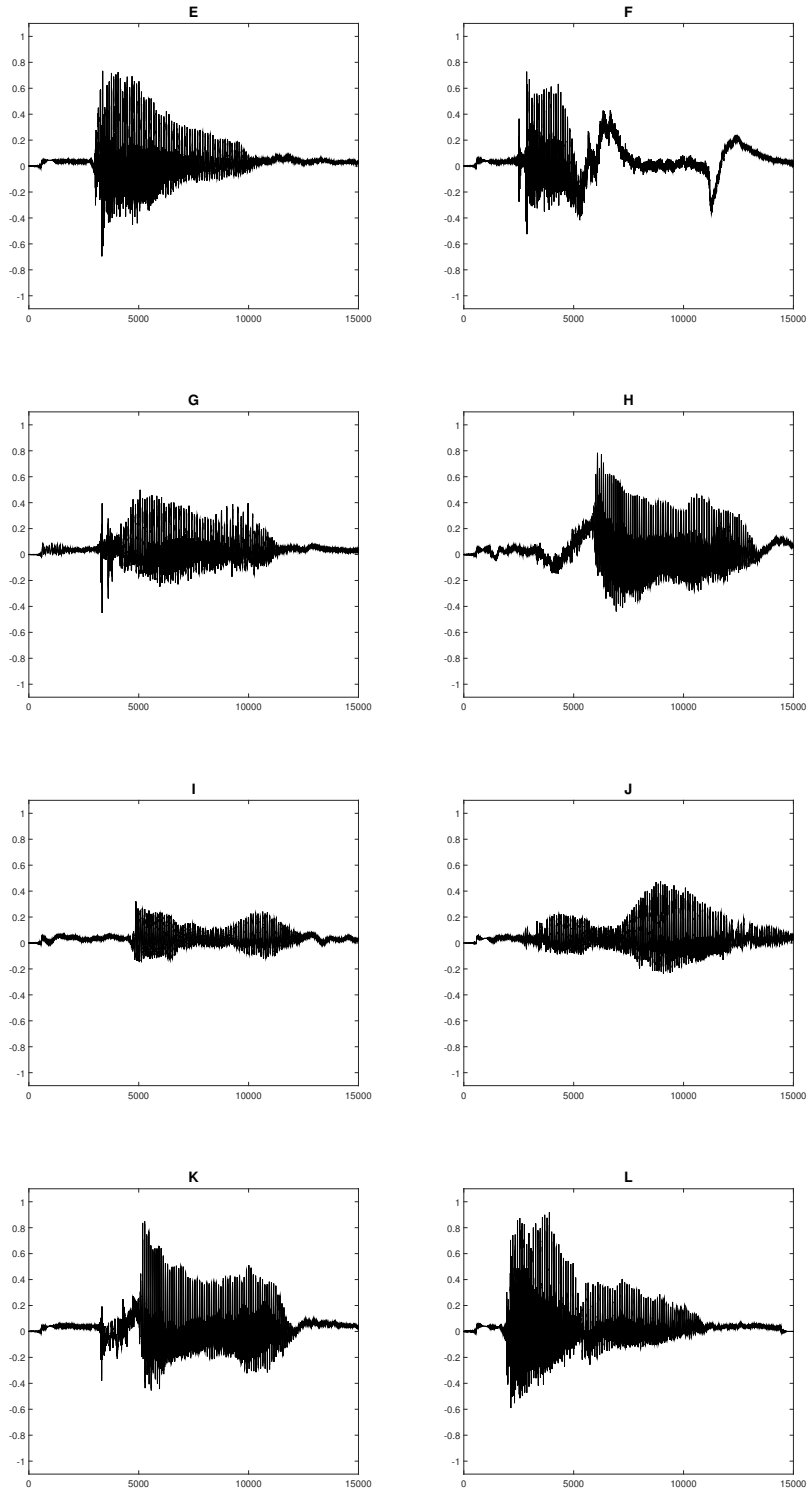


Figure A.2: Swedish alphabet in graphs: E-L

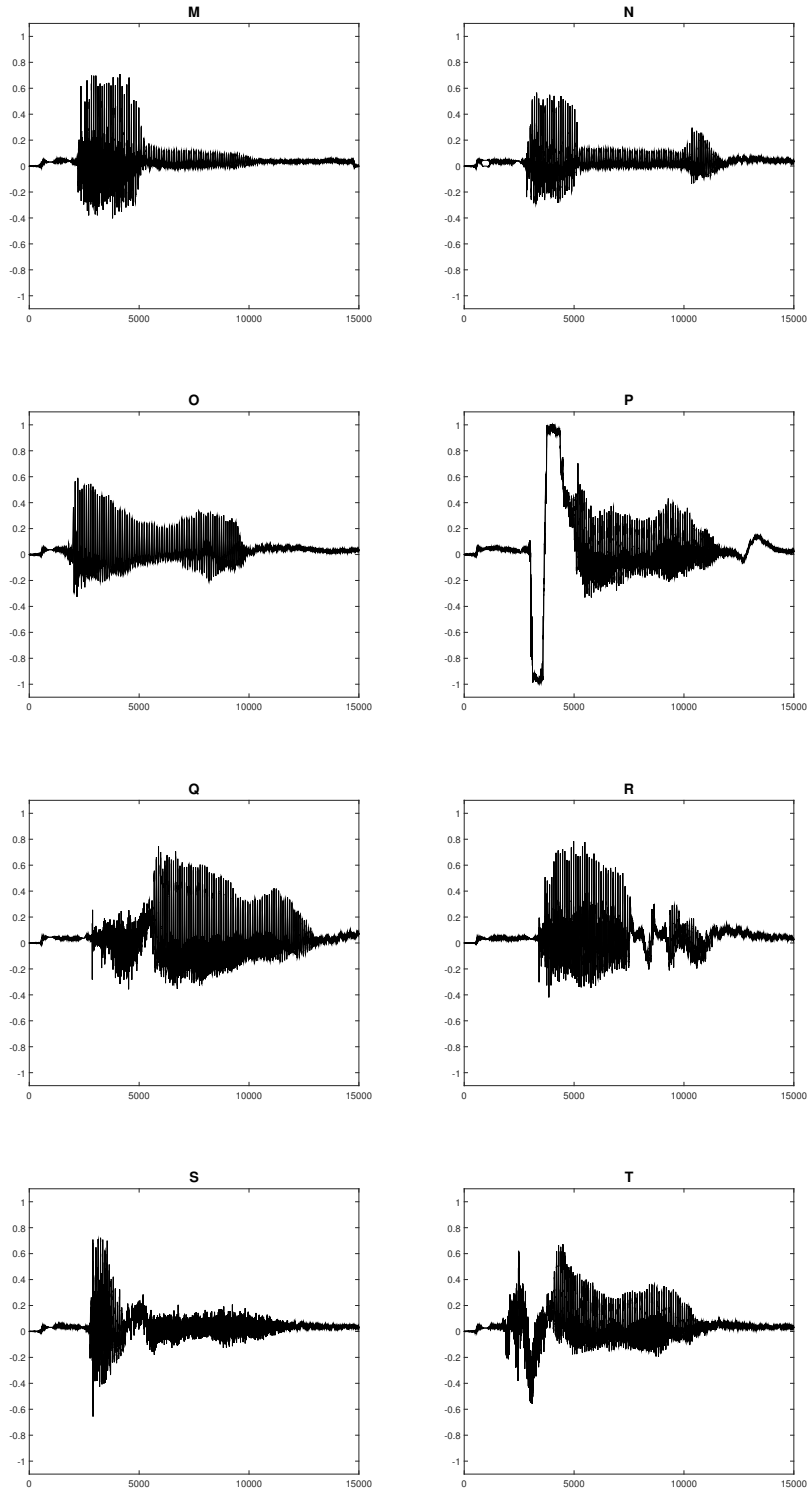
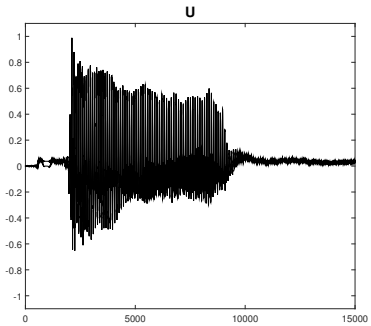
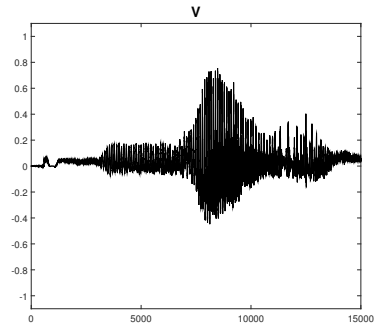
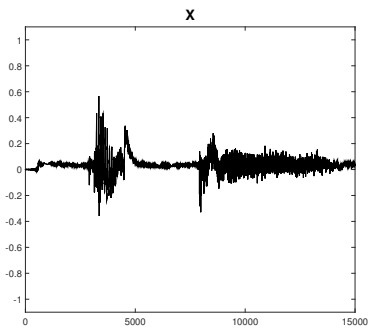
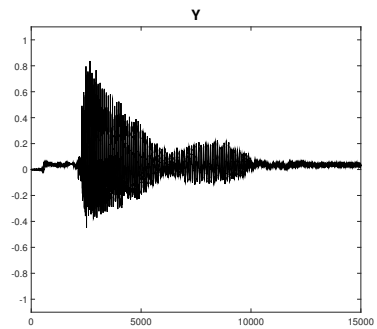


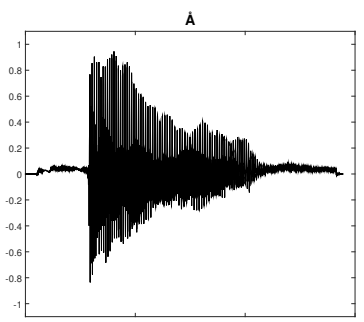
Figure A.3: Swedish alphabet in graphs: M-T

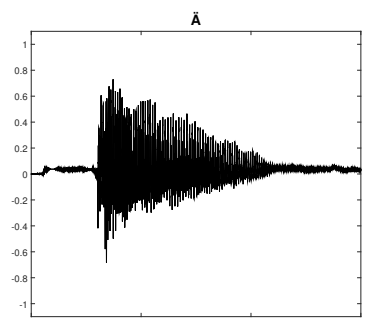












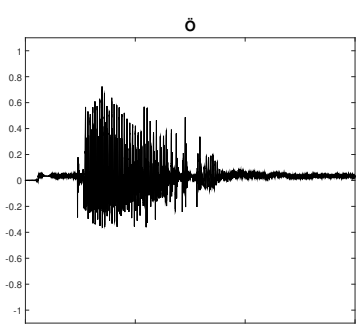


Figure A.4: Swedish alphabet in graphs: U-Ö

Appendix B

Wiener-Hopf Equations

The Wiener-Hopf equations are a set of equations which solve for the Wiener solution, which is the linear optimal estimate of a filter. The Wiener solution is an estimate of a filter which minimize the mean square error, thus the output of the estimated filter is equal to the desired signal [14, p. 108-122]. In this appendix the Wiener-Hopf equations will be given on matrix form.

The mean-square error of the estimation $e(n)$ is minimized by using the cost function $J(n)$

$$J = E[e(n)e^*(n)] = E[\|e^2(n)\|], \tag{B.1}$$

which when rewritten on matrix form becomes

$$\begin{aligned} J(w) &= E[[d(n) - w^H(n)u(n)][d(n) - w^H(n)u(n)]^*] \\ &= \sigma d^2 - w^H p - p^H w + w^H R w, \end{aligned} \tag{B.2}$$

where $w(n)$ is the filter, $u(n)$ is the input signal to the filter, $d(n)$ is the desired signal, R is the correlation matrix of the input and the desired signal, p is the cross correlation between the input and the desired signal and σ_d^2 is the standard deviation of the desired signal

$$\begin{aligned} w &= [w_0 \quad w_1 \quad \dots \quad w_{M-1}]^T \\ u(n) &= [u(n) \quad u(n-1) \quad \dots \quad u(n-M+1)]^T \\ R &= \begin{bmatrix} r(0) & r(1) & \dots & r(M-1) \\ r^*(1) & r(0) & \dots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \dots & r(0) \end{bmatrix} \\ p &= E[u(n)d^*(n)] = [p(0) \quad p(-1) \quad \dots \quad p(-(M-1))]^T \\ \sigma_d^2 &= E[d(n)d^*(n)] \end{aligned} \quad . \tag{B.3}$$

$J(w)$ is minimized by differentiation, that is, calculating the gradient of the cost function $J(w)$

$$\nabla J(w) = -2p + 2Rw. \tag{B.4}$$

When setting equation B.4 to zero one states the Wiener-Hopf solution

$$-2p + 2Rw_0 = 0 \quad \rightarrow \quad Rw_0 = p. \tag{B.5}$$

This equation assumes that the error surface is convex, that is, that there exists a minimum value of the estimation error $e(n)$.

And if solving for w_0 one calculates the Wiener solution

$$w_0 = R^{-1}p. \tag{B.6}$$

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2015-475

http://www.eit.lth.se