TAMPERE UNIVERSITY OF TECHNOLOGY

AKI TINAKARI

PHYSICAL SIZE OF MICROPHONE ARRAYS IN AD-HOC BEAMFORMING

Master's thesis

Examiners:
D.Sc. (Tech.) Pasi Pertilä
D.Sc. (Tech.) Adriana Vasilache
Examiners and topic approved by the
7/2014 Faculty Council of Computing
and Electrical Engineering on 13.8.2014

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Signaalinkäsittelyn ja tietoliikennetekniikan koulutusohjelma
AKI TINAKARI: Ad-hoc-keilanmuodostuksessa käytettävän mikrofonijärjestelmän fyysinen koko
Diplomityö, 47 sivua
Marraskuu 2017
Pääaine: Signaalinkäsittely
Tarkastajat: Pasi Pertilä, Adriana Vasilache
Avainsanat: Ad-hoc-järjestelmät, akustisen signaalin tunnistus, audio järjestelmät, keilan ohjaus, keilanmuodostus, mikrofonijärjestelmät, puheen parannus, sokea keilanmuodstus, TDOA-menetelmät, tilaäänisignaalinkäsittely, TOA-saapumisajan estimointi.

Usean mikrofonin järjestelmiä voidaan käyttää tapahtumien, esimerkiksi kokouksessa pidettävän puheen nauhoittamiseen. Itse hyötysignaalin, puheen, lisäksi nauhoituksiin tulee aina myös häiriötä useista lähteistä, kuten esimerkiksi huoneessa olevista muista äänilähteistä, huoneen itsensä aiheuttamista heijastuksista ja kaiunnasta, sekä nauhoitusvälineistöstä aiheutuvsta kohinasta. Nämä nauhoitukset voidaan kuitenkin yhdistää älykkäästi keilanmuodostusalgoritmeilla siten, että kaikista mikrofoneista nauhoitettu kohdesignaali on kohdistettu ja yhdistetty tuottaen yksittäisiä nauhoituksia paremman yhdistelmäsignaalin. Saadussa yhdistelmässä haluttu puhe on selvemmin esillä kuin yhdessäkään erillisessä nauhoituksessa.

Tässä työssä tutkitaan keilanmuodostukseen liittyviä tekniikoita, ja miten ne toimivat puhesignaaleilla kokoushuoneen kaltaisissa tilanteissa. Lisäksi selvitetään myös miten järjestelmä toimii ad-hoc -periaatteella, eli kun mikrofonien tarkkaa sijaintia ei tunneta, ja erityisesti miten mikrofonimuodostelman fyysinen koko vaikuttaa keilanmuodostuksen toimintaan. Tämä tutkimus keskittyy kokoushuoneen kaltaisiin tilanteisiin, ja käyttää sekä simulaatioita että oikeita nauhoituksia näyttämään, että mikrofonimuodostelman halkaisijalle on löydettävissä optimaalinen koko tutkituissa tilanteissa.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Signal Processing and Communications Engineering
AKI TINAKARI: Physical size of microphone arrays in ad-hoc beamforming
Master of Science Thesis, 47 pages
November 2017
Major: Signal processing
Examiner: Pasi Pertilä, Adriana Vasilache
Keywords: Acoustic Signal Detection, Ad-Hoc Arrays, Array Signal Processing, Audio Systems, Beam Steering, Blind Beamforming, Microphone Arrays, Spatial Signal Processing, Speech Enhancement, Time Difference Of Arrival, Time-Of-Arrival Estimation.

Microphone arrays can be used to record an event, for example speech during a meeting in meeting room setting. Besides the wanted signal, the speech, these audio recordings also have noise in them that is caused by various sources, including unwanted sound sources in the room, reflections and reverberations from the room itself, and noise caused by the equipment used to measure the speech. The audio recordings from the microphones can be intelligently combined with beamforming so that the information from the target event is lined up, and added up so that the result signal is an enhanced combination of all the measured recordings. This results to significantly more clear recording than any single recording of the speech.

In this thesis some of these beamforming methods are studied, as well as how they can be used with speech signals in a meeting room scenario, how the system is able to work ad-hoc when the location of the devices is omitted, and also how the physical dimensions of the array affect the beamformer output. The research focuses on a meeting room scenario, and utilizes both simulations and real recordings to show that there is an optimal size to the microphone array diameter in the studied scenario.

PREFACE

This master's thesis was done at the Tampere University of Technology in the Department of Signal Processing, and was carried out as a part of Multiple Device Audio Capture project in cooperation with Nokia Oyj. I want to thank my supervisors Dr. Pasi Pertilä, who instructed and guided me in the research whenever I needed any advice, and Dr. Adriana Vasilache for her support from Nokia. I also thank my colleagues in the Audio Research Group, especially Mikko Parviainen, Joonas Nikunen and Katariina Mahkonen for their support and help in general. Special thanks go of course to my fiancée Matilda, who supported me when I had lost all the motivation, and also to my friends and family for their support and understanding.

CONTENTS

LIST OF ACRONYMS AND SYMBOLS

COLA	Constant-Overlap-Add
DFT	Discrete Fourier Transform
DOA	Direction of Arrival
DTX	Discontinuous Transmission
ERB	Equivalent Rectangular Bandwidth
FFT	Fast Fourier Transform
GSC	Generalized Sidelobe Canceller
MVDR	Minimum Variance Distortionless Response
PESQ	Perceptual Evaluation of Speech Quality
RIR	Room Impulse Response
SAR	Sources-to-Artefacts Ratio
SDB	Sum-and-Delay Beamformer
SDR	Signal-to-Distortion Ratio
SNR	Signal-to-Noise Ratio
SSARA	Segmental Source-to-Artifact Ratio by Arithmetic mean
SSNRA	Segmental Signal-to-Noise Ratio by Arithmetic mean
SSNR	Segmental Signal-to-Noise Ratio
STFT	Short-Time Fourier Transform
TDOA	Time Difference Of Arrival
TIMIT	Corpus created by Texas Instruments and the Massachusetts Institute of Technology
TOA	Time of Arrival
VAD	Voice Activity Detection
WAV	Waveform audio file format

1. INTRODUCTION

Smartphones, tablets and other microphone equipped mobile devices have been very popular for years, and the quality of their microphones has increased significantly as well as their processing power. Yet the commercial solutions for conference calls and meeting room recording systems are often dedicated rooms equipped with a beamforming microphone array on the table or ceiling, or systems with individual crane microphones. Even though these systems might provide adequate sound quality, they have some specific drawbacks. The tabletop devices are usually quite far away from some of the speakers, which reduces their sound quality. The crane microphone systems on the other hand restrict all the speakers to their seats, effectively denying for example the possibility of keeping presentations out of their seat. Most of all, these room wide systems are more expensive and harder to renew than for example smartphones. In this thesis the viability of using an ad-hoc array of mobile devices is considered in meeting room scenarios by utilizing beamforming.

In the situations where there are multiple microphones in a single device, like in a conference call phone, often a direction based beamforming is utilized to maximize the quality of the recording. This means that the device essentially tracks the direction of the dominant sound source, and delays the audio signals received by the microphones accordingly. Adding these delayed audio signals together results in a mono signal where the audio content from the tracked position is enhanced. To delay the microphone signals so that the sound coming from a certain direction is amplified, the information about the microphone positions can be utilized. Naturally the positions are not known in ad-hoc systems, like for instance an array of smartphones.

Earlier the problem of unknown locations has been solved by first determining the position of these ad-hoc devices [20], and utilizing this information, traditional direction based beamforming can be deployed. There are multiple problems with this approach, as the positions of the devices can suddenly change, and the positions should be thus constantly and accurately tracked [19]. Recent research shows however that beamforming can be done blindly, without explicit information about the position of the devices, by tracking the relative Time Of Arrival (TOA) values [21].

This thesis contributes to the ongoing research by studying the difference that the varying microphone array size causes. In general the viability of blind steering and its issues are also studied. The thesis is structured as follows. Basics of beamforming is introduced in the beginning of the theory section in chapter 2, and the parts related to this work are studied in detail in sections 2.1 - 2.6. A suitable metric for the evaluation of the output is considered in section 2.7, which concludes the theory section. After the theory, a MATLAB implementation of ad-hoc beamforming system is presented in chapter 3, and detailed in 3.1. In sections 3.2 and 3.3 this implementation is then used to process the data for simulated and real recordings. Finally, the results are presented in chapter 4, and discussed in chapter 5. Conclusions and future directions are considered in the last part, chapter 6.

2. THEORETICAL BACKGROUND

To understand the different mechanics of an ad-hoc beamforming system, some basics about beamforming is presented here. Beamforming systems are comprised of multiple procedures, and so a block diagram of a typical system setup is provided in figure 2.1. From this overview the order of the procedures, and their relations can be observed. The scenarios considered in this thesis are based on a typical meeting room setting, where the participants sitting around a meeting room table, with their smartphones on the table in front of them. All of these smartphones are recording, and their audio signals are streamed asynchronously, although within a reasonable time, to a host device where the signal processing takes place. The reasonable time restriction is discussed in section 2.4.

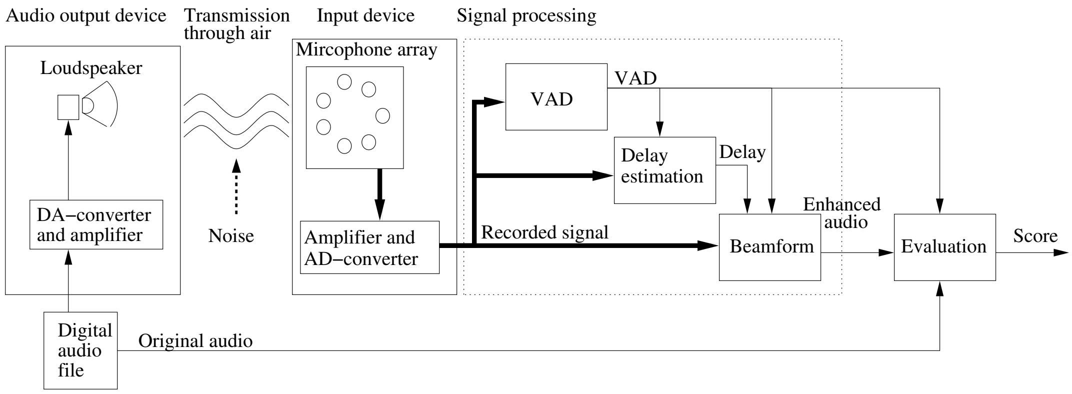


Figure 2.1 *A block diagram of the ad-hoc beamforming system and test setup. Bold paths represent multi-channel signals.*

The block diagram of the system studied here is presented in figure 2.1. The diagram shows that the original audio is first played through a loudspeaker to a microphone array for processing, and later also used in the evaluation of the beamformer output. Before the audio is captured by the microphone array, it is distorted by the room environment, and often also corrupted by other unwanted noise. The signal processing system implemented here consists of Voice activity detection, VAD, time-delay estimation block, and finally the beamformer block which combines the input signals.

As is normal for real-time audio processing, and especially speech processing, the temporal properties of the signal are modified by processing the signal in small time windows, denoted frames. The framewise processing in frequency domain is achieved by Short Time Fourier Transform, STFT, the specifics of which are explained in section 2.1.

The signal model used in this thesis can be formulated as a function of time t

$$x_i(t) = A_i(t) \cdot s(t + \tau_i(t)) + n_i(t), \qquad i = 1..N, \tag{2.1}$$

where signals $x_i(t)$ captured at microphone channel i consist of the clean source signal $s(t)$, so that they have been attenuated with $A_i(t)$, delayed with $\tau_i(t)$, and have additive noise $n_i(t)$. Note that in this work, the reflections of the original signal which are often corrupted versions of the original are here part of the noise. The goal in beamforming is enhancement of signal, which is done by estimating at least the delay values $\tau_i(t)$, although taking amplitude differences $A_i(t)$ into account can further refine the output quality. Additionally, the general goal is to suppress the noise $n_i(t)$.

Beamforming is based on signal averaging in the sense that it combines multiple signals from the same event to achieve a better quality signal. To mathematically formulate signal averaging [29], it is assumed that the signal and the noise are statistically uncorrelated, noise is random with a zero mean, and that both the signal energy

$$S = \frac{1}{N} \sum_{t=1}^{N} x^2(t), \tag{2.2}$$

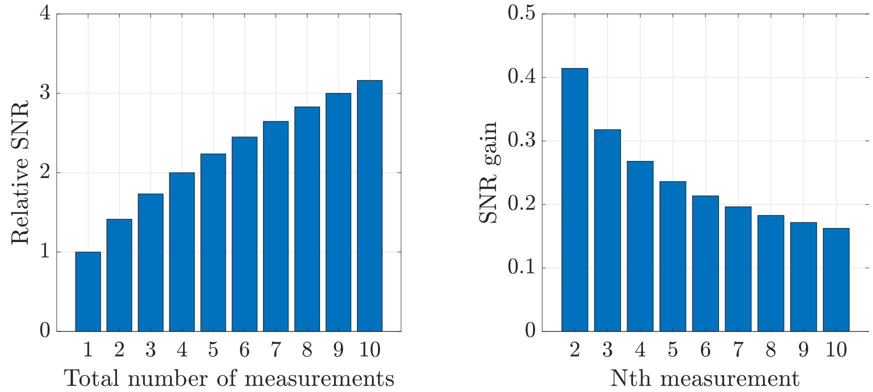
and the standard deviation of the zero mean noise in a single measurement

$$\sigma = \sqrt{\frac{1}{N} \sum_{t=1}^{N} n^2(t)}, \tag{2.3}$$

are the same in all the signals. As the standard deviation of the noise is additive, the resulting noise over k replicate measurements is $\sqrt{k\sigma^2}$. The resulting signal-to-noise ratio (SNR) is then given, as shown in [29], by

$$\mathrm{SNR} = \frac{kS}{\sqrt{k\sigma^2}} = \sqrt{k}\frac{S}{\sigma}. \tag{2.4}$$

It can be deduced that the signal averaging means, that when there are multiple replicate measurements of an event with uncorrelated additive noise, the SNR will increase in an ideal case in proportion to the square root of the number of measurements. Conversely, it can be noted that the gain from extra measurements diminishes exponentially as can be seen from figure 2.2.



(a) SNR in relation to the number of measurements.

(b) SNR gain for Nth measurement.

Figure 2.2 *Behaviour of SNR when averaging over N microphone measurements of the same event.*

The values in the figure 2.2 (a) have been calculated with equation 2.4, by setting $S/\sigma = 1$, and $k = 1..10$. Values in figure (b) are calculated by subtracting adjacent columns in (a). SNR in itself and as a metric will be discussed more in-depth in section 2.7.1.

In real world cases the uncorrelated random noise corresponds only to a certain portion of the total noise in the measurements, and so this can be seen as the upper limit for the SNR gain in this simplified case. In real world, uncorrelated noise can originate from electromagnetic interference within the used electronic recording equipment, or manifest as diffuse noise in the room environment. Partly even the transmission channel can be responsible for this kind of noise, as sound travels through air, which is not a perfect conductor because of local deviations in humidity and temperature.

In addition to dampening diffuse noise, also spatial noise that does not originate from the desired sound source is also reduced. The amount of reduction in this case is at maximum the same as with the correlating noise, but it can be also reduced further. These spatial noise sources can be much more efficiently reduced with more sophisticated algorithms, like Generalized Sidelobe Canceller (GSC), which affect the directivity pattern of the beamformer [10, 12]. Beamforming is presented more in detail in section 2.6. To know when to estimate noise correlations, some of the more complicated algorithms need Voice Activity Detection (VAD), which is a large research field in its own. The VAD used in this work is presented in section 2.2.

The problem in beamforming is to have measurements of the same event time-aligned, so that the exact same measurements are averaged over the microphones of the array. The time-alignment may change over time, and that is one of the reasons the signals need to be processed in segments, as will be presented in section 2.1. Even though digital audio is processed in discrete samples, the signals can still be aligned with sub-sample accuracy by using linear phase shift in the frequency domain. Time-alignment of the signals, also known as steering the beamformer, is presented more in-depth in the section 2.5. When discussing about beamforming, spatial aliasing is often mentioned as a significant factor. How this affects the steering of the beamformer in the context of speech signals is discussed in section 2.3.

In a traditional Direction Of Arrival (DOA) based beamformer the microphone locations are known, and using this information the delays $\tau_{i,d}$ for every direction d are easily calculated. The direction for the steering can be obtained with Acoustic Source Localization (ASL) techniques, for example by examining the energy ratios between the microphones. In general with fixed arrays the beamforming can be quite straight-forward. But when the devices differ from each other and their location is not known, as might well be in an array which is composed of pseudorandomly located smartphones of varying kind, additional things need to be taken into account. These issues with ad-hoc arrays are discussed in the section 2.4. Even though beamforming is by theory defined to increase SNR, SNR might not be the best metric to evaluate enhancements in audio, or especially speech quality. Metrics used in this work are presented in section 2.7.

2.1 Framewise processing

For one stationary speaker, and short recordings it could be viable to just use one constant steering vector to steer the beamformer. In real world however, the microphones or the sound source might move, or the recording devices could also have drifting clocks that distort the time alignment. Additionally, instead of processing everything in one batch, there are also considerations for online, real-time or live processing, which require the system to produce output with minimum delay. All of these issues can be addressed by processing the input in small segments, called frames or windows.

The window length l needs some consideration, as it defines the processing delay of the system. Qualitywise it is a choice of having more data for delay estimation per frame, but on the other hand this also corresponds to a longer time, within which the estimate might actually vary. In other words the temporal resolution is also lower.

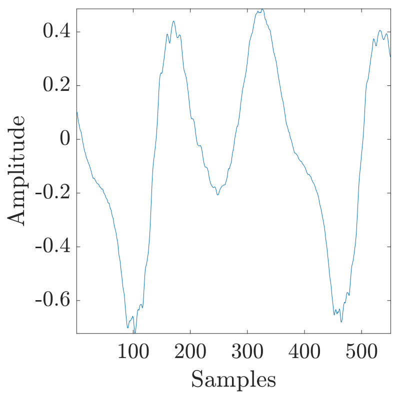
The window length of l_w = 20-100 ms is a usual choice for speech audio signal processing. The length in samples, m_w, can be calculated with

$$m_w = l_w * F_s, \tag{2.5}$$

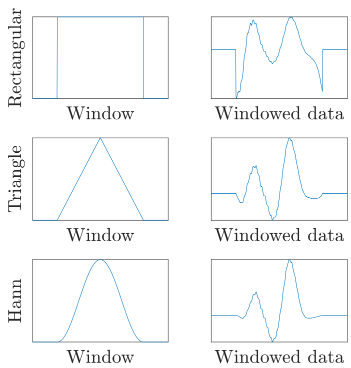
where F_s is the sampling frequency of the signal.

Using a non-weighted frame of the signal might be enough for some operations, like for example calculating cross-correlation in time domain to find out a delay of the signal. Cross-correlation will be presented in detail in section 2.5. Often however it is efficient to do the processing in the frequency domain, or also reconstruct the signal with the framewise processing. These operations have problems if nothing is done to the non-weighted frames.

To transform the signal to frequency domain, Fourier transform [7] is used. Fast Fourier transform (FFT) [4] is a fast way to compute discrete Fourier transform (DFT). DFT assumes both the time domain and frequency domain signals to be periodic, which causes problematic aliasing if a frame of signal is not periodic [27, p. 194]. An example of a non-periodic frame, where the frame does not converge to zero in the beginning and the end of the frame, is pictured in figure 2.3.



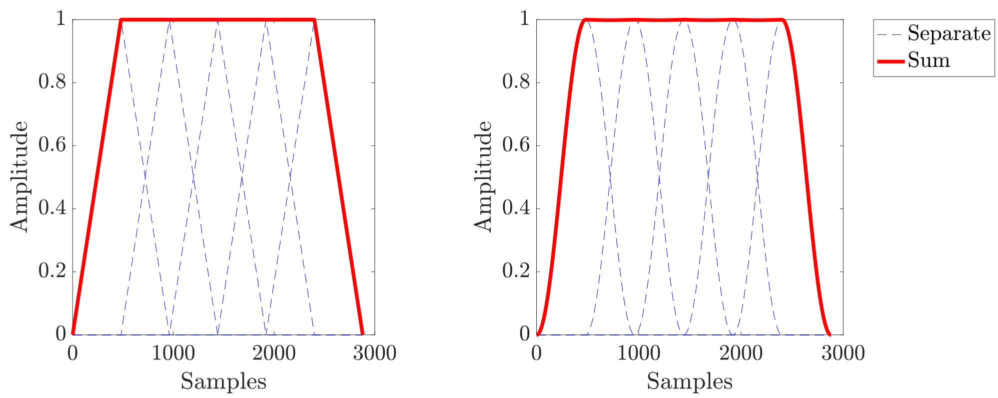
(a) Original signal.



(b) Different windows applied to the signal.

Figure 2.3 *Effect of different windows on a signal.*

To converge the frames to zero, they can be multiplied with a windowing function, for example a triangle or a Hann window, as in figure 2.3. Now however the frames do not capture the whole signal equally, so a constant-overlap-add (COLA) constraint is used to formulate overlapping windows that sum to 1 for each sample. A triangular and a Hann window with 50 % overlap are pictured in figure 2.4. An additional benefit of using overlapping windows is that longer windows can be used for the estimation, which benefits the estimation.



(a) Triangle windows. (b) Hann windows.

Figure 2.4 *Example of overlap-add windows with 50% overlap summing up to one.*

Calculating FFT of a windowed signal like this can be seen as Short-Time Fourier Transform (STFT) [26], which is a popular frequency analysis tool used to calculate for example spectrograms.

After computing the STFT for a frame, the frame is divided in the frequency domain to frequency bands, so that different frequencies can be processed separately. This would enable for example calculation of the beamformer delays separately for each frequency band. Note here that the FFT of a signal contains also the negative frequencies, in frequency bins $\omega = N_{FFT}/2..N$ (for even N_{FFT}) which can be ignored by assuming that the signal is real valued in the inverse FFT. The band division can be made similarly with overlapping windows.

To divide the STFT to frequency bands, it is useful to have different length bands for lower and higher frequencies, as human hearing is generally more accurate in the lower frequencies than in the higher frequencies. Some of the differing scales that can be used to transform the frequencies into critical bands are Equivalent Rectangular Bandwidth (ERB) [18], Bark [34], Mel [28], and octave scales. However, in this work the STFT is not divided into frequency bands, because that would introduce problems not in the scope of this thesis. These include not having enough energy in some bands for reasonable estimates, making sure the bands are steered roughly to same delays, and in general having more degrees of freedom and possibilities for noise to corrupt the estimations.

2.2 Voice Activity Detection

Voice activity detection (VAD) is used to divide a signal temporally into different sections based on its contents, in order to do different kind of processing to each section. Usually at least active speech segments and non-speech segments are distinguished, but sometimes voiced speech and non-voiced speech are separated, as well as background noise. Besides a binary decision between active speech and non-speech, some VAD implementations output likelihood or probability of the activity. A VAD is needed in many signal processing methods, especially if they utilize adaptive filtering which needs noise estimation. In the system presented earlier in the block diagram, figure 2.1, adaptive filters can be used in the beamformer and in the time aligning, and therefore obtaining VAD in this context is essential. VAD data is also useful in the evaluation of the beamformed signal. In this section the needed properties of a suitable VAD are discussed.

Voice activity detection is not only about detecting whether there is any content in the channel, it is about detecting whether there is the kind of activity the system needs to respond to. For example the sound of a projector is not usually such a thing, but if there is speech on the top of the projector noise the system should be able to detect that. So what is actually needed from VAD in this context is speech detection. It is also easy to see VAD as a classification problem: The state is often determined by certain features extracted from the signal.

Although VAD can be calculated for each sample, it is more computationally efficient to only calculate it once for each frame. Additionally, if especially speech detection is of concern, it is useful to have knowledge of previous VAD states before making a classification decision for the current frame. This is partly because speech is concentrated on certain sections and has pauses between sentences, and partly because there are non-voiced sections in speech that are hard to distinguish from background noise without the knowledge of previous states. A high level function for VAD could be defined as

$$[\mathbf{CR}_f, \mathbf{STATE}_f] = \mathrm{VAD}(\mathbf{x}_f, \mathbf{STATE}_{f-1}), \tag{2.6}$$

where for each frame f, the VAD function calculates the classification result $\mathbf{CR}_f$ and internal function state $\mathbf{STATE}_f$, by using the last state, and the current signal frame $\mathbf{x}_f$. For example a simple VAD can use the frame energy

$$\mathbf{E} = \sum_{t \in f} x^2(t) \tag{2.7}$$

as a feature for classification. Here $x(t)$ is a sample in frame f. This kind of feature would either need to use pre-defined thresholds to make classification decision, or have adaptive state parameters that are updated on each round. These parameters could be for example noise and speech power estimates.

There is a small difference in what is wanted from a VAD in different applications. For time aligning microphone signals, the speaker needs to be detected, and classifying any frames that contain such spatial information as speech is essential. However, for beamforming algorithms such as the GSC, a good noise estimate is needed for noise suppression, and none of the noise frames should be classified as speech. For a binary VAD, that classifies each frame as either true, for speech, or false, for non-speech or noise, there is a problem: which of these definitions should be the primary goal? Many of the VADs prioritize the classification of most speech segments right with the cost of misclassifying some noise frames as speech. Even though the goals are quite similar, it should be noted that there is a difference. Noise estimate can also be calculated separately like for example in [11].

A handmade VAD could be used to optimize the desired preference for different types of classification errors. However, the amount of material used in this thesis is so large that the process had to be automated. In this thesis, a standard VAD defined in ITU-T G.729 Annex B [13] is used, with the extension to work with wideband signals. ITU-T G.729 Annex B is a reference class VAD implementation for discontinuous transmissions (DTX). Even though it was originally developed for 8 kHz and relatively low quality (8 kbps) speech audio, there is an extension for wideband operation. The G.729 standard VAD uses several features for making the decision. It calculates the zero-crossing rate, estimates full-band energy, low-band energy, and the line spectral frequencies. Within this feature space, it finds a binary decision boundary, and for new frames applies adaptive correction and finally smoothes the decision estimate [1].

2.3 Effect of the spatial aliasing

Before defining spatial aliasing it is natural to begin with sampling theory and temporal aliasing. The famous Nyquist-Shannon sampling theory [24] states that when a signal is sampled at a certain rate F_s, all frequency components lower than half of the sampling rate $f < F_s/2$ can be perfectly represented with a discrete-time signal. Any components that are of a higher frequency than that, $f \geq F_s/2$, become temporally aliased and are interpreted incorrectly as a certain lower frequency.

However, with spatial aliasing, one should consider a stationary narrowband signal. Spatial aliasing starts to develop, when half of the wavelength of the signal is smaller than the distance between two microphones. For the higher frequencies which have these shorter wavelengths, the phase becomes ambiguous and can not be correctly interpreted. The direction is aliased similarly to temporal aliasing in Nyquist-Shannon sampling theory. Spatial aliasing distance d, defined as

$$f_{max} < \frac{F_s}{2} \Leftrightarrow \frac{c}{\lambda_{min}} < \frac{c}{2d} \Leftrightarrow d < \frac{\lambda_{min}}{2}, \Leftrightarrow d < \frac{c}{2f_{max}}, \tag{2.8}$$

where d is the maximum radius of how far away from each other two microphones can be, so that there is no aliasing until a certain frequency limit f_{max}. In the equation speed of sound is denoted by c, sampling frequency by F_s, and wavelength by λ. For those frequencies that are higher than f_{max} there is ambiguity between the directions.

Table 2.1 *Maximum inter-element spacing with the corresponding aliasing frequency, when $c = 343$ m/s.*

d	1.5 cm	8.5 cm	17 cm	40 cm	80 cm	140 cm	200 cm	250 cm	300 cm
f_{max}	11433 Hz	2018 Hz	1009 Hz	429 Hz	214 Hz	123 Hz	86 Hz	69 Hz	57 Hz

Table 2.1 lists a few frequencies and their corresponding aliasing distances d. As it can be seen also from the equation (2.8) the relation is inversely proportioned, and to have no aliasing for usual speech frequencies the microphones need to be very close together. However, it should be noted that most of the speech content is mainly below 4000 Hz [5, p. 149].

Spatial aliasing is a significant factor when working with narrowband signals, which is visible in radio frequency applications, where beamforming is also used. However, spatial aliasing is usually not that significant in audio beamforming. In [6] it is shown that the effect of spatial aliasing in audio signal processing diminishes because audio signals, especially speech, are generally non-stationary and wideband signals.

However spatial aliasing is relevant in interpreting the beamformer output of fixed arrays, where the beamforming is restricted to physical direction or point in space. In that scenario, when the array is steered based on spatial location or direction, the sound also gets amplified from the spatially aliased locations according to the frequency as explained, if there would be such sound sources in the aliased location. Additionally, it can not be distinguished whether the sound originates from the real location, or the aliased location.

When the beamforming is not restricted to directions, be it in two or three dimensions, and instead the beamformer is steered according to independent optimal delays rather than optimal directions, it changes the idea of ambiguous location. All the locations can be defined as a set of corresponding delays to the microphones, but this is only a small subset of all possible delays, and inversely these other delays do not map to a physical location.

In an optimum delay case, the delays would represent the sound source location, and the same aliasing interpretation would hold. But when you allow the noise to the direction or location based steering, the amplification of aliased locations would remain the same, even though their location varies with the noisy estimate. However, if there is noise directly in the delays, the spatially aliased location for the noisy delays might not exist, because it might not be in the subset of delays that represent a location. As this happens over multiple microphone pairs, it can be thought that spatial aliasing, at least in traditional sense, diminishes when steering with delays.

Spatial aliasing is basically impossible to interpret as locations with a microphone array, where the locations of the microphones are not known. In the context of this kind of arrays, the spatial aliasing can be seen as wrapping of the time delays in the sense that a delay is ambiguous to a different delay, instead of locations. It should be noted that in this sense, it can still affect the beamforming. In the next section it is defined what are the implications of the unknown locations, as well as what ad-hoc arrays mean in the context of array signal processing.

Many commercial microphone arrays utilizing beamforming are compact arrays, and as such they should not suffer much from spatial aliasing even if they used direction based beamforming. However, the interesting question here is how much the size of the array actually affects the speech quality with an ad-hoc array, and consequently, what is the optimal size for the array.

2.4 Definition of Ad-hoc

In this thesis, ad-hoc microphone array is defined as a set of $N \geq 2$ microphone-equipped devices in unknown locations within the same acoustic space that are able to transmit their microphone audio data continuously to a single device where the signals are processed. These microphone-equipped devices may also be differing in microphone quality, AD-conversion quality, and sampling rates, although not all of these aspects are in the scope of this study.

In essence, ad-hoc arrays bring three distinct sources of uncertainty to the system. The first source is the unknown location of the devices, the second is the different quality of the devices, the device performance. The third source is the lack of synchronization between the devices: The data from the devices is gathered asynchronously and their clocks might drift during the measurement. Differences in device performance are the result of differences in in-device processing, AD-converters and microphone quality. As many of these details are usually trade secrets of the manufacturers, in this work the device performance is not considered, as all the microphones are the same kind in all the recordings.

Beamforming with microphone signals that have different kinds of phase shifts and distorted frequencies is problematic. One way to solve this problem would be by calibrating each microphone, but that is very much out of the scope of this thesis.

The array geometry and sound source locations can be categorized as fixed or dynamic. This creates four distinct scenarios. In the most fixed case nothing moves, then two cases where either microphones or sound sources move, and finally where both the array geometry and the sound source locations are dynamic. The difference between the cases where either a microphone or a sound sources move can be deducted from Time Difference Of Arrival (TDOA) values, as when a microphone moves only the TDOA values related to that one microphone change, and when only the speaker moves at maximum only one TDOA can remain the same. Of course this is harder to distinguish if there is multiple moving microphones. The system described in this thesis, and its implementation are able to deal with all the scenarios, within the limits of processing frame length, although in the simulations and recordings in this thesis the locations are fixed.

Synchronization of the devices is also out of the scope of this thesis, because it is more of a computer networking problem than a signal processing problem. It is only noted that suitable accuracy can be achieved, and *sufficient synchronization* among ad-hoc devices is assumed in the rest of this thesis. For the synchronization to be sufficient in this context it is assumed that the actual offset between the signals needs to be smaller than the difference between frame length time and inter-device delay.

The effects of the unknown location of the devices however is an interesting aspect, and it is the very reason for this thesis. Even in the very basics of beamforming the objective is often to figure out the TDOA values for the signals, which in itself has traditionally been considered too noisy to be used for steering the beamformer directly, and thus have been ignored as the direct source for steering information. Usually different kinds of transformations between spatial domain and TDOA have been used to make the dimension of the problem smaller. For example by calculating the evidence for each Direction Of Arrival (DOA) the noise in the steering vector becomes smaller much like with signal averaging: the problem moves from multiple measurements to a smaller number. However, when there is no information about the microphone locations, it becomes a necessity to use TDOA more directly to figure out the steering vector of the beamformer.

2.5 Time-aligning the signals

Beamforming is in essence time-alignment of the multi-channel signals. As mentioned earlier, in DOA based systems the direction could be estimated from the energy ratios of the input signals, and the signals are delayed with the knowledge of the microphone positions. To do this alignment without that knowledge, the time difference between the signals needs to be known, and this information can be obtained via cross-correlation between all the channel pairs. So for N microphones there are $\binom{N}{2}$ combinations of pairs. In the following section 2.5.1 the TDOA is formulated, from which the relative TOA is derived. One advanced method which filters some noise from the data is also introduced in section 2.5.3.

In methods that rely on transforming the problem to physical space either near-field or far-field sound propagation needs to be assumed. In near-field propagation the sound spreads spherically whereas in far-field propagation the sound waves move as a uniform wall. This kind of assumption is not needed when such a transformation is not done, and although this transformation helps usually in keeping the steering vector noise low, the steering is not optimal in the cases where the assumption is wrong.

As there is no direct way to interpret TDOA or TOA values as a location or direction in ad-hoc systems since microphone positions are unknown, the dimension of the problem is $\binom{N}{2}$ (for TDOA) or $N-1$ (for TOA). It is at least not trivial to deduct whether the sound source moves, or whether multiple devices move by using the microphone signals. As a side note accelerometers on the devices could be used if such information would be needed. As long as the devices do not move too far away from each other, limited by the used frame length, the system should be able to follow the most dominant sound source. Similarly to what was done with VAD, manual or offline methods can be used to obtain oracle TDOA signals for a reference.

2.5.1 Time Difference Of Arrival

Assuming that there are two microphones in known locations, and a sound source, it is possible to find a parabola, where the sound source can be located. This is pictured in figure 2.5. The time difference between the signals that are received by the microphones is the TDOA.

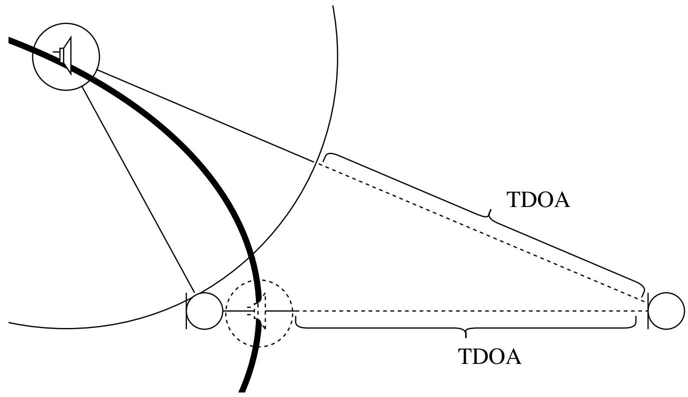


Figure 2.5 *The time difference between signals defines a parabola of possible sound source locations for a two-microphone-array.*

A rudimentary way to calculate the TDOA is to measure the power difference between the channels. This would utilize the fact that the signal power is inversely related to the squared distance from sound source. However, this is not very accurate, and does not really work for segments where the signal has not yet reached both microphones. In digital signal processing, TDOA estimation is very often based on cross-correlation that is calculated via convolution [15, p. 183]

$$(f \star g)(\tau) = \sum_{t=-\infty}^{\infty} f(t) \cdot g(\tau + t), \tag{2.9}$$

which is a measure of similarity between two signals $f(t)$ and $g(t)$ as a function of lags τ. The result is then a vector where the similarity of the two signals is evaluated when the other signal is shifted in time with the defined lag τ. As said earlier, these TDOA values can be calculated between all the $\binom{N}{2}$ unique combinations of the microphone signals.

TDOA values δ are then calculated from the lags obtained with

$$\delta = \underset{\tau}{\mathbf{argmax}}((f \star g)(\tau)), \tag{2.10}$$

where it can be seen at which delay lag the similarity is at its highest. Note that the information of how similar the signals are at that point is not taken into account here, only the delay with highest similarity is considered. As it is now known how many samples are between the microphones it would be trivial to convert these values to actual time in seconds, however it is more convenient to process these values as samples.

It is however more efficient to calculate the cross-correlation function in frequency domain

$$(f \star g)(\tau) = \mathrm{IFFT}(F(\omega) \cdot G^*(\omega)), \tag{2.11}$$

as convolution $\star$ translates to multiplication. ω in this context are the frequency bins. This leads mainly to faster calculation, because calculating convolution in time domain requires more operations than conversion to frequency domain and back with Fast Fourier Transformation (FFT).

The accuracy can be enhanced with interpolation, as it estimates the location of the peak in sub-sample level. In this thesis quadratic interpolation [26] of three adjacent samples is used, and its peak location x_{peak} is calculated from these three samples y_i by

$$x_{peak} = \frac{y_1 - y_{-1}}{2 \cdot (2 \cdot y_0 - y_1 - y_{-1})}, \tag{2.12}$$

where y_{-1} and y_1 are the values around the peak y_0 found with cross-correlation.

Noise that correlates within the input signals is very destructive to TDOA estimation, because it easily becomes dominating sound source, and directs the beamforming to the noise source. Therefore, it is critical in any more advanced methods to filter the incoming signals to attenuate the noise as much as possible. The sound quality in itself is not an issue here at this stage. What matters most is that the signal from which the TDOA is derived has the spatial information, which is the proper signal correlation related to the sound source. Before getting into noise reduction techniques however, it is useful to consider weighting of the cross-correlation function. In equation 2.13 the cross-correlation is associated with a weight W.

$$(f \star g)(\tau) = \mathrm{IFFT}(F(\omega) \cdot G^*(\omega) \cdot W(\omega)), \tag{2.13}$$

An example of a well performing weighting is the phase transform (PHAT) weighting. Even though the weighting was constructed heuristically, it was shown to be maximum likelihood (ML) optimal for arrays in reverberant low-noise environments [17, 33]. Defined in equation 2.14, PHAT weighting uses the phase information to compute the cross-correlation, and normalizes the amplitude of the spectral density between the channels.

$$W_{\mathbf{PHAT}}(\omega) = \frac{1}{|F(\omega) \cdot G^*(\omega)|}, \tag{2.14}$$

There exist a lot of different ways to weight the cross-correlation function as well, but they are out of the scope of this thesis. However, as already explained, noise suppression techniques provide also a layer of noise robustness to TDOA steering. Some of these more advanced methods are presented later in this thesis in section 2.5.3.

2.5.2 Time Of Arrival

Previously it was defined already that there are $\binom{N}{2}$ combinations of microphone pairs in an array, and subsequently in TDOA as well. Besides the obvious redundancy in having delays calculated between all the pairs, when all the N microphone signals need to be aligned for beamforming, only $N-1$ delays are needed. In this section few solutions to this are discussed.

Absolute TOA is the difference between the moment of emission to the moment of the actual arrival of the defined signal segment to the microphone, but in the context of array signal processing, it is most useful to work with relative TOA values, denoted here by $\delta_{\mathbf{TOA}}$. Relative TOA values denote the delay of the other signals relative to one of the signals, so that there are $N-1$ measurements. In this thesis the abbreviation TOA refers to specifically relative TOA unless otherwise specified. TOA can be derived from TDOA values most easily by only taking the pairwise measurements that are related to only one of the microphones. To utilize all the TDOA measurements, the overdetermined system of linear equations can be solved in least squares manner, as in Moore Penrose pseudo-inverse:

$$X\beta = y \Leftrightarrow (X^T X)\hat{\beta} = X^T y \Leftrightarrow \hat{\beta} = (X^T X)^{-1} X^T y \tag{2.15}$$

However, this is not a very robust solution, as discussed in [21], because any outliers in any of TDOA pairs introduce bigger steering errors to TOA, than just using a subset of TDOA values. In this thesis, TOA is formed by taking a subset of TDOA values related to one of the microphones.

There are also more advanced methods to filter and track TOA values, like filtering the noise from the input signals as presented in the next chapter 2.5.3, or tracking the TOA with Kalman filters as was also shown in [21].

2.5.3 Wiener filtering and noise subtraction

As stated earlier, statistically correlated noise is destructive in TDOA estimation in the sense that it easily corrupts TOA values. The correlated noise can be both static and dynamic, which differ in that the static noise spectrum does not significantly change over time.

To reduce the correlating noise, the signals can be pre-processed before the cross-correlation function. In the method presented in [22], it was proposed that before the cross-correlation, the signals can be Wiener filtered to reduce static noise, and that the noise can be reduced even further by subtracting the noise covariance Γ between a pair of channels. Calculating the noise covariance can be formulated with expectation operation $\mathrm{E}(x)$ as $\Gamma_{n_1 n_2} = \mathrm{E}(\mathrm{n} \cdot \mathrm{n}^*)$, where the noise $\mathrm{n} = [n_1, n_2]$ is composed of noise estimates for the channel pair along with the signal model stated in equation 2.1. As stated earlier, noise estimates for static noise can be acquired during speechless segments.

Gnn subtraction, detailed in [32], assumes that there is static noise recorded by the microphones, which implicates that there are one or more noise sources. In this scenario, the cross-correlation can be estimated as in equation 2.16, where the Γ is the frequency domain cross-correlation between two signals, x_1 and x_2. Along with the notations of the signal model presented in equation 2.1, the cross-correlation is composed of two parts: $\Gamma_{s_1 s_2}$ represents the cross-correlation between the clean target signal s_1 and s_2 in a channel pair, and $\Gamma_{n_1 n_2}$ represents the cross-correlation of the noise signals in the corresponding channels.

$$X_1(\omega) \cdot X_2^*(\omega) = \Gamma_{x_1 x_2}(\omega) = \Gamma_{s_1 s_2}(\omega) + \Gamma_{n_1 n_2}(\omega) \tag{2.16}$$

The estimate of $\Gamma_{n_1 n_2}$ can be obtained during speechless segments, and it can be subtracted from the cross-correlation as:

$$\Gamma^{GS}_{s_1 s_2}(\omega) = \Gamma_{x_1 x_2}(\omega) - \Gamma_{n_1 n_2}(\omega). \tag{2.17}$$

This Gnn subtraction reduces the correlating noise, but to reduce non-correlating stationary noise, Wiener filtering can be used in addition to previous methods.

$$\Gamma_{s_1 s_2}^{WF}(\omega) = W_1(\omega)W_2(\omega)\Gamma_{x_1 x_2}. \tag{2.18}$$

where for $i = 1, 2$ the weights are estimated in [22] as

$$W_i(\omega) = \frac{|X_i(\omega)|^2 - |N_i(\omega)|^2}{|X_i(\omega)|^2}. \tag{2.19}$$

and again N_i is estimated during speechless segments. By combining the Wiener filtering, and Gnn subtraction, both the stationary noise and correlated noise are reduced. These can be combined by

$$\Gamma_{t_1 t_2}^{WG}(\omega) = W_1(\omega)W_2(\omega)(\Gamma_{x_1 x_2} - \Gamma_{n_1 n_2}). \tag{2.20}$$

This method should be rather robust TOA estimator also in noisy environments, but note also that to differentiate between speech and speechless segments, a VAD is needed. VADs were already discussed in section 2.2.

2.6 Beamforming algorithms

Sum-and-delay beamforming, despite its name, first delays the channels according to calculated TDOA values, and then takes the average over the channels. As was defined in the beginning of this chapter, and especially shown in equation 2.4, when multiple replicate measurements of an event are averaged, the additive noise is reduced.

One of the simplest forms of delaying the channels is of course by shifting the samples left or right according to the TOA values, but to utilize sub sample level accuracy of some TDOA estimators, the signal is delayed in the frequency domain with linear phase shift

$$Y(\omega) = \frac{1}{M} \sum_{m=0}^{M-1} e^{j2\pi f \delta_{\mathbf{TOA}}} X_m(\omega), \tag{2.21}$$

where M signals X_m are delayed by the delays $\delta_{\mathbf{TOA}}$. This follows from the fact that multiplying the frequency domain signal by $e^{j2\pi f \delta}$, where $2\pi f$ is the affected frequency bin, is effectively modifying the phase of the signal so that it is shifted by exactly the desired amount. j is used as the imaginary unit in this thesis.

Besides this basic method, there are multiple other beamforming algorithms such as Minimum Variance Distortionless Response (MVDR) [3] and GSC. These more advanced methods are usually more equipped in dealing with noisy environments, but they are out of the scope of this thesis, where the focus is more on steering of the beamformer, rather than noise cancellation.

2.7 Audio signal quality evaluation metrics

The output of the beamformer is usually a mono signal, which should now have a better sound quality than any of the discrete microphone signals alone. To measure the performance of the beamformer system, the quality of this output signal needs to be compared to the clean source signal if possible. Listening tests where humans assess the quality are expensive and time-consuming, although there is no better way to measure subtle differences in the overall quality. However, some aspects of the sound quality can be calculated and measured with objective metrics automatically.

Automatic speech quality assessment is not a solved problem. Objective metrics are created to look into specific features within the signal, and quite often these metrics do not even attempt to reflect perceived sound quality, but only how the used features behave. Before choosing the metric it is vital to understand what it measures. Besides objective metrics, there are also perceptual metrics that are based on psycho acoustics, like for example ITU-T Recommendation P.862 that defines the Perceptual Evaluation of Speech Quality (PESQ) [14].

Metrics can also be divided into two categories based on obtrusiveness: Metrics that require a clean reference signal for comparison, and metrics that do not. It should also be noted how the metrics that compare two signals to each other behave with noise segments, or whether they should only be used during active segments. In this context it is not useful to have the score affected by comparing noise segments to each other, especially if VAD is used to process those segments differently. In the next subsections few objective metrics are are studied, starting from simple SNR and building up to Segmental Source-to-Artifact ratio, which is used as the main metric in this work.

2.7.1 Signal-to-Noise Ratio

SNR is a well-known and very widely used metric in signal processing. There are at least two distinct definitions to how to calculate it. The most often used definition in audio signal processing is to calculate SNR as the ratio of the signal power P_{signal} and noise power P_{noise} [16] as

$$\mathbf{SNR} = \frac{P_{signal}}{P_{noise}}, \tag{2.22}$$

but SNR can also be defined as the ratio of the mean μ to standard deviation σ of the signal like

$$\mathbf{SNR} = \frac{\mu}{\sigma}. \tag{2.23}$$

This latter definition is used often for example in image processing [9, 23]. Also, to be noted is that SNR is usually presented in decibel (dB) scale.

Even if the SNR is defined as a ratio of signal powers, there are still differences in how to calculate the signal power and the noise power. Theoretically SNR can be calculated sample by sample from the clean reference signal by defining

$$\mathbf{SNR_{dB}} = 10\log_{10}\left(\frac{\frac{1}{N}\cdot\sum_{i=1}^{N} y^2(i)}{\frac{1}{N}\cdot\sum_{i=1}^{N}(y(i)-x(i))^2}\right). \tag{2.24}$$

Here the $y(i)$ is the reference and $x(i)$ the processed signal at sample $i = 1..N$. In this method the signals would need to be temporally aligned infinitely accurately, and not be affected by clock drifting in the recorders AD-converter or similar effects in the processing. This method also assumes that the clean reference signal has no noise of its own and that the scaling matches as well.

2.7.2 Segmental SNR

One notable modification to traditional SNR is to calculate it segmentally to see how SNR changes over time. As usual for speech processing, 20-30 ms windows are used, however there is no need for window overlapping in this context. From these segments the average of the Segmental SNR (SSNR) over the whole signal can be calculated, and by using a VAD the noise regions can be omitted to calculate the average SSNR over active regions. After the SSNR is calculated for all the K segments, the arithmetic mean of the SSNR values (SSNRA) is calculated as

$$\mathbf{SSNRA} = 10\log_{10}\left(\sum_{k=1}^{K}\frac{10^{\mathbf{SNR}_k/10}}{K}\right). \tag{2.25}$$

SSNRA is single number, that represents the mean SNR of the recording over active speech segments.

2.7.3 Signal-to-Distortion Ratio

Signal-to-Distortion ratio (SDR) [30] is an objective metric created for blind acoustic source separation evaluation. In SNR the original signal is decomposed to desired signal and noise components, and by taking this idea of decomposition a lot further the authors of SDR proposed a decomposition (2.26) of the processed signal x into source signal s_{target}, interference between output signals $e_{interference}$, noise e_{noise}, and other artifacts $e_{artifacts}$ that do not originate from the original signals. This decomposition is designed for separation algorithms, where source interference can be calculated for each separated source, and the background noise is separated as its own estimate. SDR uses clean source signals as the reference for the decomposition, and so it really reflects the change to the original.

$$x = s_{\text{target}} + e_{\text{interference}} + e_{\text{noise}} + e_{\text{artifacts}} \tag{2.26}$$

The decomposition in itself is detailed in [30], but in short it is based on orthogonal projections of the reference source signals y_i to certain projectors from which the components in (2.26) are derived from. After the decomposition to distortion components, SDR score is then calculated as

$$\mathbf{SDR} = 10 \log_{10} \left(\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interference}} + e_{\text{noise}} + e_{\text{artifacts}}\|^2} \right), \tag{2.27}$$

where $\|s\|^2$ is the given reference signal power, and the denominator $\|e_{\text{interference}} + e_{\text{noise}} + e_{\text{artifacts}}\|^2$ represents the power of the distortion components.

2.7.4 Segmental Source-to-Artifact Ratio

In the scenario used in this work, however, there is only one sound source, and thus no need for the interference component in the SDR score. In this context there is not really a need or interest to separate noise from artifacts, and as there is no sensible estimate of the noise component at this point either, the noise component is dropped as well. What is left is the artifacts component, which now also includes the noise in the signal. The SAR score that is related to the SDR calculates exactly this, as

$$\mathbf{SAR} = 10 \log_{10} \left(\frac{\|s_{\text{target}} + e_{\text{interference}} + e_{\text{noise}}\|^2}{\|e_{\text{artifacts}}\|^2} \right). \tag{2.28}$$

This might seem to be pretty close to the definition of SNR, as there is now only the desired signal and the noise, but as explained in earlier in section 2.7.3, the decomposition is made differently.

The baseline SAR score does not take into account the noise segments, so calculating SAR for the whole recording would also compare the noise segments to each other. By calculating SAR segmentally, and omitting the noise segments with VAD, the arithmetic mean of the segmental SAR values, SSARA, can be calculated. This method is motivated by the SSNRA, and presented in [21]. To take the arithmetic mean of the logarithmic SAR scores calculated for each segment k, the scores are first cast to linear scale, averaged, and then brought back to logarithmic scale as in

$$\mathbf{SSARA} = 10\log_{10}\left(\sum_{k=1}^{K}\frac{10^{\mathbf{SAR}_k/10}}{K}\right). \tag{2.29}$$

SSARA score represents the average strength of the target signal during the active segments, in respect to the noise and artifacts that are still present in the output signal.

3. IMPLEMENTATION OF THE SYSTEM

The objective of this study is to assess which attributes affect the speech quality in ad-hoc array beamforming in contrast to fixed arrays. As was pointed out in section 2.4, where the ad-hoc array was defined, the major differences are the unknown locations of the devices, asynchronous operation, and thirdly the differing device performances. Because the device performance can not be approached in detail because of the unknown device properties, it is not in the scope of this thesis.

The focus of the study is to understand how the unknown locations of the devices affect the sound quality. Both the simulations and the recordings have seven microphones in a circle of a varying diameter, and one loudspeaker as a sound source at a fixed distance from the origin of the circle. The idea is to reflect a normal meeting room case, where there are people around a table with their handsets on the table in front of them, and there is one person talking. This scenario is pictured later in section 3.3 in figure 3.3. Note that instead of real speakers, in this work a loudspeaker is used for accurate and repeatable recordings. The exact dimensions of the rooms and device locations are presented in each section. Three main parameters of interest here are the diameter of the microphone array, the noise level in the room, and the room reverberation. These parameters are varied in the simulations and the results are compared with results from actual recordings.

A signal processing system to produce these results was constructed in MATLAB, and it implements the system described in the theory section. Even though initially implemented, the Generalized Sidelobe Canceller algorithm is not used in these tests. As there were no spatial noise sources in the recording scenarios, it actually performed equally, if not a bit worse, than simple sum-and-delay. Due to this only the results for the sum-and-delay algorithm are reported here.

The rest of the chapter is constructed as follows. First the implementation specifics are detailed in section 3.1, which is divided to subsections detailing the implementation specifics of the VAD, TOA estimation, beamformer, and evaluation blocks. This implementation of the system is then applied to simulations in section 3.2, and the recordings in section 3.3.

3.1 System configuration

Beamformer systems such as introduced in the theory section, consist of multiple blocks, and become easily complex. Consequently, there are also multiple parameters related to each of the algorithms and relating to the general setup. Most of these parameter choices are detailed next, along with some implementation specifics and solutions related to multi-channel and ad-hoc handling. First the details related to each of the beamforming system blocks are explained, and then the details of simulation and recording scenarios are presented.

Both in simulated and real recordings the sound source was a loudspeaker playing samples from Texas Instruments and Massachusetts Institute of Technology (TIMIT) corpus [8] of read speech. Although TIMIT was designed for English speech recognition research in mind, it is well suited for this use as well, because the recordings are very clean and high quality, phonetically rich speech signals. The TIMIT database consists of recorded utterances by 630 speakers with eight different English dialects. To reduce the number of recordings for the evaluation while still preserving sufficient variance, one speaker was randomly selected for each of the eight dialects available. The speakers include both male and female speakers, and they read ten sentences each. In total this equals eight minutes and six seconds of audio. TIMIT recordings are 1-channel 16-bit pulse-code modulated (PCM) digital audio capsulated in WAV audio containers.

The sample rate of the TIMIT corpus was 16 kHz, which corresponds to the Nyquist frequency of 8 kHz. While human speech rarely goes over that, it should be noted that the low sample rate might be restrictive in representing full human speech spectrum. The processing of the data was made with 48 kHz sampling rate, and all the files were either recorded at that rate, or upsampled to that rate prior to processing. However, after processing the data is downsampled so that it can be fairly compared against the original TIMIT recordings.

Resampling a signal for a new sampling rate can be done with only few steps. For example, to get from 48 kHz sampling down by a factor of n to $48/n$ kHz sampling, the signal should first be low-pass filtered for any frequency content that is not representable in the new sampling rate. Next, every second sample is dropped from the signal. For upsampling by factor of n, a zero is inserted between every sample, and then the signal is low-pass filtered so that the cutoff frequency is the Nyquist frequency. To sample at any rate, these operations can be combined: To get from 44100 Hz to 48000 Hz, the signal is upsampled by 160 and then downsampled by 147 [25, p. 139]. How to make that computationally efficient is not trivial, and also not in the scope of this thesis.

The system was constructed in an offline manner, so that it would be possible to take averages over the whole signal for constructing oracle TOA vectors. However, the core of the system should be able to function as a real time online system, given enough computing power. This is achieved by processing the data in time windows, the specifics of which were explained in the section 2.1. The implementation uses framewise processing within each of the signal processing blocks, although the blocks are processed separately and independently of each other. In other words, each of the blocks process the whole signal at once, and use framewise processing within the block. To make the system an online system, this framewise processing would need to be moved outside of these blocks.

First in MATLAB, the recorded or simulated data was read to a struct, and cut so that all the recordings were of equal lengths. The files were also roughly pre-aligned with time-domain cross-correlation over the whole file to satisfy the sufficient synchronization defined in section 2.4. After the delays were defined with the cross-correlation, the signal samples were shifted so that they lined up, by padding with zeros. Also, while calculating the cross-correlation over the whole file, it was inspected if any of the devices had inverted phase (i.e. If the maximum of the cross-correlation was negative). If such inverted phases were found, the phase was inverted by multiplying it by -1.

After these preparatory steps, the signals are inputted to the system. The system consists of blocks as shown in the figure 2.1, in the beginning of chapter 2. As in the figure, first a VAD is calculated, then the steering vector consisting of TOA values, and finally everything is combined in the beamformer block for the output. The output of the beamformer is then compared to the original signal with the SSARA metric that was defined in section 2.7.4.

3.1.1 VAD block

An implementation of the ITU-T standard G.729 Annex B, as presented in more detail earlier in section 2.2, was used here as the VAD. Only difference to the default settings of the standard VAD was that the sampling rate was raised to 48000 Hz, and the band width was doubled raised correspondingly.

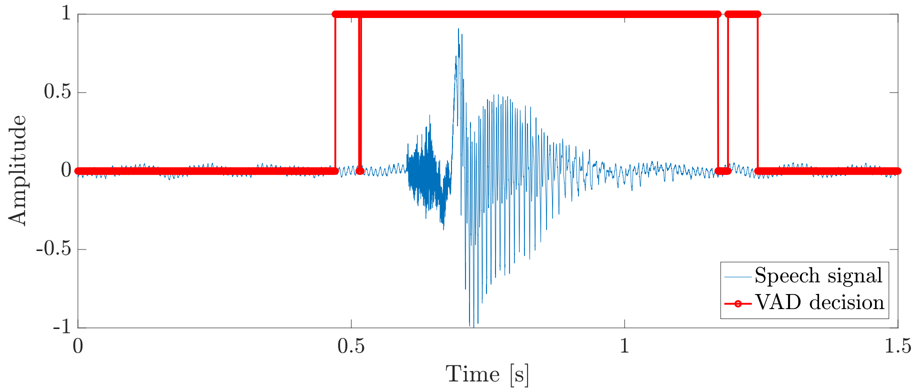


Figure 3.1 *ITU-T G.729 Annex B VAD used on a speech signal.*

Because the VAD is used before beamforming in the time-alignment, a multi-channel approach is needed. VAD is therefore calculated for each of the channels, and the decision for a mutual VAD is achieved by majority vote.

$$\mathbf{MUTUAL}_i = \left(\sum_{n=1}^{N} \mathbf{VAD}_{n,i} \right) \geq \lceil \frac{N}{2} \rceil, \tag{3.1}$$

where N is the number of channels, and i the frame. The output of the VAD is binary. This means that for each signal sample, the VAD output is zero, false, when the algorithm estimates there is no speech, and one, true, when it is estimated to contain speech. An example of ITU-T G.729 output is pictured in figure 3.1. The window size for the algorithm was the default 160 samples when used in wide band operation.

After the mutual VAD was calculated, possible amplification differences between the channels were taken into account by measuring the noise power during silent segments. Then each of the channels were multiplied with the ratio between the noise power of the first channel, and the channel's own noise power. Note that clipping would not be an issue here, since the computation is performed with double precision floating-point numbers. Still, it is to be noted that after this operation some channels might have values over 1.

3.1.2 TOA block

The steering vector for the beamformer was constructed with TOA values for every time window, as explained earlier in section 2.1. The frame length was 4096 samples, which corresponds to 85 ms at 48 kHz sampling rate. The overlap of the frames was 50 %, equating to 2048 samples. The frames were multiplied with Hann windows [2] of the same length. The implemented TOA block uses Wiener filtering and Gnn subtraction, presented in section 2.5.3. These methods were used to reduce the noise in the TOA steering vector.

Because the VAD was calculated for every sample, and the TOA is calculated for every frame, the VAD decision was set to be zero if it was zero for most of the window length. The VAD decision in the TOA block is used for switching on the Gnn noise estimation, and also for keeping the output static during silent segments. In this work it was chosen that the steering vector is kept the same while VAD is zero. By keeping the TOA static during silent segments, it is ensured that the beamformer is not steered towards noise sources, that easily become dominating sound sources during the silent segments.

The cross-correlation calculation was not only limited within the time window of 4096 samples, but also artificially with distance related variable D_s, by restricting $\delta_{\mathbf{TOA}}$. This was to ensure that irrespective of the window length or sampling rate, the distance between any two microphones would not be assumed to be larger than $D_l = 2.5$m. The distance variable can be calculated by knowing the speed of sound c, assumed here to be 343 m/s, by

$$D_s = \left\lceil \frac{F_s * D_l}{c} \right\rceil, \tag{3.2}$$

where F_s is of course the sampling rate. This equates to $D_s = 350$ samples in this work. This additional layer of protection reduces noise during segments when the steering does not work well, for example when the VAD is one, but there is no speech to track. What happens when the limit of 2.5 m is too restrictive is studied in the simulations, where the array size gets as large as 2.5 m and 3 m.

After the cross-correlation is calculated via Wiener filtering and Gnn subtraction, quadratic interpolation (defined earlier in equation 2.12) was used to determine the subsample accurate TOA value.

As reference, an oracle TOA was calculated by taking the median of the TOA values in a low noise case for each scenario. Oracle TOA is useful for finding out the baseline score for the beamformer. It tells how well the beamformer would perform if the steering was perfect.

3.1.3 Beamforming block

Beamforming algorithm used in this thesis is the simple sum-and-delay method that was presented in section 2.6. It uses the same 4096 sample long Hann windows with 50 % overlap, as did the TOA estimation block. This comes naturally from the fact that the delays were also calculated for similar window lengths. Both the dynamically calculated and oracle TOA were used for the beamforming separately, so that there are both naturally steered and oracle beamformed outputs.

3.1.4 Evaluation of the system performance

The evaluation is done with the SSARA algorithm presented in section 2.7.4. First, as the TIMIT database original files have 16 kHz sampling rate, both the VAD and the beamformer output was downsampled to that rate as well. SSARA was calculated in 4096 samples long windows, which equates to 256 ms. The windowing here did not use overlapping, because that would have only meant that the score was calculated twice for every segment. The window length here does not affect the output so much, as it mainly affects how accurately VAD limits can be obeyed.

To calculate the score, the reference signal was time-aligned to the beamformed signal by calculating cross-correlation in time-domain, and shifting the signal appropriate amount of samples. This was to preserve the exact signal values. The SDR underlying in the calculation of SSARA does take into account mismatch in time alignment by allowing time-varying distortions in the used projectors [30]. The SDR implementation used was the “BSS eval” toolbox (version 3, 2007) by the author of the author of SDR [30].

The score of speech-only segments is calculated. To account for the sample-accurate VAD, if the VAD was zero anywhere in the frame, then the score for that segment is excluded from the SSARA score. Besides calculating the SSARA score for the TOA and oracle beamformed signal, the SSARA score was also calculated for the all the channels of the array separately. This ensures that the best single channel could be determined. To accommodate this, each of the single channels were aligned with the reference signal as well. By calculating also the best single channel score, it can be compared how much the beamforming improves the result.

3.2 Simulations

For simulations the system utilized MCRoomSim [31] room acoustics simulator, which was used to create room impulse responses (RIR) for each simulated microphone. To get the audio signal for each simulated microphone, the source recording is convolved with the calculated RIRs. The impulse responses were generated for 48 kHz sampling rate, and the TIMIT data was upsampled before convolving with the RIRs.

Two rooms were created for the simulations; an office room, and an anechoic room. The office room was defined to be 4 m wide, 4.7 m long, and 2.6 m tall. Relative humidity of the room was set to 35 %, and the temperature was to 22° C. The room humidity is related to the speed of sound in the room, and the temperature magnifies the effect of the humidity. Explaining the physics behind the speed of sound in air is out of the scope of this thesis, as the temperature fluctuation in the recordings was negligible, this does not matter much for the overall system.

The wall absorption (A) and scattering (S) parameters for the office room scenario are left their default values, which are listed for all the six faces of the room in Table 3.1. In the table the X0 and X1 correspond to the walls related to the width of the room: they are the left and right walls. Similarly Y0 and Y1 correspond to front and back walls of the room, and Z0 and Z1 are related to the floor and ceiling. For absorption values, zero would mean the wall is purely reflective, and one purely absorptive. The scattering coefficients control how the reflection acts. When it is zero, the wall reflection is purely specular, and one would make reflections purely diffuse. The MCRoomSim allows all of these parameters to be set for user specifiable frequencies, but for the office room scenario, these were left to the listed default values.

Table 3.1 *Room simulator wall absorption (A) and scattering (S) coefficients for the office room simulation.*

	X0		X1		Y0		Y1		Z0		Z1	
Frequency	A	S	A	S	A	S	A	S	A	S	A	S
125 Hz	0.20	0.50	0.30	0.50	0.30	0.50	0.45	0.50	0.50	0.60	0.60	0.60
250 Hz	0.20	0.50	0.30	0.50	0.30	0.50	0.45	0.50	0.50	0.60	0.60	0.60
500 Hz	0.20	0.50	0.30	0.50	0.30	0.50	0.45	0.50	0.50	0.60	0.60	0.60
1 kHz	0.20	0.50	0.30	0.50	0.30	0.50	0.45	0.50	0.50	0.60	0.60	0.60
2 kHz	0.40	0.55	0.50	0.55	0.50	0.55	0.45	0.55	0.65	0.65	0.80	0.65
4 kHz	0.30	0.50	0.40	0.50	0.30	0.50	0.45	0.50	0.60	0.60	0.70	0.60

The anechoic room was set to be 5 m wide, 5 m long, and 3 m tall, with humidity of 25 %, and temperature of 22° C. For the anechoic room the absorption and scattering parameters were all set to 0.999, which means that all the walls absorb very much all of the sound, and the remaining 0.001 are almost purely diffuse reflections. This allows us to establish a baseline for the metrics, and also observe how much the room acoustics affect the score by comparing to a situation where the room acoustics have very small role to play.

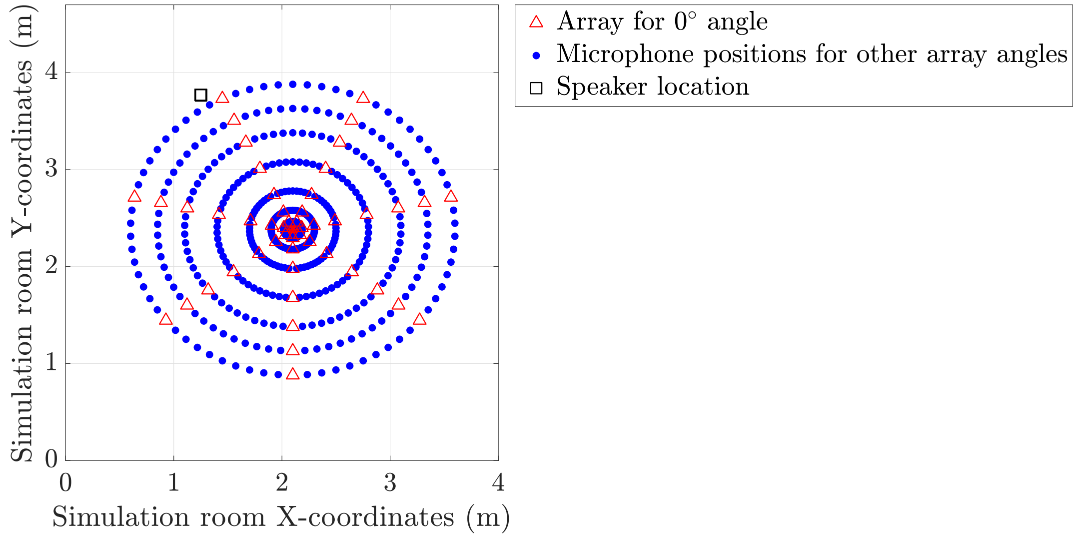


Figure 3.2 *Simulated room with all array sizes and angles..*

For both rooms, the microphones were set up in the room as pictured in figure 3.2. It should be noted, that the rooms were of different size, the figure here is picturing the office room case dimensions. Seven microphones were set up in a circle so that the center of the circle was located in coordinates (X = 2.1, Y = 2.38, Z = 0.72), the units of which correspond to meters. The diameter of the microphone circle d_i was varied from 15 mm to 3 m, as is listed on Table 3.2.

Table 3.2 *Microphone array diameters in simulations.*

Diameters	15 mm	85 mm	17 cm	40 cm	80 cm	1.4 m	2.0 m	2.5 m	3.0 m

After measuring the RIRs for initial 0° angle, the whole array was rotated, so that the average performance over the rotations can be calculated. The complex polar coordinates $z_{n,i,p}$ for such rotation can be calculated by

$$z_{n,i,p} = \frac{d_i}{2} e^{j \cdot (\frac{-2\pi n}{N} + \frac{p\pi}{PN},)} \tag{3.3}$$

where the microphone index n ranges from 1 to N, i is the index of the diameter, and $p = 1, 2...P$ is the rotation index. Here, ten equal rotations were made ($P = 10$), in total covering of the sector between the seven microphones ($N = 7$). The added microphones were set to be omnidirectional microphones.

The loudspeaker was setup to coordinates (X = 1.25, Y = 3.77, Z = 1), and the orientation was set to face the center of the microphone array in the X and Y plane, keeping the rotation in Z plane zero. The loudspeaker type was set to “tannoyv6”.

To make the situation more lifelike, so that the microphone circle would not be perfect, white Gaussian noise was added to the room measurements and the microphone positions. Pseudorandom values u drawn from standard normal distribution $\mathcal{N}(\mu = 0, \sigma^2 = 1)$ were drawn for all measurements. The noise N_{Loc} affecting each measurement was constructed from the random numbers u with

$$N_{\text{Loc}} = u * \sqrt{\frac{1}{P}}, \tag{3.4}$$

where P is used to scale the distribution's standard deviation. For the locations, P was set to 60. This roughly corresponds to standard deviation of 0.001, so in the context of locations the noise added had standard deviation of 1 mm.

Equation 3.4 also corresponds to adding white Gaussian noise to 0 decibel watt (dBW) signal so that the resulting signal would have SNR of P dB. 0 dBW signal is for example a sinusoid with amplitude of $\sqrt{2}$. Even though the noise addition here does not correspond to the real SNR of the used audio signals, this method for adding noise was used because MATLAB had built-in function **awgn** readily available.

Besides having two rooms, the simulations are also used for testing how much uncorrelated noise distorts the results. White Gaussian noise was thus also added to the microphone signals, after convolving it with the room impulse response. Thus the signal model for the room simulation signals can be written as

$$S_n = (X \star RIR_n) + u_{i=1..L} * \sqrt{\frac{1}{P}}, \tag{3.5}$$

where X is the original source signal, RIR_n the room impulse response related to microphone n, and L the length of the signal X. The noise power variable P that affects the 0 dBW SNR was varied from 200 to 2, with values shown in Table 3.3. Listed in the table there are also the standard deviations related to the 0 dBW SNR.

Table 3.3 *SNRs used in simulations.*

0 dBW SNR	200 dB	38 dB	18 dB	8 dB	2 dB
Std	$9.928 \cdot 10^{-11}$	0.0125	0.1280	0.4007	0.7862

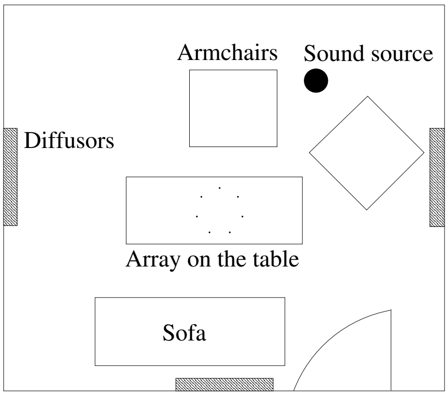
So in conclusion for simulations, in this thesis the aim is to see how the performance of the system is related to the physical size of the array. To study that, two simulation rooms were constructed so that the room impulse response's effect to the beamformer performance can be evaluated. By varying the array diameter and noise level the idea is to test and see what is the relation of the array size to the performance. The room dimensions, microphone and loudspeaker locations had noise added to them, and the simulation was made 10 times with the array rotating to 10 differing angles, so that possible imperfections in the simulator can be averaged out.

3.3 Recordings

Even though the TIMIT data was 16 kHz, the recordings were done at 48 kHz sampling rate, as well as the processing of the recordings. This also makes for a fair comparison, because the simulations were processed with 48 kHz as well.

The recordings were made at the audio laboratory of the Audio Research Group (ARG) at Tampere University of Technology. This laboratory has an acoustically insulated measurement room, and most of the equipment can be used from the outside of this room. This ensures that no outside noise interferes with the recordings, and that the measurements can be made without being inside the measurement room. The measurement room has the same dimensions as the simulated office room: 4 m wide, 4.7 m long, and 2.6 m tall. A rough blueprint of the room is pictured in figure 3.3. As can be seen from the figure, the measurement room has typical room furniture like armchairs, a sofa and a table, but the room also has three diffusors on three of the walls, and also a rather low ambient noise level due to being acoustically insulated.

(a) Photograph of the recording case setup. The used loudspeaker is circled.



(b) A rough blueprint of the room where the recordings were made.

Figure 3.3 *Recording case in pictures.*

The playback for the recordings was initiated through MATLAB running on a computer located at the monitoring station. The computer was connected with a Firewire to an RME Fireface UFX audio interface. The RME Fireface was connected with an ADAT connection to an RME Octamic - the recording interface used via Audacity software to record while the playback was going on on MATLAB. The monitoring station where the recordings were initiated was located outside of the measurement room, with the Firewire cable going through a small acoustically sealed hole into the measurement room. The amplification of all the microphones was manually set to the same level on the Octamic interface.

Inside the measurement room the Fireface was connected to a Genelec 1029a loudspeaker. The recordings were made with an array of seven omnidirectional Sennheiser MKE 2 P-C lavalier microphones. These microphones are electret condenser microphones, and they were all connected to the same RME Octamic. The microphones were laid on small foam pads, so that they can be easily positioned. Additionally the foam pads might also dampen strong reflections and vibrations from the table they were laid on. The recording setup for array diameter of 40 cm can be seen in figure 3.3.

The array size was varied from 1.5 cm to 140 cm, with the diameters listed in Table 3.4. These are the same diameters as for the simulations, except the recordings do not have the largest three diameters the simulations had. This is due to the difficulty of placing the microphones to a level circle without having to move furniture, which might in turn affect the room acoustics.

Table 3.4 *Microphone array diameters in recordings.*

Diameters	15 mm	85 mm	17 cm	40 cm	80 cm	1.4 m

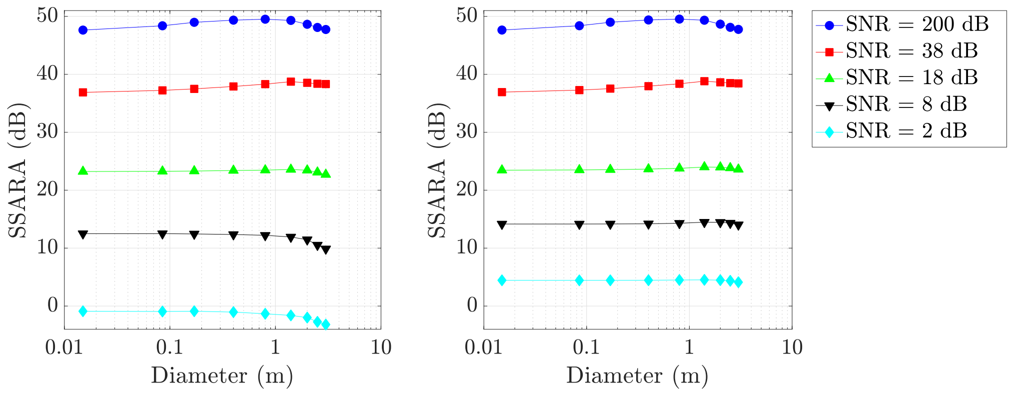
The array center and the loudspeaker were also set to the same position as in the simulated office room scenario: The loudspeaker was at coordinates (X = 1.25, Y = 3.77, Z = 1), and the array center at (X = 2.1, Y = 2.38, Z = 0.72). The microphones were placed on the table by hand, and the margin of error for the locations is estimated to be ± 1 cm. When the array size grew too large to be all on the table, microphone stands with arms were used to place the microphones to their proper spots.

4. RESULTS

Both the simulated and the real recordings produced SSARA scores reflecting the sound quality achieved in each of the scenarios. In this section both the absolute SSARA scores, and the difference gained by beamforming in comparison to the best single microphone in the same scenario are inspected. More in-depth analysis and interpretation of the scores will be done in the next section 5.

It is to be noted here, that the SNR levels shown for the simulations are not the actual SNRs of the signal, but as explained in section 3.2, they are the SNR of a signal when it is assumed that the original clean signal has power of 0 dBW. Also note that in the simulations the largest two array sizes, with diameters of 2.5 m and 3 m, are actually limited by the algorithm, as discussed in section 3.1.2, so that the steering is not able to perform at its best.

First the simulation scores are shown, first the absolute scores and then the relative scores, which are created by subtracting the best single microphone score from the absolute score. This relative score highlights the gain in the SSARA score the beamforming gives in comparison to just using the best microphone. Note that often the nearest microphone is also the best in terms of the SSARA score. The scores are shown on a semi logarithmic axis, because the chosen diameters are almost linear on the logarithmic scale.

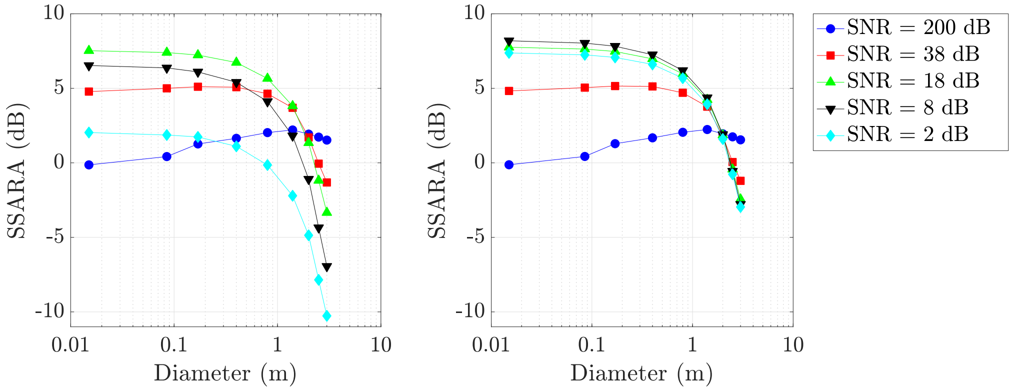


(a) TOA steering

(b) Oracle steering

Figure 4.1 *Beamforming results for the simulations in the anechoic room.*

In the figure 4.1 are the absolute scores of the simulations in an anechoic room. It is noticeable that the range of scores varies largely with the noise level, although the array size does seem to have few dB differences for differing array sizes. Both the oracle steering and the TOA steering seem to have their highest scores on 80 cm and 140 cm arrays on low noise, while the smaller arrays seem to be slightly better when the noise level is higher. It is also noticeable that the highest two noise levels show that the TOA steering starts to suffer in comparison to the oracle steering.

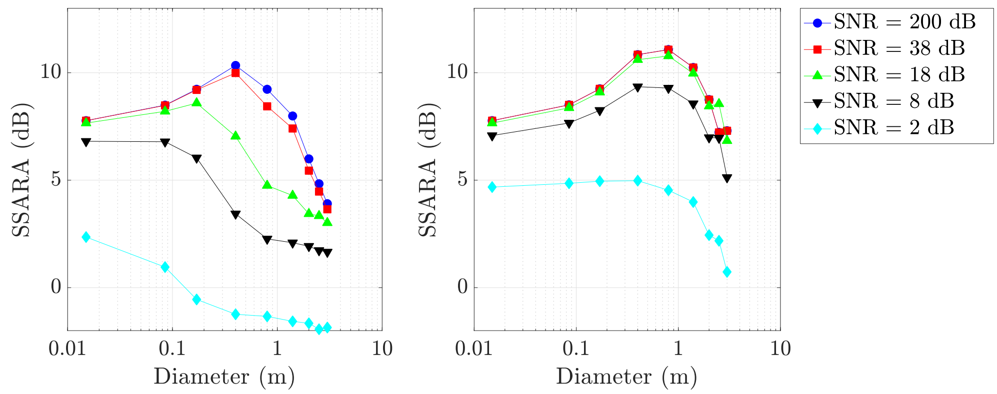


(a) TOA steering

(b) Oracle steering

Figure 4.2 *Beamforming results for the simulations in the anechoic room, normalized to the best microphone in each case.*

Interestingly, the difference to the best microphone in the simulated anechoic room, pictured in figure 4.2, shows that in contrast to the stable absolute scores, the benefit of beamforming is not so straight forward in an anechoic room. With TOA steering, the largest benefits are with medium noise, especially with smaller arrays. The larger arrays seem to yield the highest gain from beamforming in comparison to the small arrays only in the low noise conditions. The oracle steering has similar results, but again with the two highest noise levels, the gain is much higher in comparison to TOA steering.

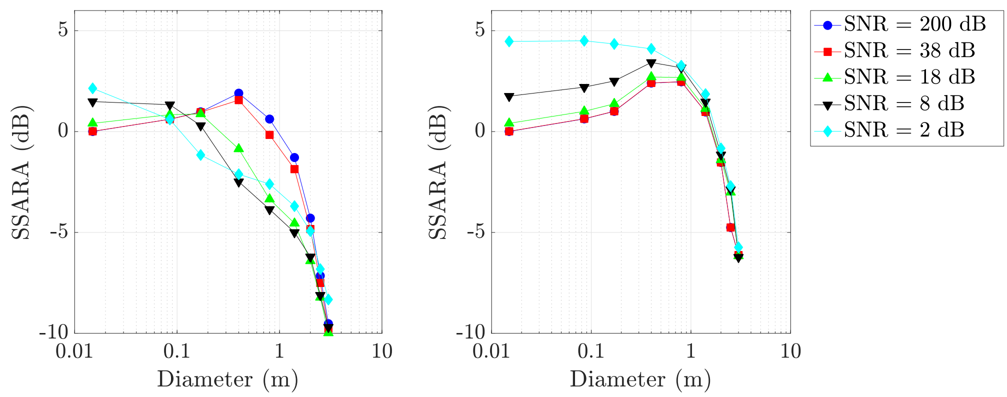


(a) TOA steering (b) Oracle steering

Figure 4.3 *Beamforming results for the simulations in an office setting.*

The absolute SSARA scores for the office room simulations, detailed in 3.2, are shown in figure 4.3. It is clearly visible in the TOA steering, that in low to medium noise conditions the arrays of sizes of 17 cm and 40 cm seem to yield the best SSARA scores. The higher the noise however, and the smaller arrays seem to be the better option again. The oracle steering shows that if it would be possible to estimate the steering vector more accurately, the larger array size of 80 cm would yield the best results in low to medium noise. In the highest noise conditions the 40 cm array would be best on oracle steering.

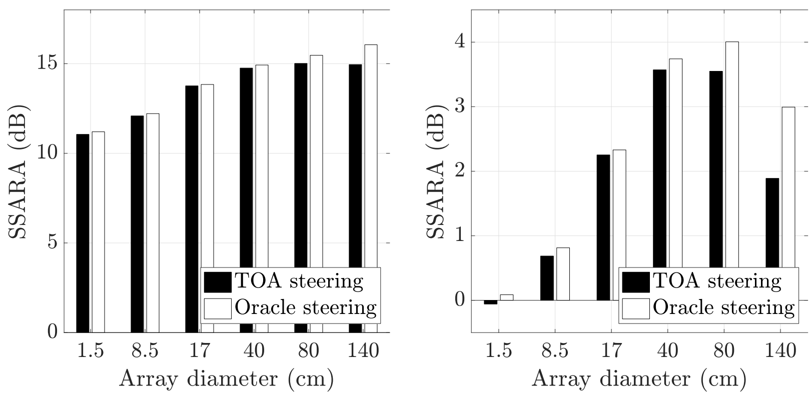
It is also notable here, that there is a nonlinearity when moving from 2 m array to 2.5 m, exactly where the steering limitation of the system is. In this case, it would seem that the score for 2.5 m is actually higher than it should be. The steering limitation was explained in section 3.1.2.



(a) TOA steering (b) Oracle steering

Figure 4.4 *Office setting results normalized to the best microphone in each case.*

Pictured in figure 4.4 are the office scenario scores normalized to the best microphone. For the oracle steering the scores here have almost identical shape, but it is notable that for the highest benefit from beamforming is achieved in the highest noise case, by using the smallest array. Looking at the TOA steering scores reveals that a clear benefit from beamforming is only gained with the 40 cm arrays in low noise conditions, and by smaller arrays for the higher noise levels. The reasons why beamforming with the smallest arrays does not get better scores in low noise conditions are discussed in the next section. It is also notable here, that when the relative score goes below 0 dB, it means that the beamformed signal is actually worse than the best microphone.



(a) Absolute scores

(b) Difference to the best microphone in each case

Figure 4.5 *Beamforming results for the real recordings.*

The results for the real life recordings are shown in figure 4.5. The absolute scores show that increasing the array size makes the SSARA score seemingly higher. However, it should be noted that simulations show decrease in performance when array diameter is larger than 80 cm. TOA steering is almost as good as the oracle steering in this case, although the difference is a bit larger with the larger 140 cm array. When comparing the difference to the best microphone, it is shown that in real life recordings, the beamforming yields highest benefits with 40 cm arrays on TOA steering, and 80 cm arrays on oracle steering.

In conclusion, it can be noted that the experimental results correspond roughly to the simulated office room results. The implications of these results are detailed in the next chapter, as well as the relationship between the scores for the simulation and the experimental setting.

5. DISCUSSION

It is imperative to keep in mind while analyzing the results for both the simulated and real recording scenario, that while the microphone array diameter was varying, the speaker was located at the same position all the time. Taking into consideration that the microphone nearest to the speaker generally also has the best speech reference, it can be deduced that the benefits of beamforming diminish when the array reaches the speaker. The simulations showed that if the array gets so large that the microphones get very near the loudspeaker, as visible from the simulation scores, the other microphones would be so far away from the speaker that beamforming do not produce good results. However, the focus here was to study a realistic office room setting, where the size of the array was varying. Otherwise, the case could have been set up differently to rule out the fact that larger arrays also get closer to the speaker, and thus the nearest microphone gets better sound quality. For future work, interesting factors could be found out if the closest microphone was kept equally distant from the loudspeaker, while growing the array size.

The office room simulations corresponded roughly to the real life experiment scores. However it should be noted that the deductions made from the simulations should be confirmed with real life experiments. The benefit of simulations is that they let us hypothesize how the system would likely perform in different conditions more easily than conducting real life experiments.

The results showed, that using beamforming allows the use of a compact ad-hoc array, instead of every speaker having their own microphone set up in front of them. As the absolute scores in figure 4.5 showed, the performance on the real recordings in this case is around the same with array sizes 40-140 cm. The increase of sound quality for arrays larger than 40 cm is observable, but marginal.

The good scores for 40-140 cm arrays are possible because of the gains from beamforming, that are shown in the normalized scores, where the score of the best microphone in the array was subtracted from the absolute score. Speculatively this means, that instead of having microphones just closer to the speakers, they can be randomly placed on a table to an array of around 40 cm diameter, and still have almost the same performance. In this study the microphones of the array were not really randomly placed, but the used algorithms did not utilize the location information.

Rather surprisingly when looking at the normalized scores, the smallest arrays do not seem to have any benefit over the best microphone, even though as discussed in the beginning of the theory section, the beamforming should yield better results because of the fact that there are multiple independent measurements made from the same event. One significant factor here is that when the microphones are close to each other, the distortions caused by the room environment (caused by the RIR in simulations) are basically the same for all the microphones, thus being correlated. When the microphones are moved farther away from each other, also the room impulse response is more different for each microphone, and so the distortions in the measurements are again sufficiently uncorrelated, as was assumed by the signal averaging in equation 2.4.

By combining these two observations, that when the microphone array is so large that the closest microphone has very good reference, and the observation that microphones very close to each other experience similarly correlated distortions, the behaviour of the normalized results in low noise cases can be explained to some extent: The benefit of beamforming is close to zero when the array is small, then reaches its optimum where the distortions in the measurements are sufficiently uncorrelated, and diminishes when the closest microphone gets too close to the loudspeaker.

Conversely there is also a similar relation when looking at the normalized scores for higher noise levels (for example in Figure 4.4): The benefit of beamforming with the small arrays is larger, because there is now uncorrelated noise in each of the measurement, and so the beamforming is beneficial.

Distortion correlation also affects the absolute scores: The small arrays suffer because of the correlated distortions in the signal. However it is not fully clear why the absolute scores suffer when the array is large: To some extent it can be explained by the steering being harder if the noise level is high, and by the fact that signals from far away microphones just are more corrupted, and might just be counter-productive to use them in the beamforming at all. Still, there are probably more factors that affect the shape of the results.

When the results were normalized to the best microphone, it is visible that the beamforming is actually worse than the best microphone on large arrays, especially with high noise and TOA steering. This is mostly due to the fact that as the array size grows, the closest microphone gets very near to the speaker, while others get farther away. The high noise overpowers both the steering, as visible in the difference between TOA and uncorrupted oracle steering, and finally the beamforming. This observation supports the theory of far away microphones just getting too corrupted for them to be of any use in the beamforming.

It is notable that the real life SSARA scores were actually better than the simulated scores for office room. However it should be noted, that the room where the recordings were made, was acoustically rather clean environment, and the default values of the simulated room might imitate harsher room acoustics, which lead to lower scores in the simulations. Also note, that the real recordings would probably have had the absolute scores drop, if the array size was grown more. However, further studied should be conducted to confirm this.

When it comes to other issues in this work, the use of the MATLAB function **awgn** to add white Gaussian noise to the signal lead to a bit confusing definition of noise level in the signals, as explained in section 3.2. It would have been more simple to define the noise level in a different manner, so that it would not imply that the SNR of the signal is the one shown in the figures. Especially when the SNR definition is studied in the section 2.7.1.

The limiting factors in regard to microphone location was the used window length, which was further reduced with the arbitrary limit to inter-microphone distances, as was discussed in 3.1.2. The use of this arbitrary limit might have been counter productive in this work, as it might be confusing as to why such a limit is introduced. However it was shown in the absolute scores for the simulated office, in figure 4.3, that the limitation actually helps keep the score higher than it would otherwise be for the 2.5 m case. This might be due to the fact that the noise in the steering signal can not be too much off the target, when the range of possible steering delays is limited. In future work, this limit could be used in part to determine if a microphone should be excluded from the beamforming.

Further studies are required to get into the bottom of why the results show the maximum gain from beamforming to settle to array diameter of 40-80 cm in both the simulated and real recordings, depending on steering accuracy. From simulations, it can be speculated that the room acoustics play a role in this, as speculated in the discussion chapter. However, this study can not show or pinpoint why exactly there seems to be an optimum array size. Note that this is optimal only in the sense that it optimizes the benefit from beamforming: The experimental case did not show lower absolute scores for larger arrays. In that case only the benefit from beamforming in comparison to best single microphone was lower than for the 40-80 cm arrays.

Instead of having the microphones in a circle of growing diameter, an interesting setup for studying the physical size of the optimum array in future work would be to have the microphones equidistant to each other, and grow the inter-microphone distance. Future work is also needed on exploring other different array geometries, because the use of circular arrays here was just an easy way to set a size for the array. The benefit of different array geometries is not explored in this work. Also, how the number of microphones affect beamforming performance was not explored in practice. In theory it was mentioned how much it could potentially help gain SNR, but it was also shown that when the steering is not correct, the beamforming might actually perform worse. The number of microphones might also affect this, and excluding some of the microphones might yield better performance.

6. CONCLUSION

In this thesis ad-hoc audio beamforming system was constructed for meeting room recording purposes. The focus was to study the performance of the audio quality enhancement while varying the physical size of the microphone array. The purpose was to find how much the physical size affects the performance in a typical meeting room scenario.

In the theory chapter, this thesis presented the algorithms used in the different blocks of the system: The VAD, TOA estimation, and the sum-and-delay beamformer blocks. Additionally, metrics for analyzing the output of the system were presented, as well as considerations for framewise processing, ad-hoc systems, and spatial aliasing.

The implementation specifics for the system using MATLAB were discussed in the implementation chapter. This included the used parameters for the algorithms, and the details of the simulation and recording setups. The results showed that the studied beamforming system is a viable solution for combining multiple audio signals from a microphone array into one superior signal in a real world room environment.

In discussion chapter it was analysed when beamforming had benefits over the closest microphone in both the simulations and real recordings, taking also into consideration varying amount of added non-correlating noise. Most importantly, it was shown that the benefit from beamforming is optimized in arrays of size 40-80 cm in the studied cases. The optimum diameter for the array found in this study is however only subject to these cases. This study did not comprehensively find all the reasons affecting this behaviour. Affecting factors could include the room environment, the used array setup, and the used equipment. The case studied here was rather specific, and further studies into these factors are required to have any general recommendations about the physical size of the microphone arrays in ad-hoc beamforming.

REFERENCES

[1] A. Benyassine, *et al.*, "ITU-T recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 64–73, Sep 1997.

[2] R. B. Blackman and J. W. Tukey, "The measurement of power spectra from the point of view of communications engineering," *Bell Labs Technical Journal*, vol. 37, no. 1, pp. 185–282, 1958.

[3] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[4] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[5] J. Davidson, J. Peters, and B. Gracely, *Voice over IP fundamentals.* Cisco press, 2000.

[6] J. Dmochowski, J. Benesty, and S. Affes, "On spatial aliasing in microphone arrays," *IEEE Transactions on Signal Processing*, vol. 57, no. 4, pp. 1383–1395, April 2009. [Online]. Available: http://dx.doi.org/10.1109/TSP.2008.2010596, Accessed: 2017-11.

[7] J. Fourier, *Théorie analytique de la chaleur.* Chez Firmin Didot, père et fils, 1822.

[8] J. S. Garofolo, *et al.*, "TIMIT acoustic-phonetic continuous speech corpus," Philadelphia, 1993. [Online]. Available: https://catalog.ldc.upenn.edu/ldc93s1, LDC93S1.

[9] R. Gonzalez and R. Woods, *Digital Image Processing.* Pearson/Prentice Hall, 2008.

[10] W. Herbordt and W. Kellermann, "Computationally efficient frequency-domain robust generalized sidelobe canceller," in *Proc. Int. Workshop on Acoustic Echo and Noise Control*, 2001, pp. 51–55.

[11] H.-G. Hirsch and C. Ehrlicher, "Noise estimation techniques for robust speech recognition," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95.,*

1995 International Conference on, vol. 1, 1995, pp. 153–156 vol.1. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.1995.479387, Accessed: 2017-11.

[12] A. Hoshuyama, O. & Sugiyama, “A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters,” *IEEE Transactions on signal processing*, vol. 47, no. 10, 1999.

[13] ITU-T, “Recommendation G.729 Annex B: A silence compression scheme for G.729 optimized for terminals conforming to recommendation V.70,” Iternational Telecommunication Union, Tech. Rep., 1996. [Online]. Available: https://www.itu.int/rec/T-REC-G.729-199610-S!AnnB/en, Accessed: 2017-11.

[14] *Recommendation P.862: Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-band Telephone Networks and Speech Codecs*, ITU-T, Geneva,Switzerland, 2001.

[15] G. James, *Advanced modern engineering mathematics.* Pearson Education, 2004.

[16] D. H. Johnson, “Signal-to-noise ratio,” *Scholarpedia*, vol. 1, no. 12, p. 2088, 2006, revision #91770.

[17] G. Knapp, C.H. & Carter, “The generalized correlation method for estimation of time delay,” *IEEE Transactions on acoustics, speech and signal processing*, vol. 24, no. 4, pp. 320 – 327, 1976.

[18] B. C. J. Moore and B. R. Glasberg, “Suggested formulae for calculating auditoryfilter bandwidths and excitation patterns,” *The Journal of the Acoustical Society of America*, vol. 74, no. 3, pp. 750–753, 1983. [Online]. Available: http://dx.doi.org/10.1121/1.389861, Accessed: 2017-11.

[19] M. Parviainen and P. Pertilä, “Self-localization of dynamic user-worn microphones from observed speech,” *Applied Acoustics*, vol. 117, pp. 76–85, 2017.

[20] P. Pertilä, M. Mieskolainen, and M. Hämäläinen, “Passive Self-Localization of microphones using ambient sounds,” in *20th European Signal Processing Conference 2012 (EUSIPCO 2012)*, Bucharest, Romania, aug 2012.

[21] P. Pertilä and A. Tinakari, “Time-of-arrival estimation for blind beamforming,” in *2013 18th International Conference on Digital Signal Processing (DSP)*, July 2013, pp. 1–6.

[22] Y. Rui and D. Florencio, "Time delay estimation in the presence of correlated noise and reverberation," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, May 2004, pp. ii–133–6 vol.2.

[23] J. Russ, *The Image Processing Handbook, Fifth Edition.* CRC Press, 2006.

[24] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[25] J. O. Smith, *Mathematics of the discrete fourier transform (DFT): with audio applications.* Stanford University, CA, USA.: W3K: Charleston, SC, USA, 2007. [Online]. Available: https://www.dsprelated.com/freebooks/mdft/, Accessed: 2017-11.

[26] ——, *Spectral audio signal processing.* Stanford University, CA, USA.: W3K: Charleston, SC, USA, 2007. [Online]. Available: https://ccrma.stanford.edu/~jos/sasp/sasp.html, Accessed: 2017-11.

[27] S. W. Smith *et al.*, *The scientist and engineer's guide to digital signal processing.* California Technical Pub. San Diego, 1997. [Online]. Available: http://www.dspguide.com/, Accessed: 2017-11.

[28] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[29] P. Tagare, "Biomedical digital signal processing," W. J. Tompkins, Ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993, ch. Signal Averaging, pp. 184–192. [Online]. Available: http://dl.acm.org/citation.cfm?id=166367.166381, Accessed: 2017-11.

[30] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 1462 –1469, july 2006.

[31] A. Wabnitz, *et al.*, "Room acoustics simulation for multichannel microphone arrays," in *Proceedings of the International Symposium on Room Acoustics*, 2010, pp. 1–6.

[32] H. Wang and P. Chu, "Voice source localization for automatic camera pointing system in videoconferencing," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Apr 1997, pp. 187–190 vol.1.

[33] C. Zhang, D. Florencio, and Z. Zhang, "Why does phat work well in low noise, reverberative environments?" in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 03 2008, pp. 2565–2568.

[34] E. Zwicker, "Subdivision of the audible frequency range into critical bands," *The Journal of the Acoustical Society of America*, vol. 33, no. 2, pp. 248–248, 1961.