Aalborg University,
School of Information and
Communication Technology

Speaker Localization and Extraction with Distributed Microphone Arrays

Martin Weiss Hansen

MSc Medialogy, 10th Semester

Supervisors: Mads Græsbøll Christensen
Jesper Rindom Jensen

Date: 30 May 2013

AALBORG UNIVERSITET

Department of Architecture, Design & Media Technology
Medialogy, 10th Semester

Title:

Speaker Localization and Extraction with Distributed Microphone Arrays

Semester Theme:

Master's Thesis

Project Period:
MED10

Project Group:
131033

Member:

Martin Weiss Hansen

Supervisors:
Mads Græsbøll Christensen
Jesper Rindom Jensen

Circulation: 3

Number of pages: 58

Number of appendices and form:
1 (DVD)

Delivered: 30th of May 2013

Abstract:

In this report a method for localization and extraction of speech signals with distributed microphone arrays is described. Microphone arrays are used to estimate the DOA of a sound source. Multiple DOA estimates are combined to form an estimate of the location. A least-squares approach is used to find the point from which the distance to the lines formed by the DOAs and the microphone array postions.

Three experiments were conducted in order to evaluate the performance of the system. The results of the experiments indicate that the system is able to localize a static sound source with a test signal used as input. Future work should include an optimization of the implementation, and the development of a method for weighting the DOA estimates when an estimate of the location is formed.

Preface

This report, with accompanying documentation and implementation, is submitted as part requirement for the degree of MSc Medialogy at Aalborg University, Denmark. It is the product of my own labour except where indicated in the text.

A DVD containing the following material is enclosed:

- MATLAB files
- PDF versions of internet resources
- An AV-production created as part of the requirements for the semester
- A digital version of this report

I would like to thank my supervisors Mads Græsbøll Christensen and Jesper Rindom Jensen for their invaluable guidance, support and feedback during this project, and for introducing me to the exciting world of microphone arrays and beamforming.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Microphones are used to capture the world surrounding us. In most cases, besides recording a sound source, a microphone will also pick up ambient noise present in the enviroment. To improve intelligibility or quality of the signal, noise reduction is often necessary. When a single microphone is used, a time- or frequency-domain filter can be used on the time-series data to remove noise from the signal. If the recording system contains multiple microphones, it is possible to treat these microphones as an array of microphones, and perform beamforming, which is also known as spatial filtering (Brandstein and Ward, 2001, p. 3).

When using beamforming it is possible to extract a signal from a certain direction using a microphone array. Examples of the application of beamforming are hearing aids, noise and echo reduction, enhancement of the spoken input for interactive systems, speech recognition and separation of acoustic signals (Brandstein and Ward, 2001, Part III). The Oticon Epoq hearing aid (Oticon A/S, 2007) and the Microsoft Kinect (Microsoft Corporation, 2013) are examples of the use of microphone arrays in commercial products. Beamforming can also be used to find the direction of arrival (DOA) of a sound source, if this is not a priori knowledge. This could be

useful in a situation where we wish to find the position of the sound source.

If several microphone arrays are present in an environment, the DOAs estimated by the arrays can be combined to result in an estimation of the position of the speaker. A system consisting of multiple microphone arrays could be used in intelligent homes, enabling users to talk to their homes and appliances. Furthermore, if the estimation of the position of the speaker is used to adjust the direction of arrival of the signal from the speaker, the system should be capable of further noise reduction, compared to a traditional system with a single microphone array. The estimate of the speaker position over time is a useful feature in itself.

In this project, focus will be on developing a method for sound source localization and extraction, using K microphone arrays, in a living room. Each microphone array consists of M microphones uniformly spaced on a line. Such arrays are called uniform linear arrays (Stoica and Moses, 2005, p. 283). See Figure 1.1. The estimation of the source location will investigated as a way of improving the performance of the system.

1.1 Problem Statement

Based on the discussion presented in this chapter, the following problem statement is phrased:

How is it possible to determine the location of a sound source using distributed microphone arrays, and use this information to improve the SNR of the output signal when extracting a signal using beamforming?

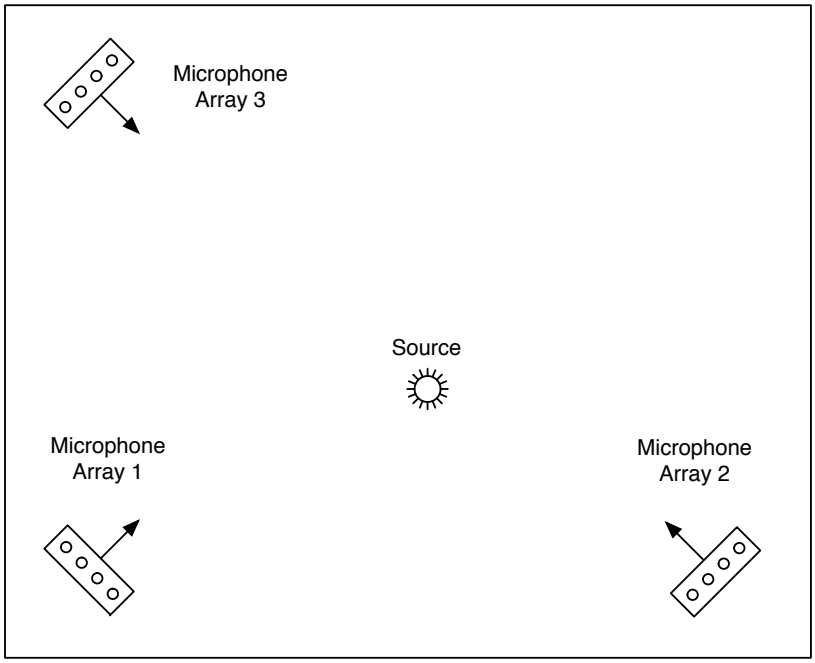


Figure 1.1: A room with three microphone arrays

Chapter 2

Background

In this chapter, the concept of beamforming will be presented. The relationship between temporal and spatio-temporal filtering will be determined. Furthermore, the problem of determining the localization of a sound source based on a number of estimated DOAs will be defined.

2.1 Beamforming

Beamforming is a technique used for extraction of a signal in a noisy environment. It is a technique which has applications within many fields. In acoustic beamforming, microphone arrays are used to extract one or more sound sources, which are contaminated with noise, from a certain direction (Brandstein and Ward, 2001, p. 88). Wave fields are sampled in both space and time by the microphone array, which is why beamforming can be called spatio-temporal filtering, as opposed to conventional temporal filtering, where only time samples are considered (Stoica and Moses, 2005, p. 275).

Two assumptions are usually made in beamforming applications (Brandstein and Ward, 2001, p. 3):

1. The narrowband assumption: The signals incident on the array are narrowband
2. The farfield assumption: The signal sources are located far enough away from the array that the wavefronts impinging on the the array can be modeled as planar waves

Array Model

If the location of a sound source is unkown, it is possible to find the direction at which the sound source is located relative to the microphone array, by performing spatial spectral estimation (Stoica and Moses, 2005, p. 275). With a model of the output signal of the receiving microphone array, the problem of estimating the DOA of the sound source is similar to the problem of temporal frequency estimation.

If $x(t)$ is value of the signal waveform at time t, τ_k is the time it takes for the wave to travel to the kth sensor, the output of sensor k can be written as

$$\bar{y}_k(t) = \bar{h}_k(t) * x(t - \tau_k) + \bar{e}_k(t), \tag{2.1}$$

where $\bar{h}_k(t)$ is the impulse response of the kth sensor, * is the convolution operation, and $\bar{e}_k(t)$ is noise (Stoica and Moses, 2005, p. 277).

If the signals received by the sensors in the microphone array are assumed to be narrowband, the array can be modeled by the frequency-domain model equation

$$Y_k(\omega) = H_k(\omega_c)S(\omega)e^{-i\omega_c\tau_k} + E_k(\omega + \omega_c). \tag{2.2}$$

The time-domain version of 2.2 is

$$y_k = H_k(\omega_c)e^{-i\omega_c\tau_k}s(t) + e_k(t), \tag{2.3}$$

where $y_k(t)$ and $e_k(t)$ are the inverse Fourier transforms of the corresponding terms in 2.2. Sampling of the signals received by the sensors in the microphone array is done using a discrete version of t in equation 2.3 (Stoica and Moses, 2005, p. 282).

Equation 2.3 can be simplified by using the so-called direction vector

$$a(\theta) = [H_1(\omega_c)e^{-i\omega_c\tau_1} \dots H_m(\omega_c)e^{-i\omega_c\tau_m}]^T, \tag{2.4}$$

where θ is the DOA of the source (Stoica and Moses, 2005, p. 282). This is the parameter we wish to determine. The simplified version of equation 2.3 become

$$y(t) = a(\theta)s(t) + e(t). \tag{2.5}$$

If the sensors are considered to be omnidirectinal over the range in which we are interested, or even identical, the direction vector can be simplified to

$$a(\theta) = [1 \quad e^{-i\omega_c\tau_2} \dots e^{-i\omega_c\tau_m}]^T. \tag{2.6}$$

Uniform Linear Array

If the sensors in the microphone array are assumed to be identical and uniformly spaced on a line, the direction vector can be further simplified under the farfield assumption by introducing the spatial frequency ω_s:

$$a(\theta) = [1 \quad e^{-i\omega_s} \dots e^{-i(m-1)\omega_s}]^T, \tag{2.7}$$

with

$$\omega_s = 2\pi f_s = \omega_c \frac{d\ sin\theta}{c}, \tag{2.8}$$

where ω_c is the frequency of the source signal. See Figure 2.1.

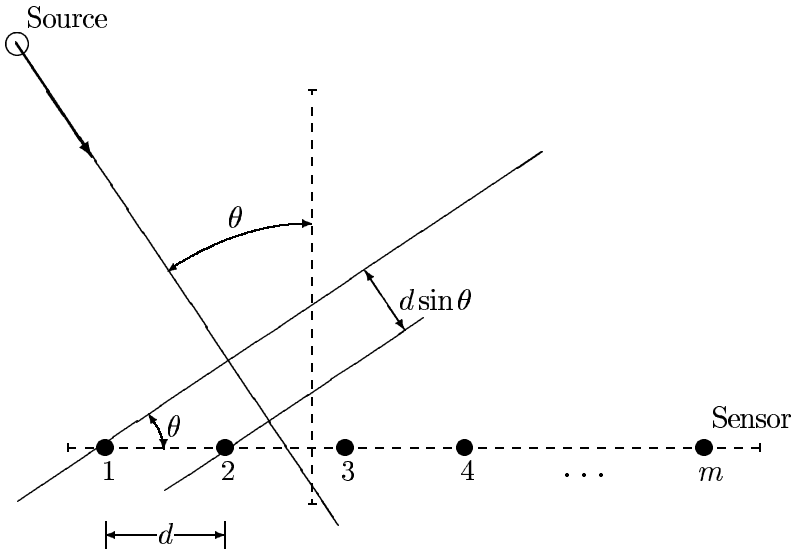


Figure 2.1: A uniform linear array (Stoica and Moses, 2005, p. 284).

Delay and Sum Beamforming

The delay and sum beamforming, also known as classical beamforming, is similar to filtering a temporally sampled signal. In Finite Impulse Response (FIR) filter design we wish to design a filter which approaximates a desired frequency response, which is unity at a frequency of interest, and zero elsewhere. In spatial filtering we are interested in receiving a signal from an angle θ_{DOA} (Van Veen and Buckley, 1988). Figure 2.2 shows the similarity between filtering a temporally sampled signal and filtering a spatially sampled signal. The output of an FIR filter is defined as

$$y(t) = \sum_{k=0}^{m-1} h_k u(t-k) = h^H x(t), \tag{2.9}$$

where $h = [h_0 \dots h_{m-1}]^H$ are filter weights, and $x(t) = [u(t) \dots u(t-m+1)]^T$ is the filter input. The output of a temporal filter for an input $u(t)$ is

$$y(t) = [h^H a(\omega)]u(t), \tag{2.10}$$

where $a(\omega)$ is a vector of complex sinusoids as defined in Figure 2.2. By selecting h such that $h^H a(\omega)$ is large, the power of $y(t)$ at frequency ω can be enhanced (Stoica and Moses, 2005, p. 287).

In the same way, spatial samples can be used to define a spatial filter

$$y(t) = h^H x(t), \tag{2.11}$$

and the spatially filtered output of an array for an input signal $s(t)$ and DOA θ_{DOA} is given by

$$y(t) = [h^H a(\theta)]s(t). \tag{2.12}$$

The spatial filter can h can be selected to enhance signals coming from a given direction θ_{DOA}, by making $h^H a(\theta)$ large.

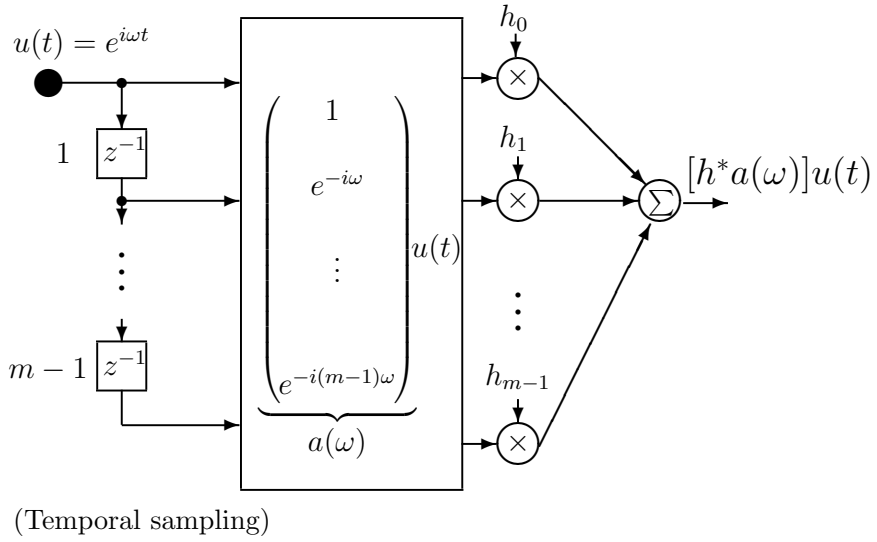
SRP Beamforming

Steered Response Power (SRP) beamforming, also known as filter and sum beamforming, is a generalization of the delay and sum beamforming. The output of the filter and sum beamformer for an N-element array can be defined as

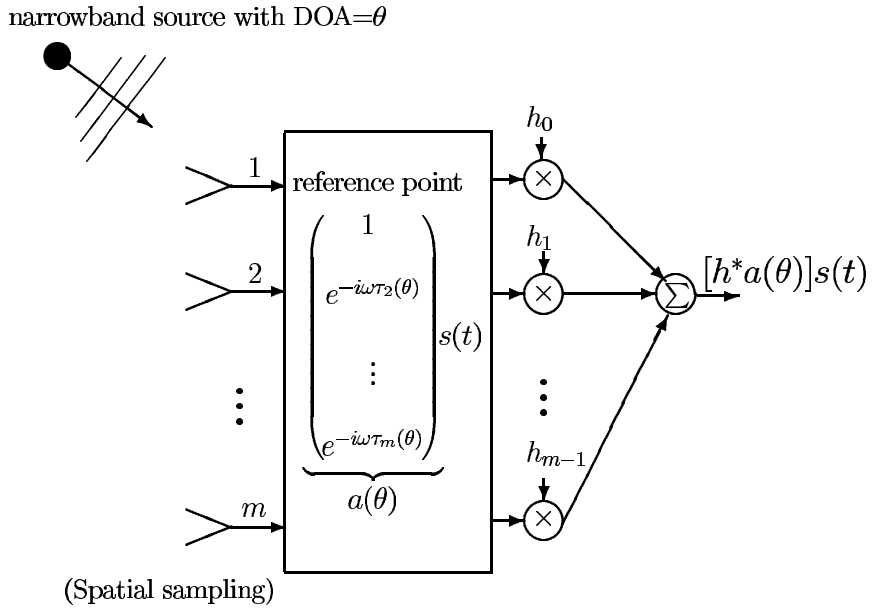
$$Y(\omega, \theta) = \sum_{n=1}^{N} G_n(\omega) X_n(\omega) e^{j\omega d_n}, \tag{2.13}$$

where d is the sensor spacing, which depends on θ using the ULA definition previously mentioned in this section. $X_n(\omega)$ is the frequency-domain input and $G_n(\omega)$ is the frequency-domain filter.

In the SRP-PHAT method, the phase transform (PHAT) weighting whitens the sensor signals to equally emphasize all frequencies in the signal (DiBiase, 2000). Brandstein and Ward (2001) have found that the PHAT weighting enhances the performance in low to moderate reverberation conditons



(a) Temporal filter



(b) Spatial filter

Figure 2.2: Similarity between a temporal filter and a spatial filter (Stoica and Moses, 2005, p. 286)

(Brandstein and Ward, 2001, p. 170). The PHAT weighting is equivalent to the use of the individual frequency-domain filters

$$G_n(\omega) = \frac{1}{|X_n(\omega)|}, \tag{2.14}$$

where $G_n(\omega)$ is the frequency-domain filter from expression 2.13.

MVDR Beamforming

Minimum variance distortionless response (MVDR) beamforming is also known as Capon beamforming (Capon, 1969, Cox et al., 1987). The particular technique used in this project is a frequency-domain MVDR (FMV) algorithm developed by Lockwood et al. (2004). The original algorithm is for a two-microphone system. In this project, the algorithm has been extended to be employed in a system containing an arbitraty number of microphones. The input signals are transformed into the frequency domain every $L = 16$ samples using a 256-point FFT, using a Hamming window. The $F = 32$ most recent FFTs are stored in a buffer, from which a correlation matrix $\mathbf{R}_k$ is calculated for each frequency bin k using

$$\mathbf{R}_k = \begin{bmatrix} \frac{M}{F}\sum_{i=1}^{F} X^*_{1k,i}X_{1k,i} & \frac{1}{F}\sum_{i=1}^{F} X^*_{1k,i}X_{2k,i} & \cdots & \frac{1}{F}\sum_{i=1}^{F} X^*_{1k,i}X_{4k,i} \\ \frac{1}{F}\sum_{i=1}^{F} X^*_{2k,i}X_{1k,i} & \frac{M}{F}\sum_{i=1}^{F} X^*_{2k,i}X_{2k,i} & \cdots & \frac{1}{F}\sum_{i=1}^{F} X^*_{2k,i}X_{4k,i} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{F}\sum_{i=1}^{F} X^*_{4k,i}X_{1k,i} & \frac{1}{F}\sum_{i=1}^{F} X^*_{4k,i}X_{2k,i} & \cdots & \frac{M}{F}\sum_{i=1}^{F} X^*_{4k,i}X_{4k,i} \end{bmatrix}, \tag{2.15}$$

where $M = 1.03$ is a regularization constant used to avoid matrix singularity. The values of L, F and M are the same as used by Lockwood et al. (2004). The matrices $\mathbf{R}_k$ are updated every $L = 16$ samples. The output

of the beamformer is

$$Y_k = \mathbf{h}_k^H \mathbf{X}_k, \tag{2.16}$$

where $\mathbf{h}_k^H$ is a conjugate transposed vector with frequency-domain filter coefficients. In the MVDR approach we want to pass undistorted the signals with a given DOA θ, and attenuate signals with all other DOAs as much as possible, an optimization goal is stated, seeking to minimize the expectation of the output for each frequency band

$$\min_{\mathbf{h}_k} \mathbf{h}_k^H \mathbf{R}_k \mathbf{h}_k \text{ subject to } \mathbf{h}_k^H \mathbf{a}(\theta) = 1, \tag{2.17}$$

where $\mathbf{a}(\theta)$ is the direction vector. This general approach is known as the Capon method (Stoica and Moses, 2005, p. 291-293). The solution to 2.17 is

$$\mathbf{h}_k = \frac{\mathbf{R}_k^{-1}\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\mathbf{R}_k^{-1}\mathbf{a}(\theta) + \sigma}, \tag{2.18}$$

where σ is a small constant that avoids division by zero. The FMV method of Lockwood et al. assumes that the DOA, and hereby the direction vector, is known beforehand. However, the Capon method can also be used to find the DOA of a source. This is done by obtaining the largest peak of the function

$$\frac{1}{\mathbf{a}^H(\theta)\mathbf{R}_k^{-1}\mathbf{a}(\theta)}. \tag{2.19}$$

In the case of n multiple speakers, the DOA estimation is a matter of finding the n largest peaks of the function 2.19. In this project, however, only one speaker will be present in the room at a time, excluding noise.

2.2 Localization

When several microphone arrays are present in a room, each providing an etimate of the DOA of the target sound source, these angles can be combined

to form an estimate of the position of the sound source. If we assume the microphone arrays are able to estimate the true DOAs, for instance in a room without any noise present, the lines drawn by the DOAs form a single point of intersection, which is the position of the sound source. See Figure 2.3.

In reality, however, this will rarely be the case, because noise in the environment will influence the estimation of the DOAs. In a noisy environment, the lines drawn by the DOAs do not form a single point of intersection. See Figure 2.4, where the gray dots indicate that acoustic noise is present in the room. Because of this, we have to find a point which we believe is an approximate position of the source. A possible solution is to weight the individual DOAs estimates to form an estimate of the location. This weighting of the DOAs should take into account the noise in the DOA estimates. DOA estimates with least noise should be preferred to estimates with more noise. Another option is to find the point from which the distance to the lines drawn from the microphone arrays is minimal.

Finding the point that has the minimal distance to the lines drawn by the microphone arrays can be solving using a least-squares approach (van der Heijden et al., 2004, p. 68-69). The perpendicular distance from a point $P = (x_1, y_1)$ to a line l with the equation $ax + by + c = 0$ is given by the following expression (Deza and Deza, 2013, p. 86)

$$dist(P, l)\frac{|ax_1 + by_1 + c|}{\sqrt{a^2 + b^2}}. \tag{2.20}$$

The objective of finding the point with minimal distance to the K lines drawn by the arrays is stated as minimizing the cost function

$$J = \sum_{k=1}^{K} \left(\frac{|a_k x_1 + b_k y_1 + c|}{\sqrt{a_k^2 + b_k^2}} \right)^2. \tag{2.21}$$

This is done by partial differentiation and equating to zero, which results in the equations

$$\frac{\partial J}{\partial x_1} = \sum_{k=1}^{K} 2a_k^2 x_1 + 2a_k b_k y_1 + 2a_k c_k = 0, \tag{2.22}$$

and

$$\frac{\partial J}{\partial b_1} = \sum_{k=1}^{K} 2b_k^2 y_1 + 2a_k^2 x_1 + 2a_k c_k = 0. \tag{2.23}$$

From 2.22 and 2.23 the following matrix equation is formed

$$\begin{bmatrix} \sum_{k=1}^{K} a_k^2 & \sum_{k=1}^{K} a_k b_k \\ \sum_{k=1}^{K} a_k b_k & \sum_{k=1}^{K} b_k^2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} -\sum_{k=1}^{K} a_k c_k \\ -\sum_{k=1}^{K} b_k c_k \end{bmatrix}, \tag{2.24}$$

to which the solution is

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{K} a_k^2 & \sum_{k=1}^{K} a_k b_k \\ \sum_{k=1}^{K} a_k b_k & \sum_{k=1}^{K} b_k^2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} -\sum_{k=1}^{K} a_k c_k \\ -\sum_{k=1}^{K} b_k c_k \end{bmatrix}. \tag{2.25}$$

By solving 2.25 the point with the minimal perpendicular distance to the lines formed by the microphone arrays is found.

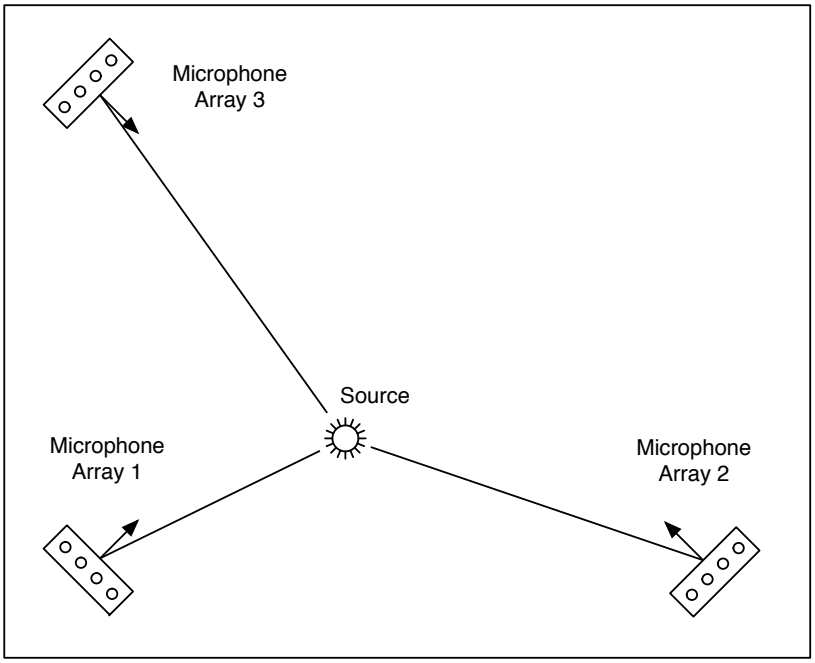


Figure 2.3: Microphone arrays detecting the true DOAs of the source

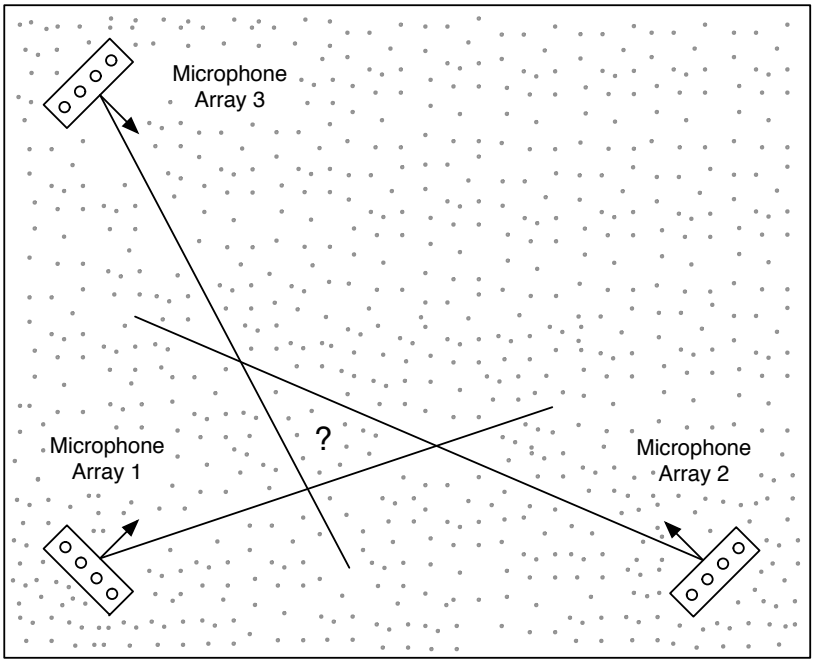


Figure 2.4: Microphone arrays detecting approximate DOAs of the source

Chapter 3

Proposed Method

In this chapter, the proposed method of using multiple microhpone arrays to enhance the extraction of a sound source using an estimate of the location, will be described.

The idea is to improve beamforming performance, compared to a system with a single microphone array, by using several microphone arrays and an estimation of the localization of the sound source based on the DOAs estimated by the microphone array beamformers. The location estimate is used to make adjustments to the DOAs detected by the microphone arrays. For each frame, the SNRs of the outputs of the microphone array beamformers are compared. The output of the array with the best SNR is saved in an output buffer. This way, the system uses the output frames that have the best SNRs. This should improve performance compared to a system that uses the output from a single microphone array.

The following steps are performed in each frame of the signal:

1. Estimate DOAs
2. Perform beamforming in the directions of the estimated DOAs

3. Estimate location of sound source using the DOAs estimated in step 1
4. Estimate SNR for each array, choose the array with the best SNR
5. Determine an adjusted angle from the chosen array to the source
6. Perform beamforming in the direction of the adjusted angle

These steps are reiterated on Figure 3.1.

Localization of the sound source is done using the FMV beamforming approach described in Chapter 2 using expression 2.19. For each microphone array, the frequency-domain output of each microphone is stored in a buffer containing the F latest FFTs. From this data a correlation matrix $\mathbf{R}_k$ is formed using expression 2.15. The FFT buffer and the correlation matrix are updated every L samples. This approach is similar to that of Lockwood et al. (2004).

A direction vector is formed for a number of DOA candidates $\theta \in [-90°, 90°]$ at which we calculate the output power of the beamformer for each frequency bin k. We do this to determine at which angle θ_{DOA} the magnitude of the beamformed signal is at its maximum. This angle θ_{DOA} is the DOA of the sound source relative to the microphone array in question. After the DOA of the sound source has been determined for the current frame, the spatial filter weights are found using expression 2.18. The weights are saved in a matrix in order to determine the SNR of the outputs after beamforming has been done.

To obtain the time-domain output of the beamformer, the spatial frequency-domain weights are applied to the buffered FFT data to obtain the Fourier

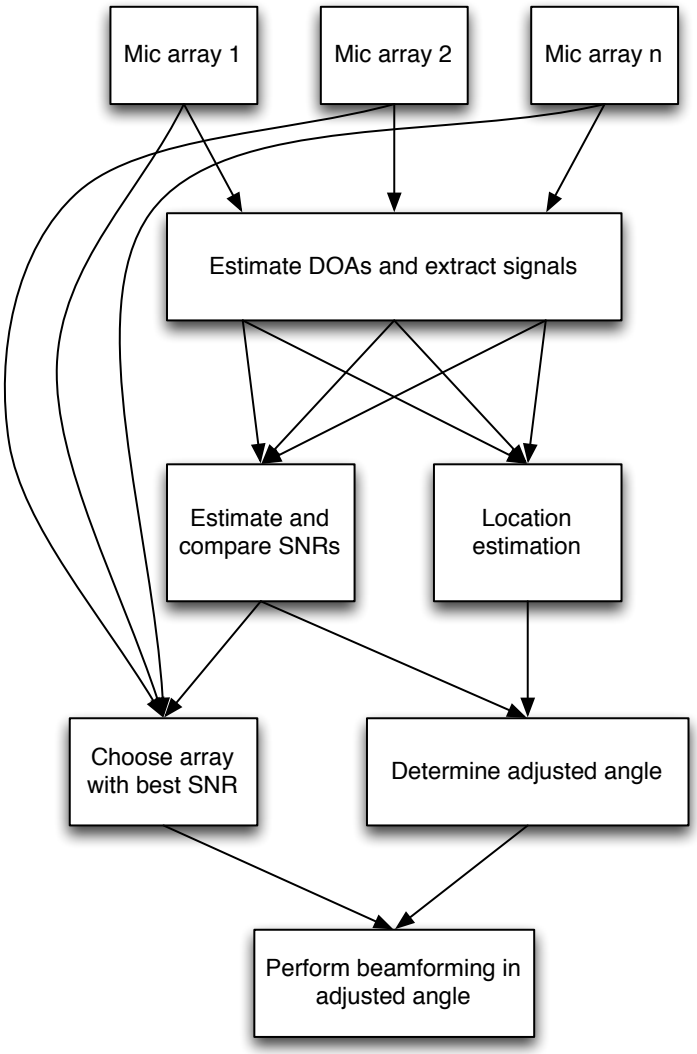


Figure 3.1: System Diagram

transform of the output. The frequency-domain output is then transformed to the time-domain using an N-point inverse FFT every L samples. This approach is similar to that of Lockwood et al. (2004). Lockwood et al. (2004) use the L central samples for the output. In this project, choosing the central L samples and using them directly for the output resulted in artifacts, which is why the overlap-add method has been used instead.

The position of the speaker is estimated using the DOAs estimated by the microphone array beamformers. The DOAs are combined to form an estimate of the location. This is done using a least-squares approach, where the goal is to find the point from which the distance to the lines drawn by the DOAs and the microphone array positions is minimal. The technique is described in the previous section.

The beamformed output of each of the microphone arrays is used to decide which microphone array to use for the current output frame. This is done by estimating the SNR of each of the outputs from the microphone arrays. The SNRs are estimated using the Multi-Pitch Estimation Toolbox for MATLAB, developed by Christensen and Jakobsson (2009). The `joint_anls()` function has been used for joint pitch and order detection. The `amp_als()` method is used for amplitude estimation. The SNR of each of the outputs from the beamformers is calculated using the expression

$$SNR_{dB,est} = 20 \cdot \log_{10} \frac{\sigma^2_{x_{est}}}{\sigma^2_{noise_{est}}}, \tag{3.1}$$

where x_{est} is the estimated input signal formed by the estimated fundamental frequency, the estimated model order and a number of estimated amplitudes corresponding to the model order.

The estimate of the location is used to adjust the DOA at which the beamformer is pointed at. Finally, beamforming is performed in the direction of the new angle using the microphone array which resulted in the best SNR estimate. The output of the system consists of the output frames with the best SNRs from the microphone arrays. In the following chapter, results from experiments carried out in this project will be described.

Chapter 4

Experimental Method

In this chapter the experimental setup and results will be described along with parts of the implementation of the system. The outputs of each of the microphone arrays are generated before processing the signals, resulting in an off-line system.

4.1 Room Setup

The experimental environment has been set up using MATLAB. A room impulse response (RIR) generator by Habets (2010) has been used to set up a virtual room in which experiments have taken place. The RIR generator is based on the image method by Allen and Berkley (1979). Using the RIR generator, the microphone arrays and the speech signal are placed in the room. A signal generator by Habets (2011) has been used to simulate a moving speaker. The test setup consists of a room of size 6 x 6 x 4 m. Three microphone arrays are placed in the corners of the room, 1 m from the walls. The arrays point toward the middle of the room in a 45° angle. See Figure 4.1. The properties of the RIR generator is shown in Table 4.1. From these room properties, a number of RIRs, corresponding to the number of microphones, are generated. The outputs of the microphones are

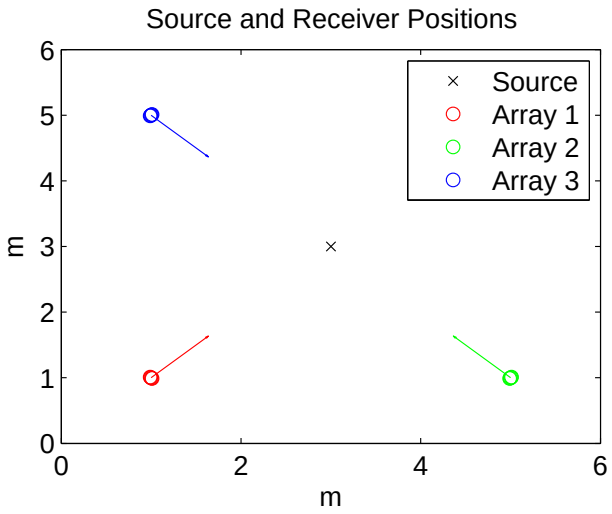


Figure 4.1: Room dimensions and position of source and microphone arrays

Table 4.1: RIR Generator Input Parameters (Habets, 2010)

Input Parameter	Description
c	Sound velocity in m/s
fs	Sampling rate in Hz
r	Matrix with (x,y,z) coordinates of receivers in m
s	Vector with (x,y,x) coordinates of source in m
L	Vector with the room dimensions in m
beta	Reverberation time (RT_{60}) in seconds
nsample	Number of samples in the RIR
mtype	Microphone directivity pattern
order	Maximum reflection order
dim	Room dimension
orientation	Microphone orientation
hp_filter	High-pass filter option

convolved with the RIRs. The resulting convolved outputs of the microphones are saved in a matrix. A sound file containing noise is then loaded. From this signal, a number of channels of diffuse noise, corresponding to the number of microphones, is generated. This noise matrix is added to the matrix with the outputs of the individual microphones in the microphone

arrays. The resulting matrix is used as input data for the beamformers. Next, DOA estimation is carried out for each of the microphone arrays. The frequency-domain filter weights are saved in a matrix for later use. The weights will be applied to the original input signal and the noise. By doing this it is possible to compare the SNR of the signal after beamforming to the SNR of the signal before beamforming. This SNR will be used as a metric by which the proposed method is evaluated.

In order to evaluate the performance of the system, the SNR of the signal is calculated at different stages of the system. The SNR is calculated for overlapping frames with a length of 1024 samples, with a hop size of 16 samples using the expression

$$\text{SNR}_{dB} = 20 \cdot \log_{10} \frac{\sigma^2_{input}}{\sigma^2_{noise}} \text{ dB}. \tag{4.1}$$

The SNR is calculated before and after beamforming is performed, and again after beamforming has been done using the angles corrected using location information.

The input signals consist of a test signal with fundamental frequency $f_0 = 200Hz$ with two harmonics at $f_1 = 400Hz$ and $f_2 = 600$ Hz. Besides this test signal, the NOIZEUS speech corpus has been used as input to the system. The speech corpus consists of 30 sentences, produced by both male and female speakers (Hu and Loizou, 2007). Furthermore, a noise database is used to add noise to the signals. The noise signals are listed in Table 4.2 (Ellis, 2002).

4.2 Implementation

In this section, parts of the MATLAB implementation are presented. The MATLAB source code can be found in its entirety on the attached DVD

Table 4.2: Noise Signals (Ellis, 2002)

Name	Description
Babble	Mixture of voices
Airport	Ambient noise from an airport lobby
Restaurant	Ambient noise from a restaurant
Exhibition	Ambient noise from an exhibition hall
Street	Ambient noise from a city street
Car	Noise inside a moving car
Subway	Noise inside a subway train
Train	Noise inside a train carriage

along with the source code to compile the toolboxes by Habets. As already mentioned, the RIR generator by Habets (2010) is used to place a static sound source in a virtual room. This is done by executing the following line of code:

```
h = rir_generator(c, fs, r2, s, L, beta, n, mtype, order, ...
        dim, array_or, hp_filter); % Generate RIRs
```

The values of the input parameters mentioned in Table 4.1 are shown in Table 4.3. The speaker position is in this chapter referred to as the source position, and the positions of the microphones in the microphone arrays are referred to as receivers. For moving sound sources, the following line of code is used:

```
[out, beta_hat] = signal_generator(in, c, fs, rp_path, ...
        sp_path, L, beta, nsample, 'o', order)
```

The function parameters that are similar to the ones for the `rir_generator`. The main difference is that `rp_path` holds the time-variable positions of the receivers, and `sp_path` holds the time-variable positions of the sources. The microphone positions are determined by a function `r = micarray(n_mics, orientation, pos, d)`, which places `n_mics` microphones with spacing `d`

Table 4.3: RIR Generator Parameter Settings (Habets, 2010)

Input Parameter	Setting
c	340 m/s
fs	8000 Hz
r	(1,1,2) m, (5,1,2) m and (1,5,2) m
s	Varies
L	(6,6,4) m
beta	0.4 s
nsample	4096 samples
mtype	'omnidirectional'
order	Varies
dim	3
orientation	45°, 135° and 315°
hp_filter	1

with a midpoint for the array at `pos`. The orientation of the microphones is determined by the variable `orientation` which contains a value for the azimuth of the microphones. After the RIRs have been generated, the source signal is convolved with the RIRs to generate the receiver outputs. Diffuse noise is then added to each of the receiver outputs. The noise signal is scaled to result in an average SNR of the noise corrupted signal. The following expression is used to compute the factor c_{noise} by which the noise should be scaled to achieve an average SNR of SNR_{dB}

$$c_{noise} = \frac{\sigma_s^2 \frac{1}{10^{\frac{SNR_{dB}}{20}}}}{\sigma_{noise}^2}. \tag{4.2}$$

After the noise has been scaled and added to the outputs of the receivers, the SNR is calculated for frames of the noisy signal, using the clean receiver signals and the noise signals, to determine the SNR of the signals before beamforming is performed.

```
1 H = 16;
2 n_win = 1024;
```

```
segment = 1:n_win;
frames = (length(x)/H)-(n_win/H)+1
for i = 1:frames
    for k = 1:12
        SNR_mic(i,k) = 20*log10(var(x_clean(k,segment))/...
            var(noise_attenuated(k,segment)));
    end
    segment = segment + H;
end
```

The average SNR for the first microphone array is the average of the SNR of the first four signals, and so on for the SNRs of the second and the third arrays. The average SNRs of the microphone array inputs are later compared to the average SNRs of the beamformed microphone array outputs to evaluate the performance of the system.

After the input SNRs have been estimated, the DOA estimation is performed for each of the microphone arrays, and beamforming is performed in the direction of the estimated DOAs. DOA estimation and beamforming is done in the same step. The FMV beamformer by Lockwood et al. (2004) is used for DOA estimation and beamforming, as described in Section 2.1. The signals from the microphone arrays are processed individually for each array. The noise corrupted sensor inputs are split into overlapping frames of 256 samples, with a hop size of $L = 16$ samples. At a sampling frequency of 8000 Hz a frame length of 256 samples corresponds to a duration of 32 ms. This corresponds to the duration of the frames used by Lockwood et al. (2004), which is $\frac{22050 \text{ Hz}}{1024 \text{ samples}} = 46$ ms. The frames are transformed into the frequency-domain, and an FFT buffer containing the latest 32 FFTs is computed

```
for i = 1:number_of_frames
    for mic_num = 1:number_of_mics
        X(mic_num,:) = fft(x(mic_num,segment).*window', n_window);
        X_buffer((mod(i-1,F))+1,:,mic_num) = X(mic_num,:);
    end
        segment = segment + L;
end
```

The number of frames is determined by `number_of_frames = ((length(x)-n_fft)/L)+1`. Next, the correlation matrix 2.15 of the FFT buffer is determined

```
R = zeros(number_of_mics,number_of_mics,n_fft);
M = 1.03;
for k = 1:n_fft % kth frequency bin
    for row = 1:number_of_mics % sensor number
        for column = 1:number_of_mics % sensor number
            if (row == column)
                R(row,column,k) = (M/F)*sum(conj((X_buffer(:,k,row)))...
                    .*(X_buffer(:,k,column)));
            else
                R(row,column,k) = (1/F)*sum(conj((X_buffer(:,k,row)))...
                    .*(X_buffer(:,k,column)));
            end
        end
    end
end
```

The output power of the beamformer is then computed for each frequency bin, for each of the candidate DOAs `index` which are in the range $\theta \in [-90°, 90°]$

```
for bin = 1:n_window/2
    k = bin-1;
    wc = k/n_window*2*pi*fs;
    for mic_num = 1:number_of_mics
        a(mic_num,:) = exp((mic_num-1)*(-1i*(wc*((d*...
        sin(theta))/c)))); % steering vector
    end

    for index = 1:length(a)
        power(index) = power(index) + 1/real(a(:,index)'...
        *inv(R(:,:,bin))*a(:,index));
    end
end
```

The DOA of the frame is found by finding the index at which `power` is greatest. This DOA is used to perform beamforming

```
for bin = 1:n_window/2
    k = bin-1;
    wc = k/n_window*2*pi*fs;
    for mic_num = 1:number_of_mics
        a(mic_num,:) = exp((mic_num-1)*...
        (-1i*(wc*((d*sin(theta))/c)))); % steering vector
    end
    temp = R(:,:,bin)\a(:,index);
    w(:,bin) = temp/((a(:,index)'*temp));
end
```

The frequency-domain filter weights are applied to the FFT of the current frame. The frame is then transformed to the time-domain using an inverse FFT transform. The output signal is formed by overlap-add. Furthermore,

the frequency-domain filter weigths in `w` are saved for each frame in order to estimate the SNR of the signal using the clean microphone array inputs and the noise signals. The filter weights define the beamformer, and can be applied to the frequency-domain version of the microphone array inputs and the noise signals in order to perform beamforming. The clean microphone array signals and noise signals are fed to the beamformer, using the filter weights saved in `w`, in order to evaluate whether the SNR of the output of the beamformer is greater than the SNR of the noise corrupted signal.

After the DOAs have been estimated and beamforming has been performed, the DOAs are smoothed using a 51-tap mean filter. The smoothed DOAs are used to estimate the location of the source, using the method described in Section 2.2. The location is smoothed using a non-linear smoothing technique by Ney (1983).

Next, the SNR of each of the outputs of each of the microphone array beamformers are compared using the functions `joint_anls`, `amp_als`, `vandermonde`, from the Multi-Pitch Estimation Toolbox (Christensen and Jakobsson, 2009). The model order is saved in `L_est`, and the estimated fundamental frequency is saved in `w0_est`. The estimated amplitudes of the sinusoids are saved in `a_est`. These features are used to form a matrix `Z_est` of complex sinusoids:

```
1 [w0_est, L_est]=joint_anls((input),w0_lim,F_search);
2 a_est = amp_als((input),w0_est*[1:L_est],F_search);
3 Z_est=vandermonde(w0_est*[1:L_est],N);
4 x_est=real(Z_est*a_est);
5 error = real(input)-real(x_est);
6 SNR_dB = 20*log10(var(real(x_est))/var(error));
```

The microphone array which resulted in the best output SNR is chosen. Using the position of this microphone array and the estimation of the location, an adjusted DOA is calculated. The chosen array and the DOA is used as input to the beamformer. If the adjusted angle calculated from the location estimate is closer to the correct DOA of the source, the SNR of the output of the beamformer using the adjusted DOA should be greater than the SNR of the output of the beamformer using the originally estimated DOA. Finally, since the output of the system switches between the outputs of the individual beamformers, discrepancies between the distances from the estimated location of the sound source to the microphone arrays might result in time delay differences. To adjust for this, the signal is shifted using indexing for the integer number of samples of delay, and an allpass filter for the fractional part of the delay. The fractional delay line is based on M-file 2.8 by Zölzer (2011) (Zölzer, 2011, M-file 2.8, pp. 75-76)

```
function y=fracdelay(x,DELAY)
x_size = size(x);
if x_size(2)>x_size(1)
    x = x';
end
y_old=0;          % previous output
LEN=length(x);    % length of input
L=2;              % delay line length
Delayline=zeros(L,1); % delay line initialization
y=zeros(size(x));     % output buffer initialization
for n=1:(LEN-1)
    TAP=1+DELAY;
    i=floor(TAP);
    frac=TAP-i;
    Delayline=[x(n);Delayline(1:L-1)];
```

Table 4.4: Experiments Overview

Experiment	Input Signal	Noise Signal
1	Sine wave signal (static)	White noise
2	Sine wave signal (moving)	White noise
3	sp10.wav	White noise

```
16      y(n,1)=Delayline(i+1)+(1-frac)*Delayline(i)-...
17              (1-frac)*y_old;
18      y_old=y(n,1);
19  end
20  end
```

4.3 Results

In this section the results of a series of experiments are presented. An overview of the experiments is presented in Table 4.4.

Experiment 1

The first experiment was carried out to test the system, the DOA and location estimation in particular. For this test, the input was a spatially static input with a fundamental frequency at $f_0 = 200$ Hz, and two harmonics at $f_1 = 400$ Hz and $f_2 = 600$ Hz respectively. The location of the sound source is shown in Figure 4.2. With the source placed in the middle of the room, we expect the DOAs to be zero.

The noise signal consists of white noise with zero mean, which has been scaled to result in an average SNR of 20 dB. The SNR is measured for overlapping frames of length 1024 samples, with a hop size of 16 samples. A number of channels of diffuse noise, corresponding to the number of microphones, is generated from the noise signal. See Figure 4.3.

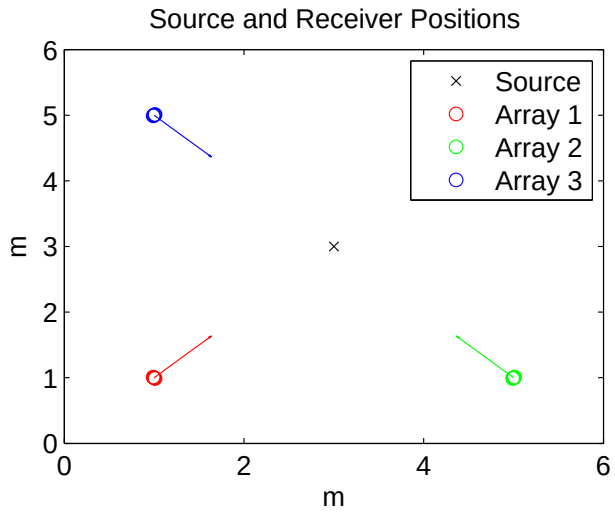


Figure 4.2: Sound source position

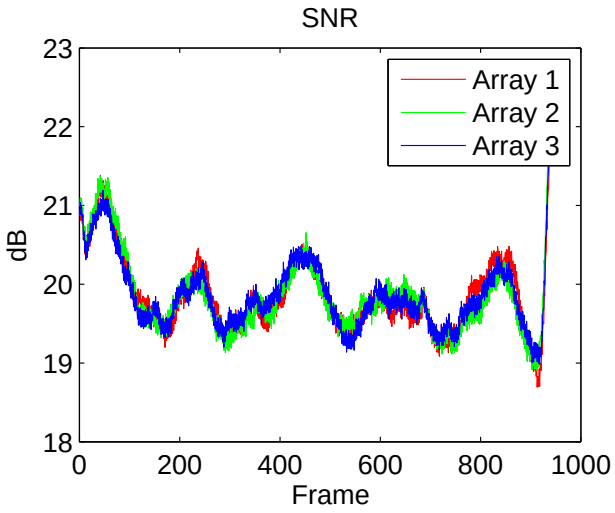


Figure 4.3: SNR before beamforming

Figure 4.4 shows the DOAs estimated by the beamformers of the microphone arrays. The DOAs have been smoothed using a 51-point mean filter. The DOAs fluctuate around zero, as expected. The estimated location of the source is shown in Figures 4.5 and 4.6. The fluctuations appear because of noise in the signal. For this test it did not make sense to form an output

based on the microphone arrays with the best SNRs, because the sound source is in the middle of the room. In this case it does not matter which microphone array is used to form the output of the system. If the position of the sound source is at another position, but still static, the system output is the output of the microphone array that results in the signal with the best SNR.

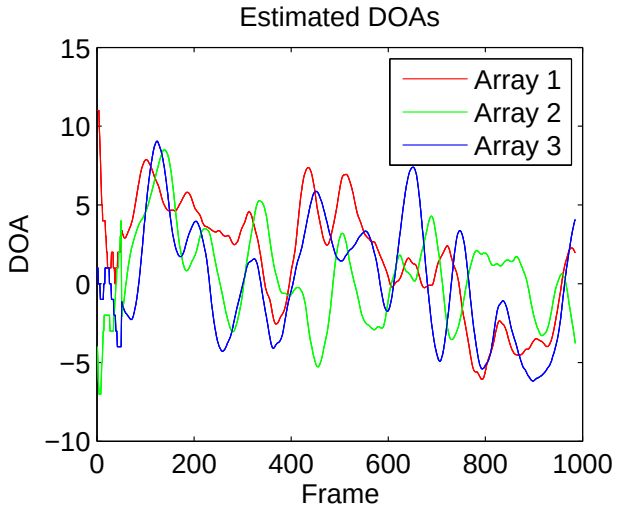


Figure 4.4: Smoothed DOAs

Figure 4.13 shows the SNR of the signal after beamforming. The SNR seems to have improved, as expected.

Experiment 2

In the second experiment the input signal is a moving version of the signal used in the first experiment. The noise signal is the same as in the first experiment. This experiment is conducted to evaluate the estimation of the DOAs and the location over time. The location of the sound source over time is illustrated in Figure 4.8.

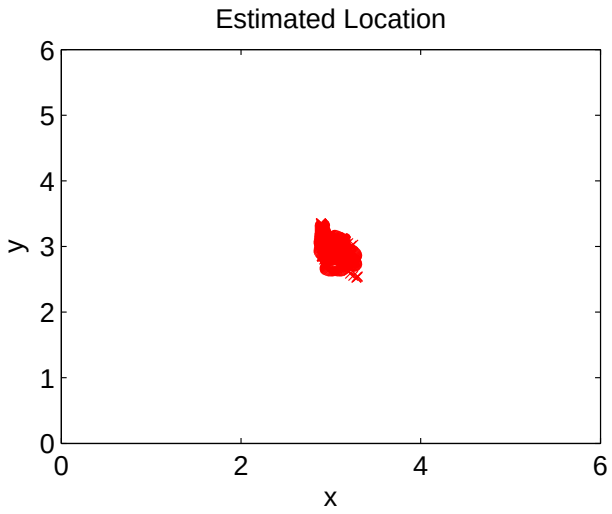


Figure 4.5: Estimated source location

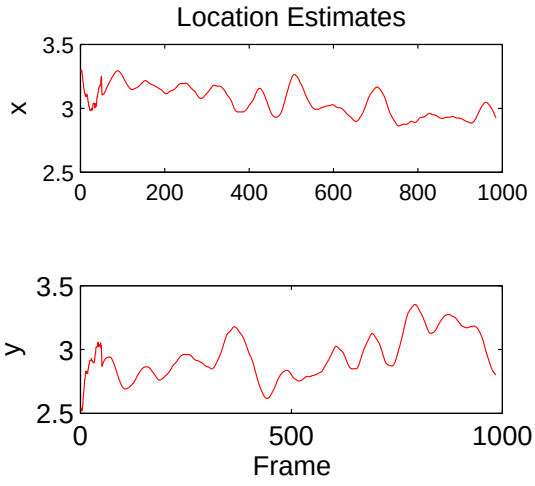


Figure 4.6: Estimated source location

Figure 4.9 shows the amplitudes of the signals recorded by the microphone arrays. We note that the amplitudes of the signals for the first array (red) reach its maximum in the middle of the signal. The amplitudes of the signals for array 2 (green) reach their maximum at the end of the signal. The amplitudes of the signals for array 3 (blue) start out at their maximum

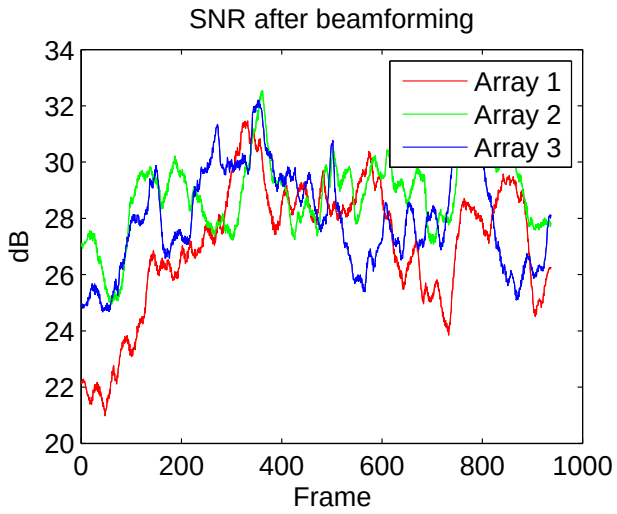


Figure 4.7: SNRs after beamforming

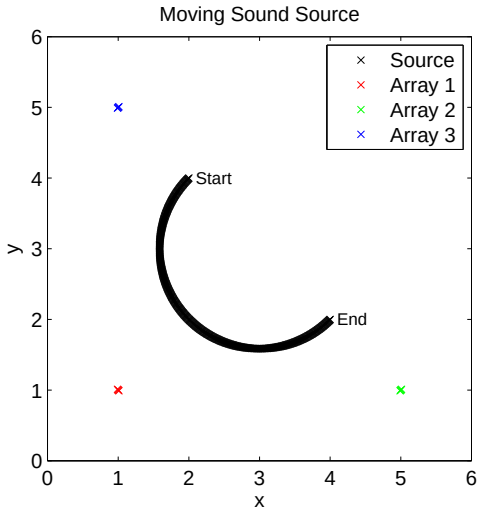


Figure 4.8: Location of moving sound source

amplitude. Since the sound source moves with constant velocity these observations correspond to Figure 4.8.

The SNRs of the signals, with noise added, are also in correspondence with Figures 4.8 and 4.9. See Figure 4.10.

Figure 4.11 shows the DOAs detected by the microphone arrays. It is noted

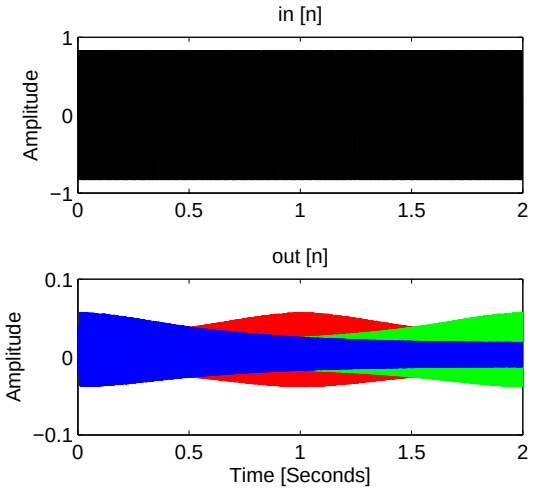


Figure 4.9: Amplitudes of microphone array signals

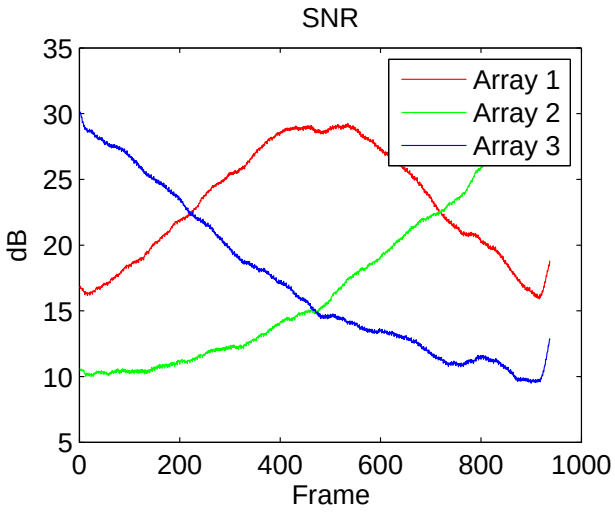


Figure 4.10: SNRs of the noisy microphone array input signals

that the DOA estimates look quite noisy. The noise in the DOA estimates could be because the input signal is corrupted with noise. However, the SNR of the signal is quite good at 30 dB. The smoothed DOA estimates are shown in Figure 4.12.

At the same time as the DOAs are estimated, beamforming is performed.

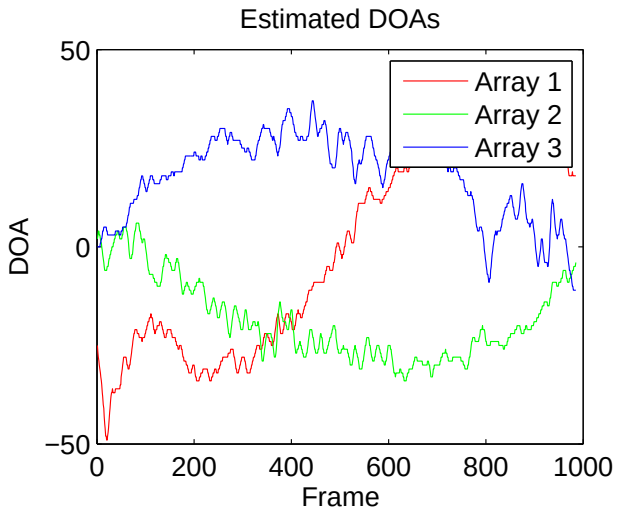


Figure 4.11: Estimated DOAs of the microphone array signals

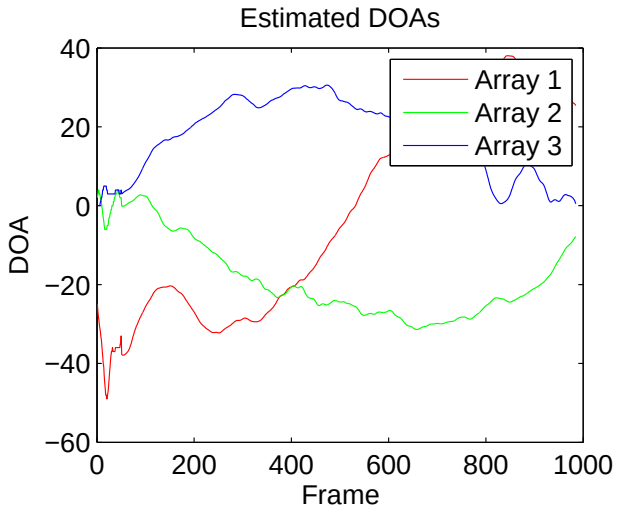


Figure 4.12: Smoothed version of Figure 4.11

The SNRs of the outputs of the beamformers are shown in Figure 4.13. The SNRs of the signals seem to have improved by 15 dB.

The DOAs are combined to form an estimate of the location of the source over time. See Figures 4.14 and 4.15.

The non-linear smoothing technique by Ney (1983) has been used to smooth

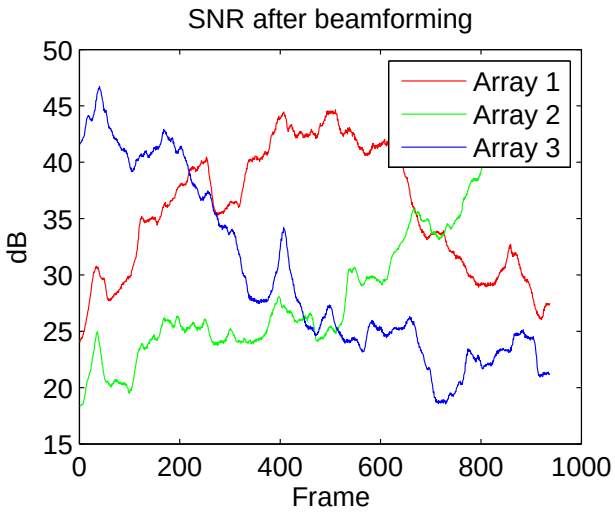


Figure 4.13: SNR after beamforming

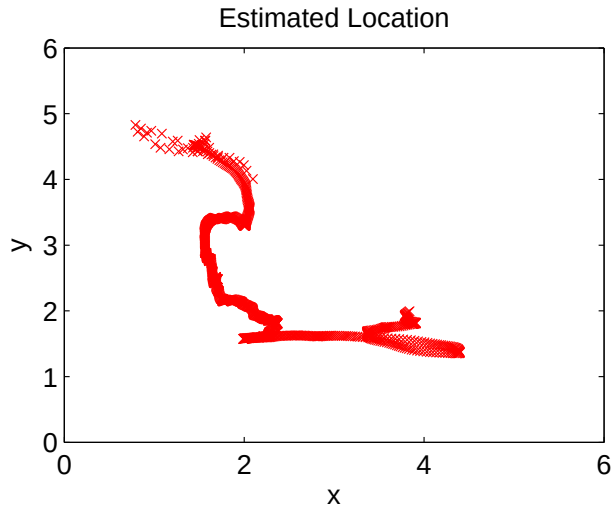


Figure 4.14: Estimated source location

the estimated location. The result is seen in Figures 4.16 and 4.17.

The outputs of the beamformers are compared by estimating the SNRs of the outputs using the Multi-Pitch Estimation Toolbox as described in Section 4.2 (Christensen and Jakobsson, 2009). Ideally the SNRs of the output signals from the beamformers should be estimated at every sample in

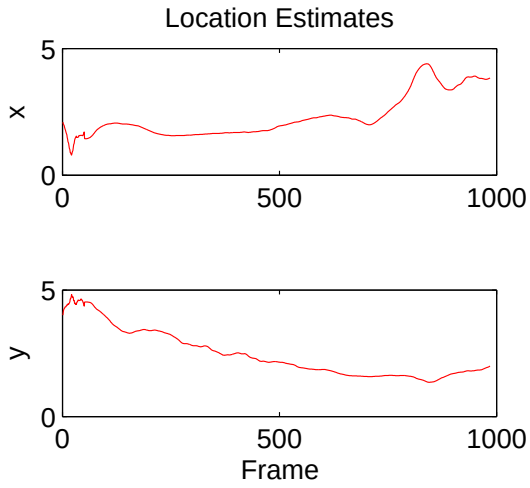


Figure 4.15: Estimated source location

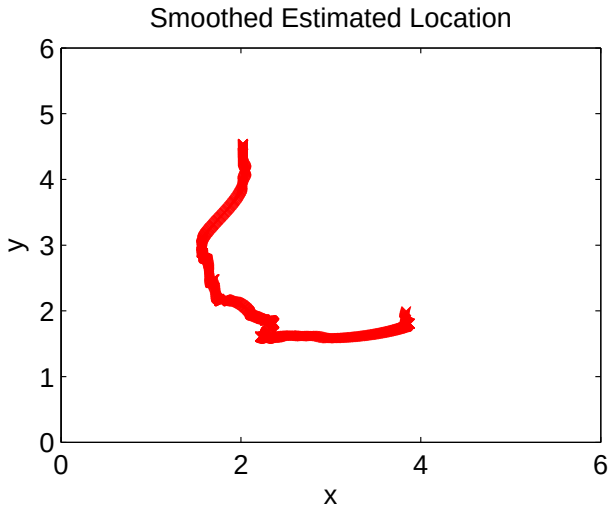


Figure 4.16: Smoothed estimate of source location

order to make meaningful decisions. However, because the order- and pitch-estimation is computationally expensive, SNR estimation and comparison between microphone arrays is only formed for three parts of the signals, a part at the beginning of the signal, a part at the middle of the signal, and a part at the end of the signal. The results are shown in Table 4.5.

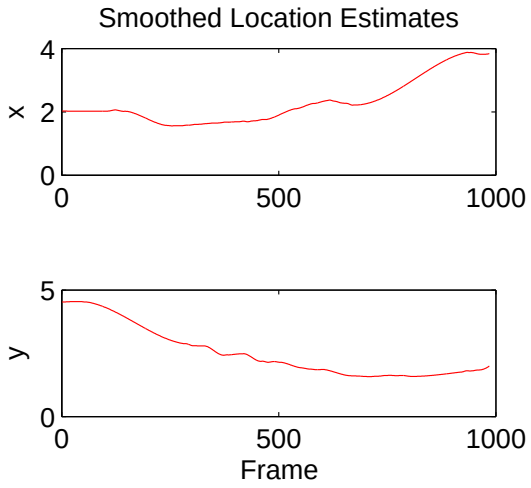


Figure 4.17: Smoothed estimate of source location

Table 4.5: SNR Estimation (Experiment 2)

Array	Sample 1-256	Sample 8001-8256	Sample 15001-15256
1	4.9 dB	44.1 dB	32.4 dB
2	3.6 dB	33.9 dB	44.7 dB
3	6.6 dB	31.5 dB	29.7 dB

The results in Table 4.5 are in agreement with the intuition that the array closest to the source acheives the greatest SNR.

Experiment 3

In the third experiment the input signal stems from the NOIZEUS speech corpus (Hu and Loizou, 2007). The file `sp10.wav` has been used. The noise signal is the same as for the two first experiments. The amplitude of the input used in the third experiment varies more than the amplitude of the signal used in the first two experiments. See Figure 4.18. The amplitude pattern seems similar to the one seen in Figure 4.9. The location of the sound source is similar to the movement seen in Figure 4.8. Because the

amplitude of the speech signal drops significantly between the spoken words, a voice activity detector is used to ensure that silent parts of the signal are excluded (Brookes, 2013).

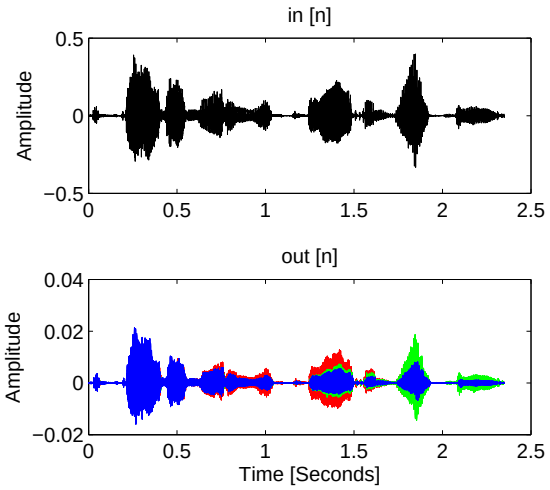


Figure 4.18: Amplitudes of microphone array signals

The SNRs of the noise corrupted signals are in agreement with the amplitude pattern in Figure 4.18. See Figure 4.19.

Figure 4.20 shows the smoothed DOAs estimated by the beamformers. We notice that the DOA estimates seem to be more noise corrupted than the DOAs seen in Figure 4.12. The frames with the greatest variation in the estimated DOAs are the same frames with poor SNRs in Figure 4.19. The SNRs after beamforming are less than the SNRs before beamforming. See Figure 4.21.

The fluctuation DOA estimates result in very fluctuating estimations of the position of the source. See Figures 4.22 and 4.23. The smoothed estimates of the location are better apart from the estimated x-coordinate. See Figures 4.24 and 4.25.

SNR estimation was also done for three parts of the signals in the third experiment. The results are shown in Tabel 4.6.

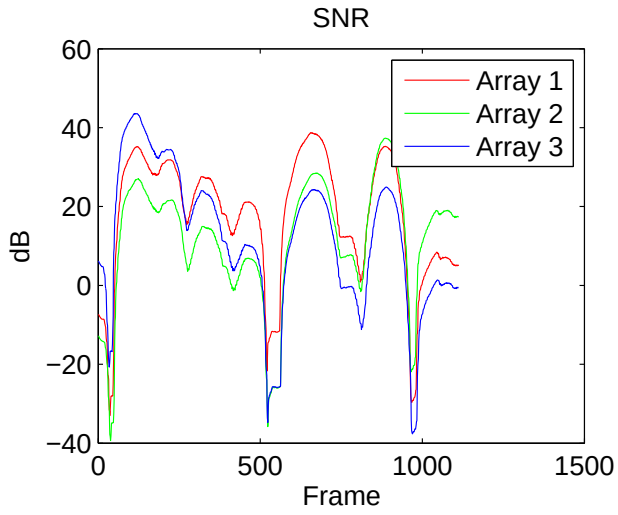


Figure 4.19: SNRs of the noisy microphone array input signals

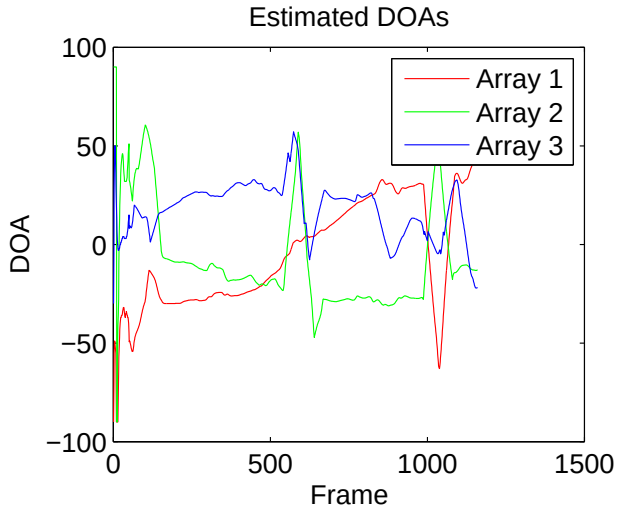


Figure 4.20: Estimated DOAs of the source (smoothed)

Table 4.6: SNR Estimation (Experiment 3)

Array	Sample 1-256	Sample 9001-9256	Sample 18001-18256
1	-18.1 dB	-23.2 dB	16.1 dB
2	-Inf dB	-12.8 dB	21.1 dB
3	-1.3 dB	-21.2 dB	14.9 dB

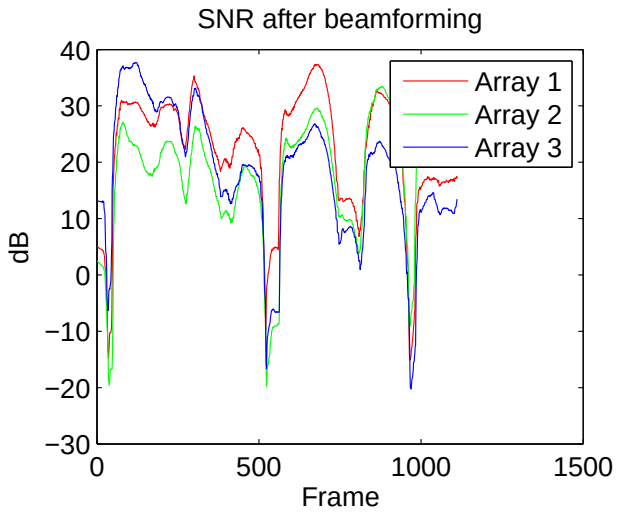


Figure 4.21: SNRs after beamforming

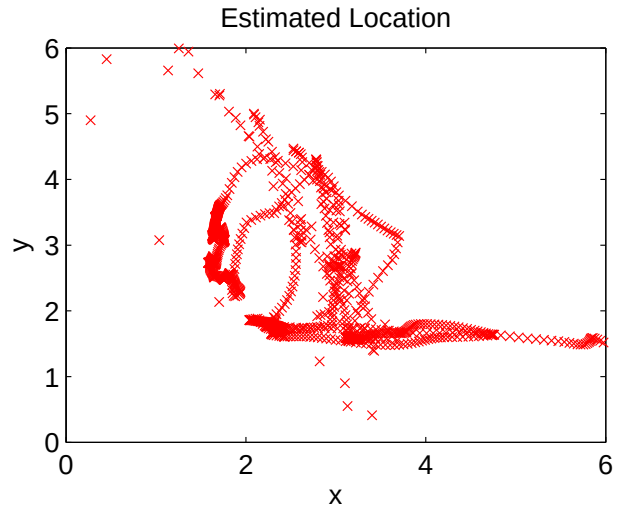


Figure 4.22: Estimated source location

The results shown in Table 4.6 do not correspond to the distances from the arrays to the source, as the results in Table 4.5 do. This is expected when the looking at the fluctuating nature of the estimated DOAs of the signals in Figure 4.20.

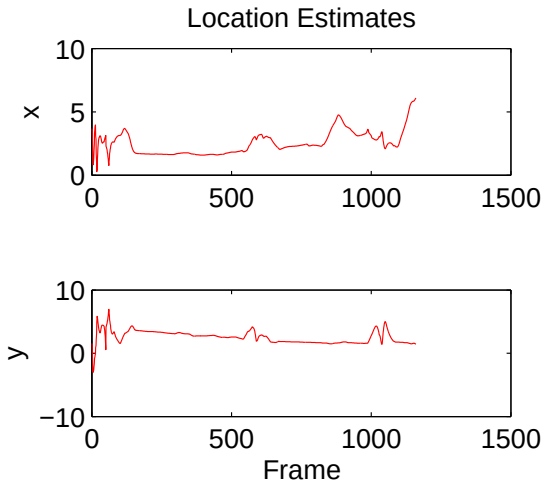


Figure 4.23: Estimated source location

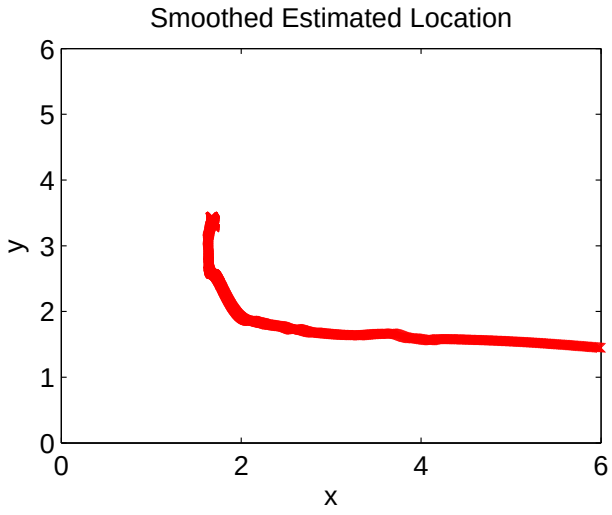


Figure 4.24: Smoothed estimated of source location

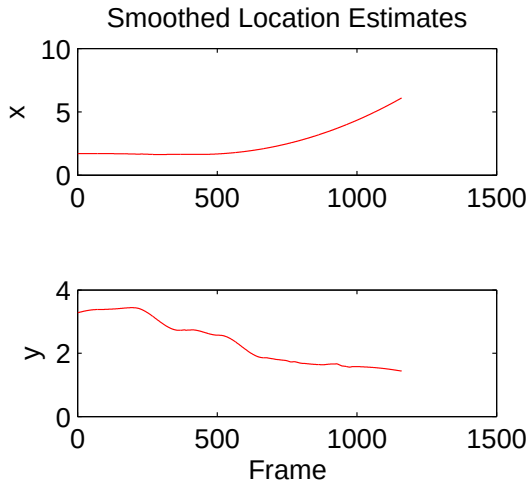


Figure 4.25: Smoothed estimate of source location

Chapter 5

Discussion

In this chapter the results from the previous section will be discussed. Furthermore, a conclusion is presented.

The proposed system has been tested in three experiments. One experiment with a static test signal consisting of three harmonically related sinusoids, one experiment with a moving version of the test signal and an experiment where an input signal from the NOIZEUS speech corpus is used.

The system seems to be able to estimate the DOAs of both the static and the moving test signal, with an increase in the SNR of approximately 15 dB for the static signal. The beamformer is struggling to estimate the DOAs of the speech signal. This is illustrated by Figures 4.4, 4.12 and 4.20. The estimations of the locations of the source follow the same trend. See Figures 4.5, 4.14 and 4.22. Furthermore, the noise in the estimated DOAs in Figure 4.11, despite a high input SNR and a static position of the source, indicate that a mistake could have been made in the implementation.

The failure of the beamformer to estimate the DOAs of the signals for the speech signal input could be attributed to improper voice activity detector

setup. Figures 4.19 and 4.20 indicate that the voice activity detection does not work as intended. Parameters need to be adjusted.

5.1 Conclusion

We recall the problem statement:

> *How is it possible to determine the location of a sound source using distributed microphone arrays, and use this information to improve the SNR of the output signal when extracting a signal using beamforming?*

The results of the experiments indicate that the implementation of the DOA estimation and beamforming part of the system is problematic. In theory a system that switches between microphone array outputs based on SNR estimates of the outputs of the individual microphone arrays. Further experiments would have to be conducted in order to determine whether the proposed method is capapble of improving the SNR of the output signal.

The execution time of the ANLS order- and pitch-estimator is quite high, approximately 90 seconds for a signal of 256 samples. This makes it time-consuming to perform SNR-estimation on signals of substantial length.

Unfortunately the parts of the system that chooses the output of the beamformer with the best SNR has not been thoroughly tested because of inconsistencies in previous experiments. In a situation where the DOA estimates are so noisy that the beamformed signal has a lower SNR than the original, noisy signal, it was considered wasteful to direct the beamformer at the adjusted angle. The location estimates depend equally on the DOA estimates at each frame. If the DOAs are noisy, the location estimate will not

be great. On the other hand, if the location estimate was weighted corresponding to the noise in the estimated DOAs, a scenario where a satisfying location estimate is found despite noise on some of the DOA estimates is possible.

Comparing Figures 4.20 and 4.22 with Figures 4.12 and 4.14 it seems like the performance of the locations estimation is closely related to the quality of the DOA estimates. The location is estimated using a least-squares approach, finding the point which has the minimum distance to the lines drawn by the DOAs and array positions. This approach does not take into consideration the fact than the output of one of the arrays might have a higher SNR that the other arrays. It would be interesting to try weighting the DOAs of an array according to the SNR of the output of the array.

A Capon method for uncertain direction vectors exist. When the DOA is imprecise, the performance could be worse than the performance of conventional delay and sum beamforming according to Stoica and Moses (2005) (Stoica and Moses, 2005, p. 306-311).

5.2 Future Work

The experiments conducted in this project leave many questions unanswered. The results indicate that the current weighting of DOAs could be improved if the quality of the DOA estimates are taken into consideration. In a future iteration of the project such an improvement could be developed.

It would be beneficial to try more additional configurations of input signal, noise signals and SNRs. The system should also be tested in a real world environment. Furthermore, experimentation should be done with the choice

of array based on SNR estimation and comparison between arrays. Experimentation should be done with the calculation of the SNR, more specifically the window size and the hop size. Another area for further investigation is the problem of how often array output SNRs should be estimated and compared.

The current system only works on off-line data. A future improvement could be a real-time implementation.

Bibliography

Allen, J. B. and D. A. Berkley (1979). Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America 65*, 943.

Brandstein, M. and D. Ward (Eds.) (2001). *Microphone Arrays - Signal Processing Techniques and Applications*. Springer.

Brookes, M. (2013). Voicebox: Speech processing toolbox for matlab. Technical report. The toolbox can be downloaded at `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`.

Capon, J. (1969). High-resolution frequency-wavenumber spectrum analysis. *Proceedings of the IEEE 57*(8), 1408–1418.

Christensen, M. and A. Jakobsson (2009). *Multi-Pitch Estimation*. Synthesis Lectures on Speech and Audio Processing. Morgan & Claypool Publishers. Toolbox available online at `http://www.morganclaypool.com/page/multi-pitch`.

Cox, H., R. Zeskind, and M. Owen (1987). Robust adaptive beamforming. *Acoustics, Speech and Signal Processing, IEEE Transactions on 35*(10), 1365–1376.

Deza, M. M. and E. Deza (2013). *Encyclopedia of Distances* (Second ed.). Springer.

DiBiase, J. (2000). *A High-Accuracy, Low-Latency Technique for Talker Localization in Reverberant Environments*. Ph. D. thesis, Brown University, Providence RI, USA.

Ellis, D. (2002). Aurora noise database. Technical report. The noise database can be downloaded at `http://www.ee.columbia.edu/~dpwe/sounds/noise/`.

Habets, E. (2010). Room impulse response generator for matlab. `http://home.tiscali.nl/ehabets/rir_generator.html`. Included on DVD.

Habets, E. (2011). Signal generator for matlab. `http://home.tiscali.nl/ehabets/signal_generator.html`. Included on DVD.

Hu, Y. and P. Loizou (2007). Subjective evaluation and comparison of speech enhancement algorithms. *Speech Communication 49*, 588–601. The speech corpus can be downloaded at `http://www.utdallas.edu/~loizou/speech/noizeus/`.

Lockwood, M. E., D. L. Jones, R. C. Bilger, C. R. Lansing, W. D. O. Jr., B. C. Wheeler, and A. S. Feng (2004, January). Performance of time- and frequency-domain binaural beamformers based on recorded signals from real rooms. *Journal of the Acoustical Society of America 115*(1).

Microsoft Corporation (2013). Microsoft kinect audio stream. `http://msdn.microsoft.com/en-us/library/jj131026.aspx`. Included on DVD.

Ney, H. (1983). Dynamic programming algorithm for optimal estimation of speech parameter contours. *IEEE Transactions on Systems, Man, and Cybernetics SMC-13*(3).

Oticon A/S (2007). The audiology in epoq - a whitepaper. `http://www.oticon.com/~asset/cache.ashx?id=10193&type=14&`. Included on DVD.

Stoica, P. and R. Moses (2005). *Spectral Analysis of Signals*. Prentice Hall.

van der Heijden, F., R. P. W. Duin, D. de Ridder, and D. M. J. Tax (2004). *Classification, Parameter Estimation and State Estimation*. Wiley.

Van Veen, B. and K. Buckley (1988). Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Magazine 5*(2), 4–24.

Zölzer, U. (Ed.) (2011). *DAFX: Digital Audio Effects* (Second ed.). Wiley.