Implementation and Evaluation of a Novel Source Localization Algorithm using Dipole Microphones

Bachelors Thesis
of
Thomas Baer

Lehrstuhl für

Multimediakommukation
und Signalverarbeitung

Lehrstuhl für Multimediakommunikation und Signalverarbeitung
Friedrich-Alexander Universität Erlangen-Nürnberg
June 2013

Lehrstuhl für Multimediakommunikation
und Signalverarbeitung
Prof. Dr.-Ing. W. Kellermann

Friedrich-Alexander-Universität
Erlangen-Nürnberg

Bachelor Thesis

Implementation and Evaluation of a Novel Source Localization Algorithm using Dipole Microphones

In acoustic signal processing, localizing acoustic sources is one of the fundamental problems. The fewer microphones are available and the more closely they are spaced, the more difficult this problem becomes. In this thesis, an extreme case is considered: Single dipole microphones should be used for finding the direction from which the source signal arrives, so that the cardioid directivity pattern, that is formed by the dipole signals, can be exploited for spatial discrimination.

In the thesis, the underlying beamforming theory and the direction-finding algorithm shall be described analytically and illustrated by simulations. Its real-time implementation shall be documented and experimentally evaluated. The experimental results shall be compared with the simulations and incongruencies shall be investigated. The evaluation shall especially focus on the sensitivity of the algorithm to ambient noise and the dependency on the source distance.

To this end, Mr. Bär will study the theoretical background, become acquainted with and use the specific real-time software development tools for the cardioid microphones under consideration, and develop measurement methods and according software for the experimental evaluation. Thoughtfully designed and well-documented experiments and a well-structured software are expected.

Start date: February 1, 2013
End date: June 20, 2013

(Prof. Dr.-Ing. W. Kellermann)

I assure I wrote the presented thesis myself and without any illegimitate help.

Erlangen, June 20, 2013

(Thomas Bär)

Acknowledgement

Special thanks to the company mh acoustics LLC [1] and the mh team, namely Gary W. Elko, Tomas Gaensler, Eric J. Diethorn and Jens M. Meyer, who provided the necessary hardware and helpful, supportive ideas.

Also special thanks to Prof. Dr.-Ing. Walter Kellermann for the arrangement of getting in contact with mh acoustics LLC and its staff.

Finally I want to thank Hendrik Barfuß, whose project was used as a solid foundation for the implementation on the real time system.

[1] `http://www.mhacoustics.com/mh_acoustics/mh_acoustics.html`

Contents

Chapter 1

Introduction

Some time ago, manufacturers of communication devices recognized the importance of several microphones in their system to improve the offered quality of speech with their devices. So the most recent smartphone of Apple, the Iphone 5, has three microphones for supressing ambient noise. Also serveral manufactures of conference phones discovered they can improve their quality of speech by using a microphone array in their systems.

One technique for supressing ambient noise with several microphones, a microphone array, is the so called *Beamforming* (c.f. chapter 2). With this technique the microphone array gains a directivity that can be exploited for supressing sound from undesired directions.

Since most beamforming algorithms require knowledge on the direction in which their directivity shall point, cost efficient algorithms for obtaining an estimation for the direction of the target direction are desirable, which do not require an evaluation of an arbitrary costly beamforming algorithm.

Since the directivity of cardioids show a unique spatial null, i.e., ideally no sound component is picked up from that direction, this spatial null can be exploited for finding a target direction.

An algorithm will be presented which uses four microphone outputs and combines them using three adaptive filter coefficients by exploiting the assembly of a virtual cardioid microphone.

This thesis will be arranged in the following way: The next chapter will introduce basic concepts of beamforming, which are relevant for the implementation of the simulation and the understanding of the proposed algorithm. Chapter 3 will introduce the direction finding algorithm as an optimization problem and solves it by using an least-mean-squares (LMS) approach. In chapter 4 the implementation of the simulation for this algorithm will be shortly discussed and its results will be presented. Chapter 5 provides some information on the implemenation on a real time system and measured results for the direction estimation. Chapter 6 will conclude this thesis by recapitulating the content of the thesis.

Chapter 2

Beamforming

Beamforming, a *Microphone Array Signal Processing* technique, describes different algorithms to form the directional characteristic of a microphone array consisting of at least two microphones.

Within these algorithms the microphone signals are combined such that signals that arrive from a desired direction gain an attuation. Therefore a beamformer leads to an increased signal to noise ratio (SNR) w.r.t. a single microphone.

In the following some basic beamforming techniques will be described, namely the sum-and-delay beamforming and a method for first order differential beamforming.

2.1 Delay-and-Sum Beamforming

A simple realization of a beamformer is the so called Delay-and-Sum Beamforming. The individual microphone signals of a microphone array with an arbitrary amount of elements and an arbitrary geometric alginment are first delayed, sucht that the time delays of a from the desired receive direction arriving signal are adjusted. Afterwards the signals are summed up

Mathematically this can be described as

$$y(t) = \sum_{m=0}^{M-1} a_m x_m(t - \tau_m) \tag{2.1}$$

where M denotes the amount of microphones and a_m denote the weighting factors of the microphones.

A simple, common approach is to choose equal weighting factors a_m, such that the microphone signals are averaged. This method is called unweighted delay-and-sum beamforming. The microphone delays τ_m can be defined as static values, if the application requires it, or can be computed according to time-delay estimation, which e.g. utilize the cross-correlation of the microphone signals (for further information c.f. [7]) for the delay computation.

Are for instance two microphones aligned according to Figure 2.1 with a constant spacing d and the defined receive direction θ the delay between the microphone $m = 0$ and microphone $\hat{m} = 0 \dots M$ can be expressed as

$$\tau_m = \frac{\hat{m} d \cos\theta}{c} \tag{2.2}$$

where c denotes the speed of sound. It can be shown that doubling the amount of microphones in the array leads to an SNR improvement of 3 dB given the interference and the desired signal are uncorrelated [6].

Note: this technique is an example for *fixed beamforming*, since the parameters do not change over time. For adaptive beamforming techniques the reader is referred to the named literature [6].

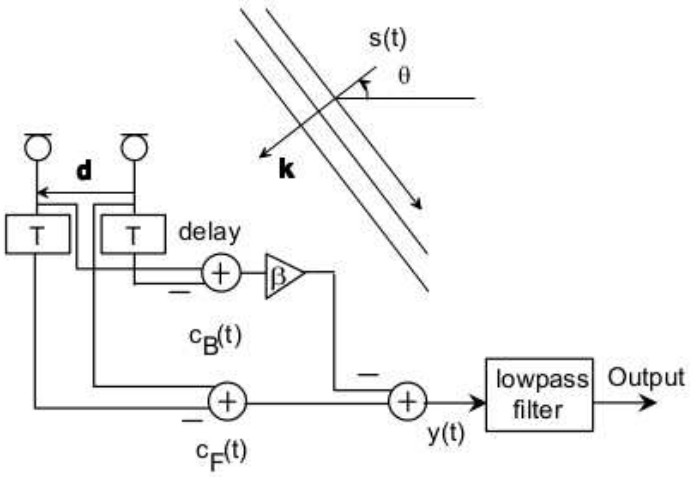


Figure 2.1: Schematic implementation of an adaptive first-order differential microphone

2.2 Differential Beamforming

An elegant method for realizing differential beamforming by using microphones with an omnidirectional characteristic was developed by Gary W. Elko and will be described in the following.

The schematic structure of a differential beamformer is depicted in Figure2.1. The basic idea is the scalar combination of two back-to-back cardioid microphones.

The distance between the microphones $d = |\mathbf{d}|$ is assumed to be small such that a low delay between the signals can be assumed ($kd << \pi$ and $\omega T << \pi$, with the wave number $k = w/c = |\mathbf{k}|$).

Choosing the sampling period $T = \frac{d}{c}$, the forward facing cardioid C_F and the backward facing cardioid C_B can be expressed as:

$$C_F(\omega, \theta) = 2jS(\omega)e^{-j\omega T/2} \sin \frac{kd(1 + \cos\theta)}{2} \tag{2.3}$$

$$C_B(\omega, \theta) = 2jS(\omega)e^{-j\omega T/2} \sin \frac{kd(1 - \cos\theta)}{2} \tag{2.4}$$

where the spatial origins are placed at the array center to simplify the expressions and $S(\omega)$ denotes the spectrum of the incoming signal.

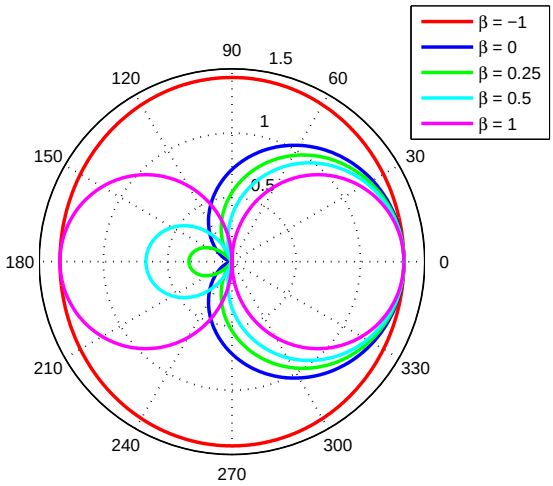


Figure 2.2: Different realizable directional patterns w.r.t. β

Note that the amplitude characteristic is increasing linearly with the frequency, such that this frequency dependency can be corrected with a first order lowpass filter.

A scalar combination with a factor β yields the to $S(\omega)$ normalized output spectrum and output signal:

$$\begin{aligned} |Y(\omega,\theta)| =& C_F(\omega,\theta) - \beta C_B(\omega,\theta) \\ =& 2\left|\sin\frac{kd(1+\cos\theta)}{2} - \beta\sin\frac{kd(1-\cos\theta)}{2}\right| \end{aligned}$$

respectively

$$y(t) = c_F(t) - \beta c_B(t) \tag{2.5}$$

where β can be a value between -1 and 1.

Evaluation of the output spectrum for different values for β reveal an omnidirectional characteristic for $\beta = -1$ and a dipole characteristic for $\beta = 1$, c.f. Figure 2.2. Therefore the characteristic of the microphone array can be changed when using an adaptive algorithm for determining β. [4, 1, 3]

Chapter 3

Adaptive Cardioid Direction Finder Algorithm

In this chapter the basic idea behind the cardioid direction finder algorithm as well as a implementation as an LMS algorithm will be presented. The algorithm uses one signal which shows an omnidirectional response and three signals which show an bipolar or dipole characteristic. These signals can be obtained by using closely-spaced omnidirectional microphones . The simple solution adaptively finds the location of the single cardioid null that minimizes the output power of a generally rotated cardioid.

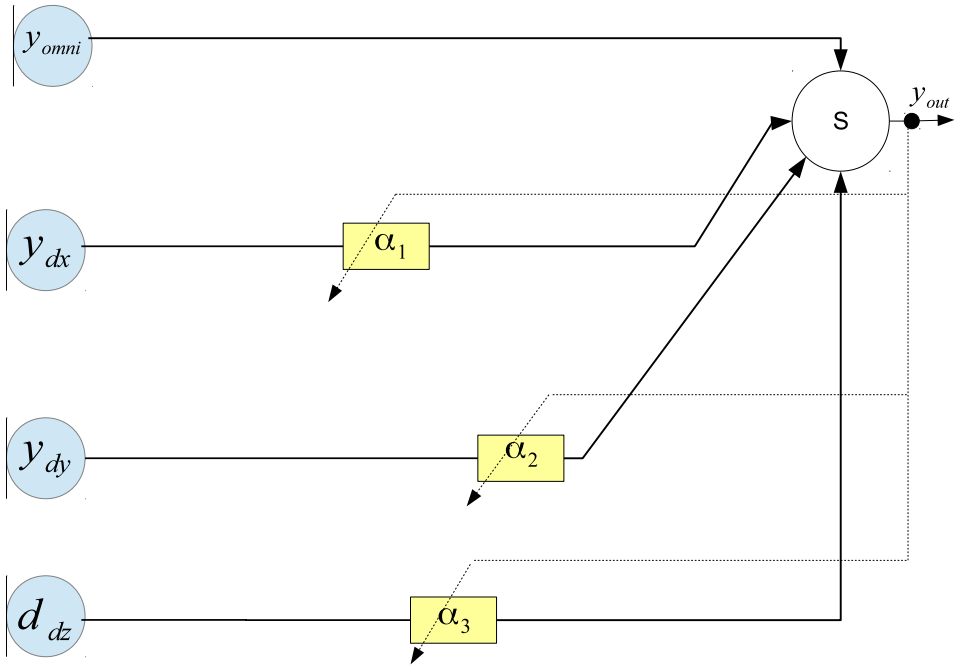


Figure 3.1: Block diagram of an adaptive general orientation cardioid array

3.1 Cardioid Direction Finder Algorithm

A first-order adaptive cardioid beamformer is shown in figure 3.1. Audio signals $\mathbf{X}(n)$ at a sampling rate f_s are input to the adaptive filter that has only three taps. The vector input signal is defined as a four element vector and is described as:

$$\mathbf{X} = [y_{omni} \quad y_{dx} \quad y_{dy} \quad y_{dz}]^T \tag{3.1}$$

where the components of $\mathbf{X}$ are the input signals consisting of processed signals, which are obtained from a beamformer that computes signals representative of an omnidirectional response y_{omni}, and the three orthogonal dipole signals y_{dx}, y_{dy}, y_{dyz} whose axes are aligned with the cartesian coordinate system. The dipole signals can be computed by subtracting closely spaced omnidirectional signals (c.f. Chapter 2.2). For an implementation where the cardioid null can be located at any spherical direction, c.f. [2], the minimum number of microphones is 4. Similarly, the minimum number of microphones for a 2D implementation is 3 microphones, where one of the orthogonnal dipole components can be dropped.

The output y_{out} is formed by a weighted sum of the input signals, where the orthogonal dipole signals y_{dx}, y_{dy}, y_{dyz} are weighted by the adaptive factors $\alpha_1, \alpha_2, \alpha_3$ respectively as shown in figure 3.1. In order to constrain the output beampattern to a cardioid beampattern, the following constraint must be met:

$$A_d^T A_d = 1, \qquad \text{where } A_d = [\alpha_1 \quad \alpha_2 \quad \alpha_3] \tag{3.2}$$

The constraint on the dot product of the dipole scalar coefficients for the combination of the three orthogonal dipole outputs leads to an also normalized dipole that can have any general orientation. Combining the omnidirectional term with a dipole in equal amounts results in a general cardioid beampattern whose solitary null direction can be steered in any spatial direction. [2]

The output of the adaptive cardioid array can then be written as

$$y_{out} = \mathbf{A}^T \mathbf{X} = y_{omni} - A_d x_d \tag{3.3}$$

where

$$A = [1 \quad -\alpha_1 \quad -\alpha_2 \quad -\alpha_3]^T, \quad x_d = [y_{dx} \quad y_{dy} \quad y_{dz}]^T \tag{3.4}$$

When solving the following optimization problem with the above constraint on the filter taps, the cardioid null is steered to the direction wwith the highest incoming sound energy:

$$\min||y_{out}||^2, \qquad \text{subject to } y_{omni} \text{ and } x_d \tag{3.5}$$

3.2 LMS Implementation

For obtaining a simple and effective time domain algorithm, the partial derivates of $y^2(n)$ are formed with respect to the filter coefficients $\alpha_i, \quad i = 1, 2, 3.$:

$$\frac{\partial y_{out}^2}{\partial \alpha_1} = -2y_{dx} \cdot y_{out}, \quad \frac{\partial y_{out}^2}{\partial \alpha_2} = -2y_{dy} \cdot y_{out}, \quad \frac{\partial y_{out}^2}{\partial \alpha_3} = -2y_{dz} \cdot y_{out}, \tag{3.6}$$

with y_{out} from equation 3.3.

With the approximation $E\{y_{out}^2(n)\} \approx y_{out}^2(n)$ a LMS algorithm can be formed to solve the optimization problem 3.5:

$$A_d(n+1) = A_d(n) + \frac{1}{2}\mu \left[\nabla(E\{y_{out}^2(n)\}\right] \approx A_d(n) + \frac{1}{2}\mu\nabla\left(y_{out}^2(n)\right) \quad (3.7)$$

where $\nabla\left(E\{y_{out}^2(n)\}\right), \nabla\left(y_{out}^2(n)\right)$ denote the gradient of the expectation of the squared output respectively the squared output with respect to the vector A_d (c.f. equation 3.2) and μ is the adaption step-size constant which is bound for stability as

$$0 < \mu < \frac{2}{tr[R_{dx}] + tr[R_{dy}] + tr[R_{dz}]} \quad (3.8)$$

where R are the corresponding dipole input correlation matrices and $tr[]$ is the trace of the matrix.

Using the partial derivates computed in equation 3.6 in equation 3.7 yield the final form of the filter coefficients update formula:

$$A_d(n+1) = A_d(n) - \mu \cdot x_d(n) \cdot y_{out}(n) \quad (3.9)$$

The corresponding spherical angles, azimuth ϕ and elevation θ, that minimize the output power of the generally steered cardioid can be derived by basic trigonometry and can be computed on a sample by sample basis. But one has to take into account, that the tap values α_i are computed, such that the spatial null is steered to the desired direction, i.e. the negative tap values have to be considered, when computing the desired estimation directions, since the desired source is located on the other side of the sphere around the microphone array.

$$\phi(n) = \tan^{-1}\left(\frac{-\alpha_2(n)}{-\alpha_1(n)}\right) \quad (3.10)$$

$$\theta(n) = \cos^{-1}\left(\frac{-\alpha_3(n)}{A_d(n)^T A_d}\right) \quad (3.11)$$

Note that in an actual implementation an 'atan2' function shall be used for computing ϕ to account for the sign of the tap values.

Chapter 4

Simulation

In this chapter the simulation of the algorithm will be discussed. This includes the microphone setup in the simulation, the creation of the desired microphone signals and presentation of the simulation result. For simplicity, the detailed discussion will only cover the 2D simulation. The necessary changes for the 3D case will be pointed out, but not discussed in detail.

4.1 Microphone Setup

For a 2D cardioid direction finder simulation the smallest number of microphones that can be used is three, c.f. Chapter 2.2: a single omnidirectional microphone with two co-located dipole microphones elements, or three closely spaced omnidirectional micrphones in an equiliteral triangle. The last setup requires that pair-wise differences to form three dipole signals that are further processed to produce the desired orthogonal dipole signals.

For the simulation a setup with four omnidirectional microphones was chosen, since the desired orthogonal dipole signals can be directly derived. The spatial setup of the microphones are depicted in Figure 4.1. Four omnidirectional microphones are arranged at the vertices of a square with a side length of d. With these four microphones the desired virtual microphones will be created: two orthogonal dipole microphones and one omnidirectional microphone, which are located in the middle of the square.

In the following, the numbering of the microphones is chosen, like it is depicted in Figure 4.1. Forward differences cfx, cfy and backward differences cbx, cby, c.f. chapter 2.2, will be computed from the microphone pairs M_1,M_2 respectively M_3,M_4:

$$cfx[1:N-1] = 0.5 \cdot (M_1[1:N-1] - M2[2:N]) \tag{4.1}$$
$$cbx[1:N-1] = 0.5 \cdot (M_2[1:N-1] - M1[2:N]) \tag{4.2}$$

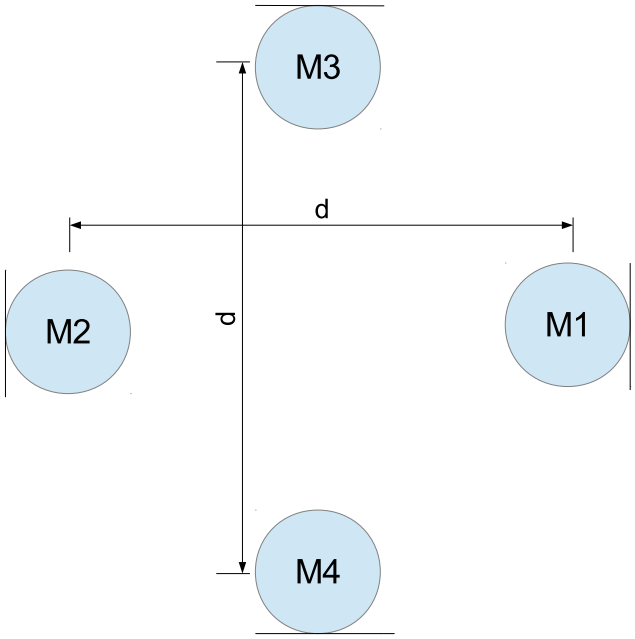
with N as the length of the microphone signals, where the sampling period was has to be chosen to $T = \frac{d}{c}$ where d denotes the spacing of the microphones and c the speed of sound.

cfy and cby will be computed by forming the differences in the y-direction of the setup, i.e. replacing M_1 by M_3 and M_2 by M_4 in equations 4.1 and 4.2.

Applying equation 2.5 with $\beta = 1$ for the forward and backward differences yields the desired dipole microphone signal M_x, M_y in x-direction respectively in y-direction, which virtual microphones are located in the middle of the square.

$$M_x = cfx - cbx \tag{4.3}$$
$$M_y = cfy - cby \tag{4.4}$$

Figure 4.1: Four-element array with omnidirectional microphones for 2D direction finding



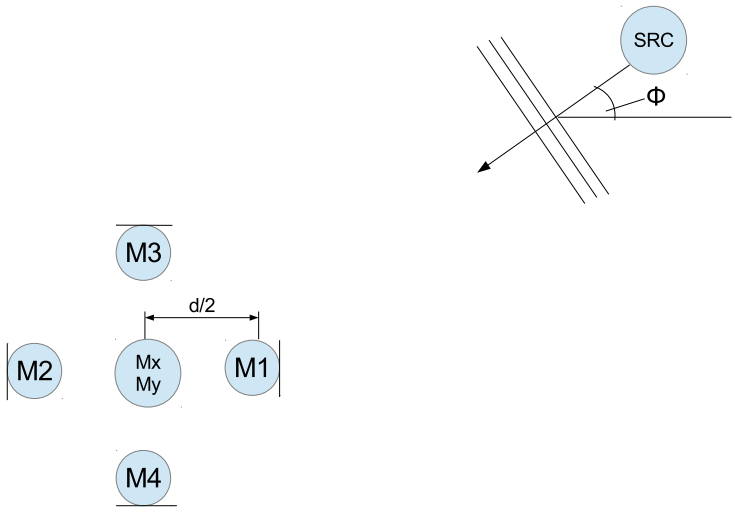
The desired centered virtual omnidirection microphone can be created according to equation 2.5 with $\beta = -1$:

$$M_{omni} = 0.5 \cdot (cfx + cbx + cfy + cby) \tag{4.5}$$

Note that the virtual omnidirectional microphone M_{omni} is derived from the forward and backward differences, so that the dipole signals and the omnidirectional signal have the same sensitivity along the main dipole axis.

For the 3D case the microphones are located on the faces of cube. For creating the orthogonal dipole in z-direction, the forward and backward differences have to be computed from the two additional microphones. For the omnidirectional virtual microphone these additional differences also have to be taken into account.

Figure 4.2: Four-element array with sine wave source at angle ϕ



4.2 Signal Creation

For the simulation a sound source will be considered which emits a sine wave of a certain frequency f and is located at angle ϕ like it is depicted in Figure 4.2.

The desired sampled microphone signals M_i can then be described as:

$$M_i = \sin(2\pi f t + \omega\tau_i) \tag{4.6}$$

where $\omega = 2\pi f$.

The sampling frequency f_c will be chosen to $f_c = \frac{c}{d}$, c.f. [1], where d denotes the square length in the spatial setup and c is the speed of sound ($\approx 343\frac{m}{s}$)

Assuming a virtual microphone in the center of the microphone setup with a relative sampling delay of 0, according to equation 2.2 the delays τ_i between the virtual microphone and the real microphones M_i can be written

as:

$$\tau_1 = -\frac{\cos(\phi) \cdot \frac{d}{2}}{c} \tag{4.7}$$

$$\tau_3 = -\frac{\sin(\phi) \cdot \frac{d}{2}}{c} \tag{4.8}$$

The values of the delays τ_2, τ_4 are the negative of the the delays τ_1 respectively τ_3, since the microphone M_2 and M_4 are located on the opposite side.

After the sampling of microphones M_1 - M_4, gaussian noise will be applied to the signals, which power is chosen accordingly to the amplitude and therefore the energy of the incident sine wave.

For the 3D case the sine wave sound source is located on a sphere around the microphone setup, such that its position can be described by its spherical coordinate ϕ and θ. The derivation of the relative delays can be done straightforward in the same way.

4.3 LMS Algorithm

Combining the sampling of the sine wave source according equation 4.6 and combining the microphone signals according equations 4.1 to 4.5 all requirements for the actual proposed LMS algorithm are met. A straight forward implementation is possible on a sample-by-sample basis over all microphone samples of the microphones M_x, M_y and M_{omni} and applying equations 3.2, 3.3, 3.9, 3.10 yield:

$$\begin{aligned} Y[i] &= M_{omni}[i] - \alpha_1 \cdot M_x[i] - \alpha_2 \cdot M_y[i] \\ \alpha_1 &= M_{omni}[i] - \mu\alpha_1 Y[i] M_x[i] \\ \alpha_2 &= M_{omni}[i] - \mu\alpha_2 Y[i] M_y[i] \\ n &= \sqrt{\alpha_1^2 + \alpha_2^2} \\ \alpha_1 &= \alpha_1/n, \quad \alpha_2 = \alpha_2/n \\ \phi &= arctan2(-\alpha_2, -\alpha_1) \end{aligned}$$

Note that in real applications the angle computation should only be evaluated for every sample if it is required. Otherwise it is sufficient to compute the angles after a certain block length, e.g. 512 or 1024 samples.

For the 3D case, the third orthogonal dipole microphone signal has to be added, which includes a third estimation parameter α_3 and the elevation θ has to be computed according to equation 3.11.

4.4 Results

For running the simulation, a few parameters have to be chosen by hand: the incident sine wave frequency, the noise power relative to the sine wave power, the microphone spacing, the incident angle of the sound wave and the step size for the LMS algorithm.

The estimated null angle location will be shown as a function of time. A microphone spacing of 20 mm was chosen for all simulations, since that spacing leads to a reasonable sampling frequency of about 17 kHz.

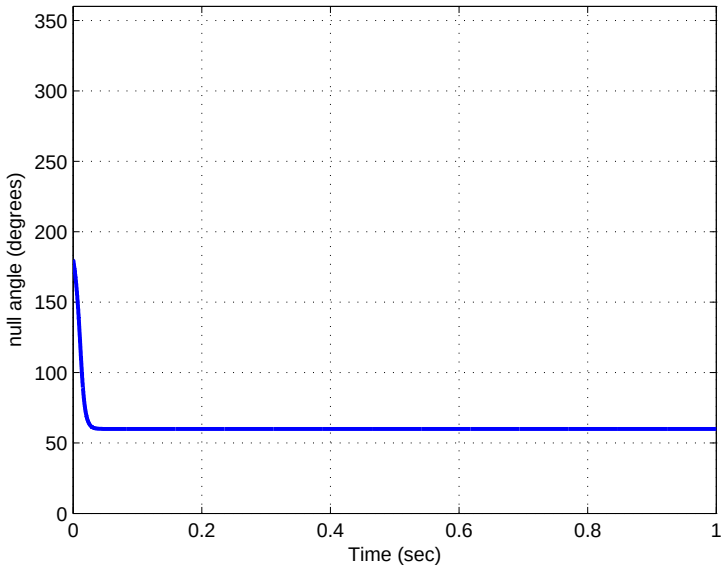


Figure 4.3: Simulated estimation for 1kHz sine at 60 degree, 60 dB SNR, step size 0.03

In Figure 4.3 the progress of the 2D estimation is shown for a 1 kHz sine wave with a incident angle of 60 degree and a very high SNR of 60 dB. The LMS step size was chosen to $\mu = 0.1$ It can be seen that the convergence of the estimation is reached mainly within the first 50 ms.

Increasing the noise power resulting in a SNR of 10 dB, results in a worse null angle estimation, as can be seen in Figure 4.4. Although the jitter within the estimation with a SNR of 10 dB is clearly visable, the estimation after 50 ms is still reasonable.

The jitter in the estimation when dealing with low SNR can be reduced by lowering the LMS step size. In Figure A.1 the impact of lowering the LMS step size from 0.1 to 0.03 can be seen. The jitter is reduced, but the time until approximate convergence is inreased. A reasonable estimate can be obtained after the first 100ms, though.

Simulations were also done for the 3D case, where six microphones are located on the faces of a cube with 20x20 mm dimensions. Similiar results were obtained by the 3D simulations.

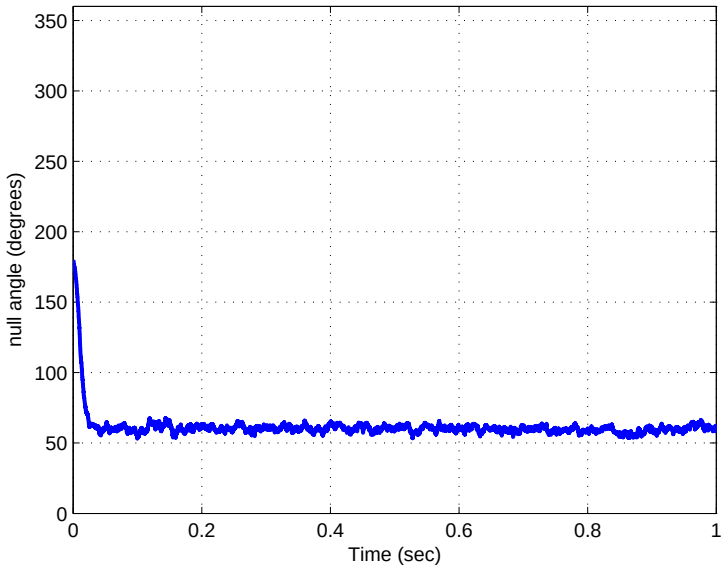


Figure 4.4: Simulated estimation for 1kHz sine at 60 degree, 10 dB SNR, step size 0.1

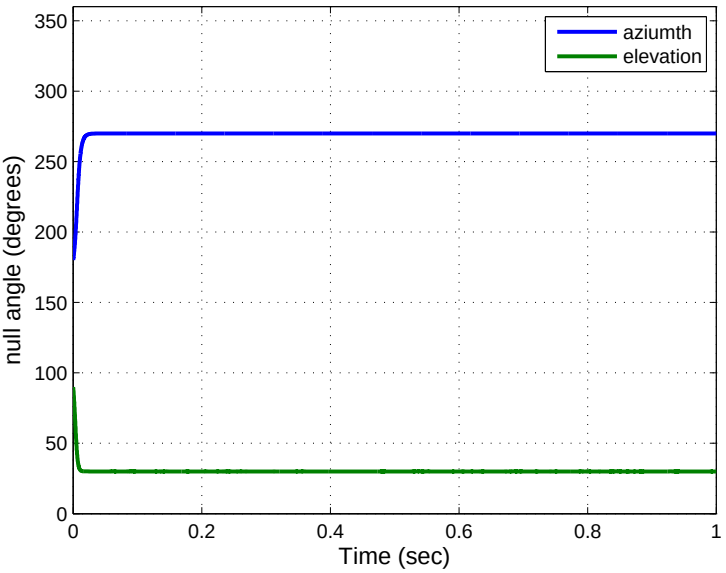


Figure 4.5: Simulated 3D estimation for 1kHz sine at (270,30) degree, 60 dB SNR, step size 0.03

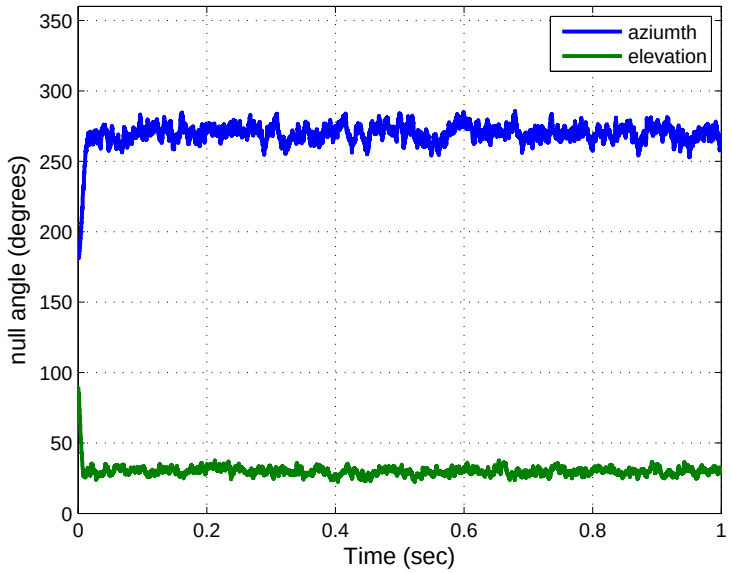


Figure 4.6: Simulated 3D estimation for 1kHz sine at (270,30) degree, 10 dB SNR, step size 0.1

As with the 2D cases, the first simulation was done with a high SNR of 60 dB and an adaption step size of 0.1. Figure 4.5 shows the estimated azimuth and elevation of a sound source that is located at an azimuth of 270 degree and an elevation of 30 degree. One can see that the estimation converges to the true direction even faster than in the 2D case.

As the 2D case was moderately fragile to noise, the 3D simulations show the same behaviour. In Figure 4.6 the same setup can be seen when the noise level rises and the SNR of 10 dB is reached.

It is interesting, that the estimated angles converge quicker in the 3D case compared to the 2D case, when the step size is chosen smaller. In Figure A.2 the step size was chosen to 0.03. While in the 2D case far more than 100 ms were needed to reach a 10 degree interval around the real angles, the simulations for the 3D case reach such a interval in less than 100 ms.

Chapter 5

Real Time Evaluation

Since the simulation shows promising results for the cardioid direction finder algorithm, the algorithm was implemented into a real time application and evaluated for different scenarios.

In this chapter the used hardware- and software will be described. Some details on the implementation will be discussed and the real time results will be presented.

5.1 Hardware and Software

For the real time implementation an Eigenmike® microphone array[1] was used for the audio processing. This 32 element microphone array first combines its microphone signals using digital signal processing to create a set of Eigenbeams. A soundfield up to the spatial order of the beamformer is captured by a complete set of Eigenbeams. Then the Eigenbeams are combined to steer multiple simultaneous beampatterns which specific directions in the acoustic field can be focused. With the software package that is delivered with the Eigenmike® a omnidirectional beam and the three spatial orthogonal dipole beams are computed from the 32 microphone signals which are then used in the cardioid direction finder algorithm.

Additionally to the Eigenmike® microphone array for the sound processing a Ladybug®2 Video Camera [2] was utilized for the evaluation of the algorithm. This video camera system has six individual 0.8 megapixel cameras that enables the system to collect video from more than 75% of the full sphere and allows a streaming to disk at 30 frames per second. The camera was used to observe the behaviour of the algorithm for a moving target.

A combination of these two systems was already implemented by Hendrik Barfuß. For the evaluation his application was extended by e.g. the cardioid direction finder algorithm and evaluation routines.

5.2 Coordinate Transformation

As already mentioned, the camera system was used for evaluating the algorithm for a moving target. The video streams from the six individual cameras can be mapped onto a sphere which allows for an almost omnidirectional view of the environment. Since the camera systems and the microphone array cannot be placed at the same place, but have some spatial offset to each other, the coordinates of a target, that can be observed with the camera system have to be converted to coordinates for the microphone array.

Since the offset $d = [d_x, d_y, d_z]^T$ of the systems is given in cartesian coordi-

[1] `http://www.mhacoustics.com/mh_acoustics/Eigenmike_microphone_array.html`
[2] `http://www.ptgrey.com/products/ladybug2/ladybug2_360_video_camera.asp`

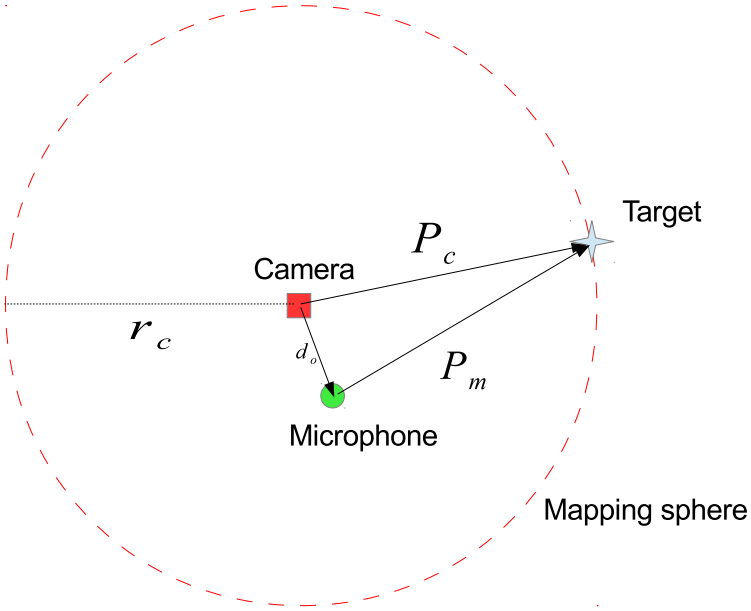


Figure 5.1: Position of camera and microphone system for the angle conversion

nates, the spherical coordinates from the systems are converted into cartesian coordinates, shifted by the offset and converted back into spherical coordinates.

The spherical coordinates P_c in the camera system are given as

$$P_c = [r_c, \theta_c, \phi_c]^T \tag{5.1}$$

where the angles ϕ_c and θ_c can be observed in the displayed video stream and the radius r_c is the sphere size the camera systems uses to map its video images.

A conversion of spherical coordinates into cartesian coordinates (c.f. [5, Addendum F2]) is given by

$$P_{c,cart}((r_c, \theta_c, \phi_c)^T) = \begin{pmatrix} x_c((r_c, \theta_c, \phi_c)^T) \\ y_c((r_c, \theta_c, \phi_c)^T) \\ z_c((r_c, \theta_c, \phi_c)^T) \end{pmatrix} = \begin{pmatrix} r_c \cdot \sin\theta_c \cdot \cos(\phi) \\ r_c \cdot \sin\theta_c \cdot \cos\phi \\ r_c \cdot \cos\theta_c \end{pmatrix} \tag{5.2}$$

Defining the offset between the offset d_o as $d_o = [d_x, d_y, d_z]^T$, c.f. Figure

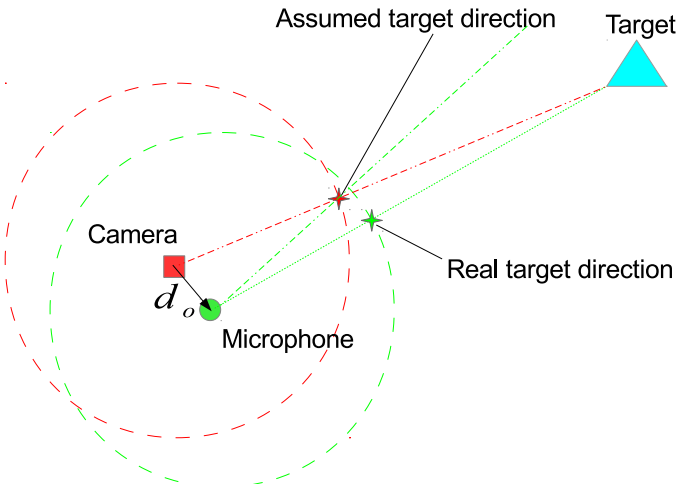


Figure 5.2: Angle transformation from camera and microphone without proper distance to the target

5.1 for a 2D sketch, the cartesian coordinates $P_{m,cart} = [x_m, y_m, z_m]^T$ can be computed by subtraction of the offset from the cartesian coordinates of the camera system:

$$P_{m,cart} = P_{c,cart} - d_o \tag{5.3}$$

The spherical coordinates P_m for the microphone system can now be obtained by the inverse transform of equation 5.2, which reads as follows:

$$P_m((x,y,z)^T) = \begin{pmatrix} r_m((x,y,z)^T) \\ \theta_m((x,y,z)^T) \\ \phi_m((x,y,z)^T) \end{pmatrix} = \begin{pmatrix} \sqrt{x^2+y^2+z^2} \\ \arccos \frac{z}{r} \\ \arctan \frac{y}{x} \end{pmatrix} \tag{5.4}$$

Since the radius component of P_m denotes the distance from the microphone to the sphere on which the camera system maps its images, the radius component r_m of P_m can be omitted for the direction comparison.

Note in the angle transformation that a point located on the mapping sphere of the camera system is transformed. If the actual source position is located much closer or far more away from the system than the mapping

sphere size of the camera system, the transformed direction will not represent the actual source position properly, as it is depicted in Figure 5.2 for a 2D case. Therefore previous knowledge on the distance has to be brought into the evaluation, when comparing the observed direction with the estimated direction, such that the mapping sphere size of the video processing can be set to a proper size.

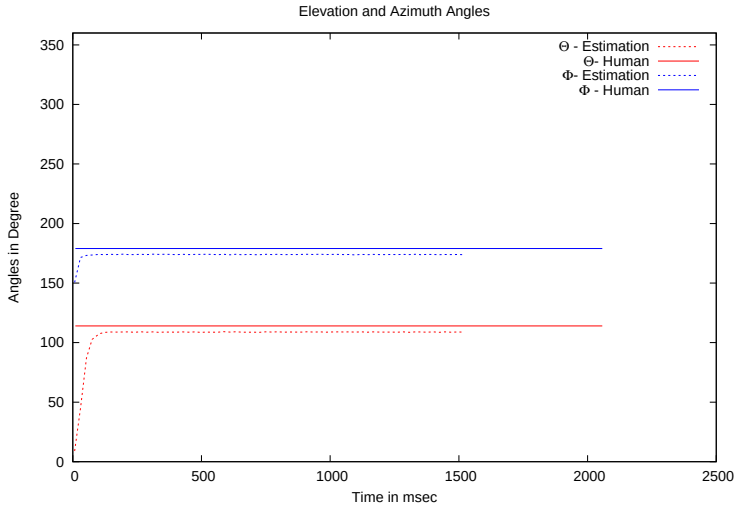


Figure 5.3: Measured estimation of a 440 Hz tone at a distance of 140cm

5.3 Non-moving source

The proposed algorithym will now be evaluated for an easy scenario, where a single source is producing sounds. The sound signals were played over a speaker, which location relative to the microphone system is known. The speaker will remain stationary, i.e. it does not change its position during the playback.

At first the speaker will play a single tone so that the real-time results can be compared with the simulation and interference effects can be analyzed more easily. Afterwards single and multiple speech signals will be played back for covering more realistic scenarios.

In the following evaluation figures the dashed lines will represent the progress of the direction estimation of the source and the solid lines show the position of the source where it could be observed by a human. The angle ϕ denotes the azimuth and the angle θ denotes the elevation of the source. This description also holds for section 5.4.

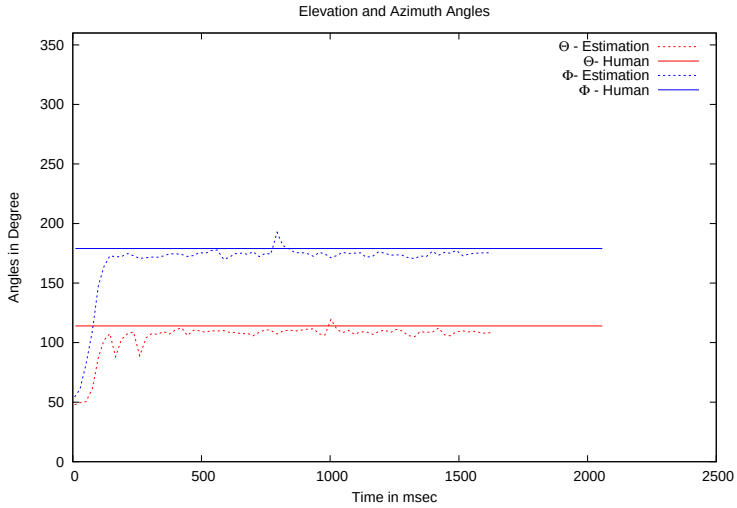


Figure 5.4: Measured estimation of a 440 Hz tone at a distance of 140cm with an SNR of 10dB

5.3.1 Single tone source

Figure 5.3 shows the source location estimation where the speaker was placed at angles of about $\phi = 180°$ and $\theta = 115°$ and at a distance of about 140cm. The speaker played a 440 Hz signal, where no noise was overlayed with the signal. The step size μ of the LMS algorithm was chosen to 0.1. One can see that the estimation converges close to the real position within the first 100 ms which is consistent with the results obtained from the simulation.

In the next scenario the recieved signals were overlayed with noise resulting in an SNR of about 10dB. The impact of that noise can be seen in Figure 5.4. The convergence of the algorithm is significantly slowed down, i.e. the estimation takes about 200 ms for reaching an interval of 10° around the actual source position. Also a clearly visible jitter in the estimation is introduced. Since the noise was added such that it represents an omnipresent noise in the scenario and the algorithm relies on the incoming signal energy, this is the expected behaviour.

One should note that the reverberation property of the room will have an impact on the estimation result. For a situation, where the speaker does not face the microphone system, but speaks in another direction, the speaker was placed at a distance of 210cm and slightly rotated, such that reflected

soundwaves arriving at the microphone system are carrying weight. As can be seen in Figure A.3 the estimation has a slight offset due to the reflected soundwaves since the most signal energy is now arriving from a slightly shifted direction.

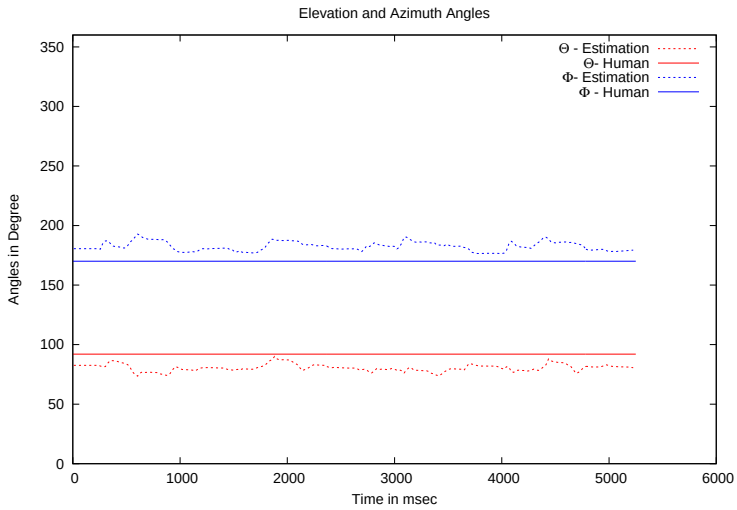


Figure 5.5: Measured estimation of speech signals with source distance of 1m

5.3.2 Speech source

For evaluation of the algorithm for speech signals the speaker was placed at a distance of 1m and at angles $\phi = 170°$ and $\theta = 95°$ and replayed a recorded speech signal at an average conversation volume level, i.e. around 50 dB(A). The estimation was recorded for about 5 seconds. One can see in Figure 5.5that the estimation does not stay as constant as in the single tone case. This can be explained by the natural pauses and different volume levels in speech.

Adding a second speech source at about $\phi = 250°$ and $\theta = 70°$ which signal energy was about 3dB lower than the desired source energy results in an estimation as it is depicted in Figure A.4. One can observe an slight offset in the direction of the additional source. The jitter towards it stays in an acceptable interval except in the pauses between the words of the original speech source, e.g. at about 900 ms, 2300ms and 4400 ms, where the estimation tends up to 60° towards the background speech direction.

To increase the difficulty for the estimation a conversation was observed, which took place at a distance of about three meters. Since the room in which the conversation took place had a width of about five meters and the speakers were neither located at the same spot nor faced the microphone system directly, the sound field of the conversation becomes a lot more diffuse

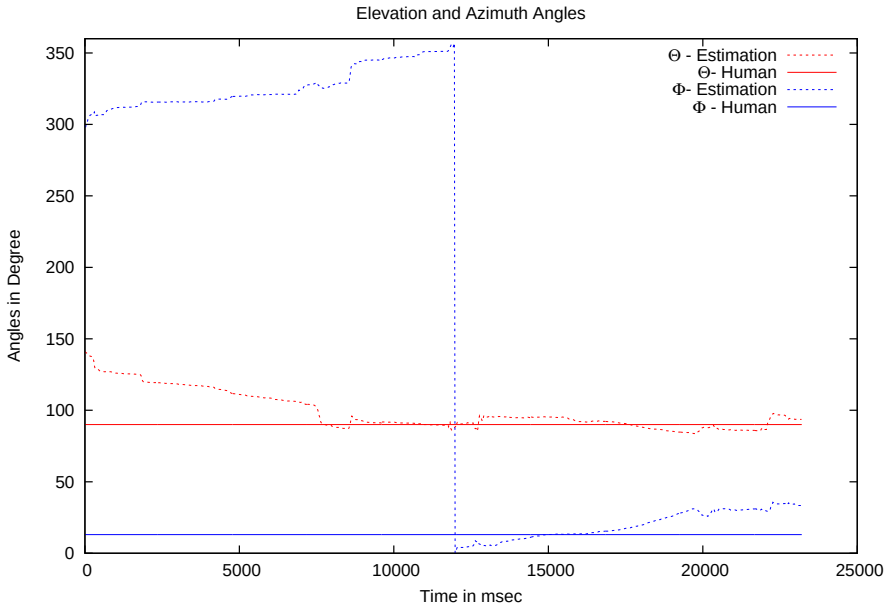


Figure 5.6: Measured estimation of conversation at a distance of about 3 meter

which results in a very slow convergence for the direction estimation, as can be seen in Figure 5.6. It takes several seconds until the estimation approximates the real direction. When increasing the distance to the conversation to about 5 meters, the convergence of the algorithm to the true direction can almost only be guessed, as Figure A.5 depicts.

One can conclude that the algorithm delivers reasonable results as long as the sound source is located sufficiently close, i.e. about 1-2 meters, to the microphone system such that the sound field does not become too diffuse. Also additional sound sources which sound energies are not neglegtible small will introduce unwanted offsets in the estimation.

5.4 Moving source

For comparing the observerable position of a moving source with the estimated position of source, the Ladybug camera system was used. The observable source position was tracked by a human operator with the cursor, which position was converted into spatial coordinates, transformed according

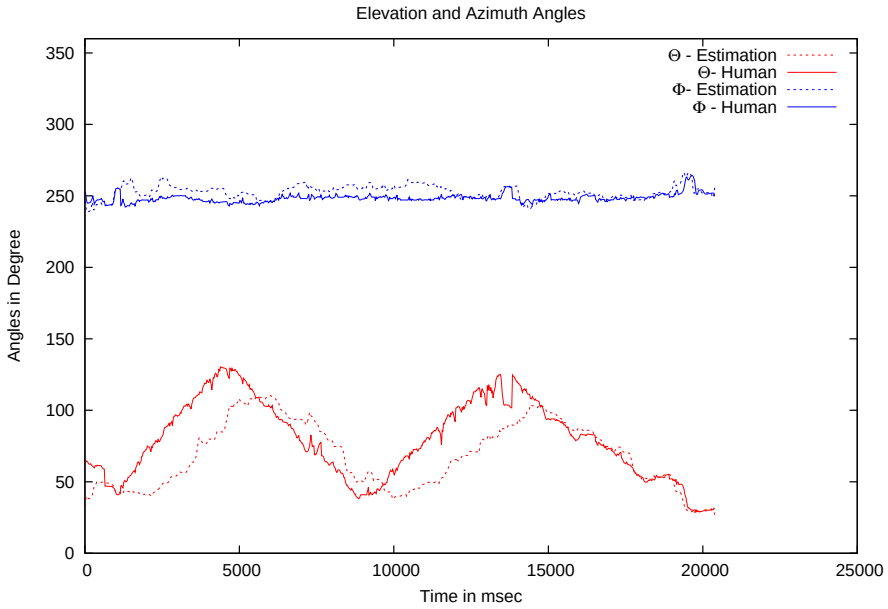


Figure 5.7: Measured vertical movement of the source

to section 5.2 and stored together with the microphone estimation.

One has to note that with this method some delays for the tracking of the human operator are introduced. The human operator can only react to the image that is displayed on the screen, which had to be processed first. Also the human reaction time has to be taken into account.

In the following figures one will see that the estimation will still 'chase after' the human observation, i.e. the natural delay introduced by the human observer is still smaller than the estimation delay of the algorithm.

Two different movement patterns will be discussed: the horizontal movement and the vertical movement where the source's position is changed by a person carrying the speaker around the microphone system, where the speaker was tried to face the microphone system at all times. Figures 5.7 and 5.8 shows the estimated and the observed direction for the vertical respectively horizontal movement. In this scenarios the time until approximate convergence is not as important as movement tracking capability of the algorithm. One can identify these scenario with the situation in a conference call where the caller walks around the phone.

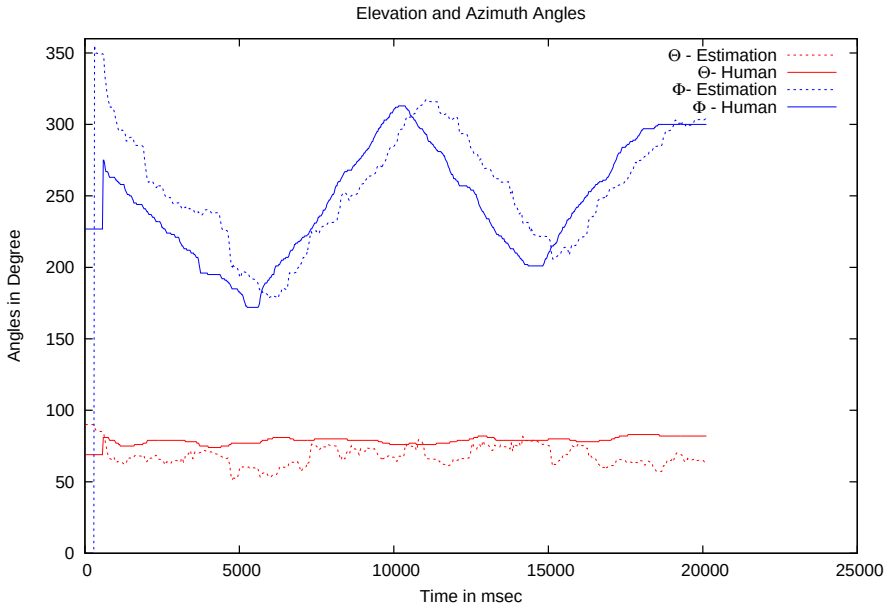


Figure 5.8: Measured horizontal movement of the source

In both scenarios, the horizontal and the vertical movement of the source, the source had a distance of about 1 meter to the microphone system and the volume was again about the volume of a normal conversation.

One can see that the estimation for the angle which is not changed, azimuth ϕ for the vertical and elevation θ for the horizontal movement, stays in a close range to the actual observed position. Note that even the observed values for these angles do not stay completely constant since the speaker was moved by a person.

The estimations for the changing angle seem to follow the observed angles quite well. The delay until the estimation reaches the observed direction fluctuates between a few hundret milliseconds and almost two seconds. Interestingly a decrease of the angle seems to be recognized much faster than an increase of the angle. This occurance was not further investigated, though.

As can be assumed from section 5.3 additional omnipresent noise would not change these observations in an recognizable way, as can be seen in figures A.6 and A.7. A second point source on the other hand would introduce an offset of the estimation to its direction and a recognizable peak in its direction every time the desired source is silent.

Chapter 6

Conclusion

This chapter concludes retrospectively the results of this thesis.

In this thesis a cost efficient, time domain source localization algorithm, that utilizes the assembly of a virtual cardioid microphone, was described.

The algorithm was simulated for simple sound signals and implemented in a real time system using the Eigenmike® microphone array and evaluated using the Ladybug®2 video camera system for moving sources.

The algorithm shows promising results for relatively close distant sources, i.e. up to two meters, and seems to be pretty robust against omnipresent noise. An evaluation where the microphone is located in the nearfield of the sound source did not to be seem reasonable since the required dipole directional patterns cannot be obtained from the microphone system in the nearfield at this point.

Additional point sources with a not negletible signal energy were identified as a problematic factor for the proposed algorithm. Taking the statistics of the source with the highest energy into account might be an aspect for future work.

Appendix A

Additional Figures

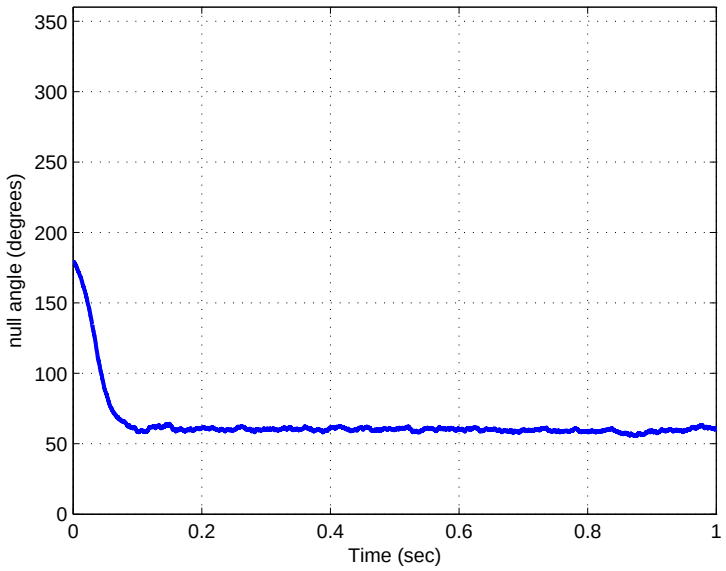


Figure A.1: Simulated estimation for 1kHz sine at 60 degree, 10 dB SNR, step size 0.03

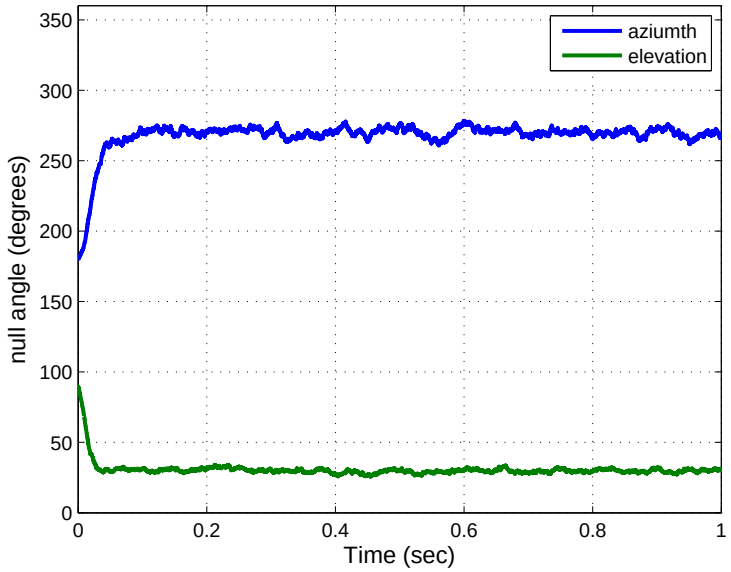


Figure A.2: Simulated 3D estimation for 1kHz sine at (270,30), 10 dB SNR, step size 0.03

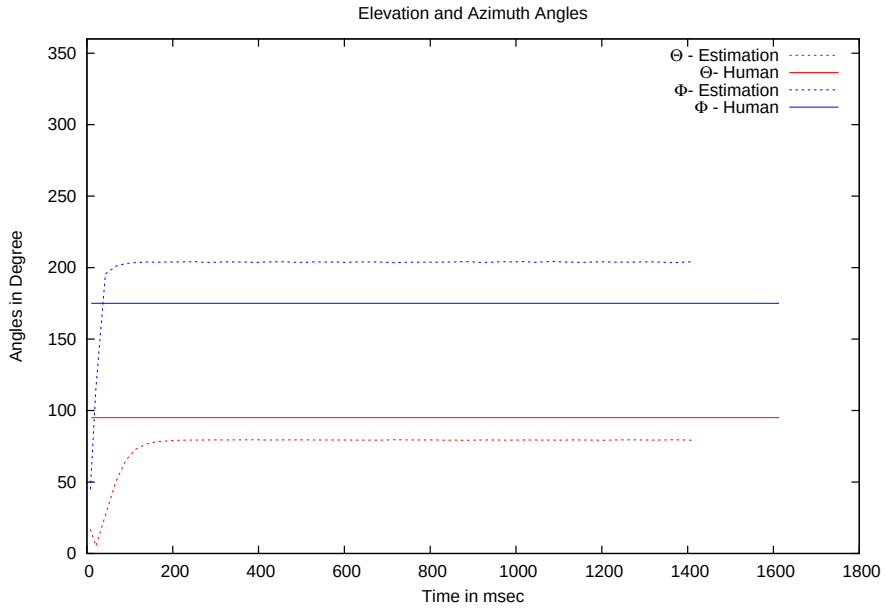


Figure A.3: Measured estimation of a 440 Hz tone from a rotated speaker at a distance of 210cm

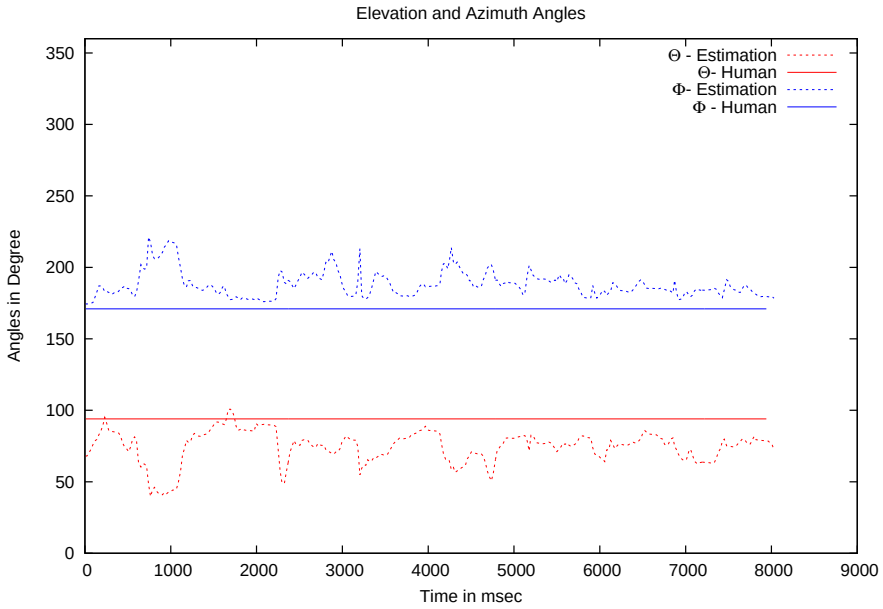


Figure A.4: Measured estimation of speech signals with source distance of 1m with a second source

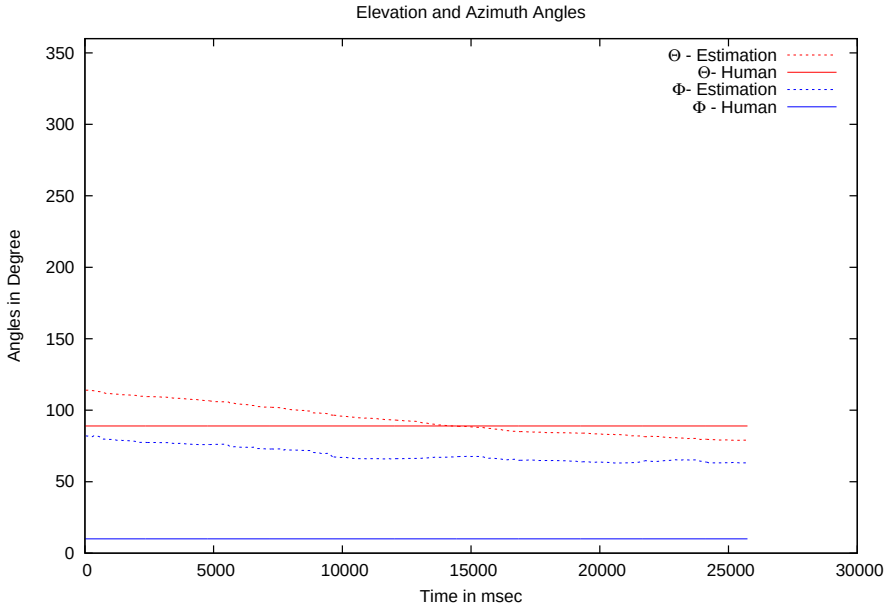


Figure A.5: Measured estimation of conversation at a distance of about 5 meter

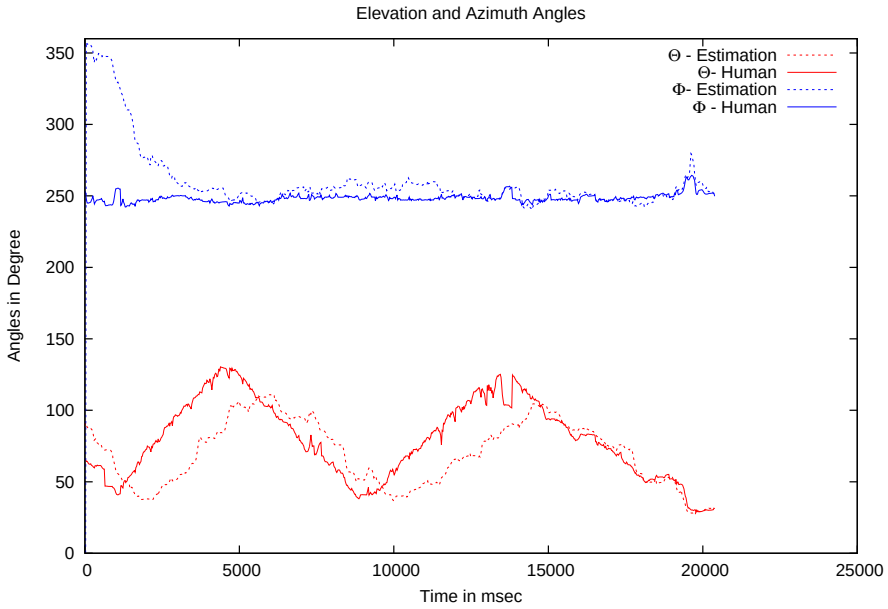


Figure A.6: Measured estimation of vertical movement at a SNR of about 10 dB

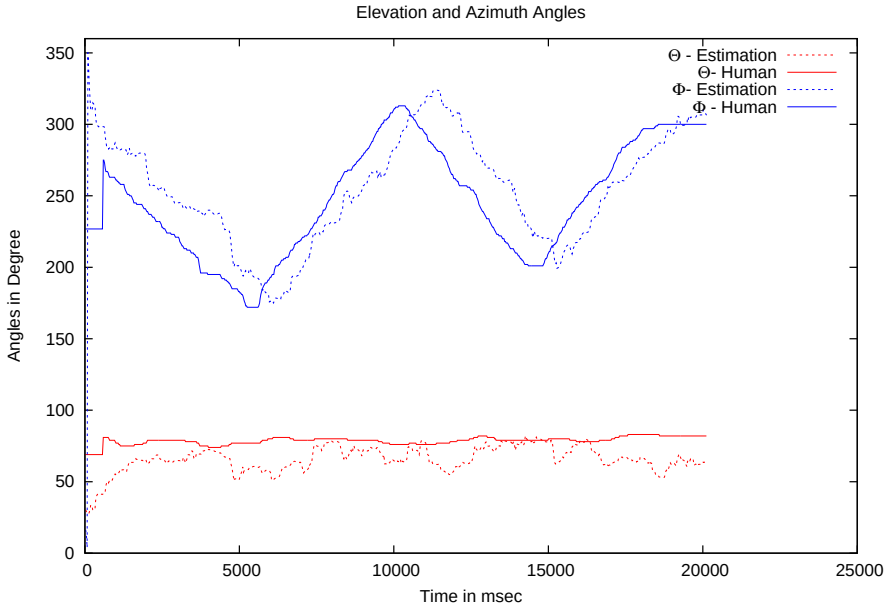


Figure A.7: Measured estimation of horizontal movement at a SNR of about 10 dB

List of Figures

Bibliography

[1] G.W. Elko. A simple adaptive first-order differential microphone. In *DARPA, Air Coupled Acoustic Microsensors Workshop*, 1999.

[2] G.W. Elko. Steerable microphone array system with a first order directional pattern. August-24-2011.

[3] G.W. Elko. Steerable and variable first-order differential microphone array. March-21-2000.

[4] G.W Elko and A.T.N Pong. A steerable and variable first-order-differential microphone array. *ICASSP-97, 1997 IEEE International Conference on Bd. I*, pages 223–226, IEEE 1997.

[5] G. Merziger, G. Mühlbach, Wille D., and Wirth T. *Formeln + Hilfen, Höhere Mathematik*. Binomi Verlag, 6th edition, 2010.

[6] M.L. Seltzer. *Microphone Array Processing for Robust Speech Recognition*. PhD thesis, Garnegie Messon University, Pittsburgh, USA, 2003.

[7] D. Ward and M. Brandstein. *Microphone Arrays: Signal Processing Techniques and Applications*. Springer, 2001.