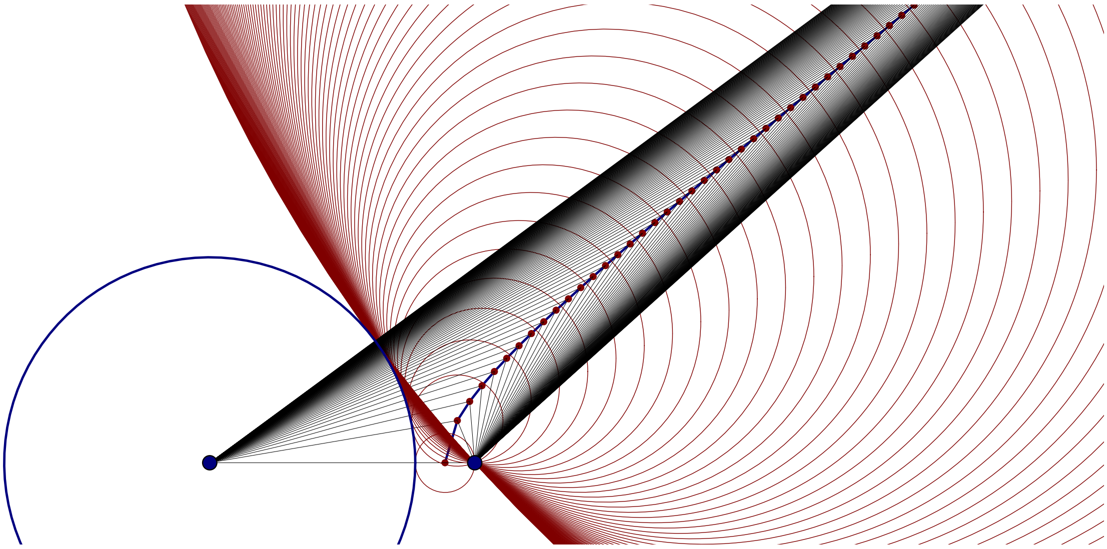


Aalborg University

Acoustics and Audio Technology

Locating Acoustic Sources with Multilateration

Applied to Stationary and Moving Sources

Author:
Daniel Dalskov

Supervisor:
Søren Krarup Olesen

June 4, 2014

Department of Electronic Systems
Acoustics
Fredrik Bajers Vej 7
Phone: +45 9940 8710
http://www.es.aau.dk/sections/acoustics

Title:
Locating Acoustic Sources with Multilateration - Applied to Stationary and Moving Sources

Topic:
Master's thesis

Group:
14gr1062

Participants:
Daniel Dalskov

Supervisor:
Søren Krarup Olesen

Pages:
76

Appendices:
4

Project period:
Spring Semester 2014

Handed in:
June 4, 2014

Synopsis:

The main topic of this report is locating acoustic sources. The underlying theory used to created the locating method, is based on multilateration, which uses multiple sensors and their TDOA (Time Difference Of Arrival) to estimate a unique position. The resulting method is able locate stationary sources as well as moving source, as long as an anechoic environment is used. A hardware platform has been developed to provide the possibility of online execution. The platform is based on a Raspberry Pi, a Wolfson Audio Card and a preamplifier circuit for the microphones. An issue with communication between the Raspberry Pi and the Wolfson Audio Card, limited the number of input channels to 2, instead of the 3 required 2D location. This confines the method to only determining the angle of the source, while using this platform.

Foreword

This report is written by Daniel Dalskov at the 4th semester of Acoustics and Audio Technologys master program under the Department of Electronic Systems at Aalborg University. It was developed in the period from the 1st of February to the 4th of June 2014. The theme of the report is locating acoustic sources, using the method multilateration. The project is proposed and supervised by Søren Krarup Olesen.

A project CD is attached to the report, containing a code example, results of measurements, datasheets and selected references.

Daniel Dalskov

Contents

Chapter 1

Introduction

This first chapter will begin with a description of the topic locating sound, what the project will focus on and present some use cases where locating sound may be helpful. Then an investigation of methods, used for locating sound or other sources with similar characteristics is made. This investigation will lead into the assembly of a method which will provide a solution.

The second chapter will look into the theory of how to locate a stationary source, first by determining the direction and then by finding a 2D position. Along with the theoretical description, both simulations and measurements are carried out in order to investigate and show the properties of the method and verify the basis of the theory.

The third chapter will expand on the second by locating a moving source and thereby adding a time dependence to the method. Simulations will again be used to show the performance of the method.

The fourth chapter will show the design of a dedicated hardware platform and the implementation of the method thereon, which will involve the interaction between the high-level language Python and the hardware.

The last chapter will contain the conclusion of the project and a discussion containing recommendations for further work and a description of where alternative choises could alter the outcome.

1.1 Project Description

The overall goal of the project is to locate an arbitrary sound source in an arbitrary environment. It is however unlikely to achieve this within the time frame of a semester. The more moderate goal to be met will be locating a talking person or a source with similar acoustic characteristics in an ideal envionment. The method will however be generalised as much as possible, so a generic use may be realised. The selection of type of sound source gives certain benefits regarding the frequency spectre and range in which to analyse, but also for how fast a person moves in the case of locating a moving source.

Efficient solutions such as microphone-arrays already exist, except the number of microphones used are often high, giving the soution a higher cost and putting high requirements on the processing hardware. The algorithms can often be crued which further increases the requirements. Another goal will be to use a limited amount of processing power and number of microphones. There will also be put emphasis on using analytical opposed to nummerical solutions where possible.

The milestones included in reaching the overall goal is first to establish a reliable method, which can produce a sufficiently accurate result in an ideal environment, with little noise, no wall-reflections and no other sound sources. To begin with, only the angle is found, the method is then expanded to a 2D position. The last part will be for the method to continuously track the postion as a person sometimes moves around while talking.

1.1.1 Possible Applications

This method is applicable in various situations when e.g. conducting a conference. First of all the current speaker can be located and perhaps identified, this can be added to a transscript or a recording of the conference. Once the speaker is identified, a camera could be rotated in his/her direction and zoom in for a video recording. Aside from recording and storage of audio and video, the method can be used in live tele-conferencing for synthesising a virtual sound source at a given position in the receiving room. If the speaker walks towards the white-board in the transmitting room, the sound source will also move towards the white-board the receiving room. Reseach into a similar project such as the Beaming project by [Madsen et al., 2011]. In the summary it is written: *"The Position and movements of participants, particularly the head, are tracked and from this sound is rendered to include binaural cues so the visitor is able to move around in a limited space while perceiving Destination sound as 'stationary'."*

1.2 Positioning of Audio in General

In this section four of the more known methods for locating a sound source will be described. These four methods are: multilateration, trilateration, binaural recording and beamforming.

1.2.1 Multilateration

Multilateration uses time-difference of arrival (TDOA) to determine likely positions of the source for multiple sensor pairs. The TDOAs are illustrated as the dashed lines in Figure 1.1. TDOA describes the time difference between the arrival of the signal to one sensor and another. This difference will be the same for all source locations along a hyperboloid and in order to find the position of the source in a 2D scenario, another sensor pair is required. The position of a source can be determined by the point where the hyperbola from the first sensor pair and the hyperbola from the second sensor pair intersect. It should be noted that one microphone from each of the two sensor pairs can be the same microphone, but the microphones must be distributed in both dimensions for a completely unique solution. Additionally it is possible to use a third hyperbola between the first and third microphone, which will intersect the two first hyperbolas at the same point, allowing for higher accuracy. For a unique solution in the 3D case, four microphones distributed in all three dimensions are required. [Gustafsson and Gunnarsson, 2003] and [Tellakula, 2007] describe the method in more detail and some of the complications related to the method.

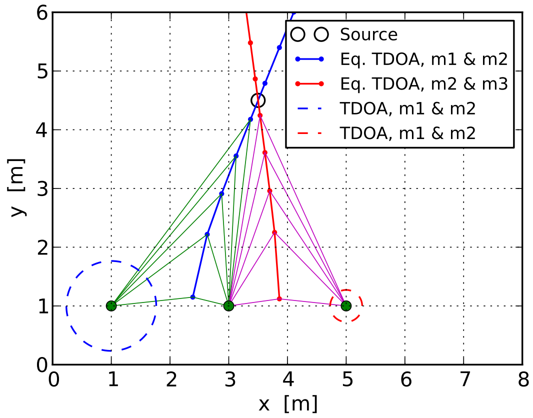


Figure 1.1: Multilateration - The red and blue curves show the described hyperbola and the point of intersection at the source (Only the positive part of the lines are shown). The dashed circles illustrates the TDOA.

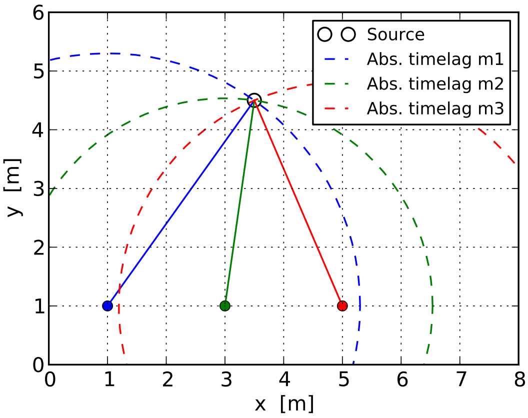


Figure 1.2: Trilateration - The solid lines show the absolute distance traveled by the signal. The dashed circles show the possible source locations for each sensor. The intersection of the three circles show the source position. (Two results are given because the sensors are on a line, but only the positive result is shown)

1.2.2 Trilateration

Trilateration relies on knowing the original signal or the time of departure to calculate the absolute distance from two or more reference points. The distance is aquired from the absolute difference between time of departure and time of arrival. The difference is then used to draw the circles shown in Figure 1.2. This results in two or more circles depending on the number of sensors, which positions the source at the intersection of the circles. For the 2D case using two sensors results in two intersections, the two possible positions can be narrowed down by describing a valid domain. Using three sensors will give only one result as long as the source is located in the same plane as the sensors, for a 3D case the three sensors will give two result but can again be narrowed down as in the 2D case or by adding a fourth sensor. [Jiménez and Seco, 2005] shows the use of trilateration for precise positioning in archaeology.

1.2.3 Binaural Recording

The binaural recording technique uses dummy heads as in Figure 1.3, with artificial pinnas in order to mimic the way humans localize audio. Humans use difference cues to distinguish the direction and distance from which the sound is originating. As for the direction, the major cues are interaural level difference (ILD), interaural time difference (ITD) and monaural cues. The ILD and ITD are used to determine azimuth (from which horisontal angle it is originating), by looking at the difference in level (ILD) from one ear to the other and difference in time of arrival between the two ears (ITD).

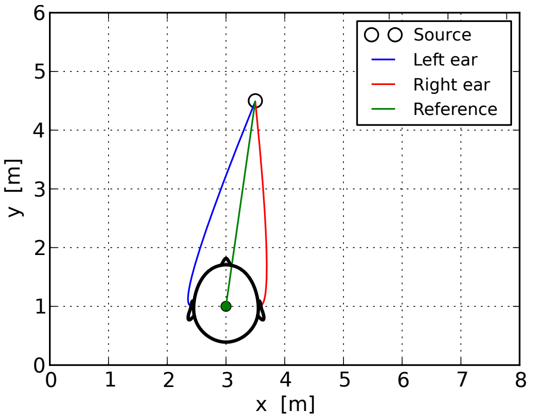


Figure 1.3: Binaural recording - The blue and red lines show the pathway of the sound to the left and right ear respectively. The green line shows the reference, from which the positions will be relative to.

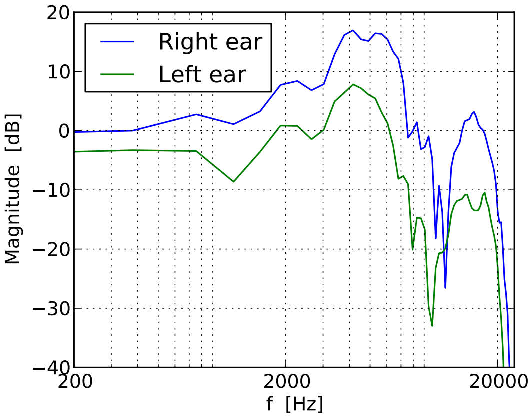


Figure 1.4: Binaural recording - The frequency response measured at the right and left ear of a valdemar dummy head with a sound source at 30° to the right at 2 m distance. The general shape of the response is determined by the monaural cue and the level difference is the ILD cue.

The monaural cue determines the elevation (from which vertical angle it is originating) by looking at the spectal shape of the sound. The sound is reflected by the pinna and shoulders, changeing the spectral shape the direction is changed. At some directions, these cues can be inconclusive, but a small turn of the head will result in a change of the cues, which is enough to remove the ambiguity. The distance cues are a lot more complicated and among other things rely on the reverberation of the room or surroundings, which means it is based on experience with the room and the sound in question. Algorithms have been developed to try and mimic the behavior of humans as described by [Bronkhorst and Houtgast, 1999] and [Colburn, 1996], but it is outside the scope of the project.

1.2.4 Beamforming

Beamforming is one of the prefered methods in sensor arrays and supports the use of many inputs, giving high angular accuracy. Traditional beamforming disregards distance and assumes plane waves, which is only possible when the source is far away from the sensor array. The simplest form of beamforming is the delay-and-sum version where a certain direction corresponds to a specific time difference between two sensors. If the signal from the first microphone has been delayed by that timelag and the two signals are sumed up, they will double in strength, whereas the sound coming from all other directions will be diminished. The method can sweep all the possible angles in predetermined steps and finds the maximum signal strength. The found time difference is then converted to its corresponding direction as the result. In order to find an azimuth, the sensors should be located on an axis, and can be positioned in various patterns. These patterns could be linear, logarithmic or semi-randomly distributed depending the purpose, but the position must always be known.

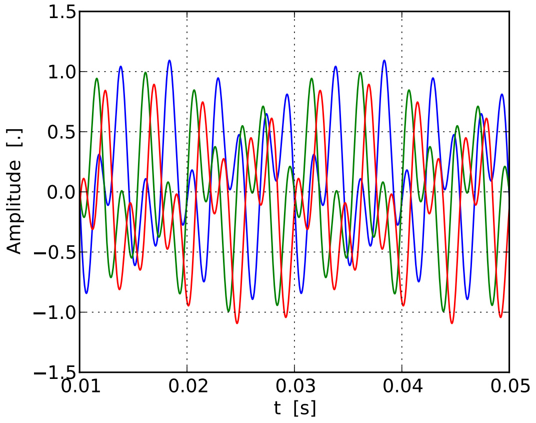


Figure 1.5: Beamforming - A signal recorded at each sensor with a small offset for easier discrimination. It can be seen that the individual signals have different offsets.

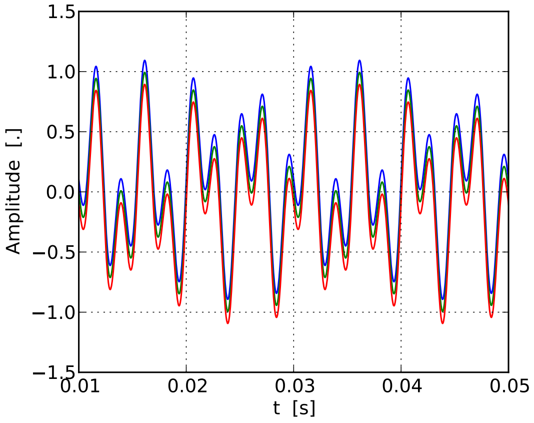


Figure 1.6: Beamforming - The same signals as in Figure 1.5 but the signal at microphone 1 has been moved to the left by 2.23 ms and at microphone 3 by 0.79 ms.

To determine both an azimuth and elevation, a two dimentional array is required, which again can be positioned in different patterns. These patterns include x-shape, circle, square, circular-grid, square-grid, spiral and semi-random distribution. [Johnson and Dudgeon, 1992]

1.3 Discussion of Methods

Each of these methods use different techniques to locate a source using two sensors or more. Some of the methods required more complicated analysis of the aquired data, which is beside the purpose of this project, where a simplere method is prefered. Trilateration calculates in absolute distances which in this case is not possible as the signal at the source is unknown. Binaural recording requires the use of a dummy head which is large and impractical, moreover the analysis required to determine the elevation from the monaural cue and the distance from a reverberant environment requires far too much processing time and power. Beamforming discards the distance to the source and is most often used with large arrays and can have a large computational complexity which also is beside of the scope of this project.

That leaves multilateration, which does not require information from the source other than what is received through the sensors. With an ideal noise-free signal, a precise position of the source can be obtained. If however the signal has little energy, is noisy or the distance to the source is much larger than the separation of the microphones, the result may become inaccurate and require extra consideration.

Chapter 2

Locating a Stationary Source

This chapter will focus on locating a stationary source. First the theory is derived of how a direction of the source may be obtained. This theory is then expanded to obtaining a 2D position. Some of the issues and limitations involved in using the theory will be explained during the development to provide a better understanding of the method. Simulations are used to verify the theory and identify further problems that may arise in the physical measurements. As a conclusion to this chapter on locating stationary sources the performance of the method in a physical envionment is measured analysed and compaired with the simulations.

2.1 Direction of a Stationary Source

In this section only the direction will be determined and for that, two microphones are required. Figure 2.1 show an overview of the principle. The two microphones pick-up the sound emitted by the source, which is then sampled by a soundcard and send for processing. The processing consist of a method to extract the timing information from the two signals, which is then compaired by subtracting them from eachother resulting in a difference in time (TDOA). This TDOA can be used to describe a hyperbola which will go through the position of the source. The hyperbola in itself does not provide a direction, but the asymptote of the hyperbola will. The difference between the hyperbola and the asymptote is that the former assumes the waves to be spherical or circular and the latter assumes that the waves are plane.

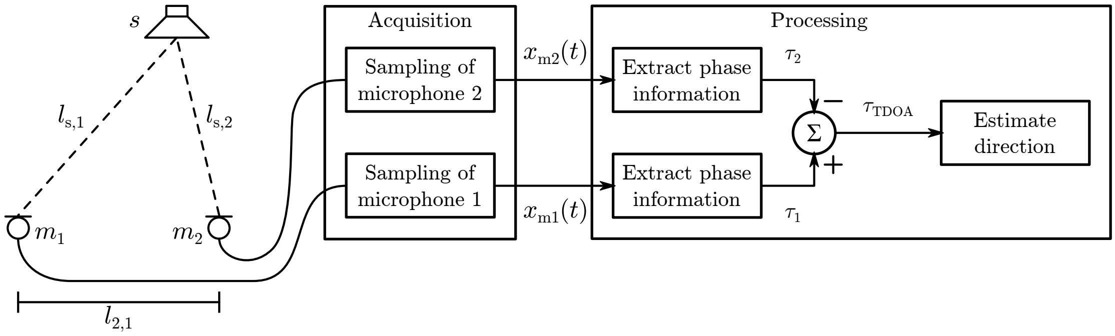


Figure 2.1: Illustration of the method as an overview of the main components.

2.1.1 Assumptions

In the case of two microphones where only the angle is found, the following assumptions apply:

- Plane sound waves
- Anechoic and noise-free environment
- Phase-matched, linear and noise-free equipment
- No other sources are pressent

It is required that the sound waves are plane, because the direction would otherwise become inaccurate. As it can be seen in Figure 2.2 when the separation of the microphones is decreased, the distance to the source has less influence on the accuracy of the direction. This is a result of the section of the spherical waves seen by the microphones is so small that it behaving more like plane waves. The waves become plane when the source is adequately far away compaired to the separation of the microphones.

The asymptote will always go through the center of the two microphones and point in the direction of the source. When the source rotates, the asymptote rotates along with it and the angle of the asymptote can therefore be used to tell the angle of the source relative to the axis of the microphones.

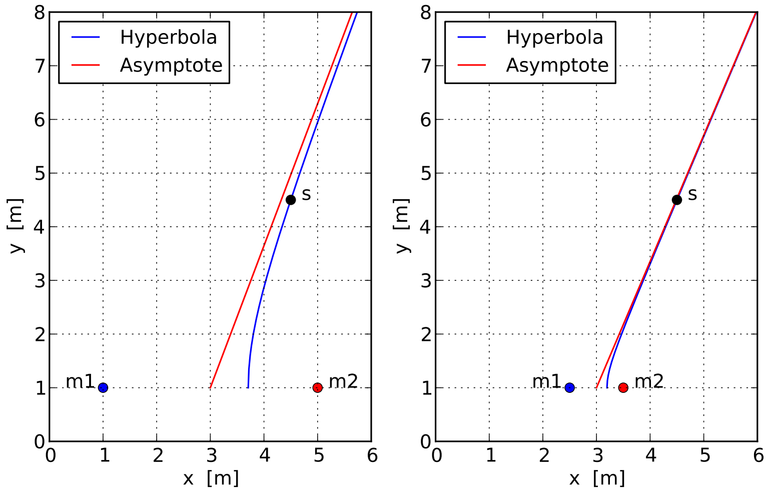


Figure 2.2: Exaggerated illustation of the asymptote in two cases with equal parameters except for the microphone separation, which in the lefthand side is 4 m and in the righthand side 1 m.

Ideal envionment and equipment is used to show the best case scenario, the envionment should be anechoic so that no reflections are picked up by the microphones. It is assumed that there are no other sources than the expected and that it is constantly active.

2.1.2 Calculation of TDOA

There are more than one way of calculating the TDOA from the two signals. One intuitive and commonly used way is the general cross-correlation (GCC) method, where the two signals are cross-correlated and the argmax of the correlation will be the offset in time. An unfortunate problem with the GCC method is that a high sampling rate and large microphone separation is required to get a good angular resolution, especially for angles close to the axis of the microphones. If e.g. the sampling rate is too low, the angle will be quantized into small steps when perpendicular to the microphone axis and getting larger and larger as the the angle approaches the microphone axis.

An alternative to the GCC is calculating a complex Fourier transform of the signal and extract the phase information. As the phase describes a frequency dependent offset or delay, it is possible to convert it to a time value which will have a much better resolution largely independent of the sampling rate, as opposed to the GCC.

To calculate the TDOA an analysis of the wave propagation from the source the microphones is made. The theory used is based on [Kinsler, 2000]. Let the signal at the source be given as a sinusoidal signal in the euclidean form:

$$\begin{aligned} x_0(t) &= A_0 \mathrm{e}^{j(\omega_0 t + \theta_0)} \\ \omega_0 &= 2\pi f_0 \end{aligned}$$

with amplitude A_0, frequency f_0 and phase θ_0. The signal is picked up at microphone 1 with a delay τ_1 and a lowered amplitude A_1 corresponding to the distance from the source to the microphone:

$$\begin{aligned} x_1(t) &= x_0(t-\tau_1) = A_1 \mathrm{e}^{j(\omega_0(t-\tau_1)+\theta_0)} \\ &= A_1 \mathrm{e}^{j(\omega_0 t-\omega_0\tau_1+\theta_0)} \\ \tau_1 &= \frac{l_{s,1}}{c} \end{aligned}$$

where c is the speed of sound, $l_{s,1}$ is the distance from the source to the microphone and $-\omega_0\tau_1$ can be seen as a negative shift in phase and will be denoted as θ_1

$$= A_1 \mathrm{e}^{j(\omega_0 t+\theta_0+\theta_1)} \qquad \theta_1 = -\omega_0\tau_1$$

The Fourier transform of $x_1(t)$ is:

$$\begin{aligned} X_1(\omega) = \mathcal{F}\{x_1(t)\} &= \int\limits_{-\infty}^{\infty} x_1(t)\mathrm{e}^{-j\omega t}\mathrm{d}t \qquad \omega = 2\pi f \\ &= \mathrm{e}^{j(\theta_0+\theta_1)}\sqrt{2}A_1\delta(\omega-\omega_0) \end{aligned}$$

where $\delta(\omega)$ represents the Dirac delta unit pulse. The magnitude of $X_1(\omega)$ is a pulse of $\sqrt{2}A_0$ at $\omega = \omega_0$ and the angle contain the phase information:

$$\begin{aligned} |X_1(\omega)| &= \sqrt{2}A_1\delta(\omega-\omega_0) \\ \angle X_1(\omega) &= \theta_0+\theta_1 = \theta_0-\omega_0\tau_1 \end{aligned}$$

It should be noted that:

$$\pi \geq \theta_0+\theta_1 > -\pi$$

so for any angle above π, it is wrapped around to $-\pi$ and vice versa. It is therfore impossible to find τ_1 from the angle as the initial phase of the signal at the source θ_0 is unknown. This problem can however be solved by including the signal of the second microphone:

$$\begin{aligned} x_2(t) &= x_s(t-\tau_2) = A_2\mathrm{e}^{j(\omega_0 t+\theta_0+\theta_2)} \qquad \theta_2 = -\omega_0\tau_2 \\ X_2(\omega) &= \mathcal{F}\{x_2(t)\} = \mathrm{e}^{j(\theta_0+\theta_1)}\sqrt{2}A_2\delta(\omega-\omega_0) \\ |X_2(\omega)| &= \sqrt{2}A_2\delta(\omega-\omega_0) \\ \angle X_2(\omega) &= \theta_0+\theta_2 \end{aligned}$$

By subtracting the angle of X_2 from X_1, the inital angle is removed:

$$\begin{aligned}\angle X_1(\omega) - \angle X_2(\omega) &= (\theta_0 + \theta_1) - (\theta_0 + \theta_2) \\ &= \theta_1 - \theta_2 \\ &= -\omega_0\tau_1 + \omega_0\tau_2 \\ &= \omega_0(\tau_2 - \tau_1)\end{aligned}$$

where $\tau_2 - \tau_1$ is the TDOA

$$= \omega_0\tau_{\text{TDOA}}$$

and can be found as:

$$\tau_{\text{TDOA}} = \frac{\angle X_1(\omega) - \angle X_2(\omega)}{\omega_0}$$

It is important to remember that:

$$\pi \geq \theta_1 > -\pi \qquad \text{and} \qquad \pi \geq \theta_2 > -\pi$$

which means the phase difference will be:

$$2\pi \geq \omega_0\tau_{\text{TDOA}} > -2\pi$$

This will also result in large abrupt changes in the phase when the frequency is changed. This excess phase needs to be be removed, otherwise the TDOA will suffer from the same issues. This can done by wrapping around whenever the $|\theta_1 - \theta_2|$ exceeds π. However a more efficient way is to find the angle after dividing the two complex Fourier transforms:

$$\begin{aligned}Y(\omega) &= \frac{X_1(\omega)}{X_2(\omega)} \\ &= \frac{\mathrm{e}^{j(\theta_0+\theta_1)}\sqrt{2}A_1\delta(\omega-\omega_0)}{\mathrm{e}^{j(\theta_0+\theta_2)}\sqrt{2}A_2\delta(\omega-\omega_0)} \\ &= \frac{A_1\mathrm{e}^{j(\theta_0+\theta_1)}}{A_2\mathrm{e}^{j(\theta_0+\theta_2)}} \\ &= \frac{A_1}{A_2}\mathrm{e}^{j(\theta_0+\theta_1)-j(\theta_0+\theta_2)} \\ &= \frac{A_1}{A_2}\mathrm{e}^{j(\theta_1-\theta_2)} \\ |Y(\omega)| &= \frac{A_1}{A_2}\end{aligned}$$

and if the two amplitude A_1 and A_2 are assumed to be equal

$$\begin{aligned}|Y(\omega)| &= 1 \\ \angle Y(\omega) &= \theta_1 - \theta_2\end{aligned}$$

where:

$$\pi \geq \angle Y(\omega) > -\pi$$

The advantage of this second method is that no wrapping is required as wrapping can be very time consuming. Extracting the TDOA from the phase is a simple matter of dividing with the angular frequency ω_0:

$$\tau_{\text{TDOA}} = \frac{\angle Y(\omega)}{\omega_0} \tag{2.1}$$

2.1.3 Calculation of the Source Direction

As described in the beginning of this section, the TDOA gained when using two microphones will describe a hyperbola. The equation of the hyperbola is shown in Equation (2.2) and is derived in Appendix A.

$$y(x) = \frac{1}{2}\sqrt{\frac{(4x^2 - \Delta l^2)(l_{2,1}^2 - \Delta l^2)}{\Delta l^2}} \tag{2.2}$$

where $l_{2,1}$ is the microphone seperation and Δl is the length corresponding to the TDOA:

$$\Delta l = \tau_{\text{TDOA}} \cdot c$$

Two examples of the hyperbolas are shown in Figure 2.2. The x-axis of the hyperbola will always be parallel with the axis running through both microphones and have zero at the center of the microphones. The hyperbola describe all the possible positions of a source with the given TDOA and is in itself not enough to get the angle of the source. If the sound waves from the source are plane waves, the direction of propagation can be described by the asymptote of the hyperbola and the slope of the asymptote will then describe the direction relative to the axis of the microphones. The asymptote is described by Equation (2.3) which is derived in Appendix A.

$$y(x) = x\frac{\sqrt{l_{2,1}^2 - \Delta l^2}}{\Delta l} \tag{2.3}$$

where the angle of the asymptote is:

$$\theta_a = \tan^{-1}\left(\frac{\sqrt{l_{2,1}^2 - \Delta l^2}}{\Delta l}\right) \tag{2.4}$$

With Equation (2.4) it now possible to determine the angle the source, concluding the calculation of the source direction.

2.1.4 Frequency and Physical Limitations

The method for calculating the TDOA described in Section 2.1.2 is valid for all frequencies that have a wavelength above $2 \cdot l_{2,1}$ or put in another way:

$$f_{max} < \frac{c}{2 \cdot l_{2,1}} \tag{2.5}$$

The reason for this limitation is that the maximum absolute phase difference occurs when the source is located on the x-axis to either side of the microphones. At the maximum frequency, the wavelength will be equal to $2 \cdot l_{2,1}$ resulting in a phase of $+\pi$ or $-\pi$. If a frequency above this limit is used, the phase will wrap around and the calculated source direction will be opposite the actual direction for the that frequency.

If e.g. the maximum frequency is set to 3.3 kHz, which is often viewed as the upper frequency limit for what is required to understand speech, the microphone separation is required to be at maximum 5.2 cm. This imposes a problem as the closer the microphones are placed, the less accurate the direction becomes at larger distances. Small pertubations in the TDOA can cause large fluctuations in the direction, which is shown in Figure 2.3. The microphone separation is here set to 0.5 m and hyperbolas has been plottet for three individual TDOAs with a range of ±50 μs. As a reference, 1 sample at 48 kHz is 20.8 μs.

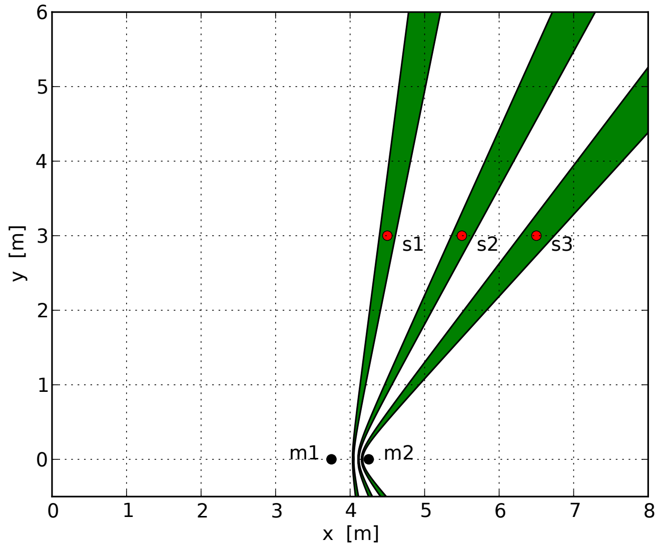


Figure 2.3: Uncertainty in direction for three different source positions. The green area between the black lines illustrate where the hyperbolas could end up with an error up to ± 50 μs. The microphone separation is set to 0.5 m.

In order to increase to accuracy when small fluctuations occur, the TDOA is calculated for several frequencies and the mean of these TDOAs will provide a more steady result.

2.2 Position of a Stationary Source

The method for determining the direction of a stationary source, described in Section 2.1, will here be expanded to finding a 2D position.

2.2.1 Assumptions

The assumptions used in determining a 2D position are largely the same as for the direction, but with one exception, namely that the sound waves are not required to be plane but spherical:

- Spherical sound waves
- Anechoic and noise-free environment
- Phase-matched, linear and noise-free equipment
- No other sources are pressent

The sound waves are no longer required to be plane, as the hyperbola itself will be used to determine the position, instead of the asymptotes. The best case scenario is still desirable and so the remaining assumptions apply.

2.2.2 Calculation of the Source Position

The determination a 2D position is based on the intersection of two hyperbolas. Another hyperbola is therefore required, and also another microphone. The use of three microphones will not only provide the possibility of two hyperbolas, but three. There will be one for each microphone pair; 1-2, 2-3 and 3-1. These three hyperbolas will ideally intersect in the same position, namely that of the source.

The implementation of a third hyperbola can however prove somewhat difficult. As explained in Section 2.1.3 the x-axis of the hyperbola is described by the axis of the microphones. If the microphone-axis of one pair is rotated relative to the second pair, one of the hyperbolas must be rotated in order to be in the same coordinate system. With an emphasis is on an analytical solution, this will be very difficult when the angular difference between the two microphone-axes are not integer multiplications of 90°.

Shown in Figure 2.4 are two kinds of hyperbolas; East-West opening hyperbolas, shown as the green curves and North-South hyperbolas shown as the blue curves.

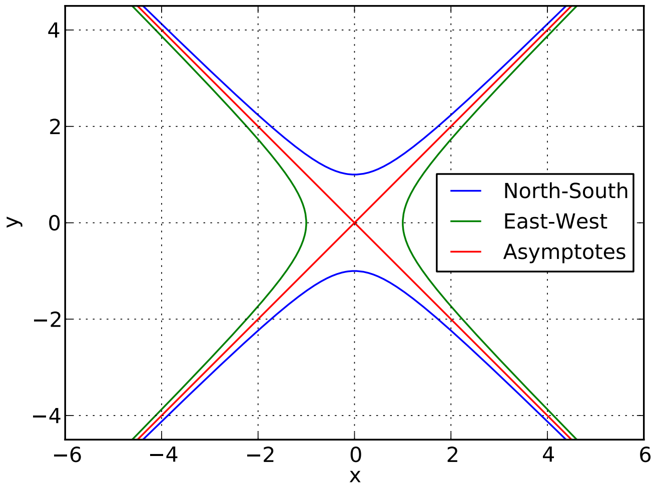


Figure 2.4: Two hyperbolas with coefficients $a = 1$ and $b = 1$.

The equations of the two cases can be seen in (2.6) and (2.7) respectively. Figure 2.4 is of course a parametric representation of the equations as more than one y-value is present for each x-value. The East-West opening hyperbola is the one used when the microphone-axis is parallel to the x-axis. If this hyperbola is to be rotated 90°, the inside of the square-root in Equation (2.6) is multiplied by -1 as it has been done in Equation (2.7), resulting in a North-South hyperbola.

$$y_{ew}(x) = \sqrt{\frac{\frac{x^2}{a^2} - 1}{b^2}} \tag{2.6}$$

$$y_{ns}(x) = \sqrt{\frac{1 - \frac{x^2}{a^2}}{b^2}} \tag{2.7}$$

Arbitrary rotation of the hyperbolas is less simple and require discretised vector representation and the use of a rotation matrix as shown in Equation (2.8).

$$M(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$\begin{bmatrix} x_{rot} \\ y_{rot} \end{bmatrix} = M(\theta) \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.8}$$

This method is more computationally heavy and does not guarantee an exact point of intersection. It is therefore advantageous to place the microphones parallely or orthogonally if possible, ensureing the highest precision and efficiency. The only way of placing the three microphones so that all microphone-axes are parallel or orthogonal is by having them on a line. This will however mean that the separation for the two outermost microphones will be twice the separation of the two other pairs and will halve the maximum frequency usable by that pair. If only two axes are required to be orthogonal or parallel an L-shape can be used, giving an orthogonal crossing of the axes. A different way entirely is to find the intersection of the asymptotes described by Equation (2.3). The asymptotes are straight lines and easily rotated, they will however introduce an inaccuracy in the estimates of the position, mostly for sources near the microphones. An examination of the repercussions involved with using the asymptotes can be carried out if this approach is later deemed necessary. It is instead decided to focus the attention on obtaining the position using the intersection of the first two hyperbola with the microphones placed on a line and omit the implementing the third hyperbola.

The intersection of the two hyperbolas can be found as two equations with two unknowns. It is here done by setting one hyperbola equal to the other and isolating the x-coordinate.

$$y_1(x) = \frac{1}{2}\sqrt{\frac{\left(4\left(x-\left(m_x-\frac{l_{2,1}}{2}\right)\right)^2-\Delta l_1^2\right)(l_{2,1}^2-\Delta l_1^2)}{\Delta l_1^2}} \tag{2.9}$$

$$y_2(x) = \frac{1}{2}\sqrt{\frac{\left(4\left(x-\left(m_x+\frac{l_{2,1}}{2}\right)\right)^2-\Delta l_2^2\right)(l_{2,1}^2-\Delta l_2^2)}{\Delta l_2^2}} \tag{2.10}$$

The $(m_x \pm \frac{l_{2,1}}{2})$ is used to place the $x = 0$ of the two hyperbolas at the center of the individual microphone pairs, m_x represents the x-coordinate of the center microphone.

$$\begin{aligned}
y_1(x) &= y_2(x) \\
\frac{1}{2}\sqrt{\frac{\left(4\left(x-\left(m_x-\frac{l_{2,1}}{2}\right)\right)^2-\Delta l_1^2\right)(l_{2,1}^2-\Delta l_1^2)}{\Delta l_1^2}} \\
&= \frac{1}{2}\sqrt{\frac{\left(4\left(x-\left(m_x+\frac{l_{2,1}}{2}\right)\right)^2-\Delta l_2^2\right)(l_{2,1}^2-\Delta l_2^2)}{\Delta l_2^2}}
\end{aligned}$$

From here x is isolated:

$$\begin{aligned}
x =& \frac{-2\cdot\Delta l_1^2\cdot\Delta l_2^2\cdot l_{2,1}+2\cdot\Delta l_1^2\cdot l_{2,1}^2\cdot m_x+\Delta l_1^2\cdot l_{2,1}^3-2\cdot\Delta l_2^2\cdot l_{2,1}^2\cdot m_x+\Delta l_2^2\cdot l_{2,1}^3}{2\cdot l_{2,1}^2\cdot(\Delta l_1^2-\Delta l_2^2)} \\
&\pm\frac{\sqrt{\Delta l_1^2\cdot\Delta l_2^2\cdot l_{2,1}^2\cdot(\Delta l_1^2+\Delta l_2^2-2\cdot l_{2,1}^2)^2}}{2\cdot l_{2,1}^2\cdot(\Delta l_1^2-\Delta l_2^2)}
\end{aligned} \tag{2.11}$$

The ± in the second part of Equation (2.11) is determined by the sign of $\Delta l_1 \cdot \Delta l_2$. The expression used to evaluate the x-coordinate is rather long however all the variables are determined beforehand except for Δl_1 and Δl_2, making those the only true variables. Once the x-coordinate is evaluated, it can be used to find the y-coordinate by inserting the found x into either Equation (2.9) or (2.10). This result in an estimated 2D position of the source ready for use in a variety of use cases.

2.2.3 Distance and Accuracy

In Section 2.1.4 the uncertainty of one hyperbola is shown. When using the intersection of two hyperbolas to determine the postion, the error of both the hyperbolas will have an influence. Figure 2.5 show the intersection of two hyperbolas, each with an uncertainty of ±50 μs (same error as in Section 2.1.4).

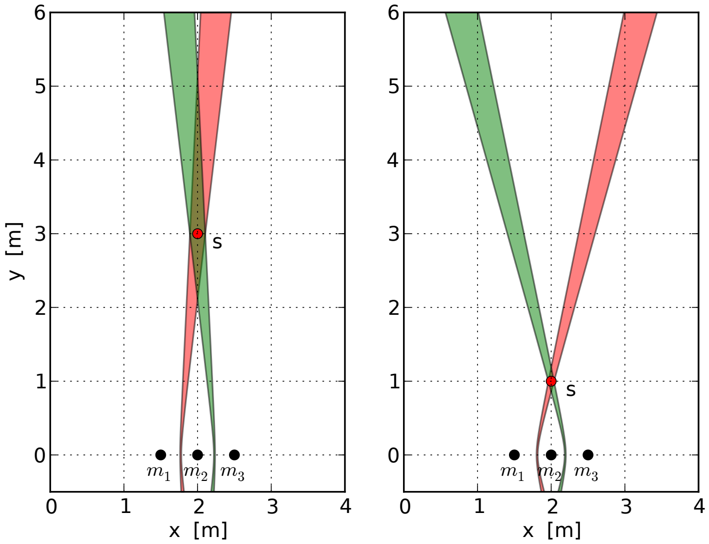


Figure 2.5: Uncertainty in the position for two different sources. The green and red area illustrate where the hyperbola could be with an error up to ±50 μs, and the intersecting brown area is where the source position would be estimated to be. The microphone separation is set to 0.5 m

There is a large difference between the size of the brown areas when the source is close and far away from the microphones. This happens because the further out the source is, the more parallel the two hyperbolas become. When the hyperbolas are almost parallel, small variations in the TDOA for one, will cause the intersection to move very far along the other hyperbola. From this it is clear that the accuracy of the method is highest for sources close to the microphones. When the microphones are placed on a line as shown in Figure 2.5, the same inaccuracy will occur when the source is on the axis of the microphones, because the hyperbolas become parallel lines. This can be avoided by e.g. placing the microphones in an L-shape, letting the axes of the two microphone-pairs become orthogonal. An L-shape will however decrease the accuracy for sources located in the front and is currently discarded.

2.3 Simulation with Stationary Sources

A method has now been developed which can provide a 2D position based on two TDOAs calculated from the signal of three microphones. This method and the underlying theory will now be tested through simulations involving stationary sources. The software platform for implementation of the simulation is chosen to be Python. Python is chosen for having a good compromise between hardware independance, fast calculations, versatility and ease of use. It can be used for a wide range of numerical calculation techniques as well hardware interaction, easing implementations with external hardware. Python can therefore be used for both simulations and online implementation which is very convenient.

Figure 2.6 shows a detailed block diagram made with the general method in Figure 2.1 as the basis.

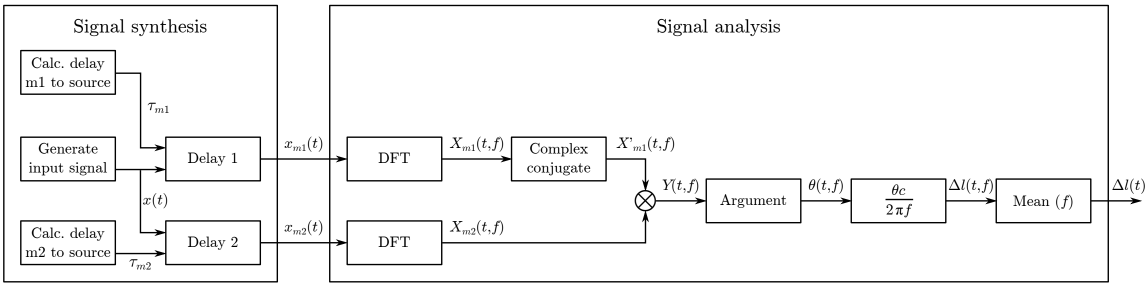


Figure 2.6: Block diagram illustating the processes used to extract the TDOA or corresponding length Δl. A larger version of the figure with the appendage of a windowing function can be found in Appendix C.

First the output of each microphone is synthesised by generating and delaying the input signal with the appropriate time values calculated from the distance between the source and microphones. Then the synthesised signals are converted to the frequency domain with a Fourier Transform and instead of dividing the two frequency signals, as described in Section 2.1.2, the first signal is complexly conjugated and then they are multiplied. This will result in a squaring of the magnitude instead of going to 1, but the argument will yield the same angle as if the signals had been divided.

The process is done for both microphone pairs, giving two TDOAs that can be used for one hyperbola each. The intersection of these two hyperbolas is then found by solving two equations with two unknowns as shown in Section 2.2.2.

Common for all the following simulations are that the source is placed at (5.0, 3.0), the microphones are placed on a line and separated with 10 cm giving a total of 20 cm from m1 to m3. This results in an upper frequency limit of 1.73 kHz and the lower limit is chosen to be 30 Hz. The simulated time frame is 0.5 s and the simulations are repeated $5 \cdot 10^4$ times.

Figure 2.7 show a simulation where the source signal contains a sum of sinusoids with frequencies matching the ones checked by the method. It is the most ideal case, as all checked frequencies have full signal strength. It should also be mentioned that before the sinusoids were summed, they are given a random phase uniformly distributed between $-\pi$ and $+\pi$, otherwise the sum would look like a series of pulses, which would be an unrealistic situation. This offset is of course different for each of the $5 \cdot 10^4$ simulations.

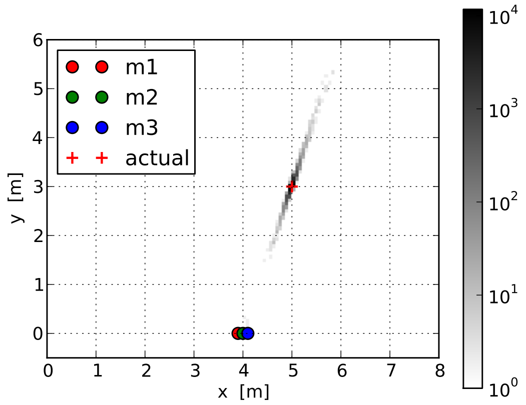


Figure 2.7: 2D histogram of simulation with sinusoids as input. The color-scale is logarithmic.

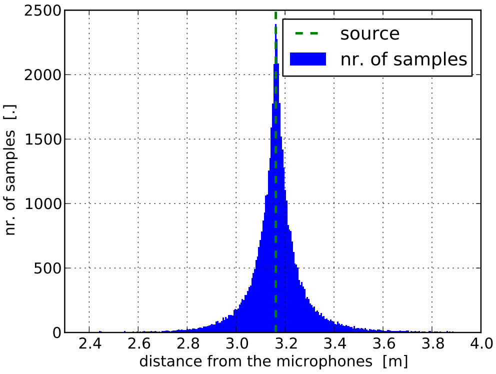


Figure 2.8: 1D histogram along the line going through the center microphone and the source.

It is clear that with the ideal case the method provides a very accurate result with regard to the angle, however the distribution along the line going through the center microphone and the source, is showing clear signs of the problem described in Section 2.2.3. This distribution is shown as the histogram in Figure 2.8.

Evidently the shape of the distribution resembles an exponential distribution. This means that a logarithmic y-axis will show linearly decaying sloaps. As it happens, the x-axis must also be logarithmic in order to have symmetric linear sloaps. This may be explained using Figure 2.5, where the brown area in the graph on the left side has an unequal distribution over and under the source. This is a result of the "angular" error being constant no matter the distance, meaning that the further away from the microphones, the more prominent the error will be. A histogram with double-logarithmic axes can be seen in Figure 2.9

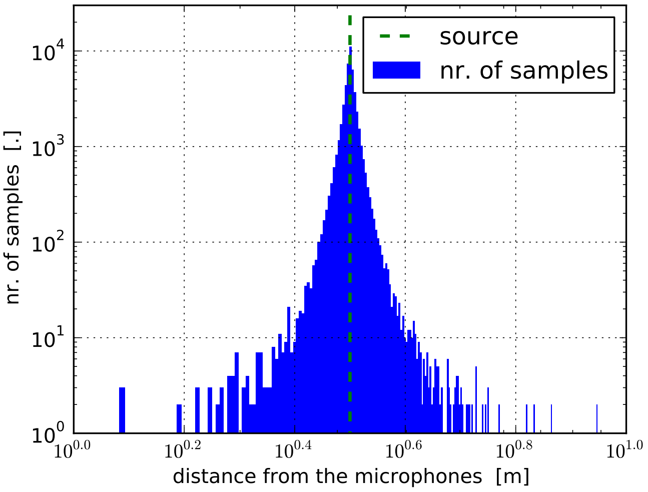


Figure 2.9: Double logarithmic 1D histogram along the line going through the center microphone and the source.

The sides of the distribution in Figure 2.9 still seems to have a slightly exponential decay, but also more erratic, indicating that a smoothing of the sloaps might reveal a more linear decay at the sides.

In Figure 2.10 the same case has been made as in Figure 2.7, but where the source signal is Gaussian white noise, created with a discrete random sequence, seeded for reproducability. In order to increase the resolution of the delays when creating the delayed signals arriving at the microphones, the noise is oversampled by a factor of 20 over the main sampling frequency f_s. If no oversampling is used, the time-delays would be quantized to e.g. steps of 20.8 μs if $f_s = 48$ and result in visible errors in the locating method. This problem is of course only present in the simulations as the delays will be produced natually in a real-world scenario. The generated noise is low-pass filtered to $0.4 \times f_s$ and then sampled at the original sampling frequency. The reason for choosing $0.4 \times f_s$ is in accordance with the Shannon/Nyquist sampling theorem, that no content should be sampled above $0.5 \times f_s$. As the method will not be using the high frequency content anyway, the 0.5 is reduced to 0.4. A 10th order Butterworth filter is used for the purpose.

The simulation has been done with white noise and repeated $5 \cdot 10^4$ times with uncorrelated samples. As it can be seen from Figure 2.10 the variance is much larger than the sinusoidal case in Figure 2.7, but the shape is otherwise very similar. The large variances in both cases must be reduced and can be by increasing the time period simulated, resulting in an improved mean for the TDOA.

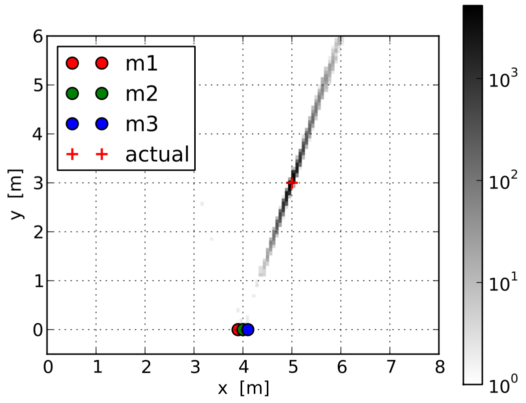


Figure 2.10: 2D histogram of simulation with white noise as input. The color-scale is logarithmic.

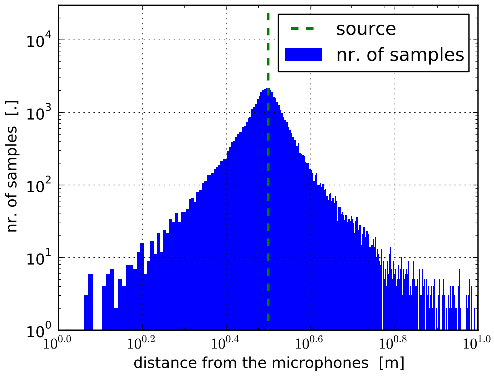


Figure 2.11: Double logarithmic 1D histogram along the line going through the center microphone and the source.

2.3.1 Weighted mean

A more viable method than using longer periods, may instead be to use a weighting of the TDOA contributed by each frequency.

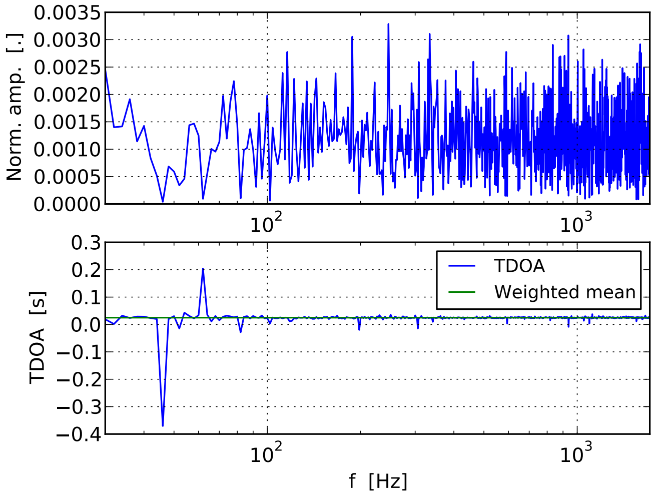


Figure 2.12: Normalised frequency response and corresponding TDOA calculations for a white noise source with an actual TDOA of 25.06 ms.

The top part of Figure 2.12 show the frequency response of one simulation limited to the frequency range used by the method. The response is plottet on linear y-axis and has been scaled so that the sum of all the elements equals 1. Below the frequency response is the TDOA calculated from each of those frequencies. The case shown is for a TDOA of 25.06 ms. The TDOA in the graph has some large fluctuations from an otherwise flat line. The mean of the shown TDOAs is 24.42 ms which is 0.64 ms off the correct TDOA and as mentioned in Section 2.2.3, this may result in a large difference from the correct position. The shown TDOA is only one of the two TDOAs, and the second may contain equally large errors.

If the response at one frequency is adequately low, there will not be phase to measure, only random noise which can be very destructive in the calculation of the TDOA. It may be noticed that whenever a large peak occurs in the TDOA, it almost always coincides with a very low value in the frequency response. The error is largest at low frequencies as the error occurs in the phase which is divided by the frequency to obtain a time value. To account for this, a weight based on the frequency response can be multiplied onto the TDOA. There are many ways applying a such weight, the simplest is a linear weighting as illustrated in Figure 2.13. The linear weighting of the frequency reponse is shown in the top of Figure 2.12.

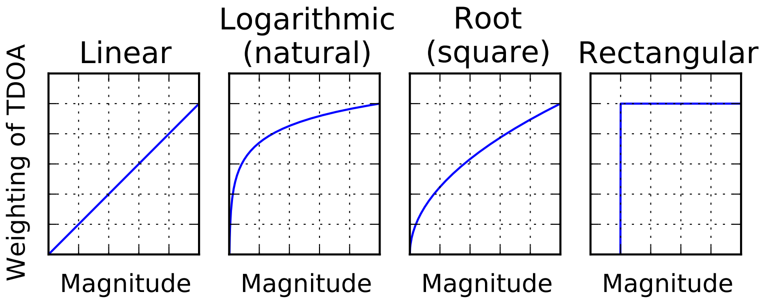


Figure 2.13: Four different forms of weightings. The x-axis is the inserted frequency response and the y-axis is the resulting weighting factor.

It is important that the weighting is normalized so the sum equals 1, otherwise there be a scaling of the TDOA, which is obviously undesirable. Beside the linear weighting there is also the logarithmic, nth-root and rectangular weighting as shown in Figure 2.13. These are just some of the different weightings that can be applied and the ideal choise depends on the use. In this case poor TDOA values only occur at very low values of the frequency response, making the logarithmic and rectangular weightings good choises, as these only sort out the worst cases. If all the elements of the TDOA is equal, it will make no difference whether a normal or a weighted mean is used, but as the peaks of the TDOA coincides with small values in the frequency response, these will have less influence on the result in the weighted mean.

However for the sake of simplicity the linear weighting is used and with the linear frequency response applied, the mean value is calculated to 24.94 ms, only 0.12 ms off. The error is not removed, but it is greatly reduced, which can also be seen in Figure 2.7 and 2.10 showing the same scenarios as Figure 2.14 and 2.15, but using the weighted mean instead and with values seeded for reproducibility.

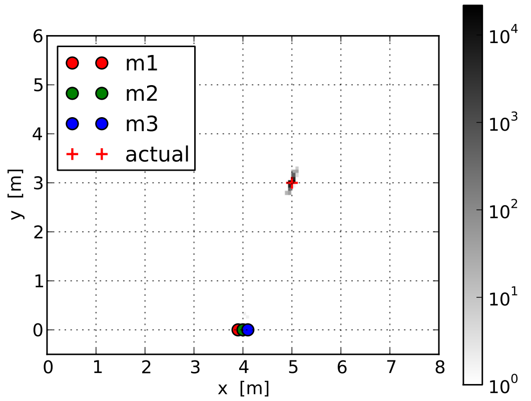


Figure 2.14: 2D histogram of simulation with sinusoids as input. The calculations include a weighted mean and the color-scale is logarithmic.

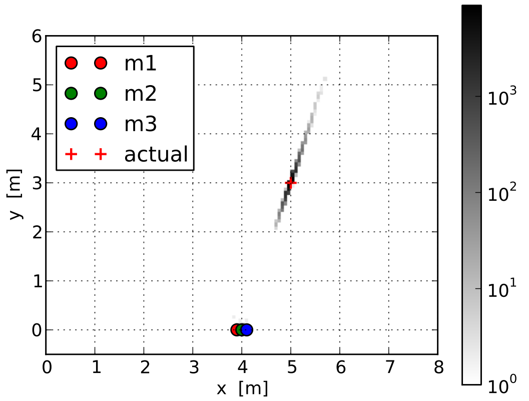


Figure 2.15: 2D histogram of simulation with white noise as input. The calculations include a weighted mean and the color-scale is logarithmic.

The variance is largely reduced, giving a more narrow and and accurate position as a result. As it may be noticed from Figure 2.12, that the largest fluctuations in the TDOA occurs at the lower frequencies as a result of the phase being divided by the frequency. The lower frequencies will always suffer from this problem and by increasing the lower frequency limit from 30 Hz to 300 Hz, a considerable decrease in variance is achieved. This can be seen in Figure 2.16 which compared to Figure 2.15 has an even smaller spead in the 2D histogram.

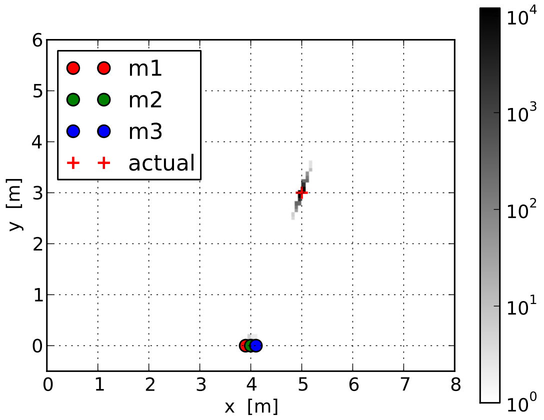


Figure 2.16: 2D histogram of simulation with white noise as input. The calculations include a weighted mean, a lower frequency limit of 300 Hz and the color-scale is logarithmic.

2.3.2 Disruptive Sources

The previous simulations are based on ideal environments where there is nothing to interfere with the main signal. This section will look at what happens when the method is subjected to non-ideal environments, e.g. the effect of a reflection from a wall, a secondary uncorrelated source containing its own TDOA and internal noise containing no TDOA.

Reflections

A reflection of the main source can be simulated as a secondary source with a reduced and delayed version of the main signal. An example is shown in Figure 2.17 where the black line illustrates a wall and the green + is a mirror-source, simulating the reflection. The mirror source has the same traveling distance as the reflection of the main source in the wall. As the source is reflected in the wall, the amount of signal reaching the microphone will be reduced, some of the signal is absorbed in the wall and some is lost as a result of the extra traveling distance. As mentioned earlier, the method does not take the signal level into account, but when two sources are present at the same time, with different phase and levels, the one with the highest level is more likely to dominate the phase information. A coefficient (A) is used to describe the amount of signal transfered to the microphones from the mirror source. The coefficient will be a number between 0 (no signal) and 1 (full signal) and is multiplied to the signal of the mirror source. As the content of the mirror source is always the same as that of the main source, the coefficient also describes the level ratio between the two signals. A frequency dependent reflection coefficient could have been used along with diffraction, but is not deemed necessary to illustrate the concept.

The simulations using white noise as input has the largest variance of the two signal types used and will visually show the effect more clearly and will therefore be used as the source signal. The figures are based on the same simulations as in Figure 2.16 with $5 \cdot 10^4$ samples of 0.5 s and using the weighted mean. The dashed blue lines illustrates the traveling path of the reflection and how that is accomplished with the mirror source. Figure 2.17 show the main source being reflected in a wall where $A = 0.2$. The influence with a coefficient of 0.2 has no visual effect other than a few results ending up in front of the microphones, but there is no considerable difference from Figure 2.16.

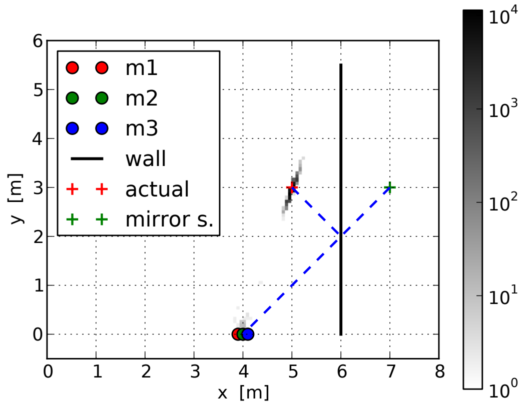


Figure 2.17: Simulation of a source and its mirror source with white noise as input, $A = 0.2$.

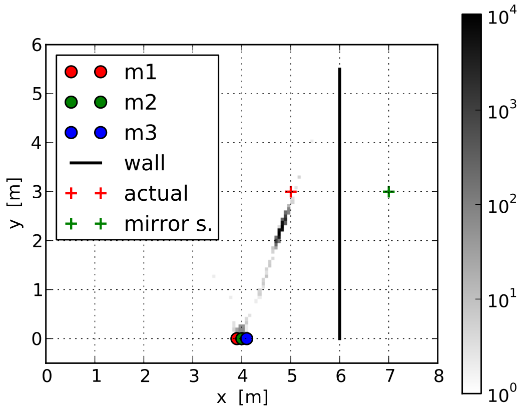


Figure 2.18: Simulation of a source and its mirror source with white noise as input, $A = 0.4$.

In Figure 2.18 where $A = 0.4$, the effect is much more prominent, the results have been pulled closer to the microphones and more samples have been moved almost all the way towards the microphones. The angle of the results has also changed a little towards the mirror source. In Figure 2.19 where $A = 0.6$, the results are pulled further towards the microphones. Even more of the samples have clustered close to the microphones and the variance has been substantially increased.

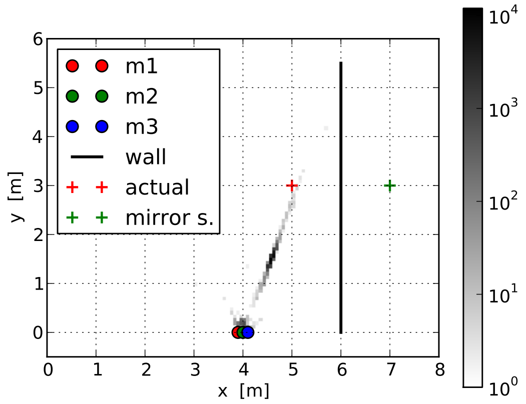


Figure 2.19: Simulation of a source and its mirror source with white noise as input, $A = 0.6$.

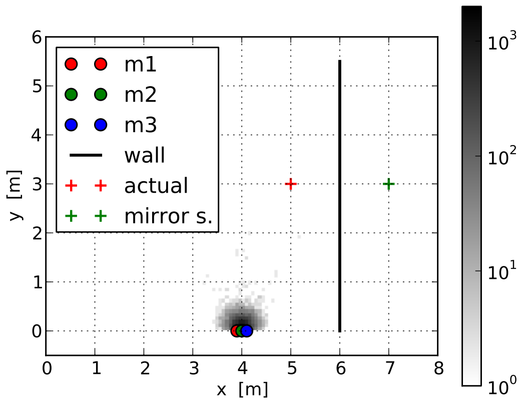


Figure 2.20: Simulation of a source and its mirror source with white noise as input, $A = 1.0$.

The last simulation in Figure 2.20 show the unrealistic example of the secondary source being the same level as the main source. The locating method is completely unable to determine the direction and places all the results close to the microphones. This shows that for a continuous source signal, reflections at levels close to the main signals level can be disruptive to the method and should if possible be avoided.

Uncorrelated Sources

The simulations of reflections in the previous section was based on a secondary source signal which is highly correlated with the actual source. In these simulations the signals will be uncorrelated, simulating an independent source and not a reflection. Both source signals will be white noise with the same initial variance, but the secondary will be scaled with a value $A < 1$. The first case shown in Figure 2.21 has the secondary source scaled to $A = 0.05$, a few result have moved close to the microphones, but other than that the difference from Figure 2.16 is very little.

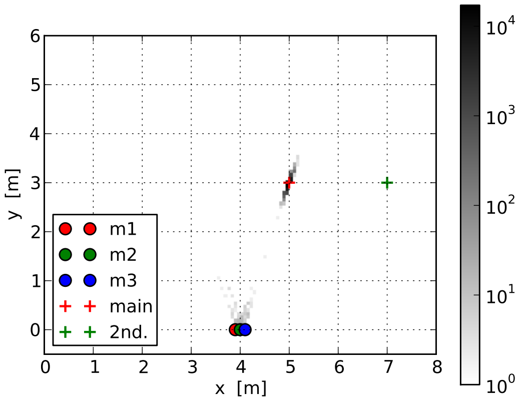


Figure 2.21: Simulation of a main source and a second uncorrelated source, both with white noise as input, $A = 0.05$.

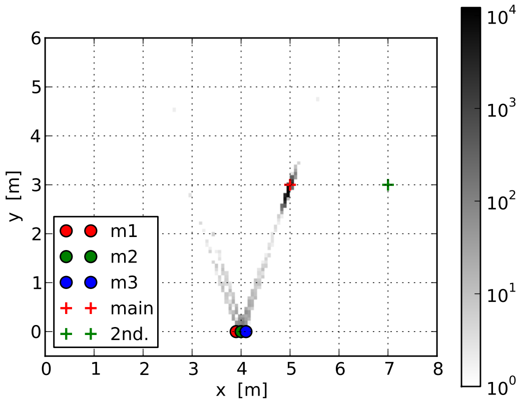


Figure 2.22: Simulation of a main source and a second uncorrelated source both, with white noise as input, $A = 0.1$.

The effect of the increased level of the secondary source in Figure 2.22 has a more distinguishable effect. The methods estimates traces a line from the actual position of the source and down to the microphones, from there the line is reflected in the axis of the microphones. This is likely because the method assumes that all sources are located in front of the microphones and the line would have continued through the microphones had this not been the case. Increasing the secondary source even more, further diminish the methods ability to locate the main source as shown by Figure 2.23 and Figure 2.24.

Comparing these simulations with those of a reflection, the amount of damage caused by an uncorrelated source is much greater than that of the correlated one. The amount of error caused by an uncorrelated source with $A = 0.1$ bears closer resemblance to the correlated source with $A = 0.6$ as can be seen when comparing Figure 2.22 with 2.19.

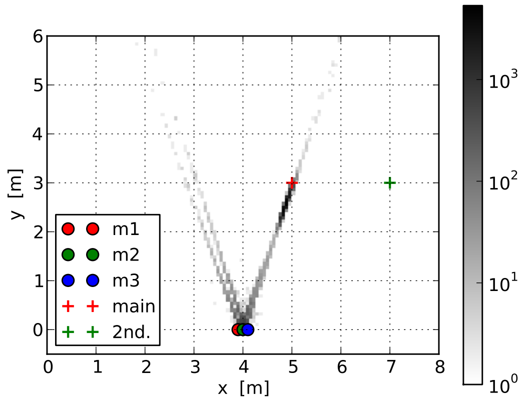


Figure 2.23: Simulation of a main source and a second uncorrelated source, both with white noise as input, $A = 0.2$.

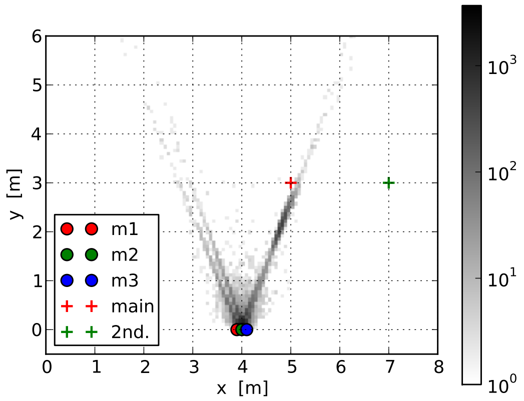


Figure 2.24: Simulation of a main source and a second uncorrelated source, both with white noise as input, $A = 0.4$.

Internal Noise

The two previous types of simulations are based sources placed in a room, which means there is a correlation between the different channels from the microphones. Internal noise on the other hand will not have any correlation between the channels as it could be e.g. noise created in each microphone and therefore have no relation to that of another microphone. Figure 2.25 show a simulation using white noise in the main source. Once the sampling of the signal is simulated by the individual delay for each channel, a white noise sequence is added. As explained the added white noise is different for each channel.

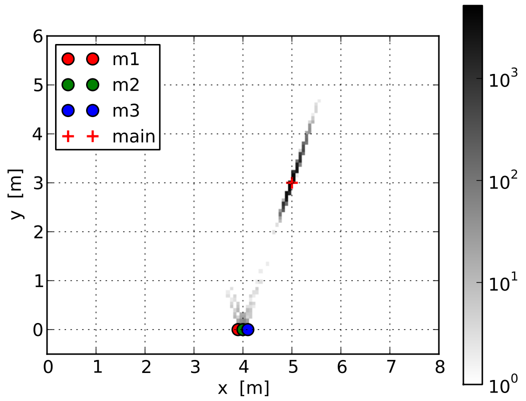


Figure 2.25: Simulation of a source using white noise added to each input channel with amplitude $A = 0.01$.

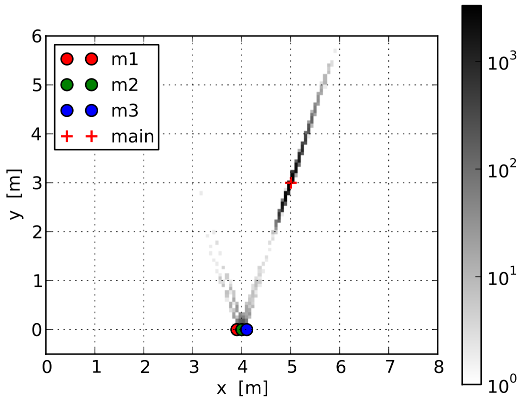


Figure 2.26: Simulation of a source using white noise added to each input channel with amplitude $A = 0.015$.

The relative amplitude of the added noise in Figure 2.25 is $A = 0.01$ and has resulted in a sligtly larger variance than in Figure 2.16, but otherwise only a few results have moved close the microphones. The result shown in Figure 2.26 of a simulation with amplitude of the added noise $A = 0.015$ show larger variance then for $A = 0.01$ and the clustering at the microphones has also increased. Internal noise with an amplitude of $A = 0.04$ shown in Figure 2.27 can probably be best compared with an uncorrelated source where $A = 0.2$ as in Figure 2.23. Looking the effect of internal noise and a second uncorrelated noise source, the method reacts stronger to the internal noise.

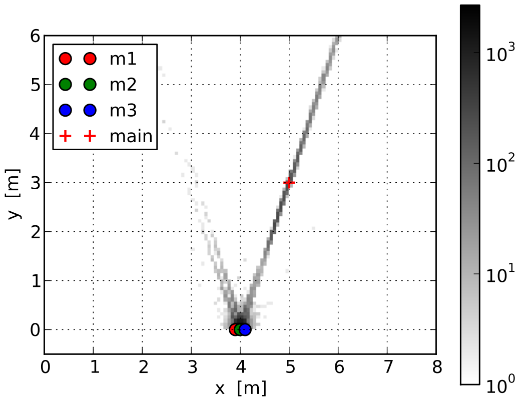


Figure 2.27: Simulation of a source using white noise added to each input channel with amplitude $A = 0.04$.

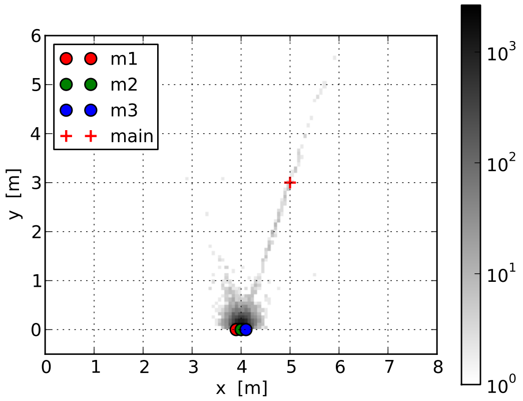


Figure 2.28: Simulation of a source using white noise added to each input channel with amplitude $A = 0.06$.

Of the three disruptive sources simulated throughout Section 2.3.2, the method is clearly most sensitive to internal noise, with amplitude values in the area of 10^{-2}, where the uncorrelated source ranges closer to 10^{-1} and a reflection also around 10^{-1}, but closer to 10^{0}. Thankfully internal noise is a relatively small problem as modern microphones have little internal noise and with a properly constructed ADC and preamplification circuit little to no considerable internal noise can be expected. Reflections and uncorrelated source are harder to avoid as these are a common part of almost every ordinary room. For the method to be able to separate the main source from a reflection or a different source, the method must be further refined. Using an always-active signal as input is also complicating the matters further as the disruptive signal will always be present together with the main signal, making it virtually imposible to separate them without prior knowledge.

2.4 Measurement of Stationary Sources

Section 2.3 describes the simulations of stationary sources and provides an indication of how well the method performs under different circumstances. This section will focus on measurements, which are done under the same conditions as described in Section 2.2.1 to confirm the simulations ability to show the functionality of the method.

2.4.1 Measurement Setup

The mesurement setup will be closely related to the one used in the preceding simulations and as explained the envionments are expected to be anechoic. The measurements are therefore conducted in an anechoic room. Figure 2.29 illustrate the arrangement of the equipment.

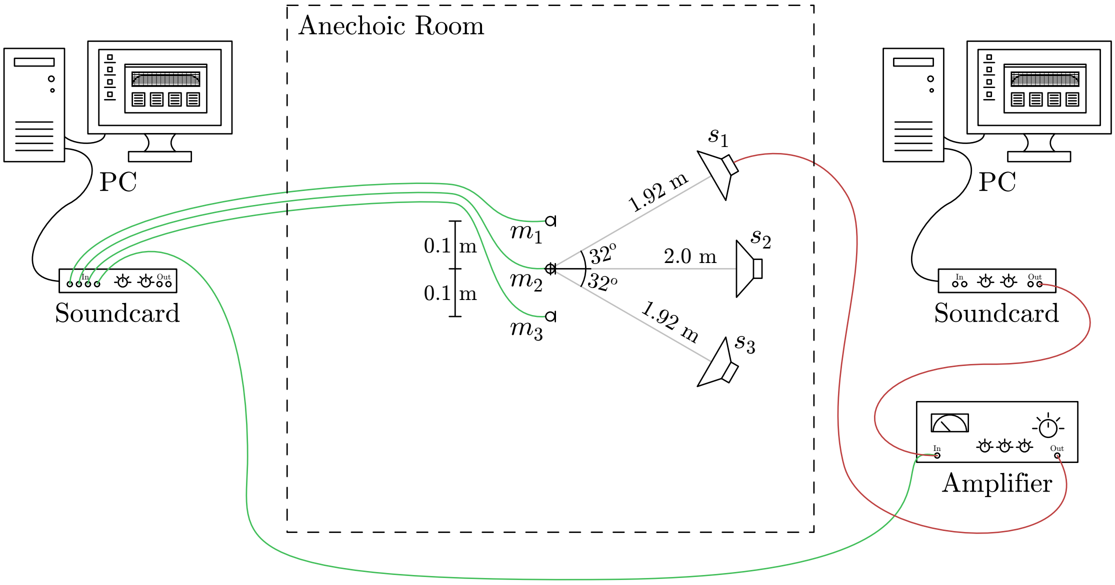


Figure 2.29: Setup of the measurement equipment in- and outside the anechoic room.

The sources and microphones are placed 1.2 m above the floor. The separation of the microphones is set to 0.1 m, which according to Equation (2.5) allows the method to analyse up to 1.73 kHz. The positions of the sources are part of a permanent setup in the used anechoic room and is the reason for the somewhat peculiar distances and angles. It is however unimportant where the sources are placed as long as the positions are known. The green lines in Figure 2.29 illustrate input, while red lines illustrate output. The output signal fed to the power amplifier is recorded along side the signals from the microphone to monitor the output. Separate playback and recording systems are used out of necessity as 3+ channel recording could not be made possible on a system with synchronised playback. Synchronisation of playback and recording is not a necessity, but will give a better comparison between measurements and especially because the recording system uses the odd recording sampling frequency 51.2 kHz. Instead of using only the three sources present in the horizontal part of the setup, the measurements will be repeated after the microphones are rotated 16° clockwise. From the view of the microphones, this will give an additional 3 source positions, resulting in those given by Table 2.1

Source	Distance [m]	Angle [°]	Coordinate [m]
1	1.89	48.0	(-1.40, 1.26)
2	1.89	32.0	(-1.00, 1.60)
3	2.00	16.0	(-0.55, 1.92)
4	2.00	0.0	(0.00, 2.00)
5	1.89	-16.0	(0.55, 1.92)
6	1.89	-32.0	(1.00, 1.60)

Table 2.1: List of source positions included in the measurements (Angles are relative to directly in front, perpendicular to the axis of the microphones).

From this array of sources the setup will provide an 80° view of the frontal performance and assuming that the performance is symmetrical around 0°, this may be expanded to 96°.

According to the simulations carried out in Section 2.3.2, uncorrelated sources or background noise can be very destructive and so a measurement of the background noise is analysed to make sure no unexpected sources are present.

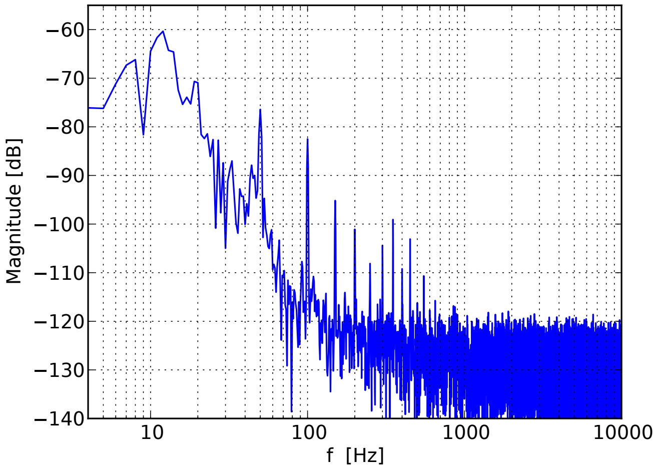


Figure 2.30: Frequency analysis of background noise.

Figure 2.30 show a frequency analysis of 30 s of backround noise within the anechoic room. The measurement system is setup to highpass filter the input at 10 Hz. The beginning of the roll-off can be seen in the figure, but is still leaving a great deal of low-frequency background noise. The steep increase in noise when going downwards in frequency from approximately 150 Hz is expected for the given anechoic room as it is only designed to be anechoic down to 200 Hz. Beside the high noise floor at low-frequencies, there is also a number of high peaks corresponding to 50 Hz hum and harmonics thereof. Minor peaks can be seen up to 550 Hz, but the major peaks are at 100 Hz and below. As the method only concentrates on the frequency range from 300 Hz to 1.7 kHz, anything outside this range will have no influence on the performance. As the noisefloor including the peaks are acceptably low within the given range, no intervention is required.

Just as in Section 2.3, both sinusoids and white noise is used to test the methods performance using an idealised source and one closer to reality. Beside these two types of signal, an additional type is used, namely a recording of speech which may give a more authentic response from the method. Figure 2.31 show a 2 s window of the three input signals.

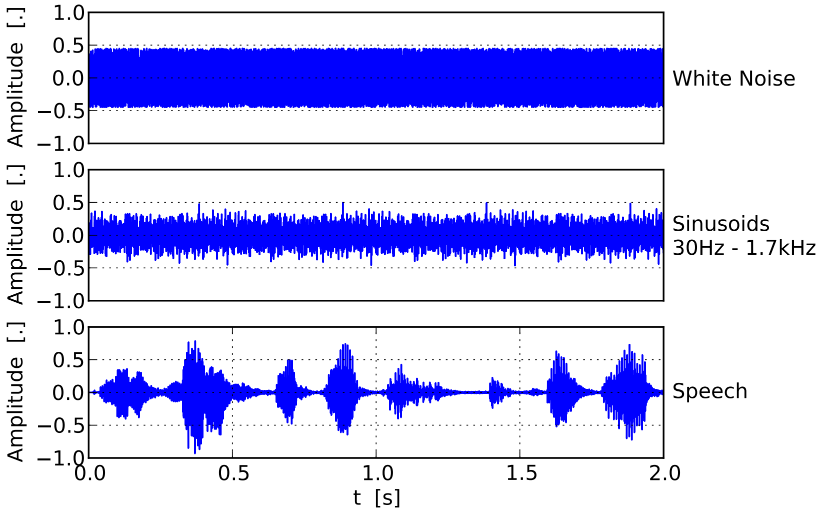


Figure 2.31: Transient representation of a 2 s sample from the three input signals. (The signals in the figure has been downsampled and may appear slightly different from the originals.)

From the figure it is clear that the three signals are not scaled to the same level. The reason for this unequal normalization is that the signal does not contain the same amount of energy and the white noise and sinsoids are therefore normalised to 0.5 instead of 1.0 for the speech. This will make the results more comparable.

2.4.2 Results

Figure 2.32 shows the results of the six measurements using the sinusoidal signal as input. Each of the colors represents a histogram of the estimates using 100 samples of 0.5 s.

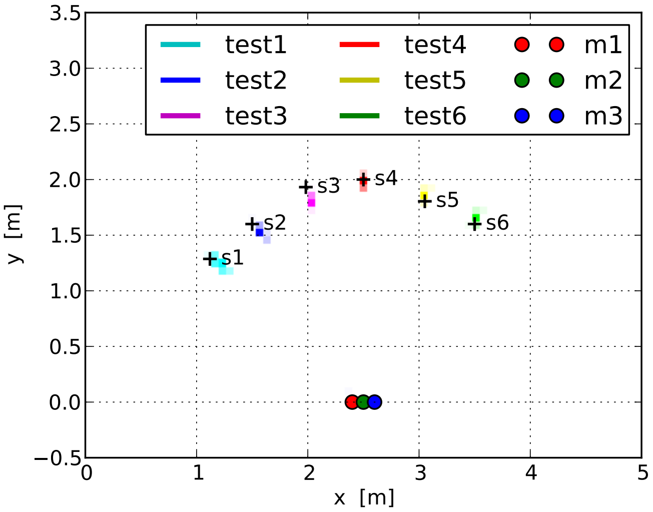


Figure 2.32: 2D Histogram of the six measurements using sinusoidal signals.

The results of these measurements show that the method succeeded in locating the source and the small deviations from the correct positions are more likely to be inaccurately measured source positions or an inconsistency in the positions of the microphones. Only a small offset in the microphone positions is required to produce a large and constant change in the estimated positions. Figure 2.33 show a measurement session with microphone 2 placed 5 mm closer to microphone 1, effectively decreasing the distance and therefore also TDOA of the first microphone pair and inversely for the second pair.

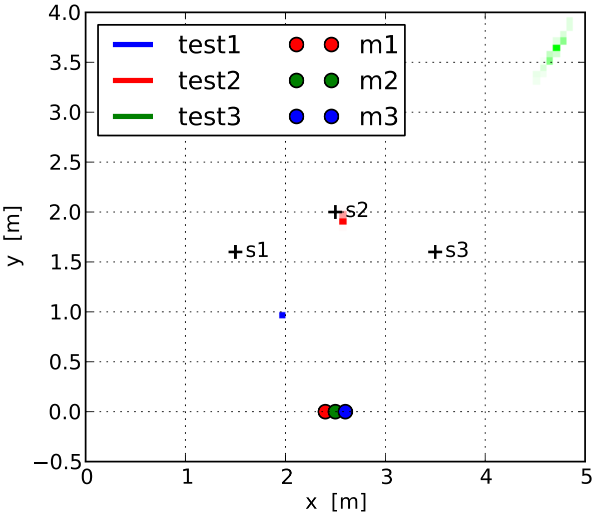


Figure 2.33: Measurement with an incorrectly placed microphone, resulting in a skewing of the estimates.

The misplaced microphone causes sources in the left side to move closer and in the right side to move away. There is no considerable effect in the center, because both the TDOAs are almost zero and will be same in the front almost no matter the horisontal offset of the microphones. A readjustment of the microphones solves the problem.

Figure 2.34 shows the measurements using white noise, with the same conditions as in Figure 2.32.

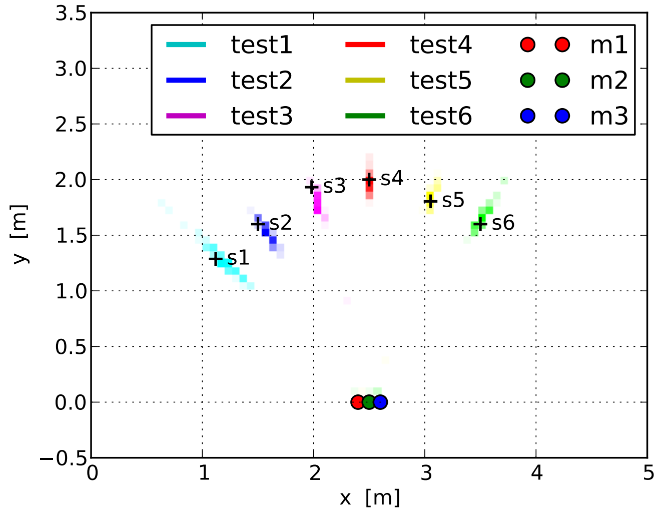


Figure 2.34: 2D Histogram of the six measurements using white noise.

Again the method succeeds in determining the positions of the source, although as expected with a somewhat larger uncertainty. This behavior closely resemble that predicted by the simulations, showing good consistency of the method. The variances of the six measurements are not completely the same. As described in Section 2.2.3 the hyperbolas become more and more parallel as they approach the axis of the microphones, making it more and more susceptible to small changes, which can be alleviated with an alternative setup of the microphones. The last series of measurements are done using the speech signal. The result thereof is shown in Figure 2.35.

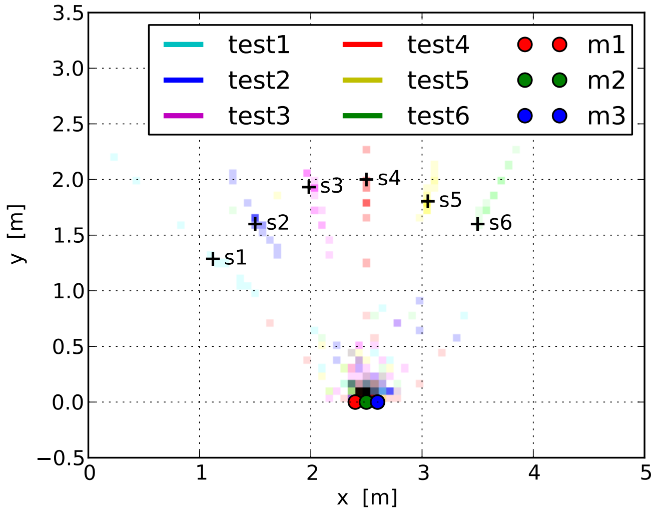


Figure 2.35: 2D Histogram of the six measurements using a recording of speech.

The results of these measurements show a lot of poor estimates by the method and out of the 100 samples only an average of 64.67 gave a valid result. The main reason for this is expected to be the large number silent parts in the signal, which can be seen in Figure 2.31, unlike the two other signal types, which are always on. This indicates that the method has not been able to ignore the background noise.

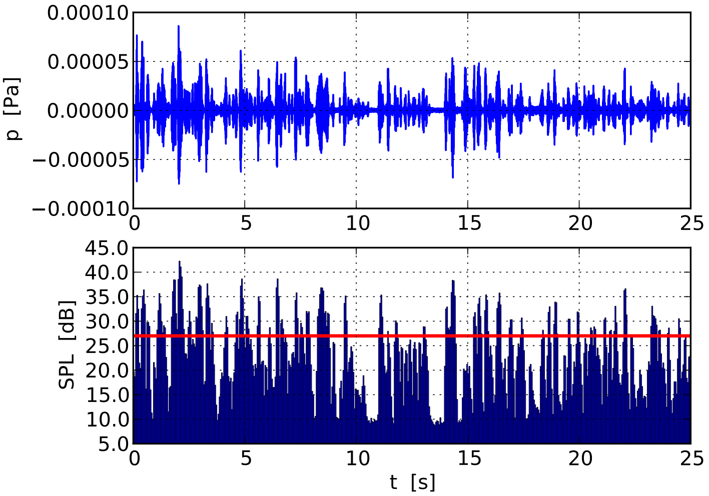


Figure 2.36: Transient view and corresponding SPL of the measured signal at microphone 2 with the speech signal played at source 6. (The transient signal in the figure has been downsampled and may appear different from the original.)

It is possible to eliminate some of the bad estimates by decreasing the window size, thereby increasing the likelyhood of one window containing only signal or only background noise. This is of course a tradeoff as the mean of the TDOA will be calculated over a smaller number of samples. Figure 2.36 show a transient view of microphone 1 while source 6 is active. Below the transient view, the SPL of 0.1 s windows is shown. It should be mentioned that the SPL value is calculated based on the frequency range used by the metod only and is not fully representative of the actual signal. The windows containing only background noise can be separated by making a threshold for the SPL and whenever the SPL is below, the estimate is excluded. By reducing the window size from 0.5 s to 0.1 s and including a threshold at 27 $\mathrm{dB_{SPL}}$, the result shown in Figure 2.37 is acheived.

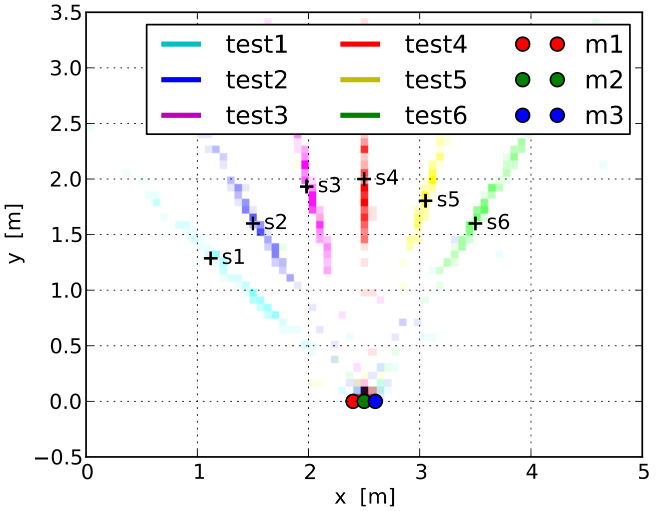


Figure 2.37: 2D Histogram of the six measurements using a recording of speech. Using 0.1 s window and a threshold at 27 $\mathrm{dB_{SPL}}$.

The variance in Figure 2.37 is larger than in the previous analyses, but more of the estimates are located around the actual source positions and not clustered near the microphones, which is a large improvement. The threshold will depend on several variables such as the window size, signal strength and background noise inorder to present a good result.

The overall result of the measurements show that with the method, it is possible to locate a source in an idealised envionment. Combining the results gathered in this section with the results of Section 2.3.2, where the method is simulations in environments containing disruptive sources, it is judged that measurements in envionments with less idealised properties may be able to produce usable results.

Chapter 3

Locating a Moving Source

The process of locating a moving source and a stationary source has many things in common, however a distinctive difference is of course that the position of a moving source is time dependant. Hence it is not enough to make one averaged measurement of the whole time frame. A division into smaller time-steps is necessary and analysing the steps as the source moves. No alterations of the basic method is required, as it will be given a window of data for each time-step and determine a position. As the source moves, the method will be able to draw the path in which the source is moving.

Because the source is moving, the TDOA within a window will change. It is therefore important that the amount of movement of the source within one window is kept at a minimum. The window will of course also have to be large enough to provide a reliable result. In order to increase the amount of positions calculated without changing the size of the window, they have been made overlapping.

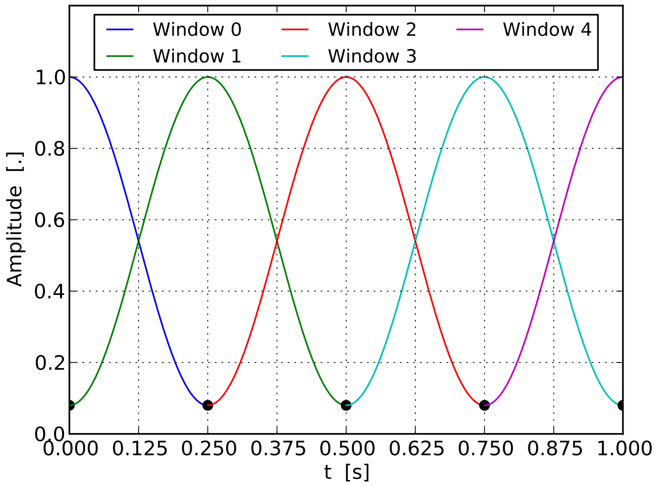


Figure 3.1: Illustation of the overlapping Hamming-window functions.

As shown in Figure 3.1, 50% overlapping Hamming-windows are used to double the number of steps.

3.0.3 Time Dependence

The block diagram in Figure 3.2 is a slightly modified version of the block diagram in Figure 2.6. The difference is the addition of windowing, which means that this process is executed twice within the frame of a window to produce the TDOA. There are however elements in the process which can be reused, so when calculating the TDOA for microphone 1 + 2 and then 2 + 3 there is no reason to calculate the DFT of microphone 2 twice.

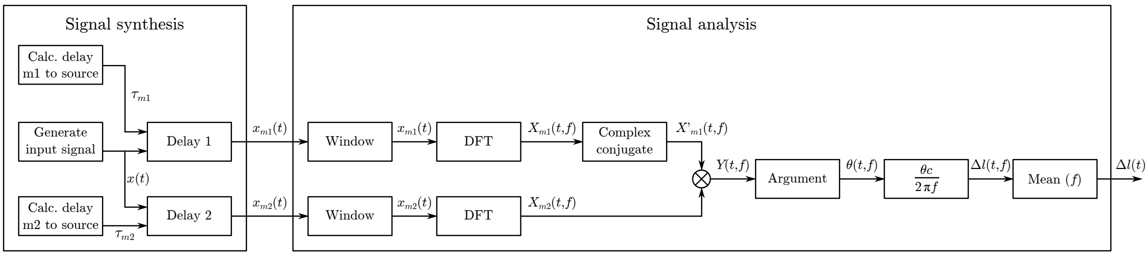


Figure 3.2: Block diagram illustating the processes used to extract the TDOA or corresponding length Δl for a moving source. A larger version of the figure can be found in Appendix C.

Just as in Section 2.2.2 the two TDOA are used to find the intersection of the two corresponding hyperbolas, providing a 2D position.

3.1 Simulation

For the simulation of a moving source to be realistic, the speed at which the source is moving should be related to that of a human being. [Bohannon, 1997] has measured the maximum and comfortable gait speeds of men and women at the ages 20s to 70s. A mean of his findings show a comfortable speed of approximately 1.4 m/s or 5.0 km/h, which is used as the source speed in the simulations. The walking pattern will be a circle with a radius of 1.5 m giving a circumference of 9.4 m. the circle is centered at (4.0, 4.0) in front of the microphones. The reason for choosing a circle as moving pattern is that all directions of movement is included and the source will at one point be close to the microphones and far away at another. This way several different aspects of the method can be tested. The source starts at the top of the circle and moves clockwise until it has made one revolution. As mentioned, the circumference of the circle is 9.4 m and the speed with which the person will walk around the circle is 1.4 m/s giving a walking time of 6.7 s. Beside a walking speed of 1.4 m/s, an additiononal simulation is done at half the speed, 0.7 m/s. This will show show the performance at lower speeds where the TDOA within one window is closer to constant. First simulation will be the slow 0.7 m/s and is shown in Figure 3.3 and 3.4.

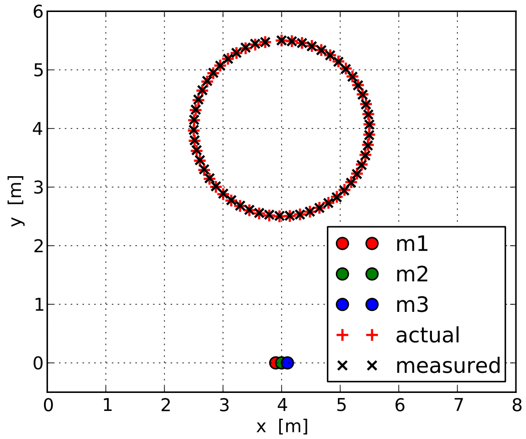


Figure 3.3: Simulation of moving source with sinusoids as input, the speed used is 0.7 m/s.

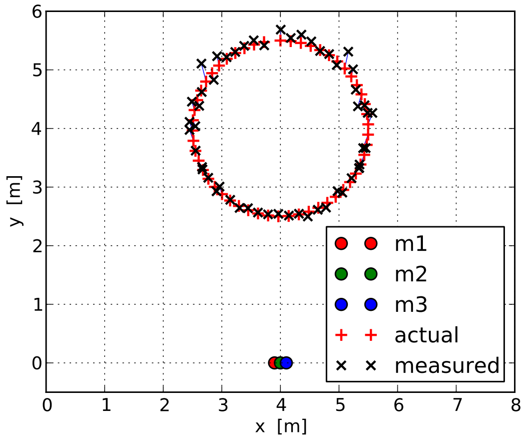


Figure 3.4: Simulation of moving source with white noise as input, the speed used is 0.7 m/s.

Each + and × show the position of the source in the middle of a window, the red + show the actual position and the black × show the position estimated by the method. Blue lines are used to illustrate the error of the calculated poistion in each time-step as a line connecting the two points. Figure 3.5, showing the simulation containing sinusoids as input and reveals a virtually exact tracking of the source, there is no visible difference between any of the estimated and the actual positions. The largest error in any of the estimates is 1.45 cm and is considered a very good estimate when comparing to the distances between the sources and the microphones which reaches up to 5.5 m. Figure 3.6 where the input is white noise, the accuracy is not quite as good as the idealised sinusoidal case. However the majority of the estimates corresponds well with the actual positions of the source. The largest error is 31.7 cm, larger than the sinusoidal case, but the shape of the tracking is circular and bears good resemblance to the actual shape.

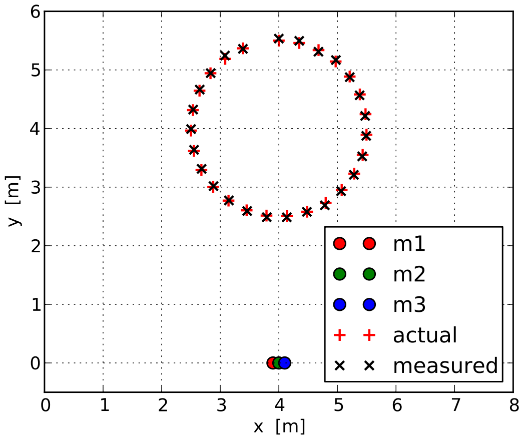


Figure 3.5: Simulation of moving source with sinusoids as input, the speed used is 1.4 m/s.

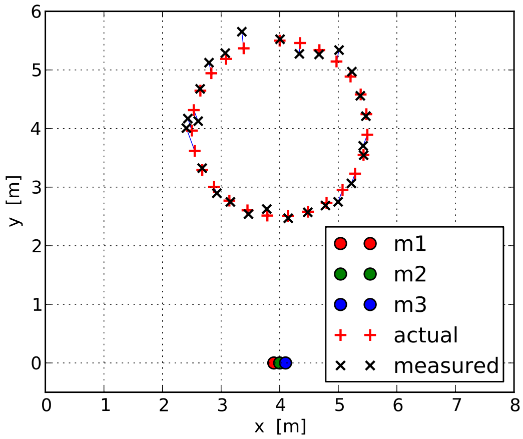


Figure 3.6: Simulation of moving source with white noise as input, the speed used is 1.4 m/s.

In Figure 3.5 where sinusoids have been used, the result of the method is virtually the same as the real positions. The largest error occuring is 5.7 cm, which is slightly larger than the case with a speed of 0.7 m/s. In Figure 3.6 where the input is white noise the distance is, just as in the stationary case in Section 2.3, the parameter with the largest issue, the angle on the other hand is very sturdy. The largest error is 41.5 cm, a relatively small increase considering that the amount of movement within each window has doubled.

In Appendix D the error in each time-step is shown for the four previous simulations. Figure 3.7 is one of those, more specifically the case of the fast moving source (1.4 m/s) with white noise shown in Figure 3.4. Unlike the other three figures, the error in Figure 3.7 is visibly dependent on the source position. This should have been even more distinct in the case of fast moving noise, but it is not showing as characteristic a shape.

As mentioned earlier, the source starts at the top of the circle, which is indicated by the medium sized error in the beginning of Figure 3.7. When the source moves to the right, it moves away from the center, which means the accuracy drops, shown by the increase in size of error. As the source moves below $y = 4$, the distance between the source and microphones is decreased and so is the error until it reaches $x = 4$, where the distance is the smalest and the source is in the center, giving the highest accuracy of the simulation. Past this point, the shape of the of the error is mirrored, which is to be expectead as the movement is simply the opposite.

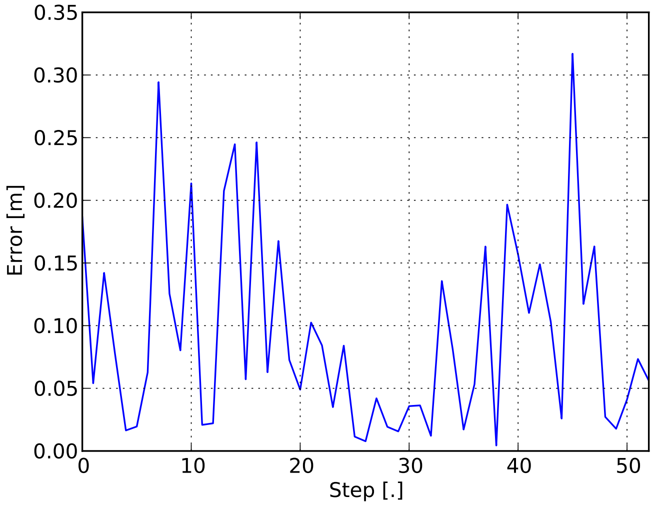


Figure 3.7: Error in the estimate when simulation a moving source with white noise as input and a speed of 0.7 m/s.

These simulations show that it is theoreticaly possible to locate a source moving at up to 1.4 m/s. The tracking can be done with a reasonable accuracy which will depend on several factors such as the quality of the input signal, the speed of the source, the general distance to the source and various envionmental factors such as those described in Section 2.3.2.

Chapter 4

Implementation

This chapter aims at describing the construction a hardware platform and the implementation of the method described throughout the previous chapters.

4.1 Hardware

The hardware used consists of a Raspberry Pi as the main processing unit, a Wolfson Audio Card as codec and a preamplifier circuit with an electret microphone for each channel. The analog part of the system is designed to handle 4 analog channels, thereby being able to process the at least 3 channels required to find a source in 2D and to be forward compatible if improvements are added. An overview of this system can be seen in Figure 4.1 where only the important connections are included.

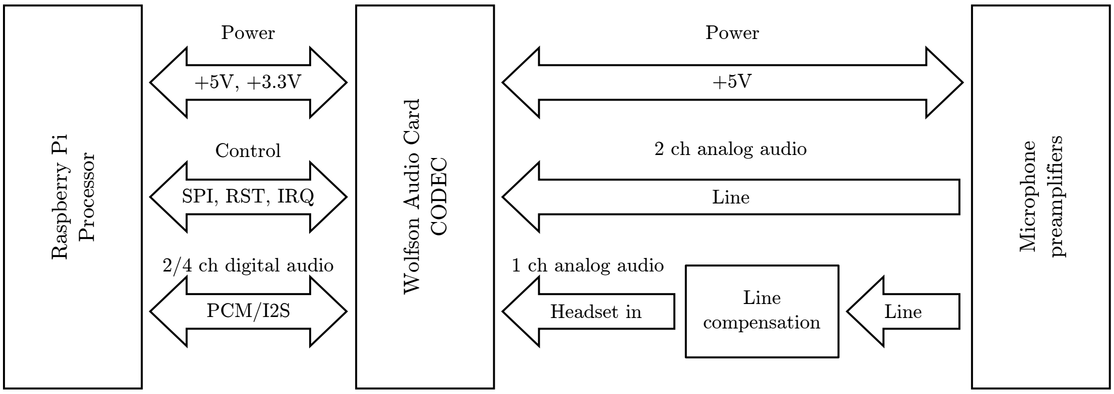


Figure 4.1: Block diagram of how the hardware is connected. The figure is based on the blockdiagram used in the schematic of the Wolfson Audio Card.

4.1.1 Microphones and Preamplifiers

In this section a description of the microphones used and the circuitry used power and amplify their signal to line level will be presented.

Requirements

The preamplifiers will be connected to a line input and will therefore have to comply with the standards that apply to a line in- and output. According to clause 6.2 of [IEC, 1997] a line output shall deliver a maximum of 2 V_{RMS}, corresponding to 2.8 V_P. Additionally the output impedance should be $\leq$ 1 kΩ for consumer equipment.

The preamplifiers should be capable of amplifying an average speaking level up to an adequate voltage chosen to 0.2 V_{RMS}. According to [Poulsen, 2005] the average speaking level of a male speaker is 65 dB_{SPL} and for a female speaker it is 62 dB_{SPL}. 62 dB_{SPL} being the lowest level will be the minimum required for a 0.2 V_{RMS} output.

The 4 preamplifiers with microphones are made as shown in Figure 4.2.

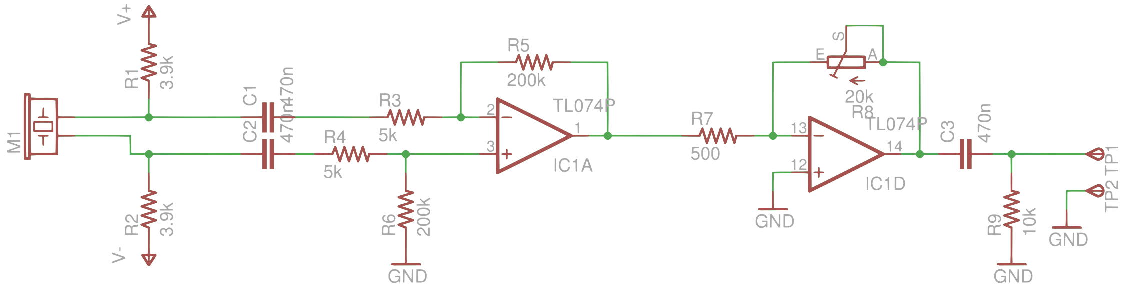


Figure 4.2: Schematic showing one of the four channels.

The electret microphones are 4 mm omni directional microphones made by RS Essentials. The important characteristics are stated in Table 4.1. A datasheet with more information on the microphones can be found on the project CD `Datasheets/microphon.pdf`.

Specification	Value	Unit
Operating voltage	2.0	V
Max current	0.5	mA
Output impedance	2.2	kΩ
Frequency range	50 - 16000	Hz
Sensitivity (94 dB_{SPL} @ 1 kHz)	8 ± 3 dB	mV
SNR	60	dB

Table 4.1: Specified electrical characteristics of the RS electret microphones.

The information in Table 4.1 is not completely consistent as a voltage of 2 V over 2.2 kΩ results in a current of 0.9 mA. Datasheets of other electret microphones with similar characteristics states an operating voltage between 1 and 10 V. The circuit is therefore designed to give 0.5 mA at an impedance of 2.2 kΩ. With the circuit shown in Figure 4.2 the microphones will be supplied with:

$$V_{M1} = V_{CC} \cdot \frac{R_{M1}}{R_{M1} + R_1 + R_5} \tag{4.1}$$

where R_{M1} is the output impedance of the microphone.

$$= 5 \text{ V} \cdot \frac{2.2 \text{ k}\Omega}{2.2 \text{ k}\Omega + 3.9 \text{ k}\Omega + 3.9 \text{ k}\Omega} \tag{4.2}$$

$$= 1.1 \text{ V} \tag{4.3}$$

With an impedance of 2.2 kΩ, the current will then be 0.5 mA.

The reason for using two resistors to supply the microphones is to allow the next stage to be differential, eliminating as much unwanted noise as possible. The first stage is set up as a standard differential amplifier circuit with equal amplification in the inverting and noninverting side. The second stage is a standard inverting amplifier with variable gain to accommodate for inaccuracies in the individual levels and to adjust the overall gain in case of low/high levels from the source.

The distribution of gain over the two amplifer circuits are 32 dB in the first and -∞ to 32 dB in the second. The total gain is then -∞ to 64 dB. This means that at the highest gain, an input level at 58 dB_{SPL} can be amplified to 0.2 V_{RMS} and 78 dB_{SPL} can be amplified to 2 V_{RMS} corresponding to the maximum input voltage of the line input of the Wolfson Audio Card.

Resistor values of the first stage is chosen relatively high, in order to keep the input impedance as high as possible without having a too large resistor in the feedback loop as they generally contribute with noise. The values in the second stage are chosen to be a factor 10 smaller than in the first as the output impedance of the first stage is much lower than the impedance of the microphones.

To keep the circuit simple, it will be supplied with 5 V from Raspberry Pi, which is then split to 2.5 V to create a virtual ground. The virtual ground is created with the rail-splitter shown in Figure 4.3 where two transistors are centered by a voltage divider created by the two 10 kΩ resistors. The bases are however separated by two diode voltages, one for each transistor. Additionally, some capacitors are added to remove potential digital noise coming from the Raspberry Pi and Wolfson Audio Card. Because the opamps are only supplied with 5 V, they need to be rail-to-rail, meaning that the output can go almost from 0 to 5 V. For that the TS464 are chosen, they can output at least 2 V_{p} with a supply of ±2.5 V relative to the virtual ground. The datasheet does not state the output impedance of the TS464, but is often around 1 Ω for general purpose opamps, which is much less than the required 1 kΩ.

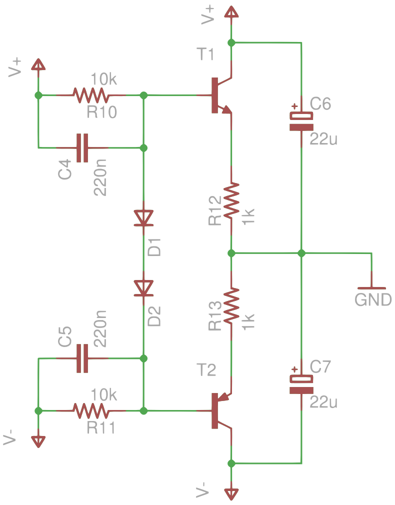


Figure 4.3: Schematic showing the rail-splitter used to create a virtual ground.

4.1.2 Codec

The Wolfson Audio Card is chosen for its codec with multichannel capabilities and for being compatible with a Raspberry Pi, together making a low-cost acquisition and processing unit.

The codec used on the audio card is a 24 bit Wolfson WM5102. The codec has 6 ADCs and 7 DACs whos data can be transfered through three digital audio interfaces (AIF), which all support TDM and four predefined interface modes including I2S. They all support sampling frequencies between 4 and 192 kH and AIF1 additionally supports timeslots with up to 8 mono channels in and out. The codec can be controlled through I2C, SPI and SLIMbus, but only SPI is connected to the Raspberry Pi. The codec has several other features which are not relavant to the project, but can be seen in its datasheet on the project CD `Datasheets/WM5102_datasheet.pdf`.

The audio card has a stereo line input and output and it has a connection for headset with one input for a microphone and stereo output for headphones. Two onboard digital MEMS microphones have been connected and a SPDIF stereo digital input and output is added through a WM8804 digital interface tranceiver. The schematic of the audio card along with a guide on how to connect and use it with the Raspberry Pi can be found on the project CD `Datasheets/WolfsonAudioCardSchematicDiagram.pdf` and `Datasheets/WolfsonRaspberryPiSoundcardManual.pdf`.

Analog input	**Use**
A_IN1 L/R	Line in L/R
A_IN2 L/R	MEMS mic Clock/Data
A_IN3 L/R	TP/Headset mic

Digital input/output	**Use**
AIF1	RPi header
AIF2	WM8804 (SPDIF)
AIF3	Exp header
I2C control	Exp header
SPI control	RPi header
SLIMbus control	N/C
$\overline{\text{RST}}$	RPi header
$\overline{\text{IRQ}}$	RPi header
MCLK2	RPi header
GPIO3	Exp + RPi header
GPIO4	Exp header
GPIO5	RPi header

Table 4.2: Important analog and digital connections from the WM5102 codec. TP refers to testpads on the PCB, RPi header is the pinheader connecting to the Raspberry Pi, Exp header is an expansion pinheader, overlining $\overline{\text{XXX}}$ means active low and N/C means not connected.

In Table 4.2 the most important analog and digital channels and pin connections are shown. It can be seen that A_IN2 is occupied by the digital microphones, but A_IN1 is accessable through the line input and the right channel of A_IN3 through the headset connector. It is also possible to access the left channel of A_IN3 by soldering a wire to the testpads enabling four channels, but as only three channels are required at the moment, this will not be necessary. The circuitry of the line input and the headset input are not the same, but can made so be attaching some external components to headset input.

4.1.3 Raspberry Pi

The Raspberry Pi (Model B) uses a Broadcom BCM2835 system on a chip, the CPU is a ARM1176JZF-S running at 700 MHz with 512 MB RAM. The BCM2835 include a VideoCore IV GPU which support 1080p over HDMI, but can also be used for parallel computing if that will be required in order to run the program in real-time. Several Linux based operating systems have been made compatible with the Raspberry Pi, but Raspbian is the closest to an official release and has the most GPIO and floating point calculation features available, which are the determining factors in this case. Raspbian is based on Debian which is a well known Linux distribution and Python tools are well supported and documented.

An image of a Raspbian build including the driver can be downloaded from the makers of the audio card at the element 14 community [Wolfson Audio and Element 14, 2014]. This image will be used as the basis for the development of the system. A few procedures has been found for compilation of the kernel to use with a fresh system that does not include a lot of irrelevant software, but none were able to boot after installation.

As it can be seen from Figure 4.1 the audio data is transferred over the I2S bus as PCM. It is done through AIF1 on the wolfson audio card which support up to 8 channels. Unfortunately the hardware I2S bus on the Raspberry Pi only supports stereo and therefore only 2 channels. Because the I2S bus on the Raspberry Pi supports up to 32 bit samples for each channel, it may be possible to circumvent this problem by transferring 4×16 bit samples and reading them with 2×32 bit samples. Unfortunately the framesize on the Raspberry Pi is adjusted to fit the sample size of the channels, which means the registers on the audio card must be changed without using the audio driver on the Raspberry Pi. The registers should then be accessed through the SPI control interface of which it has two, meaning two chip selects SPI0-0 and SPI0-1. The chip select of SPI0-0 is attached to reset on the WM8804 SPDIF transiever and SPI0-1 to the WM5102 codec. When installing the driver for the audio card, the SPI0-1 device as seen by Raspbian is seized by the driver and made unavailable to the user, making it virtually impossible to access the audio card without using the driver. It may however be possible if the I2C bus on the expansion header is connected to the I2C bus used to control the WM8804, but this has yet to be attempted.[Raspberry Pi, 2014]

4.2 Software

This section will describe the software implemented on the hardware described in Section 4.1 or more specifically on the Raspberry Pi. Figure 4.4 show a flow chart of the major processes which will be executed during the initialisation phase and the actual locating of the source.

A large part of the hardware initialisation is described during Section 4.1. As long as the provided Raspbian image is used, the basic setup of the codec and peripherals is taken care of in the upstart of the operating system. As the provided image does not support more than 2 channel audio over the I2S bus natively, a reconfiguration of the bus is required to expand beyond the 2 channels. For the time being the method is reduced to finding the direction of the source as this requires only 2 channels. The analog section of the hardware requires no initialisation exept for an optional calibration of the microphone preamplifiers.

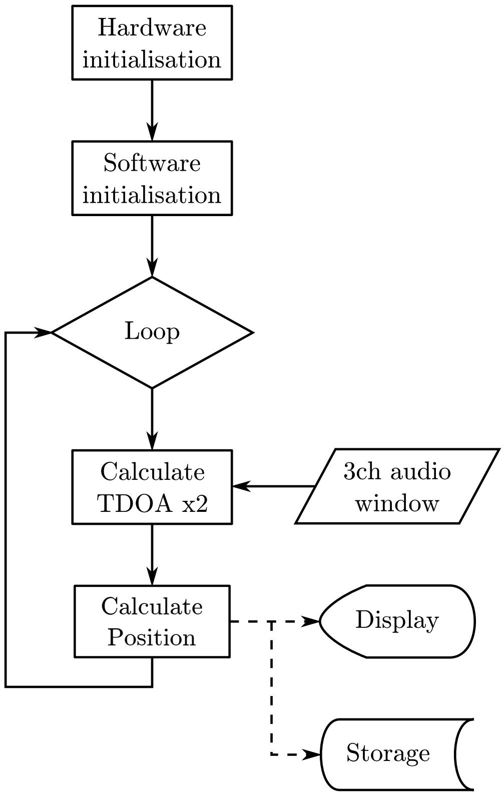


Figure 4.4: Flow chart of when the internal processes of the software is executed.

The software initialisation consists mainly of:

- Preset microphone positions
- Preset window size
- Calculate/preset frequency range
- Configure interface with soundcard

The positions of the microphones are determined by the physical setup and is simply specified for the software. The window size is also a predetermine value which is specified. The frequency range can be partially or fully specified, depending on whether a limited range or the maximum determined by the separation of the microphones is desired. The interface with the soundcard is achieved throught the PyAudio package, which has both recording an playback capabilities, although only recording is utilised.

The configuration of PyAudio is relatively easy and is explained in the following code excerpt:

```python
# First the library is imported
import pyaudio

# Default configuration variables
T = 5
BLOCK_SIZE = 2400
FORMAT = pyaudio.paInt16
CHANNELS = 2
RATE = 48000
DEVICE = 1

# Then the interface is opened
p = pyaudio.PyAudio()

# Then an input stream is opened
stream = p.open(
    format = FORMAT,
    channels = CHANNELS,
    rate = RATE,
    input = True,
    frames_per_buffer = BLOCK_SIZE,
    input_device_index = DEVICE
)
```

Most of the above configuration variables are self-explanatory, however the device number `DEVICE` is dependant on the hardware and selects the soundcard to use. The `BLOCK_SIZE` determines the size of the buffer and can e.g. be set to the length of the window that will be processed or if the windows are overlapping a readout will be convenient every half of a window length. For 0.1 s overlapping window at 48 kHz, a `BLOCK_SIZE` of 2400 would fit. The retrieval of data can be done in a way similar to the one stated below.

```python
# Run for "T" seconds
for i in range(T*RATE/BLOCK_SIZE):
    # Data is retrieved with the function call
    data = stream.read(BLOCK_SIZE)

    # Processing of data will be done here

# Once the recording is done, the interface is closed
stream.stop_stream()
stream.close()
p.terminate()
```

In the above example the code will run for 5.0 s returning data every 50 ms. When the `stream.read()` function is called it waits until the buffer is filled up and then returns the data. This makes real-time processing straightforward, as it basically works as an interrupt, called whenever a window of data is ready. The `stream.read()` function simply returns a the audio data as interlaced samples, formated as a string. With a reformating to integers or floats and a reshaping of the data to a 2x2400 array, the processing can start. The main processing function is the calculation of the TDOA, shown here with pseudo code including both TDOAs. The matter of assembling the two halfs of the window, selecting the desired frequencies and whether the signal level is above the desired threshold has been left out.

```
# DFT of the three input signals from the microphones
X1 = dft(x1)
X2 = dft(x2)
X3 = dft(x3)

# The phase difference is calculated
ph_diff_12 = angle(X1*conj(X2))
ph_diff_23 = angle(X2*conj(X3))

# The phase difference is converted to a TDOA
tdoa_12 = ph_diff_12/(2*pi*f)
tdoa_23 = ph_diff_23/(2*pi*f)

# A weighting is made from the DFT of x2
weight = abs(X2)
weight = weight/sum(weight)

# The weighted mean of the TDOA is calculated
mean_tdoa_12 = sum(tdoa_12*weight)
mean_tdoa_23 = sum(tdoa_23*weight)

# The two TDOA are returned
return array([mean_tdoa_12, mean_tdoa_23])
```

The weight is only based on microphone 2, as it is assumed the frequency content is the same for all three microphones. These calculations are essentially the same as the ones described by Figure 3.2 and as stated in Section 3.0.3, the calculations are executed for each window, returning a TDOA for each microphone pair. These two TDOAs are then handed over to the function determining the intersection of the corresponding hyperbolas. The function is just an evaluation of Equation (2.11) and (2.9), returning an (x, y) coordinate. Using the 2 channel input only, the evaluation is reduce the functionality of finding the direction through Equation 2.4.

The results of this chapter can be summarised to being: the construction of a hardware platform which is able to record two channels instead of the intended three channels. The bottleneck is the bus used for transferring data between the codec and the computer. This causes the implementation of the software to be limited to using only two inputs. The software is then regressed to only determining an angle of the source.

Chapter 5

Conclusion and Discussion

This chapter contains the conclusion which will state the major results of the information presented throughout the report and the discussion which will describe further work needed to realise some of the goals that were not fulfilled during the project.

5.1 Conclusion

The overall goal of the project was to be able locate an arbitrary source in an arbitrary environment. As this was considered beyond the frame of a semester, the goal was changed to be able to locate a talking person or a source with the same characteristics. The place for where to locate this source was constrained to an anechoic environment with minimum noise and no unwanted sources. Beside being able to determine the position of a stationary source, the method was expected to be able to locate a moving source in an online situation. It was also decided to use limited processing power and thereby reduce the demands for the platform on which the implementation would take place. To expand on that goal, the method to be used was decided to be mainly analytical instead of nummerical.

Of these goals only one did not entirely succeed and that was the implementation of the method on the hardware platform. The remaining goals did however succeed. In the developing stage of the method, it managed to determine the angle of a stationary source, the position of a stationary source and the position of a moving source. The underlying theory used to create the method was multilateration, a method for locating a source with multiple sensors using the TDOA (Time Difference Of Arrival) of the signal between the sensors. To determine the angle of the source two microphones were used. This yielded one TDOA, which could be used to create a hyperbola that describes all the possible positions the source can have, with the given TDOA. The asymptote of this hyperbola could then be used as a vector pointing in the direction of the source, at least for sources adequately far away from the microphones.

To determine the 2D position of the source a total of three microphones was required. From three microphones, it was possible to describe three hyperbolas, which all contained feasible positions of the source. Finding the intersection of just two of the three hyperbolas provides a unique position which can describe the location of the source. To increase the performance of the 2D method, a weighted mean was integrated, for when not all of the frequency range was used by the source. A threshold was also included for determining whether the source is active and therefore if a valid location can be obtained. The 2D method was tested with various simulations such as an ideal case with no disruptive sources, background noise or reflections to disturb the method. Similar measurements were carried out to verify the validity of these simulations. The simulations were indeed verified and additional simulations were made to study the effect of imperfect environments. These simulations include reflections from walls or other objects, sources with signals independent from the main source, such as the noise from a fan in the ceiling or hum from some household appliance and finally, internal noise, which is independent for each channel and could e.g. originate from the microphones themselves. Each of the disruptive sources changed the results of the method in their own way and to different degrees, however the internal noise proved the most destructive, then the independent sources and lastly the reflections, which turned out to be the least destructive.

To locate a moving source the 2D method was expanded to include a windowing of the signal, thereby in time-steps being able to estimate the position within this window. This was simulated with a source moving at a speed up to 1.4 m/s or 5 km/h, which was estimated to be the average comfortable speed of a person. The method showed good results for a source moving in a circle, between 2.5 and 5.5 m away from the microphones, resulting in an accuracy op to 1.4 cm and down to 41.5 cm, depending on the speed of the movement and the type of signal used to represent the source.

A hardware platform was developed using a Raspberry Pi as the processing unit and expanded with a Wolfson Audio Card which was used for capturing the signal from the electret microphones that were selected for the purpose. In order to get a large enough signal from the microphones, a microphone preamplifier was designed. The preamplifier included the powering of the electret microphones, a differential input and a gain from $-\infty$ to $+64$ dB. The platform was able to function with two channels where three were expected. This limited the amount of information the method had available and therefore also its functionality. Since only two channels were available, the implementation of the method could only determine the angle of the source. The bottleneck limiting the number of channels to two was the connection between the Raspberry Pi and the Wolfson Audio Card and more specifically the hardware from the Raspberry Pi side of the connection, which only supports stereo. Although the hardware platform did not succeed in supporting the three channels that were expected, measurements were possible though a 01dB Harmonie measurement system, using high grade measurement microphones. It was however not done as an online measurements forcing the method to work from a recording located on the filesystem.

5.2 Discussion

In the conlusion it is stated that all but one goal was reached. Eventhough a goal was reached it may still be possible to improve on several of the result, obtained in the process of reaching those goals. Different approches could be taken to receive an improved outcome, while others may produce inferior or indifferent outcomes, nevertheless several possibilities have not been explored during the development of this project.

First of all a hardware platform working with 3+ inputs would be a large improvement, as it would open up the possibility of online execution of the 2D locating method. The obstacle in obtaining that, is the hardware of the Raspberry Pi not supporting multichannel audio, over the connection to the Wolfson Audio Card. It does however support up to 32 bit samples for both the stereo channels and if it can be mislead into receiving two 16 bit samples as one 32 bit sample, the number of channels would be increased to four. To do so, the driver controlling the audio card and the communication between the two devices, would have have to be circumvented, so a manual setup of the codec and the communication can be done. Using the I2C bus as a control bus may be the key to solving that problem.

Regarding the locating method, many choises could have been made differently. One thing is the use of the weighted mean. The weighted mean could have been implemented differetly using e.g. the logarithmic or rectangular shape and thereby potentially improve the performance. Instead of the weighted mean, a linear regression of the phase could be used, before it is converted to a time value. This will result in the coefficients for the straight line, which is fitted to the given phase data. These coefficients may be converted more easily to a time value than an array, which afterwards would have to be averaged. An entirely different approach would be to not convert the result of the individual frequencies to a singular value. Instead the frequencies with an adequate level could be converted to an individual position. Using many frequencies this would likely end up in a clustering of estimates at various places. If multiple sources were present with signal content at different areas of the frequency range, the clustering could be used to determine the position of several sources.

The setup of the microphones could have been done in many ways. The use asymptotes to find the intersection, instead of the hyperbolas themselves, would enable the setup of the microphones in an almost arbitrary shape. The two main shapes that would be interesting to further investigate, is the L-shape, as explained in Section 2.2.2, where the microphone-axes are orthogonal and can therefore be used to find the intersection of the hyperbolas, in the same manner as it is currently done. The second is an equilateral triangular shape, where the lengths of the three sides are equal. This would mean equal separation of the microphones and easy implementation of the third microphone pair. It also means that the microphone-axes are non-parallel and non-orthogonal and would therefore require the use of the asymptotes instead of the hyperbolas.

Bibliography

Richard W. Bohannon. Comfortable and maximum walking speed of adults aged 20—79 years: reference values and determinants. *Age and Ageing*, 26(1):15–19, 1997. URL `http://dx.doi.org/10.1093/ageing/26.1.15`.

Adelbert W. Bronkhorst and Tammo Houtgast. Auditory distance perception in rooms. *Nature 397*, February 1999. URL `http://dx.doi.org/10.1038/17374`.

H. Steven Colburn. *Computational Models of Binaural Processing*, volume 6 of *Springer Handbook of Auditory Research*. Springer New York, 1996. ISBN 978-1-4612-8487-1. URL `http://dx.doi.org/10.1007/978-1-4612-4070-9_8`.

F. Gustafsson and F. Gunnarsson. Positioning using time-difference of arrival measurements. *Proceedings. (ICASSP '03). 2003 IEEE International Conference*, pages VI–553–6 vol.6, April 2003. ISSN 1520-6149. URL `http://dx.doi.org/10.1109/ICASSP.2003.1201741`.

IEC. DS/EN 61938-1: Audio-, video and audiovisual systems - interconnections and matching values - preferred matching values of analogue signals, 1997. URL `http://webshop.ds.dk/da-dk/standard/ds-en-61938corr-1997`.

A.R. Jiménez and F. Seco. Precise localisation of archaeological findings with a new ultrasonic 3d positioning sensor. *Sensors and Actuators A: Physical*, pages 224 – 233, 2005. ISSN 0924-4247. URL `http://dx.doi.org/10.1016/j.sna.2005.03.064`. Eurosensors {XVIII} 2004 The 18th European conference on Solid-State Transducers.

Don H. Johnson and Dan E. Dudgeon. *Array Signal Processing: Concepts and Techniques*. Simon and Schuster, 1992. ISBN 0130485136.

L.E. Kinsler. *Fundamentals of acoustics*. Wiley, 2000. ISBN 9780471847892.

Charles Lee and Frank Ling. Your genes your brain, 2014. URL `https://archive.org/details/groks631-1`.

Esben Madsen, Søren Krarup Olesen, Milos Markovic, Pablo F. Hoffmann, and Dorte Hammershøi. Setup for demonstrating interactive binaural synthesis for telepresence applications. *Acustica United with Acta Acustica*, 97(Supplement 1):S 90, 2011. ISSN 1610-1928.

Torben Poulsen. *Acoustic Communication. Hearing and Speech. Version 2.0.* 2005. Lecture note number: 31230-05.

Raspberry Pi. User community, 2014. URL http://www.raspberrypi.org/documentation/.

Ashok Kumar Tellakula. Acoustic source localization using time delay estimation, thesis submitted for the degree of master of science, August 2007.

Wolfson Audio and Element 14. User community, 2014. URL http://www.element14.com/community/community/raspberry-pi/raspberry-pi-accessories/wolfson_pi.

Appendices

Appendix A

Derivation of hyperbola with Equal TDOA

In Figure A.1 an illustation of sound source eminating a signal which is picked up by two microphones. Because the distance from the source to each of the microphones are not the same, one of the signals will be delayed by that difference relative to the speed of sound also known as TDOA. This TDOA is the same for all points along a hyperbola which will be derived in this section.

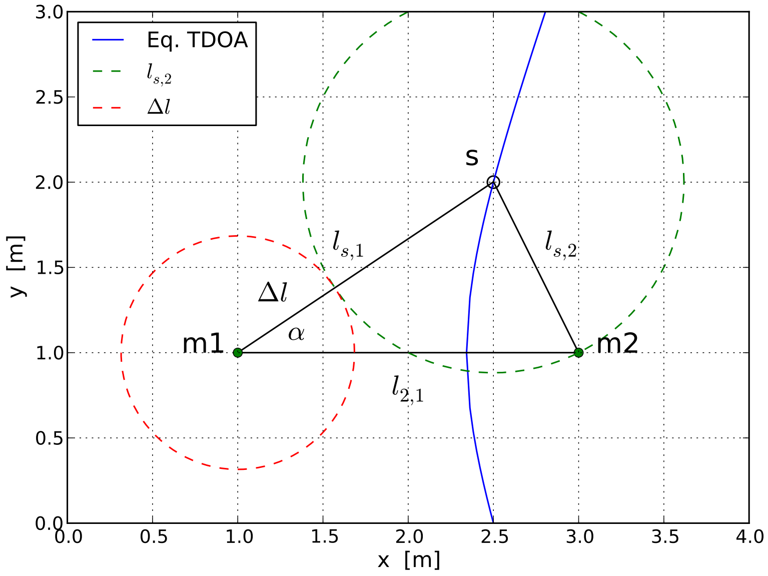


Figure A.1

A.1 Hyperbola

The hyperbola will be made relative to the center of the two microphones, so the microphones are placed at $\pm\frac{l_{2,1}}{2}$, where $l_{2,1}$ is the distance between microphone 1 and microphone 2. The distances between the source and microphone 1 and 2 will be denoted as: $l_{s,1}$ and $l_{s,2}$ respectively.

Let α be the angle between the axis of the two microphones and the line between the source and microphone 1. α can be described by the the law of cosines, when the length of all three sides of a triangle is known:

$$\begin{aligned}\cos(\alpha) &= \frac{b^2+c^2-a^2}{2bc}\\ \alpha &= \cos^{-1}\left(\frac{l_{s,1}^2+l_{2,1}^2-l_{s,2}^2}{2l_{s,1}l_{2,1}}\right)\end{aligned}$$

with α known, it is possible to get the x and y coordinate as:

$$\begin{aligned}x &= l_{s,1}\cos(\alpha)-\frac{l_{2,1}}{2}\\ &= l_{s,1}\cos\left(\cos^{-1}\left(\frac{l_{s,1}^2+l_{2,1}^2-l_{s,2}^2}{2l_{s,1}l_{2,1}}\right)\right)-\frac{l_{2,1}}{2}\\ &= l_{s,1}\frac{l_{s,1}^2+l_{2,1}^2-l_{s,2}^2}{2l_{s,1}l_{2,1}}-\frac{l_{2,1}}{2}\\ &= \frac{l_{s,1}^2+l_{2,1}^2-l_{s,2}^2}{2l_{2,1}}-\frac{l_{2,1}}{2}\\ y &= l_{s,1}\sin(\alpha)\\ &= l_{s,1}\sin\left(\cos^{-1}\left(\frac{l_{s,1}^2+l_{2,1}^2-l_{s,2}^2}{2l_{s,1}l_{2,1}}\right)\right)\end{aligned}$$

using the relation:

$$\sin(\cos^{-1}(z)) = \sqrt{1-z^2}$$

y can be written as:

$$y = l_{s,1}\sqrt{1-\left(\frac{l_{s,1}^2+l_{2,1}^2-l_{s,2}^2}{2l_{s,1}l_{2,1}}\right)^2}$$

$l_{s,1}$ is always positive, so y can be written as:

$$= \sqrt{l_{s,1}^2 \left(1 - \left(\frac{l_{s,1}^2 + l_{2,1}^2 - l_{s,2}^2}{2l_{s,1}l_{2,1}}\right)^2\right)}$$
$$= \sqrt{l_{s,1}^2 - \left(\frac{l_{s,1}^2 + l_{2,1}^2 - l_{s,2}^2}{2l_{2,1}}\right)^2}$$

Let Δl be the difference in length between $l_{s,1}$ and $l_{s,2}$:

$$\Delta l = l_{s,1} - l_{s,2}$$

Δl can however also be related directly to the TDOA, but is left out this derivation.

$$\Delta l = \tau_{TDOA} \cdot c$$

Isolating $l_{s,1}$:

$$l_{s,1} = l_{s,2} + \Delta l$$

where c is the speed of sound. $l_{s,1}$ is then inserted into x and y which are then simplified:

$$\begin{aligned}
x &= \frac{(l_{s,2} + \Delta l)^2 + l_{2,1}^2 - l_{s,2}^2}{2l_{s,2}} - \frac{l_{2,1}}{2} \\
&= \frac{\left(l_{s,2}^2 + \Delta l^2 + 2l_{s,2}\Delta l\right) + l_{2,1}^2 - l_{s,2}^2}{2l_{2,1}} - \frac{l_{2,1}}{2} \\
&= \frac{\Delta l^2 + 2l_{s,2}\Delta l + l_{2,1}^2}{2l_{2,1}} - \frac{l_{2,1}^2}{2l_{2,1}} \\
&= \frac{\Delta l^2 + 2l_{s,2}\Delta l}{2l_{2,1}} \\
y &= \sqrt{(l_{s,2} + \Delta l)^2 - \left(\frac{(l_{s,2} + \Delta l)^2 + l_{2,1}^2 - l_{s,2}^2}{2l_{2,1}}\right)^2} \\
&= \sqrt{(l_{s,2} + \Delta l)^2 - \left(\frac{\left(l_{s,2}^2 + \Delta l^2 + 2l_{s,2}\Delta l\right) + l_{2,1}^2 - l_{s,2}^2}{2l_{2,1}}\right)^2} \\
&= \sqrt{(l_{s,2} + \Delta l)^2 - \left(\frac{\Delta l^2 + 2l_{s,2}\Delta l + l_{2,1}^2}{2l_{2,1}}\right)^2}
\end{aligned}$$

$l_{s,2}$ can then be isolated from x:

$$l_{s,2} = \frac{2l_{2,1}x - \Delta l^2}{2\Delta l}$$

and inserted into y and simplified:

$$\begin{aligned}
y &= \sqrt{\left(\frac{2l_{2,1}x - \Delta l^2}{2\Delta l} + \Delta l\right)^2 - \left(\frac{\Delta l^2 + 2\Delta l \frac{2l_{2,1}x - \Delta l^2}{2\Delta l} + l_{2,1}^2}{2l_{2,1}}\right)^2} \\
&= \sqrt{\left(\frac{2l_{2,1}x + \Delta l^2}{2\Delta l}\right)^2 - \left(\frac{2l_{2,1}x + l_{2,1}^2}{2l_{2,1}}\right)^2} \\
&= \sqrt{\frac{1}{4}\left(\frac{2l_{2,1}x}{\Delta l} + \Delta l\right)^2 - \frac{1}{4}\left(2x + l_{2,1}\right)^2}
\end{aligned}$$

the squared parentheses are expanded and simplified

$$\begin{aligned}
&= \frac{1}{2}\sqrt{\left(\left(\frac{2l_{2,1}x}{\Delta l}\right)^2 + \Delta l^2 + 2\frac{2l_{2,1}x}{\Delta l}\Delta l\right) - \left(4x^2 + l_{2,1}^2 + 4l_{2,1}x\right)} \\
&= \frac{1}{2}\sqrt{4\left(\frac{l_{2,1}x}{\Delta l}\right)^2 + \Delta l^2 - 4x^2 - l_{2,1}^2}
\end{aligned}$$

factorising this leaves the final equation:

$$y = \frac{1}{2}\sqrt{\frac{(4x^2 - \Delta l^2)(l_{2,1}^2 - \Delta l^2)}{\Delta l^2}} \tag{A.1}$$

As Δl is always:

$$l_{2,1} \geq \Delta l \geq -l_{2,1}$$

it is required that:

$$|x| \geq \frac{\Delta l}{2}$$

A.2 Asymptote

The asymptote of the hyperbola can be found by rewriting Equation (A.1) to the standard form of a hyperbola:

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \tag{A.2}$$

where a and b is found to be:

$$a = \frac{\Delta l}{2} \tag{A.3}$$

$$b = \frac{\sqrt{l_{2,1}^2 - \Delta l^2}}{2} \tag{A.4}$$

The equation of the asymptote is then given by:

$$y = \frac{b}{a}x = \frac{\frac{\sqrt{l_{2,1}^2 - \Delta l^2}}{2}}{\frac{\Delta l}{2}}x \tag{A.5}$$

$$= \frac{\sqrt{l_{2,1}^2 - \Delta l^2}}{\Delta l}x \tag{A.6}$$

Appendix B

Measurement Journal

Measurement of a Stationary Source

Purpose

The purpose of this measurement is to determine wether a measurement of a real world scenario will yield the same result as the simulations. The senario will be an ideal environment, with a very low noise floor and where no reflections are encountered.

List of Equipment

Type	AAU-number	Brand	Model
Microphone 1	75535	G.R.A.S.	40AZ
Microphone 2	75521	G.R.A.S.	40AZ
Microphone 3	75530	G.R.A.S.	40AZ
Pre-amplifier 1	75566	G.R.A.S.	26CC
Pre-amplifier 2	75554	G.R.A.S.	26CC
Pre-amplifier 3	75556	G.R.A.S.	26CC
Calibrator	08373	B&K	4230
USB-soundcard (playback)	-	Lexicon	Alpha
PC (playback)	-	Lenovo	Thinkpad X200t
Acquisition-unit (recording)	56524	01dB	Harmonie
PC (recording)	47220	HP	OmniBook 6000
Speakers	02017-44,46,48	Vifa	MD10 Ball
Amplifier	33981	Rotel	RB-976 MkII

Table B.1: List of equipment.

Measurement Setup

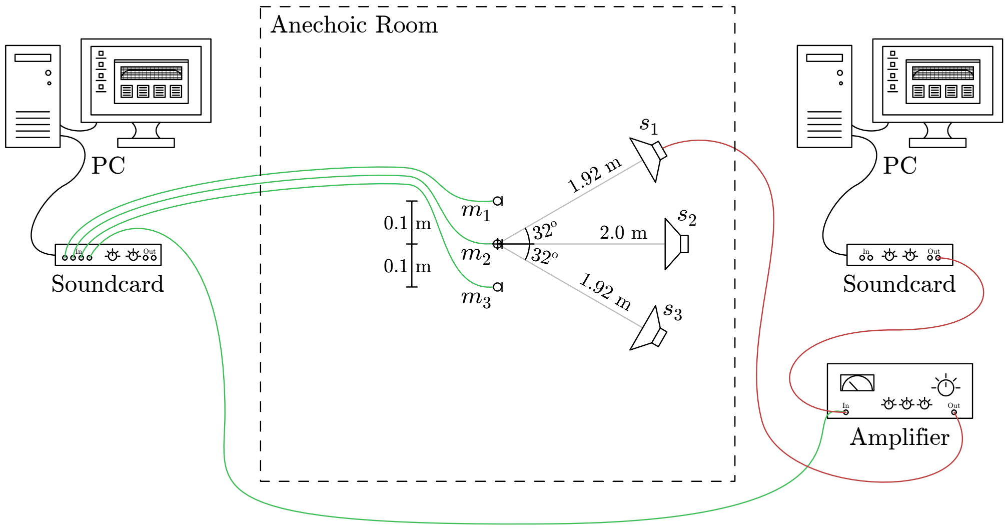


Figure B.1: Measurement setup in the ideal case with a very low noise floor and where no reflections are encountered.

In Figure B.1 the setup can be seen. The green lines illustrate the inputs and red lines illustrate output. The three speaker positions are shown. The angle of the incoming signal is aproximately 32° to either side of the microphone normal for the two side speakers. The three positions of the speakers are based on a permanent setup in the anechoic room and fits well with the pupose of this setup. The center speaker is placed at a distance of 2.0 m from the center of the microphones, perpendicular to the axis of the microphones. The two side speakers are placed aproximately 1.0 m to each side on the x-axis and 40 cm closer to microphones on the y-axis (They are not completely fastened, letting them move a little). Converting these Cartesian coordinates to polar, places the side speakers ±32° with a radius of 1.89 m.

The distance between the microphones is set to 0.1 m and a measurement with each of the three speakers is done. The microphone axis is then rotated 16° to the right and the same three measurements are carried out. This will produce another 3 source positions with angles: 48°, 16° and -16°. As stated by Equation (2.5), a distance of 0.1 m corresponds to a maximum frequency of 1715 Hz for the algorithm.

Method

The equipment is connected as shown in Figure B.1. Even though the sensitivity of the microphones has no direct impact measurements, a calibration of the microphones is carried out. The level of the amplifier is set to an adequate level, a high SNR is desired to avoid the background noise eventhough it is almost non-existent. THD should of course be avoided when adjusting the level. A pre measurment show 62 $\mathrm{dB_{SPL}}$ at microphone m_2 for 1 kHz at the maximum level without THD.

In order to determine the effectivenes of the algorithm, three diffents signals are tested. First an idealised signal comprised of 843 sinusoids linearly distributed between 30 and 1725 Hz. Each sinusoid has been given a random offset to avoid have a signal with large peaks at intervals. The second signal type is white noise and is chosen because it represents the whole frequency range and is an adequate representation of various types of sounds such as speech and music. Lastly, an excerpt of an interview from a radio show [Lee and Ling, 2014] is used to show the effect of actual speech. The full radioshow is published under the Creative Commons licence BY-NC-SA 3.0, allowing noncommercial use and distribution. The three signals can be found on the project CD `Measurements/Sourcesignals` and in Figure B.2 the signals are shown. It may be noted that the signals containing white noise and sinusoids have been normalised to 0.5 and the speech has been normalised to 1.0. The two former signals contain more energy and are therefore lowered to create a more equal measurement.

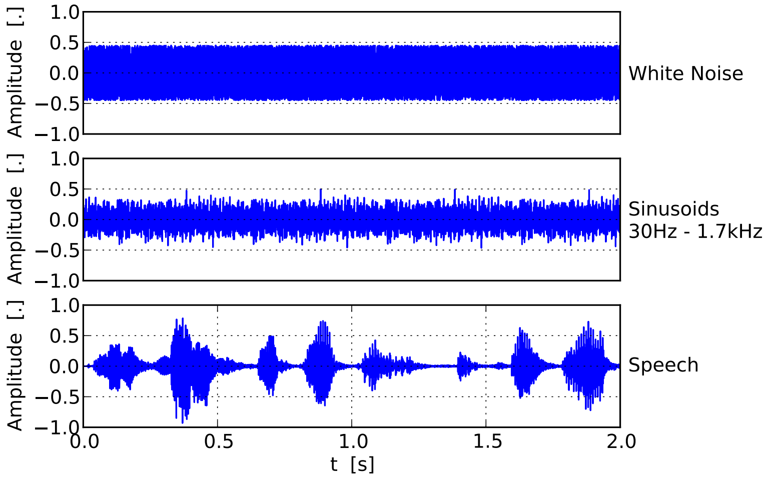


Figure B.2: Test signals. (The signals in the figure has been downsampled and may appear different from the originals.)

In Figure B.3 one of the 18 measurements is shown. The source signal-type in the figure is white noise, the other recordings look similar and are available on the project CD `Measurements/`.

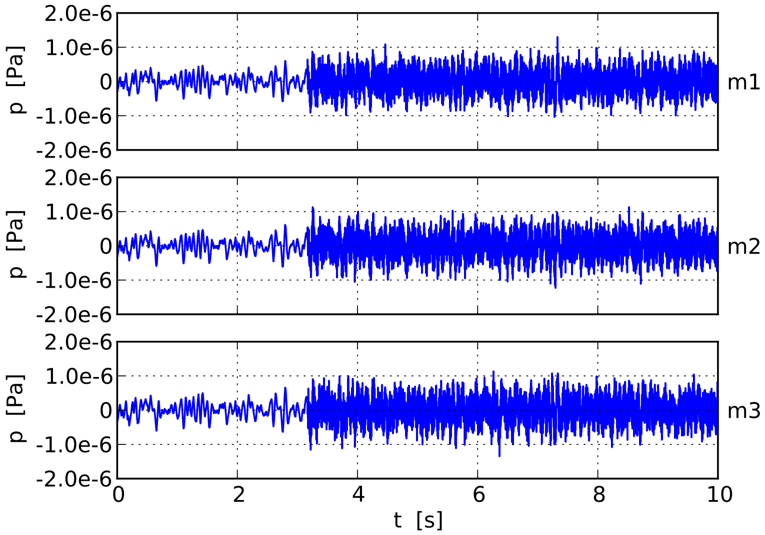


Figure B.3: 10 s sample of a raw recording for all three microphones with the source placed 1.92 m away at an angle of +16°. (The data has been downsampled.)

As it can be seen in Figure B.3 the measurement contain some low frequency background noise. This is a result of the anechoic properties of the room being designed to attenuate down to only 200 Hz. The Harmonie system is setup to highpass filter at 10 Hz, but does not remove it all. An investigation into the frequency domain of the background noise reveals a large amount below 150 Hz, the frequency analysis can be seen in Figure 2.30.

A highpass filterering of the incoming signal removes the major part of the noise. Figure B.4 show the signal filtered with a 4th order Butterworth filter with a cutoff at 80 Hz, this also attenuate some of the 50 Hz hum which also can be seen in Figure 2.30.

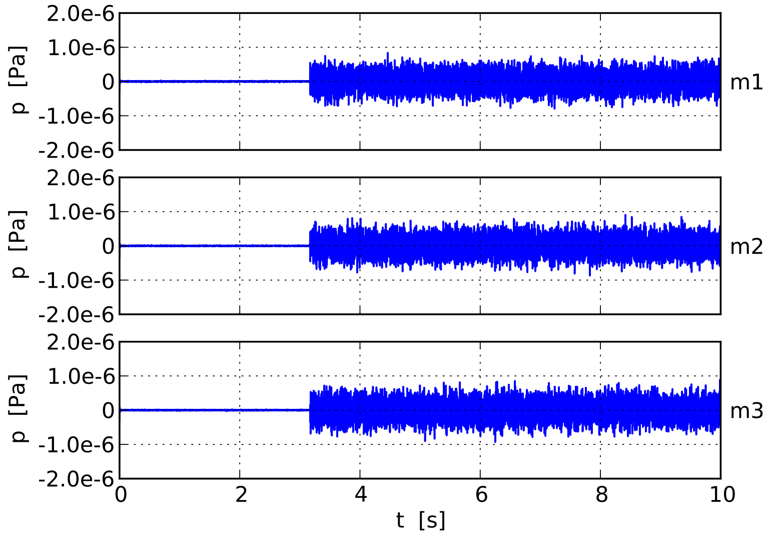


Figure B.4: High-pass filtered version of the signal in Figure B.3. (The data has been downsampled.)

A 2.5 ms exerpt of the signal shown in Figure B.3 can be seen in Figure B.5.

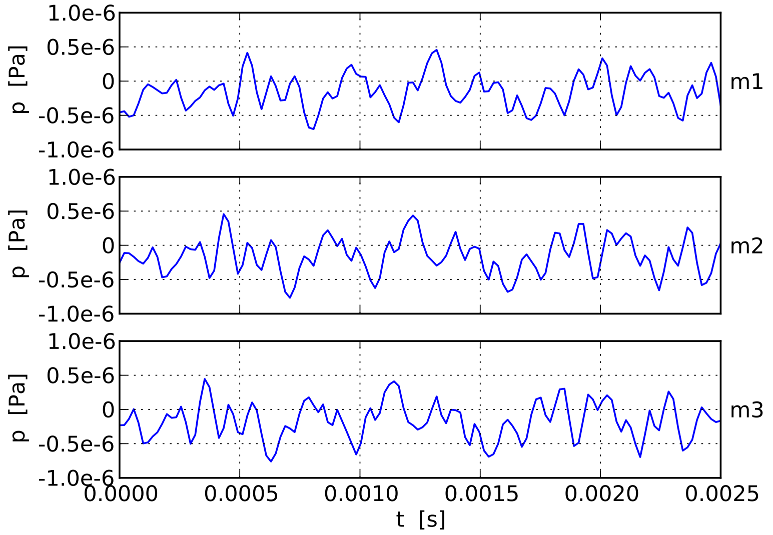
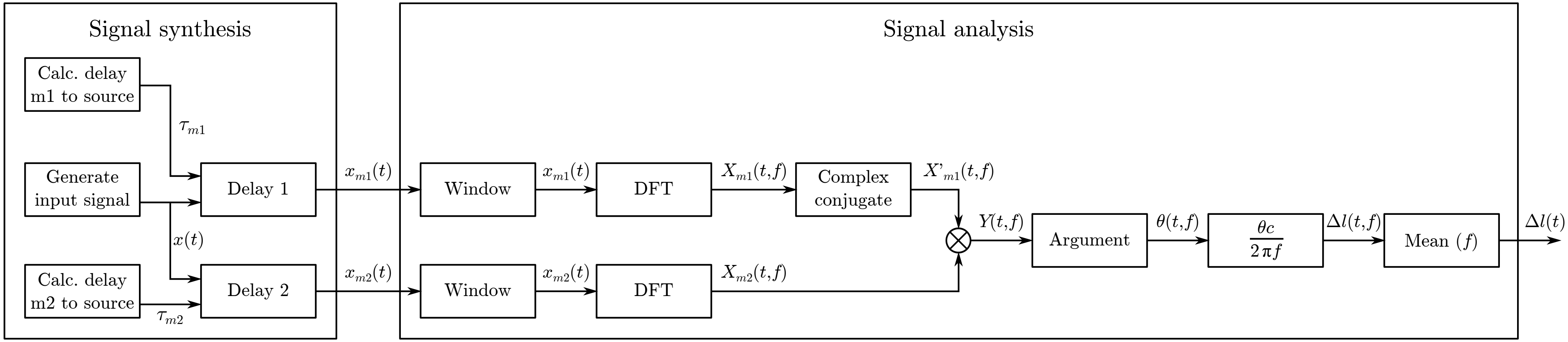


Figure B.5: 2.5 ms excerpt of the signal shown in Figure B.3.

It can here be seen that the signal arrives at microphone 3 first, it the reached microphone 2 and lastly microphone 1. The delay between microphone 3 and 2 is smaller than between 2 and 1, which corresponds well with a spherical propagation.

Appendix C

Block Diagram of TDOA Calculation for Moving Sources


Signal synthesis
Calc. delay m1 to source
τ_{m1}
Generate input signal
$x(t)$
Calc. delay m2 to source
τ_{m2}
Delay 1
Delay 2
$x_{m1}(t)$
$x_{m2}(t)$
Signal analysis
Window
$x_{m1}(t)$
DFT
$X_{m1}(t,f)$
Complex conjugate
$X'_{m1}(t,f)$
Window
$x_{m2}(t)$
DFT
$X_{m2}(t,f)$
$Y(t,f)$
Argument
$\theta(t,f)$
$\frac{\theta c}{2\pi f}$
$\Delta l(t,f)$
Mean (f)
$\Delta l(t)$

Appendix D

Errors in Simulation of Moving Sources

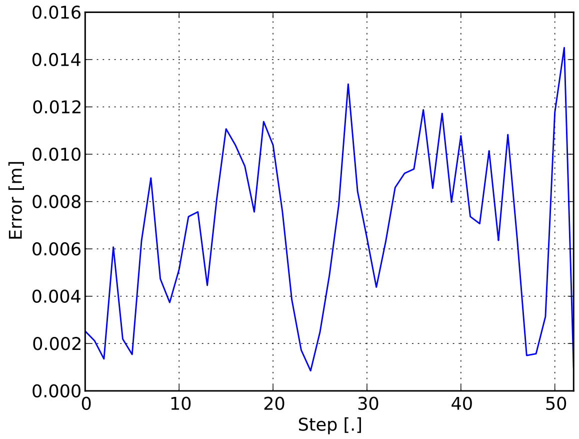


Figure D.1: Error in the estimate when simulation a moving source with sinusoids as input and a speed of 0.7 m/s.

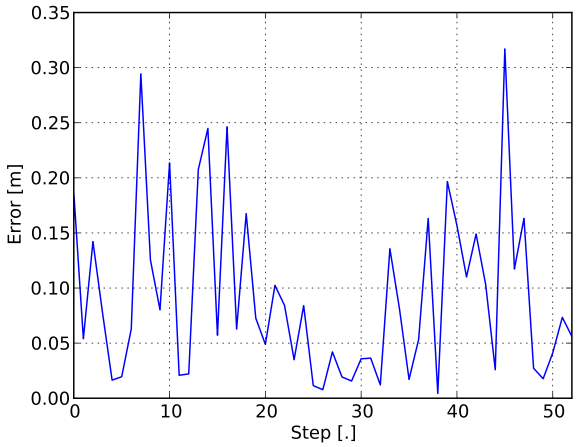


Figure D.2: Error in the estimate when simulation a moving source with white noise as input and a speed of 0.7 m/s.

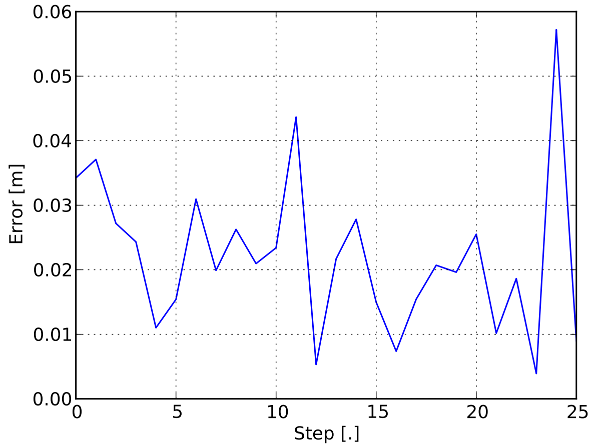


Figure D.3: Error in the estimate when simulation a moving source with sinusoids as input and a speed of 1.4 m/s.

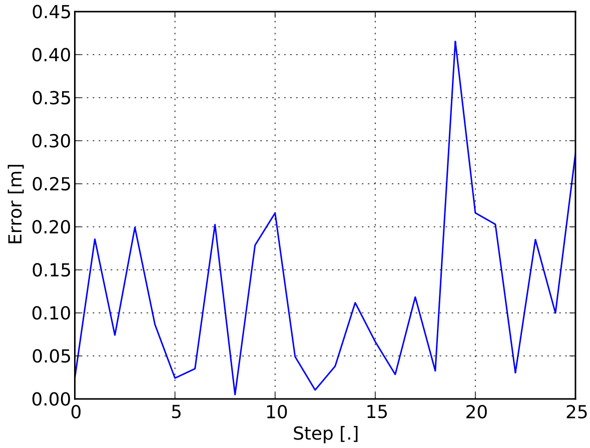


Figure D.4: Error in the estimate when simulation a moving source with white noise as input and a speed of 1.4 m/s.