Sound ranging using multilateration and Kalman filter

Tobias Samuelsson

LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6116
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2020

Abstract

Sound ranging is a method to locate sources of sound waves using microphones or other types of receivers at known reference positions. Nowadays, these receivers can be made both very small and powerful, lasting for a long time on a single battery charge, making them suitable for longtime outdoor purposes such as detecting artillery fire or other loud sounds.

In this master thesis, we present a sound ranging algorithm based on three-dimensional multilateration, Kalman filtering and sound detection. Results from simulations are given. Results from field experiments are presented. Furthermore, we compare the two and and evaluate their differences. Finally, suggestions on future work are provided.

The evaluation shows that while the Kalman filter and detection algorithm performs well even with high levels of measurement noise, the multilateration algorithm can provide an accurate source positioning, but it is very sensitive to errors, with performance degrading heavily due to low sampling resolution, too close node positioning, undefined sound peaks and a lack of robustness in the multilateration algorithm.

Acknowledgements

I would like to thank my academic supervisor Anton Cervin at the Department of Automatic Control at LTH for his time and dedication throughout this project. I would also like to thank OCTECH AB, who made this master thesis possible, for motivation and ideas as well as many interesting discussions. Thank you also to Anna Espmarker at Fontänhuset. A final thank you goes to my family and friends for their support.

Contents

1

Background

1.1 Introduction

Sound ranging is a method of determining the position of an unknown sound source by recording the sound with several receivers at different positions. The science of locating a sound source was originally developed around the First World War, and is intricately tied to its military applications. During the First World War, this was put to use in locating artillery fire, and rapidly developed from the efforts of among others, Nobel Laureate Lawrence Bragg.[Kloot, 2005]

The same principle of detecting electromagnetic waves was used to detect the direction of radio transmitters, so called radio direction finders. During the Second World War, this became crucial with regards to radio navigation, e.g. when directing bombers to enemy territory. One of these radio navigation systems was LORAN, developed by the United States and based on a hyperbolic system with fix radio beacons with known positions and an unknown receiver position. It was largely obsoleted by satellite based systems such as GPS (Global Positioning System) in the early 1990s.[Bowditch, 2019]

GPS makes use of satellites, synchronized with each other, that continually transmit radio signals about their position and current time. When these radio signals reach a receiver on ground, the difference between these times (TOT - time of transmission) and the current time (TOA - time of arrival) of the receiver are called time of flight (TOF). From four satellites, a 3D coordinate of the receiver's position is calculated. [Blewitt, 1997]

With the continuing development of miniaturized low-voltage components, along with improved bandwidth and lower production costs, one can now use small, portable nodes, e.g. sensors or SoC (system-on-a-chip) solutions. These work both as a receiver and a transmitter, communicating with each other (or a main unit) via e.g. Bluetooth.

With the use of low-voltage RTC clocks (keeping track of time with a very low power usage, typically at 32 KHz), these nodes can receive data on "stand-by" continually over long time periods without needing a power supply. This makes them attractive for field purposes such as detecting and locating sudden sound events

(e.g. artillery fire). However, the RTC clock is sensitive to drift (where the clock frequency deviates from the specification), eventually causing the nodes to fall out of sync. Therefore some kind of continuous synchronization method between the nodes (or to a main unit) will be needed. In this thesis, the nodes transmit calibration pulses to a main hub, where the drift is estimated by a Kalman filter. [STMicroelectronics, 2008]

This thesis was done at OCTECH AB, who kindly supplied the field recordings from which the results in Chapters 4-5 were derived. OCTECH is a startup company formed in 2019, involved in the field of sound ranging and sound detection.

1.2 Scope and problem formulation

The purpose of this master's thesis was to find a method to determine the position of an unknown, high-volume sound event, first using multilateration to calculate the position of sound sources based on the differences in propagation time to several receivers, then using a Kalman filter to synchronize the nodes, and at last detecting the sound to determine the local time of arrival. The hypothesis is that the accuracy of the position estimates is mostly dependant on either the temporal resolution of the recording apparatus, or the physical position of the receivers.

Originally there were field tests that included the Kalman filter, but the hardware originally planned for this purpose proved very difficult to implement the filter on, as we needed not just to implement the RTC clock and detection algorithm, but also write software for a main hub (usually an Android phone) to enable synchronization. This proved too difficult and time-consuming, so this approach was abandoned and instead a set of common microphones were used, with the synchronization being done a posteriori.

1.3 Method

The thesis combines and applies a number of previously proposed algorithms to solve the sound ranging problem.

Multilateration Multilateration is a method of determining the position of a sound source using the difference in times of arrival (TDOAs) between the receiver nodes. Knowing the time of transmission is not needed, but depending on the number N of dimensions at least $N+1$ receivers are required. The solution of this problem is non-linear.

Kalman filter The Kalman filter is a way of estimating the amount of time error of the sensors so that they can be synchronized. This is done by transmitting calibration pulses from the sensor at regular intervals.

Detection The detection algorithm handles samples recorded by the receiver and determines on-the-fly if they are indicative of a loud sound event.

1.4 Outline

Chapter 2 describes the theoretical background of multilateration methods and how an estimated position is derived from the recorded samples. Chapter 3 covers the implementation of the methods from Chapter 2 along with results from the simulation examples in Matlab. Chapter 4 describes the design of the field tests along with the recording setup and conditions for each test. Chapter 5 presents the calculated position estimates from the field tests, along with comparisons to the actual positions. In Chapter 6, these results are discussed, with possible improvements and points to future work in Chapter 7.

1.5 Related work

Several previous studies concentrate on the non-linear solutions of the multilateration algorithm, e.g. Kitic et al and Larsson et. al. Other focus mainly on the synchronization aspect, e.g. Stefanski and Sadowski, Gröönroos et. al.

Stefanski and Sadowski proposes a variant of TDOA where the position of a continously moving (mobile) emitter object is tracked at random times by at least five spatially fix and asynchronous nodes, forming a set of non-linear equations that are solved by Taylor LS-algorithms and genetic algorithms, contrary to our approach where the nodes were independently synchronized to a main unit to reduce drift and convert their measured TOA to a common time system.[1]

Kitic et. al is a comparative study on several different approaches of the non-linear multilateration problem, presenting and evaluating methods such as maximum likelihood estimation, iterative solutions, convex interpretations and least squares cost functions. It does not focus on the synchronization aspect however.[2]

Grönroos et. al uses a similar multilateration approach to ours on a set of low-cost nodes consisting of Raspberry Pi computers. These nodes track FM radio signals, cross correlating each possible pair of nodes with regards to their measured TDOA values, contrary to our approach where we used a Kalman filter to reduce the drift of the nodes.[3]

Larsson et. al approaches the multilateration problem from a matrix theory point of view, tackling it as an eigenvalue problem.[4]

[1] Stefanski and Sadowski, 2018.
[2] Kitic et al., 2020.
[3] Grönroos et al., 2017.
[4] Larsson et al., 2019.

A very early pioneering study on the sound ranging problem was done by Harry Bateman in 1918, tackling sound ranging as a triangulation problem, also focusing on meteorological aspects such as temperature and wind.[5].

[5] Bateman, 1918.

2

Theory

Multilateration is based on measuring the times of arrival (TOA) among several receivers of a wave with known propagation speed, but unknown time of transmission. Most often the TOAs are used to form time differences of arrival (TDOA), using the TOA of one of the node as a reference node which we subtract from the other TOAs. This system of TDOA is called a hyperbolic system, or pseudo-range multilateration. Several previous approaches to solving this non-linear problem can be seen in Section 1.5. Other ways of determining position include triangulation, where the angles to the source are known. However, to acquire information about the angles, directional receivers are required, which our hardware did not support. The method below (from [Misra and Bucher, 2002]) was used as it seemed straight-forward to implement.

As we want to use our sensors in differing weather conditions, we have to contend with the possibility that their characteristics may change, inducing drift (different clock speed due to temperature differences) or offset (unavoidable due to clock restarting). A detection at cycle 1000 for different sensors is meaningless if we do not know the real clock frequency and how long the clocks have been running. In order to minimize the effects of these inhomogenities we need to estimate the "true" properties of each clock. This is done here using a Kalman filter.

2.1 Multilateration in 3D

Given a signal at unknown position r_s and a receiver at known position r_r, a general expression for the distance between then, given a sound wave propagating for an unknown time, is given by

$$r_r - r_s = v\tau \tag{2.1}$$

where τ is the propagation time and v is the speed of sound. After [Cramer, 1993], the speed of sound in air is mainly dependant on the temperature, and is approximated by

$$v_{air}(m/s) = 331.3 + 0.606 \cdot T \tag{2.2}$$

where T is the temperature in C°.

The following multilateration algorithm is based on [Misra and Bucher, 2002]. We introduce the following generic terms for any $i, j \in \{0,1,2,3\}$:

$$R_{ij} = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2)} - \sqrt{(x_j - x_s)^2 + (y_j - y_s)^2 + (z_j - z_s)^2)} \tag{2.3}$$

$$x_{ij} = x_i - x_j$$

$$y_{ij} = y_i - y_j$$

$$z_{ij} = z_i - z_j$$

By squaring all R_{ij} terms and separating the terms that contain the sensor distances to the source, we end up with a set of intersecting hyperboloids[1] that can be expressed as the intersection of two planes

$$y = Ax + Bz + C \tag{2.4}$$

$$y = Dx + Ez + F \tag{2.5}$$

This results in two separate solutions:

$$\hat{r}_+ = \begin{bmatrix} Gz + H \\ Iz + J \\ z = \frac{N}{2M} + \sqrt{(\frac{N}{2M})^2 - \frac{O}{M}} \end{bmatrix} \tag{2.6}$$

$$\hat{r}_- = \begin{bmatrix} Gz + H \\ Iz + J \\ \frac{N}{2M} - \sqrt{(\frac{N}{2M})^2 - \frac{O}{M}} \end{bmatrix} \tag{2.7}$$

where

$$\left\{ \begin{array}{c} G = \frac{E-B}{A-D} \\ H = \frac{F-C}{A-D} \\ I = AG + B \\ J = AH + C \\ K = R_{ik}^2 + x_i^2 - x_k^2 + y_i^2 - y_k^2 \\ + z_i^2 - z_k^2 + 2x_{ki}H + 2y_{ki}J \\ L = 2(x_{ki}G + y_{ki}I + 2z_{ki}) \\ M = 4R_{ik}^2(G^2 + I^2 + 1) - L^2 \\ N = 8R_{ik}^2(G(x_i - H) + I(y_i - J) + z_i) + 2LK \\ O = 4R_{ik}^2((x_i - H)^2 + (y_i - J)^2 + z_i^2) - K^2 \end{array} \right\}$$

There is no fool proof way of choosing the correct solution, as this depends on the situation. If one solution has a z-coordinate significantly < 0, then we choose the other solution.

[1] A hyperboloid is a surface generated by rotating a hyperbolic curve around one of its axes.

2.2 Kalman filter

Each of the sensors are independently synchronized to a main unit that acts as a reference clock. The clock circuits of the sensors are subject to changes in frequency due to changes in temperature and to a lesser degree humidity. This will cause the circuits to drift relative to the reference clock.

The following approach is based on [Oliveira Jr. et al., 2009]. The drift (relative difference between the sensor clock and reference clock) is modeled as a state D, which is the proportional deviation of the cycle length. Ideally $D = 0$. If too fast (too short length), $D < 0$. For example, $D = -0.1$ means that the cycle length is 10% too short (and vice versa for $D = 0.1$).

The offset (absolute difference between the sensor clock and reference clock) is modeled as a state T which ideally is 0. We now form a state space model:

$$\begin{bmatrix} \dot{T} \\ \dot{D} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T \\ D \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{2.8}$$

where w_1 and w_2 are the Gaussian process noise terms. Having measurements t^s of the sensor clock and t^r of the reference clock, the state vector x_k at time step k is defined as

$$x_k = \begin{bmatrix} T_k \\ D_k \end{bmatrix} \tag{2.9}$$

where

$$T_k = t_k^s - t_k^r \tag{2.10}$$

$$D_k = \frac{t_k^s - t_{k-1}^s}{t_k^r - t_{k-1}^r} - 1 \tag{2.11}$$

We are able to measure the momentary offset, and model this by:

$$y_k = H_k x_k + v_k \tag{2.12}$$

where

$$H_k = \begin{bmatrix} 1 \\ \Delta t^s \end{bmatrix} \tag{2.13}$$

is our observation matrix, with Δt^s being time step between updates of the sensor clock ("calibration pulses"), and v_k is the measurement noise.

The dynamics of the Kalman filter itself are described by

$$\dot{x} = Ax + Gw \tag{2.14}$$

where w is the process noise vector from Equation 2.8, assumed to be Gaussian with variance Q. G is the process noise transformation matrix.

The Kalman filter is updated each time step as follows:

$$\dot{x} = A\bar{x} \tag{2.15}$$

$$\dot{\bar{P}} = A\bar{P} + P\bar{A} + GQG^T \tag{2.16}$$

where (2.16) is the matrix Riccati equation, $\bar{x}$ and $\bar{P}$ are the estimates from the previous time step. The state transition matrix is given by

$$\emptyset_{k+1,k} = \begin{bmatrix} 1 & t_k^s - t_{k-1}^s \\ 0 & 1 \end{bmatrix} \tag{2.17}$$

resulting in

$$x_k = \begin{bmatrix} T_k \\ D_k \end{bmatrix} = \begin{bmatrix} 1 & t_k^s - t_{k-1}^s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T_{k-1} \\ D_{k-1} \end{bmatrix} \tag{2.18}$$

$$\bar{P}_k = \emptyset_{k+1,k}\hat{P}_{k-1}\emptyset_{k+1,k}^T + \begin{bmatrix} 1 & \frac{\Delta t}{2} \\ 0 & 1 \end{bmatrix} Q \begin{bmatrix} 1 & \frac{\Delta t}{2} \\ 0 & 1 \end{bmatrix} \Delta t \tag{2.19}$$

where Q is the process noise matrix:

$$Q = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \tag{2.20}$$

where σ^2 is the amount of process noise in each (estimated) state.

$$K_k = \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R)^{-1} \tag{2.21}$$

$$x_k = \bar{x}_k + K_k(y_k - H_k\bar{x}_k) \tag{2.22}$$

$$\bar{P}_k = (I - K_k H_k)\bar{P}_k \tag{2.23}$$

where R is our specified measurement noise. Depending on the amount of measurement noise, we can tune Q and R to select how much noise we wish to have in our state estimation.

3

Implementation of theory

To see how well the multilateration algorithm and the Kalman filter from Chapter 2 performed in various situations and to evaluate the their robustness, they were implemented in Matlab, then each tested in several different scenarios. In Section 3.4. we also introduce the detection algorithm, from which we determine the TOA values for the position estimations.

3.1 Simulation setup

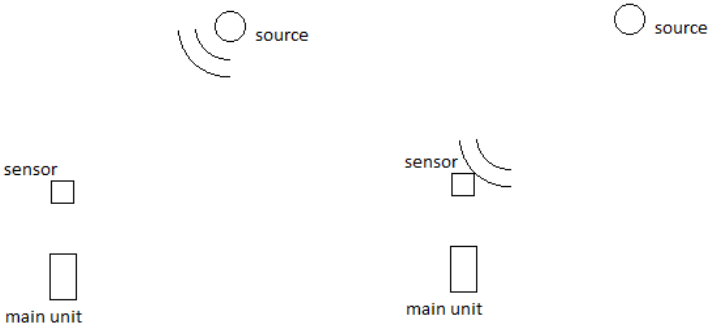
For the Kalman filter, we simulated an RTC clock of frequency 32 768 Hz. The sensors sent out a series of 100 calibration pulses with a time period of 100 RTC cycles per pulse, or every $\frac{100}{32768}$ s. The Kalman filter was updated when a pulse was received. To minimize the effect of missed pulses, each sensor used its own filter.

The multilateration algorithm was tested using four nodes at fix positions (one being in the origin), first in given formations, then at random positions to test the robustness of the algorithm. A number of edge cases were also reviewed to give an idea of the sensitivity of the algorithm to possible error propagation. The positions used are shown in Section 3.3.

3.2 Kalman filter

In Figures 3.1 and 3.2, a small overview of the setup is shown. In Figure 3.1, the source emits a wave at an unknown transmission time TOT (Figure 3.1a), which arrives at the sensor at the time TOA (in RTC cycles), at which it is detected (Figure 3.1b). The sensor immediately starts sending out calibration pulses to the main unit as shown in Figure 3.2a, where the Kalman filter is updated at the main hub after every new received pulse. In Figure 3.2b, the calibration pulses are finished and the main hub calculates the estimated drift along with the drift-adjusted TOA.

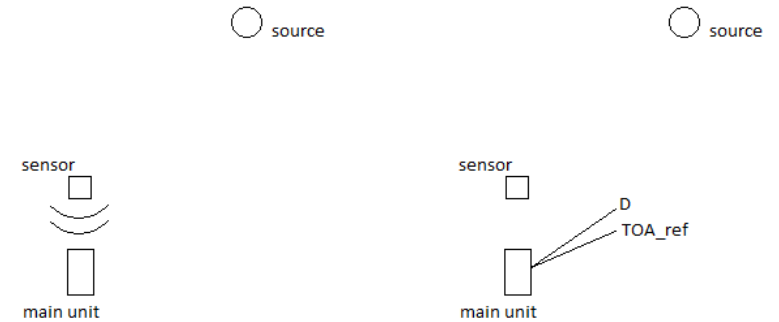
Depending on when we initialize the RTC circuits, the offset will always be non-zero. As long as the set interval is fix, This however is not relevant until we



(a) Sound transmitted at TOT

(b) Detection by sensor at TOA

Figure 3.1: Transmission and detection of sound waves



(a) Calibration pulses to main hub to update Kalman states

(b) Calculation at main hub after completed calibration series

Figure 3.2: Calibration and calculation of drift and time-of-arrival

transmit the calibration series. Luckily, in our TOA calculation we will only need the initial offset (when the first calibration pulse is received). The filter is updated when the main unit receives a calibration pulse. We cannot use fix time steps, as this would imply that we already know D. In practice, this may result in the state estimation using outdated information. As the reference clock is read in each state update, this will quickly result in very large values of T and poor state estimations.

The measurement noise is dependent on the delay between the sensor transmit-

ting its calibration pulse and the main unit acknowledging it. It it assumed to be Gaussian and stationary as long as the sensors and main unit are stationary.

When the state estimations have converged, they are then used in the calculation of the TOA by converting the sensor-specific time in RTC cycles to the equivalent time in the time system of the reference clock:

$$TOA_{ref} = (1+\hat{D})\frac{C_{TOA} - C(1)_{calib}}{f_{RTC}} - \hat{T}(0) \tag{3.1}$$

where C were the timestamp cycles and $\hat{T}(0)$ was the first measured offset.

Evaluation examples

To tune the Kalman filter so that it would perform well in situations with differing amounts of measurement noise (by finding suitable initial values for P, K and x) it was at first trained on a set of simulated time series. These were copies of a reference series, modified by adding drift (by multiplying the entire reference by $1+D$), offset (a known T_0) and measurement noise.

The Kalman filter was at first test on a set of simulated time series, same as the reference series but with an added drift (by multiplying the entire series by $1+D$) and offset (adding T as similar). This series can be seen in Figure 3.3, where we used $D = 0.1$ and $T = 10$ RTC cycles ($\frac{10}{32768}$ s). We also tested the algorithm with $D = -0.1$ and $T = -100$ RTC cycles, which can be seen in Figure 3.4.

The convergence to the a priori coefficients was quick and successful in both tests. To the test the robustness of the filter, we used the previous values of D and T, but now added Gaussian noise of variance $\sigma^2 = \frac{0.1}{32768}$.

Using the final matrices as initial guesses and rerunning the simulation did not change the outcome. However, the added noise was significantly reflected in the estimated D, as can be seen in Figure 3.5 and 3.6. By changing (the coefficients) of Q and R, we can specify how noisy we determine the process or measurement to be, so that we can choose between either a fast response (introducing a lot of noise into our estimate) or a non-noisy estimate (which will response in a long convergence time. Generally using a high value of Q is recommended, since the convergence time will be within the calibration interval anyway. R should ideally be set low to avoid overshooting.

In Figures 3.7-3.8, the process noise variance term for state D in Q has been increased, penalizing high variance due to noise but in the same time increasing the convergence time.

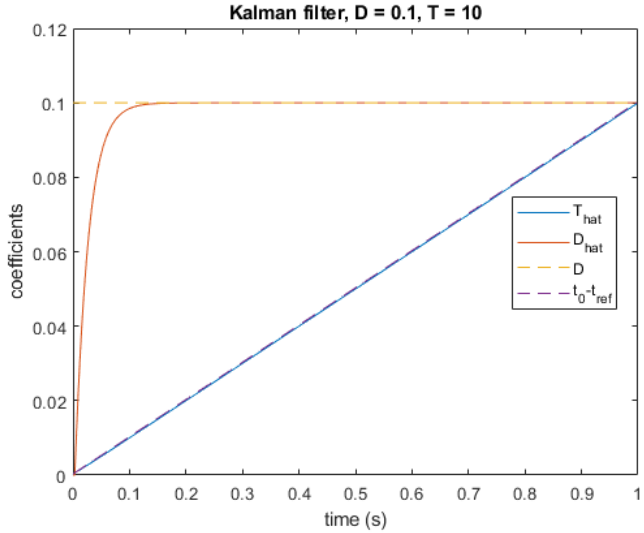


Figure 3.3: Convergence of Kalman filter states with $D = 0.1$, $T = 10$

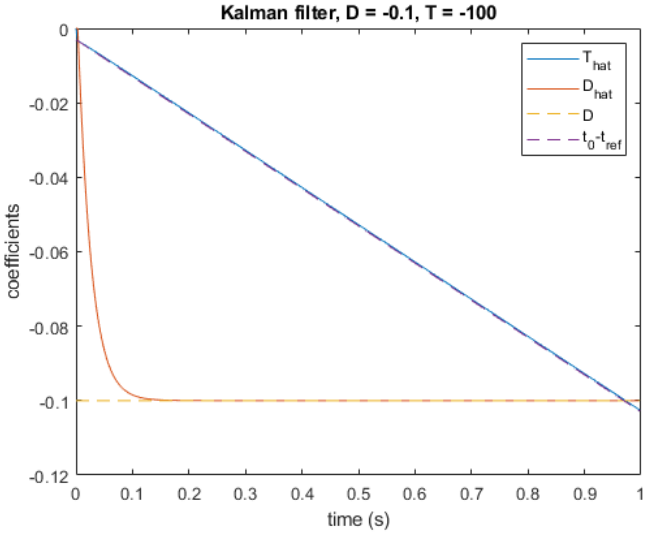


Figure 3.4: Convergence of Kalman filter states with $D = -0.1$, $T = -100$

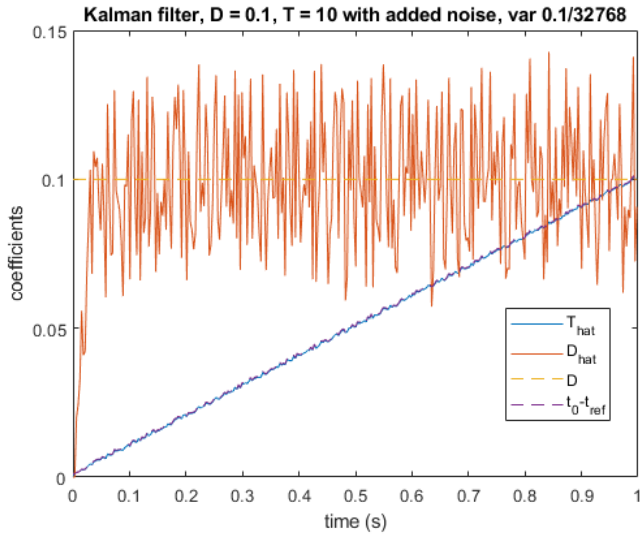


Figure 3.5: Convergence of Kalman filter states with $D = 0.1$, $T = 10$ and added noise

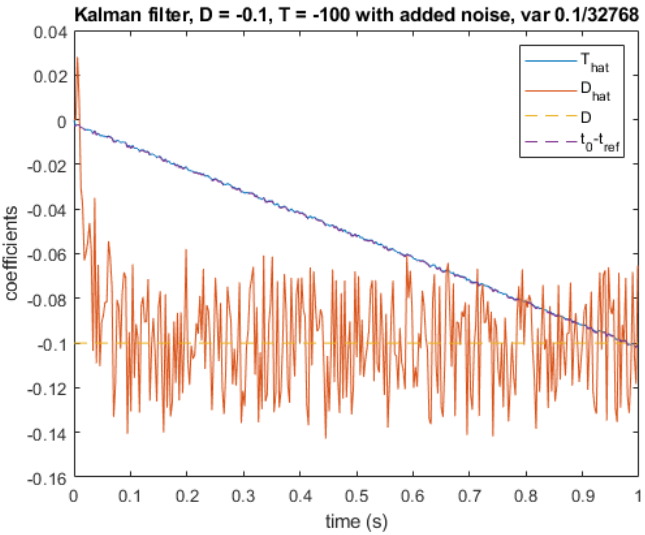


Figure 3.6: Convergence of Kalman filter states with $D = -0.1$, $T = -100$ and added noise

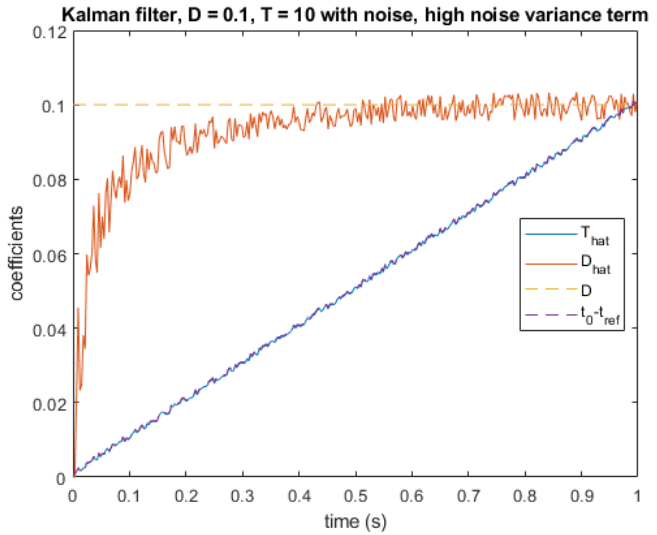


Figure 3.7: Convergence of Kalman filter states with $D = 0.1$, $T = 10$ and added noise, high noise variance penalty

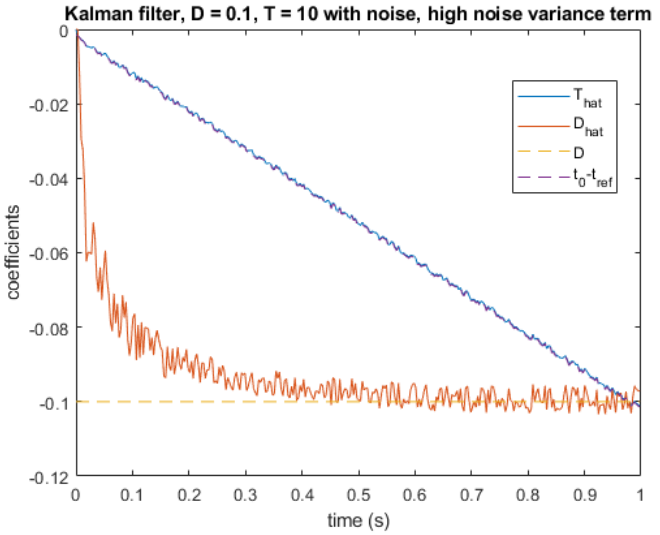


Figure 3.8: Convergence of Kalman filter states with $D = -0.1$, $T = -100$ and added noise, high noise variance penalty

3.3 Multilateration in 3D

Multilateration of time difference of arrival problems entails solving a series of non-linear least-squares equations and is computationally demanding. Since our devices are limited in memory and processing power, it is important to find a simple and least computationally demanding algorithm as possible. The algorithm from Section 2.1 was implemented here, as it does not use any matrix inverses that may be computationally expensive, however due to time constraints we could not delve deeper here, for example determining worst-case scenarios.

Assume a signal-emitting source in N dimensions with unknown position. In order to accurately determine its position, at least $N+1$ receivers are needed. Here, we use our sensors as receivers. One of the sensors is assumed to be placed in the origin, here known as s_0.

As we only know the time point of arrival (and not the propagation time from the sound source to the sensors) we make use of TDOA (time difference of arrival) estimation, where the TOAs of each sensor is subtracted by the TOA of the origin receiver. These values are multiplied by the speed of sound (Equations 2.1-2.2) in order to get the difference in distance from the origin sensor to each of the other sensor (subtracted by the unknown distance from the origin to the sound source). From these differences in distance we now calculate the three dimensional position of the source of the sound according to Equations 2.3-2.8.

From Equation 2.2, we see that the speed of sound in air is strongly dependant on the temperature. Not accommodating this will result in the calculated values varying up to 5% from ground truth. In the field the air temperatures and the relative humidity at the moment of detection was measured by the sensors and transmitted along with the calibration pulses, but the humidity ended up not being used in the calculation as the effect of the difference in humidity was negligible. For the simulations, $v_{air} = 340\ m/s$ was assumed from Equation 2.2.

Evaluation

The method was evaluated was tested using a known source position, calculating the time of propagation from the source to the sensors. The propagation time of the origin sensor was subtracted from the other sensors.

First we tested the algorithm with the nodes as unit vectors on each axis: (0,0,0), (1,0,0), (0,1,0) and (0,0,1). However, we found out that if there are more than two sensors sharing the same coefficients of an axis, this will result in a singularity in the algorithm and no solution will be found.

Putting the nodes in (0,0,0), (1,1,0), (1,0,1) and (0,1,1) and the source in (5,5,0) resulted in the estimates (-0.3577, -0.3577, 0.8925) and (4.7379, 4.7379, -0.0067). Here we see that the closest solution had a negative z-coordinate, which however is not significantly below zero. With the nodes in the same position, we evaluated a grid (20 x 20 in steps of 1 m) of possible source positions on the xy-plane and plotted these on Matlab, as shown in Figure 3.9. Here the singularity problems are

apparent as any source position on a line through origin will have no possible solution and show up as a white line on the plot. Also, we see that although the accuracy quickly drops off away from origin, there are still scattered points where the estimated solution is close to the actual source.

Moving the nodes to (0,0,0), (2,2,0), (2,0,2) and (0,2,2) respectively, we might expect better stability as the increased distances between the nodes decreases the risk of singularity. This seems to be true as the accuracy is improved according to Figure 3.10. The same "corridor" of non-solutions seems to remain though.

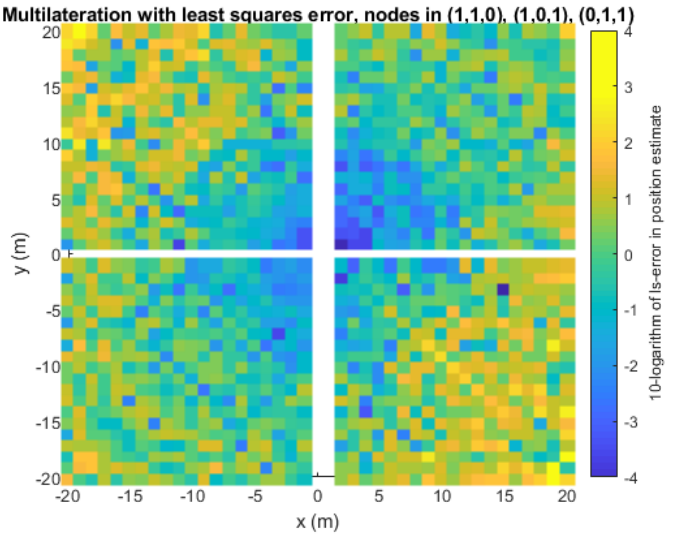


Figure 3.9: 10-logarithm of least squares error of position estimates. Nodes in (0,0,0), (1,1,0), (1,0,1) and (0,1,1). The x- and y-axes show the position of the source.

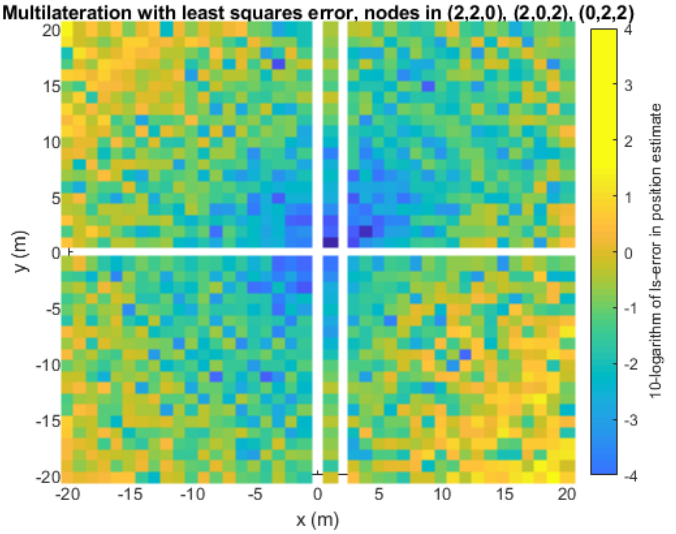


Figure 3.10: 10-logarithm of least squares error of position estimates. Nodes in (0,0,0), (2,2,0), (2,0,2) and (0,2,2). The x- and y-axes show the position of the source.

3.4 Detection of events

When a loud sound is made, a sound wave is propagated. We continually record samples from the microphone. To record events such as explosive sounds (gunfire) we want to distinguish it from the ambient noise. Preferably this will be done on the fly inside the sensor.

Our own method was at first to calculate the absolute difference between the current and previous sample. When field recording, we split the samples into noise windows of size N starting from the beginning, and calculated the average of the absolute differences within a window. From this measure of ambient noise we calculate a threshold defined as a multiple of this average.

When the absolute difference in amplitude is larger than the threshold, the noise window is terminated and a recording window of size N is initialized. If at least k samples within this window exceed the threshold, this event is flagged as a detection, with the time of the first sample above the threshold declared as the time of detection. If not, the event is discarded and we continue recording samples.

When an event is definitely detected, the sensor transmits the time of detection and initializes a calibration series, where the sensors transmits a time pulse each 100 RTC cycles to the main unit. Under this calibration interval, the sensor does not record any incoming sound.

At the end of this calibration series and a further wait, if the unit has received input (ongoing or finished) from all the sensors, the event is regarded as true (non-false positive) and the calibration series are fed into the Kalman filter. If the unit has not received input from all sensors, the event is regarded as a false positive and the RTC clocks are restarted.

The algorithm does not discern between different possible sources of the sound. For example, a hand clap or a balloon being popped will both be detected as an event.

Evaluation examples

We evaluated this method in Matlab with a set of sound files of gunshots (sampled at 44-96KHz, 32-bit, converted from stereo to mono) from http://freesound.org. A fix threshold value of 0.2 with a window of 500 sample values was used. In Figures 3.11-3.13, the amplitude, difference in amplitude and absolute difference in amplitude, respectively, from one of the sound files are shown. It was sampled at 44.1 Khz, 32-bit stereo, with the left channel used.

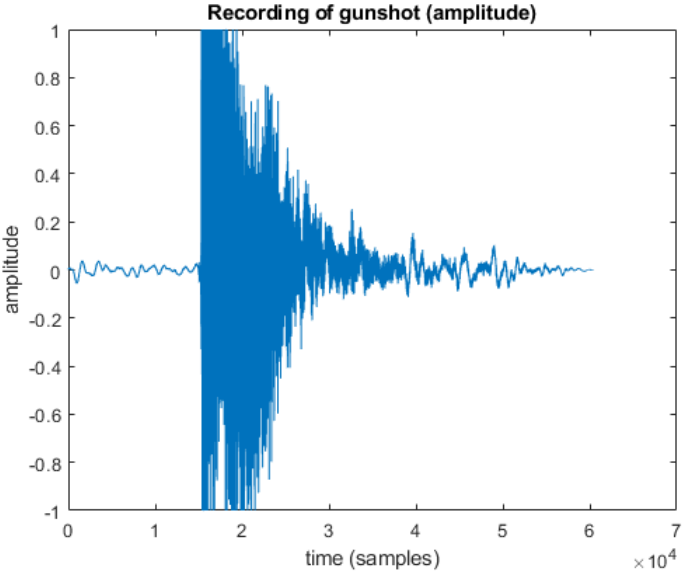


Figure 3.11: Recording of gunshot, unfiltered

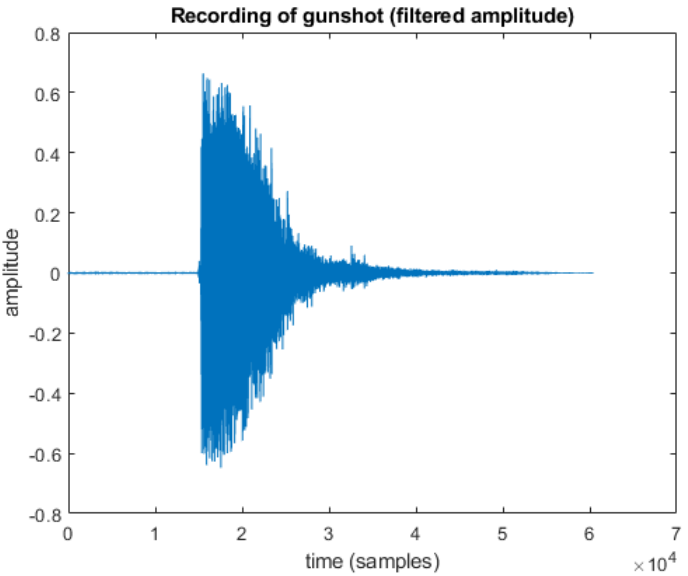


Figure 3.12: Recording of gunshot, filtered by differentiating the amplitude

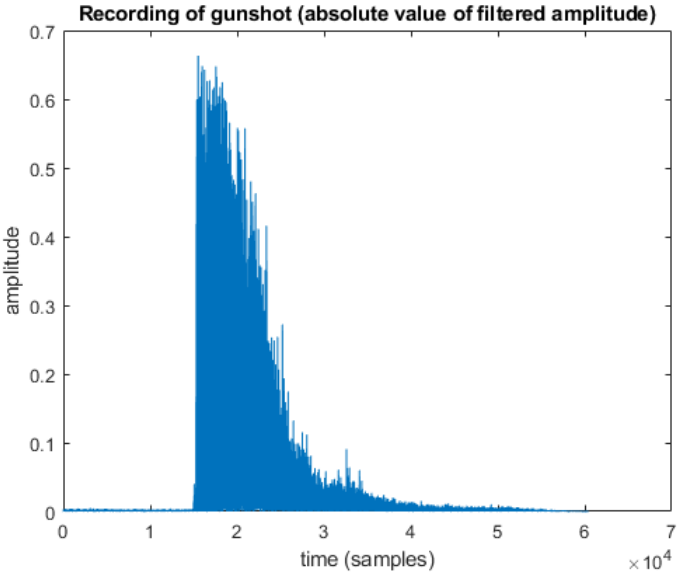


Figure 3.13: Recording of gunshot, absolute values of the filtered sound values

Most events were correctly detected, but the difference in volume between the recordings resulted in some events not being detected at all. A lower threshold would be needed here, but at the same time this increases the chance of false positives from background noise. Using a noise window was not possible as the time of preceding background noise was almost non-existent in several recordings. We instead assumed the threshold to be user selectable, so that one of 4 different thresholds can be chosen depending on the perceived background noise level. The final detection algorithm code (in Matlab) printed below.

Detection algorithm code

The following code shows the detection algorithm discussed in Section 3.4. It shows the case for the threshold window of 500 samples, with a threshold of 0.2 and a minimum cluster of 2.

```
% s1 is our sound vector, imported from elsewhere.
% it is assumed to have at least than 1000 values
ds1 = s1(2:end)-s1(1:end-1);
abs_ds1 = abs(ds1);
thrs = 0.2;           % threshold of detection
cluster_thrs = 2;     % nbr of required peaks within detection window
TOA = 0;
tempvec = [];
vec_len = 500;       % length of detection window
start = 1;
finish = 500;

for i = 1:floor(length(s1)/500)
     tempvec = abs_ds1((start:finish));
     if sum(tempvec > thrs) > cluster_thrs
         first_el = find((tempvec > thrs),1);
         TOA = start-1 + first_el;
         start = start + vec_len;
         finish = finish + vec_len;
         break
     else
         start = start + vec_len;
         finish = finish + vec_len;
     end
end
```

4

Field experiments

To test the theory out and find potential error sources, two field sessions were done, testing the detection algorithm by firing gunshots at stationary positions in series. Three microphones (m_l, m_c and m_r; left, center and right respectively) were used, continuously recording at either 44Khz or 48Khz, 32-bit in stereo. The left channel was used, as the difference between the two channels was minimal, except the right channel being delayed by 2 samples. In total both recording sessions lasted around 30 min, with the analysis being done in post using MATLAB. As originally intended, this analysis would be online, where upon detection the sensors would send their TOA and calibration pulses to the main hub where the Kalman filtering and multilateration calculation would be done automatically once all sensors had sent their calibration pulses.

The testing environment was a flat grassfield (to minimize reflections), making it possible to simplify the positioning algorithms to 2D. However, the original 3D positioning algorithms were not easily convertible to 2D, so instead an exhaustive search algorithm was used, evaluating each possible solution in a grid x by y meters centered in (0,0), with resolution dx and dy in the x- and y-axes, respectively.

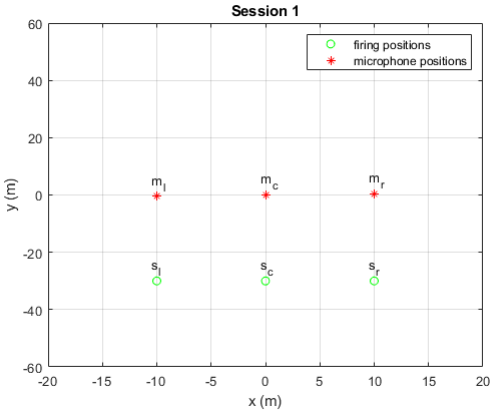


Figure 4.1: Session 1, firing positions and microphone positions

Series	Source and direction	Shots
1	s_c towards m_c	10
2	s_c towards m_r	5
	s_c towards m_l	5
3	s_l towards m_l	5
	s_l towards m_c	5
	s_l towards m_h	5
4	s_r towards m_r	5
	s_r towards m_c	5
	s_r towards m_l	5

Table 4.1: Session 1, firing series

4.1 Session 1

The temperature during the first session was 15°C.
The microphones and the sound sources were oriented according to Figure 4.1, with each firing series according to Table 4.1.

After detection processing the recordings, a fixed threshold of 0.4 was used for the TDOA analysis, as the peaks corresponding to the shots all exceeded this number. A detection window of 1000 samples was used, with a cluster value of 2. The TDOA values from each recording were normalized to actual time (in s) by dividing with the sample frequency.

Although the recordings were originally synced with four short beeps, this approach was hard to utilise in post due to gaps in the recordings. Instead the record-

Source	Microphone	Relative distance (m)
s_c	m_l	1.670
	m_r	1.575
s_l	m_l	-1.573
	m_r	4.391
s_r	m_l	4.474
	m_r	-1.673

Table 4.2: Session 1, distance from source to microphone relative to m_c

ings were calibrated from the first firing series, aligning the recordings to correspond with the theoretical TDOA values from the first shot in Series 1, to exactly match those expected from ground truth. Due to the lack of any continuous periodic syncing, the Kalman filter was not in use. From Equation 2.2, the speed of sound was estimated at 340.4 m/s. The difference in distance between the microphones and sound sources (compared to m_c) can be seen in Table 4.2.

4.2 Session 2

The temperature during the second session was 12°C.
The microphones and the sound sources were oriented as in Figure 4.2, with each firing series according to Table 4.3.

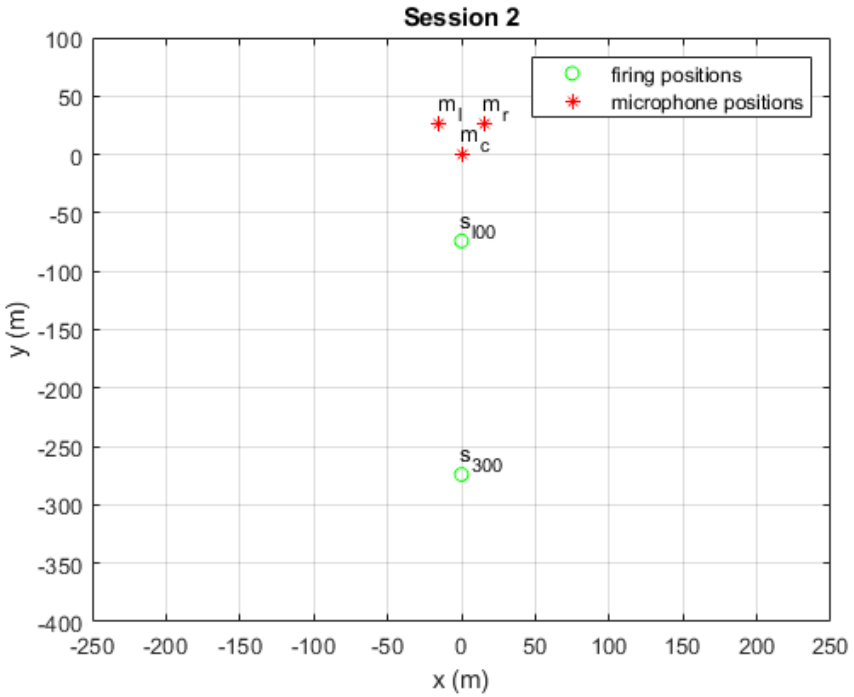


Figure 4.2: Session 2, firing positions and microphone positions

Series	Source and direction	Shots	Type (silenced)
1	s_{100} towards m_c	5	Low
2	s_{100} towards m_c	3	High (no)
3	s_{100} towards m_c	3	High (yes)
4	s_{300} towards m_c	5	Low
5	s_{300} towards m_c	3	High (no)
6	s_{300} towards m_c	3	High (yes)

Table 4.3: Session 2, firing series

The type signifies the calibre. To minimise the mirroring effects of Session 1, the microphones were placed as in an equilateral triangle with side length 30 m, with m_c being placed at the bottom.

At the beginning of the recording, the microphones were situated at the same location, synced with a series of hand claps, then placed at the given locations. This was repeated at the end of the recording, the microphones gathered together and synced with a series of hand claps. These hand claps were visible in the detection processing, and the TOA values were used to align the recordings. The Kalman filter was not used, as there was no continuous, periodic syncing.

From Equation 2.2, the speed of sound was estimated at 338.6 m/s. The expected relative distances from m_l and m_r to m_c was $(27.12, 27.12)$ for s_{100} and $(26.37, 26.37)$ for s_{300}. The number of samples between the first hand clap of the opening syncing and the last hand clap of the end syncing differed between the recordings, indicating drift within the microphones. (The left microphone had been operating at a voltage too low)

With regard to the recording from the center microphone (containing 83 184 922 samples inside the interval between the claps), the left microphone and right microphone had 793 and 264 samples too many, respectively, corresponding in a recording frequency drift of $D = 9.533 \cdot 10^{-6}$ and $3.174 \cdot 10^{-6}$ respectively. The TOA values of each recording were divided by $1 + D$ to match that of the center recording, and then aligned with both hand clap series.

As in Session 1, an exhaustive search algorithm in 2D was used, within a grid of resolution 1 m in the interval $\{x \in (-300, 300), y \in (-1000, 1000)\}$. Any solution outside this interval was deemed invalid and marked as N/A.

5

Results

In this chapter, the positions as estimated from the detected TOAs of the field tests are shown. The Figures, 5.1 from Session 1 and 5.2-5.6 from Session 2, show the estimated positions for each session (and possible calibration). The detected TDOA for each shot and microphone, together with the derived data (estimated relative distance, estimated position) and ground truth position are shown in Tables 5.1-5.2 for Session 1 and Tables 5.3-5.7 for Session 2.

5.1 Session 1

In this section, we display the setup of Session 1, along with the estimated positions, in Figure 5.1. In Tables 5.1-5.2, the raw data from the TOA analysis of Session 1 are displayed, along with the estimated distance (given the speed of sound from Equation 2.2 and the temperature according to Section 4.1) and the estimated position derived from it. We also display the ground truth position. Since there was no available sync, the first shot in series 1 was aligned according to expected ground truth.

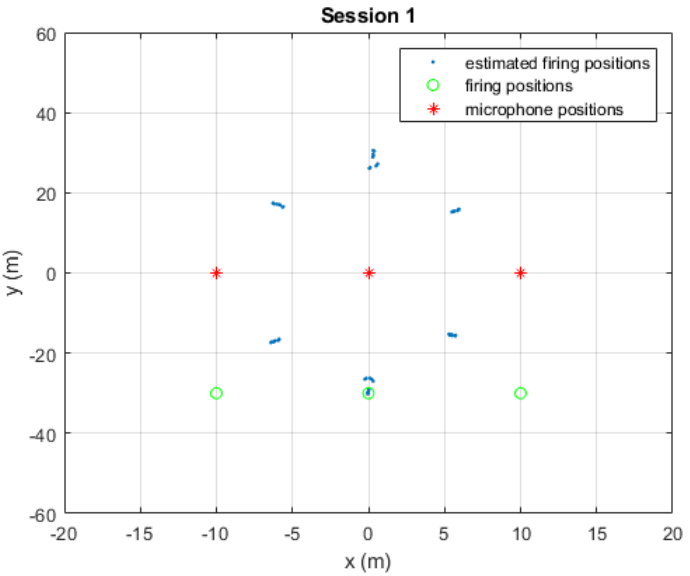


Figure 5.1: Session 1, estimated firing positions from TDOA

Series	$TDOA(m_l, m_r)$	$dr(m_c)$	est. pos	g.t.
1	$(4.912, 4.633) \cdot 10^{-3}$	(1.670, 1.575)	(0.00,-30.00)*	(0,-30)
	$(4.822, 4.558) \cdot 10^{-3}$	(1.639, 1.550)	(0.30, 30.56)	(0,-30)
	$(4.844, 4.662) \cdot 10^{-3}$	(1.647, 1.585)	(-0.06,-30.14)	(0,-30)
	$(4.890, 4.724) \cdot 10^{-3}$	(1.663, 1.606)	(-0.06,-29.78)	(0,-30)
	$(4.980, 4.682) \cdot 10^{-3}$	(1.693, 1.592)	(0.30, 29.62)	(0,-30)
	$(4.912, 4.504) \cdot 10^{-3}$	(1.670, 1.531)	(0.38, 30.44)	(0,-30)
	$(4.935, 4.692) \cdot 10^{-3}$	(1.678, 1.595)	(-0.02,-29.74)	(0,-30)
	$(5.071, 4.802) \cdot 10^{-3}$	(1.724, 1.633)	(0.28, 28.94)	(0,-30)
	$(5.071, 4.759) \cdot 10^{-3}$	(1.724, 1.618)	(0.02,-29.08)	(0,-30)
	$(5.048, 4.709) \cdot 10^{-3}$	(1.716, 1.601)	(0.32, 29.30)	(0,-30)
2	$(5.706, 4.948) \cdot 10^{-3}$	(1.940, 1.682)	(0.50,26.70)	(0,-30)
	$(5.728, 5.024) \cdot 10^{-3}$	(1.948, 1.708)	(0.20,-26.44)	(0,-30)
	$(5.661, 5.170) \cdot 10^{-3}$	(1.925, 1.758)	(0.10,-26.24)	(0,-30)
	$(5.728, 4.829) \cdot 10^{-3}$	(1.948, 1.642)	(0.30,-26.96)	(0,-30)
	$(5.706, 4.755) \cdot 10^{-3}$	(1.940, 1.617)	(0.60,27.20)	(0,-30)
	$(5.253, 5.477) \cdot 10^{-3}$	(1.786, 1.862)	(-0.24,-26.50)	(0,-30)
	$(5.366, 5.447) \cdot 10^{-3}$	(1.824, 1.852)	(0.10,26.28)	(0,-30)
	$(5.411, 5.412) \cdot 10^{-3}$	(1.840, 1.840)	(-0.14,-26.26)	(0,-30)
	$(5.366, 5.486) \cdot 10^{-3}$	(1.824, 1.865)	(0.08,26.18)	(0,-30)
	$(5.366, 5.514) \cdot 10^{-3}$	(1.824, 1.875)	(0.06,26.10)	(0,-30)

Table 5.1: Session 1, table of TDOA values, estimated distance from origin node, estimated position and ground truth for Series 1-2

Series	$TDOA(m_l, m_r)$	$dr(m_c)$	est. pos	g.t.
3	$(-2.208, 15.766) \cdot 10^{-3}$	(-0.751, 5.360)	(-6.28,17.44)	(-10,-30)
	$(-2.140, 15.824) \cdot 10^{-3}$	(-0.728, 5.380)	(-6.22,17.24)	(-10,-30)
	$(-2.117, 15.784) \cdot 10^{-3}$	(-0.720, 5.367)	(-6.38,-17.22)	(-10,-30)
	$(-2.140, 15.741) \cdot 10^{-3}$	(-0.728, 5.352)	(-6.40,-17.32)	(-10,-30)
	$(-2.162, 15.728) \cdot 10^{-3}$	(-0.735, 5.347)	(-6.42,-17.38)	(-10,-30)
	$(-1.890, 15.657) \cdot 10^{-3}$	(-0.643, 5.323)	(-6.24,-17.20)	(-10,-30)
	$(-1.800, 15.693) \cdot 10^{-3}$	(-0.612, 5.336)	(-6.18,-17.04)	(-10,-30)
	$(-1.822, 15.659) \cdot 10^{-3}$	(-0.620, 5.324)	(-6.02,17.18)	(-10,-30)
	$(-1.754, 15.734) \cdot 10^{-3}$	(-0.596, 5.349)	(-6.14,-16.92)	(-10,-30)
	$(-1.528, 15.550) \cdot 10^{-3}$	(-0.514, 5.287)	(-5.84,17.04)	(-10,-30)
	$(-1.504, 15.571) \cdot 10^{-3}$	(-0.512, 5.294)	(-5.82,16.98)	(-10,-30)
	$(-1.164, 15.626) \cdot 10^{-3}$	(-0.396, 5.313)	(-5.60,16.52)	(-10,-30)
	$(-1.391, 15.574) \cdot 10^{-3}$	(-0.473, 5.295)	(-5.92,-16.80)	(-10,-30)
	$(-1.278, 15.754) \cdot 10^{-3}$	(-0.434, 5.356)	(-5.66,16.42)	(-10,-30)
	$(-1.346, 15.725) \cdot 10^{-3}$	(-0.458, 5.347)	(-5.88,-16.50)	(-10,-30)
4	$(16.409, -1.431) \cdot 10^{-3}$	(5.580, -0.486)	(5.72,-15.54)	(10,-30)
	$(16.273, -1.540) \cdot 10^{-3}$	(5.533, -0.524)	(5.94, 15.78)	(10,-30)
	$(16.182, -1.508) \cdot 10^{-3}$	(5.502, -0.513)	(5.94, 15.92)	(10,-30)
	$(16.296, -1.412) \cdot 10^{-3}$	(5.542, -0.480)	(5.72,-15.72)	(10,-30)
	$(16.318, -1.381) \cdot 10^{-3}$	(5.548, -0.470)	(5.84, 15.56)	(10,-30)
	$(16.273, -0.976) \cdot 10^{-3}$	(5.533, -0.332)	(5.46,-15.34)	(10,-30)
	$(16.160, -1.073) \cdot 10^{-3}$	(5.494, -0.364)	(5.52,-15.60)	(10,-30)
	$(16.137, -1.062) \cdot 10^{-3}$	(5.487, -0.361)	(5.52,-15.62)	(10,-30)
	$(16.160, -0.981) \cdot 10^{-3}$	(5.494, -0.334)	(5.62, 15.46)	(10,-30)
	$(16.228, -0.948) \cdot 10^{-3}$	(5.517, -0.322)	(5.60, 15.32)	(10,-30)
	$(16.160, -0.748) \cdot 10^{-3}$	(5.494, -0.254)	(5.48, 15.24)	(10,-30)
	$(16.001, -0.778) \cdot 10^{-3}$	(5.440, -0.264)	(5.36,-15.58)	(10,-30)
	$(16.092, -0.756) \cdot 10^{-3}$	(5.471, -0.257)	(5.34,-15.40)	(10,-30)
	$(16.160, -0.690) \cdot 10^{-3}$	(5.494, -0.234)	(5.30,-15.24)	(10,-30)
	$(16.114, -0.721) \cdot 10^{-3}$	(5.479, -0.245)	(5.32,-15.34)	(10,-30)

Table 5.2: Session 1, table of TDOA values, estimated distance from origin node, estimated position and ground truth for Series 3-4

5.2 Session 2

In this section, we display the setup of Session 2, along with the estimated positions, in Figures 5.2-5.6. In Table 5.3, the raw data from the TDOA analysis of Session 2 are displayed, along with the estimated distance (given the speed of sound from Equation 2.2 and the temperature according to Section 4.2) and the estimated position derived from it. We also display the ground truth position.

Without calibration

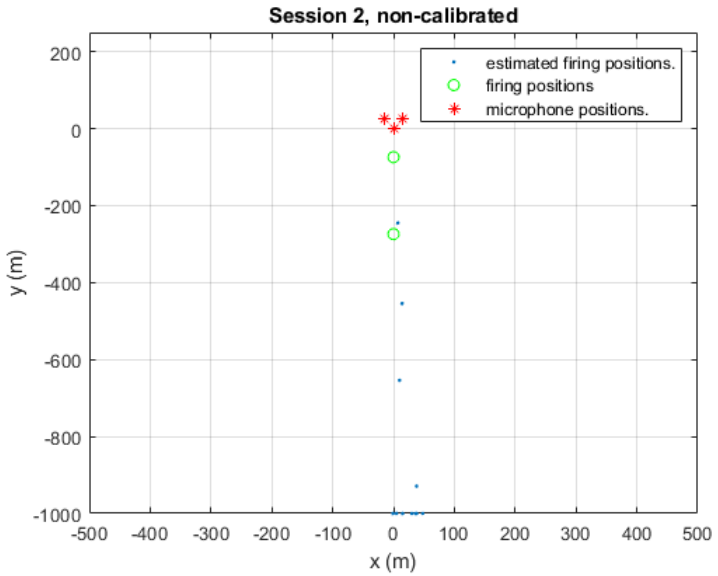


Figure 5.2: Session 2, estimated firing positions from TDOA without calibration

Series	$TDOA(m_l, m_r)$	$dr(m_c)$	est. pos	g.t.
1	(no det.)	N/A	N/A	(0,-74)
	(no det.)	N/A	N/A	(0,-74)
	(no det.)	N/A	N/A	(0,-74)
	(no det.)	N/A	N/A	(0,-74)
	(no det.)	N/A	N/A	(0,-74)
2	$(7.181, 6.939) \cdot 10^{-2}$	$(24.34, 23.52)$	N/A	(0,-74)
	$(7.237, 6.927) \cdot 10^{-2}$	$(24.53, 23.48)$	N/A	(0,-74)
	$(7.220, 7.025) \cdot 10^{-2}$	$(24.48, 23.81)$	N/A	(0,-74)
3	$(7.188, 7.196) \cdot 10^{-2}$	$(24.37, 24.40)$	N/A	(0,-74)
	$(7.248, 7.215) \cdot 10^{-2}$	$(24.57, 24.46)$	N/A	(0,-74)
	$(7.266, 7.160) \cdot 10^{-2}$	$(24.63, 24.27)$	N/A	(0,-74)
4	(no det.)	N/A	N/A	(0,-274)
	(no det.)	N/A	N/A	(0,-274)
	(no det.)	N/A	N/A	(0,-274)
	(no det.)	N/A	N/A	(0,-274)
	(no det.)	N/A	N/A	(0,-274)
5	$(7.636, 7.352) \cdot 10^{-2}$	$(25.89, 24.92)$	N/A	(0,-274)
	$(7.874, 7.522) \cdot 10^{-2}$	$(26.70, 25.50)$	(38,-929)	(0,-274)
	$(7.757, 7.509) \cdot 10^{-2}$	$(26.30, 25.45)$	N/A	(0,-274)
6	$(7.903, 7.678) \cdot 10^{-2}$	$(26.79, 26.03)$	(7,-245)	(0,-274)
	$(7.864, 7.607) \cdot 10^{-2}$	$(26.66, 25.79)$	(14,-454)	(0,-274)
	$(7.783, 7.654) \cdot 10^{-2}$	$(26.38, 25.95)$	(10,-654)	(0,-274)

Table 5.3: Session 2, without calibration. TDOA values, distance from origin node, estimated position and ground truth

Calibrating to Series 2 and s_{100}

In this section, the TDOA values are the same as in Table 5.3, but with the relative distance differences in Series 2 aligned to match the ground truth of source position s_{100}. The results of this can be seen in Figure 5.3. No new TDOA-values are measured and so they are removed from Table 5.4, where the estimated distance (given the speed of sound from Equation 2.2 and the temperature according to Section 4.2) and the estimated position derived from it are displayed. We also display the ground truth position.

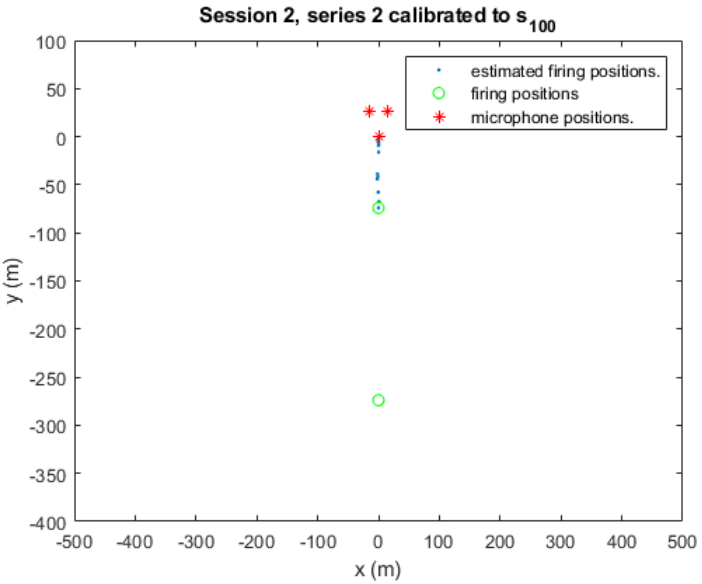


Figure 5.3: Session 2, estimated firing positions with Series 2 calibrated to microphone position s_{100}

Series	$dr(m_c)$	est. pos	g.t.
2	$(27.12, 27.12)$	(0,-74)*	(0,-74)
	$(27.31, 27.08)$	(0.7,-67.5)	(0,-74)
	$(27.25, 27.41)$	(0.5,-57.9)	(0,-74)
3	$(27.14, 27.99)$	(-2,-44.1)	(0,-74)
	$(27.34, 28.06)$	(-1.6,-38.8)	(0,-74)
	$(27.41, 27.87)$	(-1.1,-41.6)	(0,-74)
5	$(28.67, 28.52)$	(0.2,-16.0)	(0,-274)
	$(29.47, 29.10)$	(0.4,-6.4)	(0,-274)
	$(29.08, 29.06)$	(0,-9.2)	(0,-274)
6	$(29.57, 29.63)$	(-0.1,-3.4)	(0,-274)
	$(29.44, 29.39)$	(0.1,-5.2)	(0,-274)
	$(29.16, 29.55)$	(-0.5,-5.6)	(0,-274)

Table 5.4: Session 2 (without Series 1 and 4), Series 2 calibrated to s_{100}. Distance from origin node, estimated position and ground truth

Calibrating to Series 2 and s_{300} (contradicting ground truth)

In this section, the TDOA values are the same as in Table 5.3, but with the relative distance differences in Series 2 aligned to match source position s_{300}, which however contradicts ground truth. The results of this can be seen in Figure 5.4. No new TDOA-values are measured and so they are removed from Table 5.5, where the estimated distance (given the speed of sound from Equation 2.2 and the temperature according to Section 4.2) and the estimated position derived from it are displayed. We also display the ground truth position.

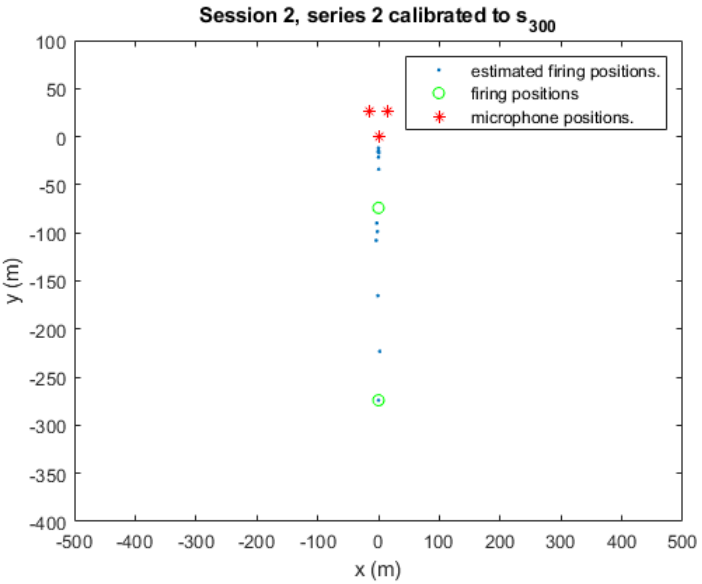


Figure 5.4: Session 2, estimated firing positions with Series 2 calibrated to microphone position s_{300}

Series	$dr(m_c)$	est. pos	g.t.
2	$(26.37, 26.37)$	(0,-274)*	(0,-74)
	$(26.57, 26.33)$	(1.9,-223.1)	(0,-74)
	$(26.51, 26.67)$	(-1,-165.3)	(0,-74)
3	$(26.40, 27.24)$	(-3.8,-107.8)	(0,-74)
	$(26.60, 27.31)$	(-2.8,-89.9)	(0,-74)
	$(26.66, 27.13)$	(-1.9,-98.7)	(0,-74)
5	$(27.92, 27.78)$	(0.3,-33.8)	(0,-274)
	$(28.73, 28.36)$	(0.6,-16.8)	(0,-274)
	$(28.33, 28.31)$	(0,-21.3)	(0,-274)
6	$(28.83, 28.89)$	(-0.1,-11.9)	(0,-274)
	$(28.70, 28.64)$	(0.1,-14.8)	(0,-274)
	$(28.42, 28.81)$	(-0.6,-15.6)	(0,-274)

Table 5.5: Session 2 (without Series 1 and 4), Series 2 calibrated to s_{300}. Distance from origin node, estimated position and ground truth

Calibrating to Series 5 and s_{100} (contradicting ground truth)

In this section, the TDOA values are the same as in Table 5.3, but with the relative distance differences in Series 5 aligned to match the source position s_{100} which however contradicts ground truth. The results of this can be seen in Figure 5.6. No new TDOA-values are measured and so they are removed from Table 5.7, where the estimated distance (given the speed of sound from Equation 2.2 and the temperature according to Section 4.2) and the estimated position derived from it are displayed. We also display the ground truth position.

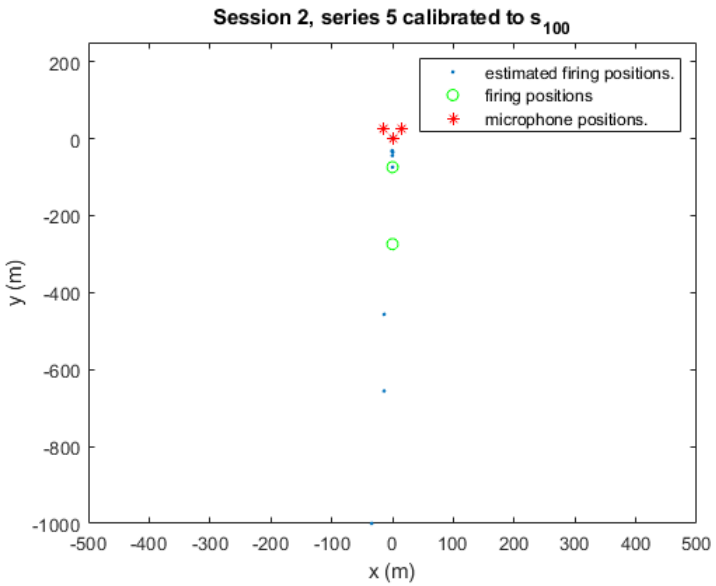


Figure 5.5: Session 2, estimated firing positions with Series 5 calibrated to microphone position s_{100}

Series	$dr(m_c)$	est. pos	g.t.
2	$(25.56, 25.71)$	(-0.1,2.4)	(0,-74)
	$(25.75, 25.66)$	(0.1,2.3)	(0,-74)
	$(25.69, 26.00)$	(-0.3,2.2)	(0,-74)
3	$(25.58, 26.58)$	N/A	(0,-74)
	$(25.79, 26.65)$	(-13.8,-456.5)	(0,-74)
	$(25.85, 26.46)$	(-13.8,-655.7)	(0,-74)
5	$(27.11, 27.11)$	(0,-74)*	(0,-274)
	$(27.92, 27.69)$	(0.5,-35.1)	(0,-274)
	$(27.52, 27.65)$	(-0.3,-43.9)	(0,-274)
6	$(28.02, 28.22)$	(-0.2,1.0)	(0,-274)
	$(27.89, 27.98)$	(-0.2,-31)	(0,-274)
	$(27.61, 28.14)$	(-1.1,-32.5)	(0,-274)

Table 5.6: Session 2 (without Series 1 and 4), Series 5 calibrated to s_{100}. Distance from origin node, estimated position and ground truth

Calibrating to Series 5 and s_{300}

In this section, the TDOA values are the same as in Table 5.3, but with the relative distance differences in Series 5 aligned to match the ground truth source position s_{300}. The results of this can be seen in Figure 5.5. No new TDOA-values are measured and so they are removed from Table 5.6, where the estimated distance (given the speed of sound from Equation 2.2 and the temperature according to Section 4.2) and the estimated position derived from it are displayed. We also display the ground truth position.

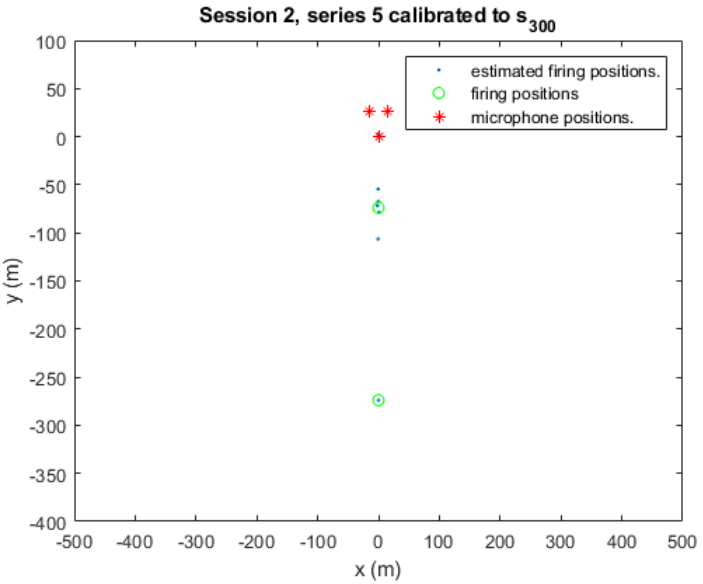


Figure 5.6: Session 2, estimated firing positions with Series 5 calibrated to microphone position s_{300}

Series	$dr(m_c)$	est. pos	g.t.
2	(24.82, 24.96)	(-0.1,2.8)	(0,-74)
	(25.01, 24.92)	(0.1,2.6)	(0,-74)
	(24.95, 25.25)	(-0.3,2.6)	(0,-74)
3	(24.84, 25.84)	(-1,2.4)	(0,-74)
	(25.04, 25.90)	(-0.8,2.4)	(0,-74)
	(25.11, 25.71)	(-0.6,2.4)	(0,-74)
5	(26.37, 26.37)	(0,-274)*	(0,-274)
	(27.18, 26.95)	(0.8,-78.5)	(0,-274)
	(26.78, 26.90)	(-0.6,-106.4)	(0,-274)
6	(27.28, 27.48)	(-0.6,-54.5)	(0,-274)
	(27.15, 27.24)	(-0.3,-67.4)	(0,-274)
	(26.87, 27.40)	(-1.8,-71.7)	(0,-274)

Table 5.7: Session 2 (without Series 1 and 4), Series 5 calibrated to s_{300}. Distance from origin node, estimated position and ground truth

6

Discussion

6.1 Field experiments

Session 1

As seen in Figure 5.1, the calculated source positions matched the actual positions with respect to angle, underestimating s_l and s_r in absolute distance, possibly due to non-compensated drift. However, there were heavy mirroring artifacts. Since the microphones were positioned along a line, it was not possible to discern whether the source was located above or below the line, resulting in the calculated positions of shots from the same series being mirrored seemingly at random. Possible measuring noise (due to the limited recording frequency) did not seem to affect the result too severely. If there were any reflecting artifacts they were not too significant. The direction in which the shots were fired did affect the TDOA slightly, but these differences were not significant enough to be immediately noticeable in the position estimation.

Session 2

The shots in Series 1 and 4 used a low-speed calibre which was mostly too quiet to be distinguishable from background noise in the detection analysis; these were discarded from the position estimations.

Without calibration, the algorithm results were poor. For Series 2-3, the TDOA values for each channel look too small, and do not correspond to any valid source position. Series 5-6 more closely resemble those expected from ground truth, but their TDOA values are larger than those in Series 2-3, contrary to ground truth. The angle from the microphones to the source roughly matched ground truth with a slight bias to the right, but the magnitude was far too large except for shot 1 of Series 6.

We suspected the source of these errors to be some kind of non-linear drift (due to non-constant speed of sound in the air, low voltages of the microphones etc.) or due to supersonic effects of the calibre.

To test how a possible calibration (similar to that in Session 1) might improve the results, we aligned the recordings of m_l and m_r to match a High-type series (i.e. Series 2 or 5), either with respect to s_{100} or s_{300}. This resulted in 4 possible calibrations. The TDOA values along with Series 1 and 4 in there entirety are left out from these calibrations.

Calibrating to Series 2 and s_{100} From Table 5.4, all TDOA values resulted in valid positions, but contrary to ground truth, the shots from s_{300} ended up much closer to the microphones.

Calibrating to Series 2 and s_{300} From Table 5.5, all TDOA values resulted in valid positions. The distances match better to ground truth, but in the wrong order (Series 2-3 better match 5-6 and vice versa). Some kind of drift between shots in the same series is very noticeable here, primarily in m_r.

Calibrating to Series 5 and s_{100} From Table 5.6, we see that the results are not realistic and do not resemble ground truth.

Calibrating to Series 2 and s_{100} From Table 5.7, Series 2-3 are too close to the m_c. Series 5-6 more closely match ground truth, but there is a noticeable drift here as well.

In conclusion, the results from calibration 2 seemed to correspond best to ground truth, although with the results in wrong order. Some kind of non-linear drift seems probable in m_r, even with the global drift correction. To more completely ameliorate effects from drift, some kind of synchronization pulse is needed, preferably with short, high-frequency pulses.

The results from the field tests and the results from the MatLab simulations in Chapter 3 are, unfortunately, not directly comparable due to the difficulty, of implementing the methods, evaluated in the simulations, on the originally intended hardware. As the field tests were conducted on flat ground, it was not strictly necessary to use more than three microphones, which made a simplified algorithm using only three inputs attractive. Also, the introduced degeneracy in the z-axis would have made the 3D multilateration algorithm particularly susceptible to singularity problems, which would have reduced the amount of usable field data.

6.2 Limitations

The multilateration algorithm from Chapter 2.2 was very sensitive to noise and quantization due to low sampling frequency as shown in Chapter 3.4, Figures 3.7-3.8. It was not very robust as the node positions heavily influenced the possible solutions, and therefore the accuracy of the estimations.

The Kalman filter assumed that the drift changed slowly enough that it could be considered constant during the calibration interval. It is possible that the dirft might be subject to fluctuations even within this interval, especially in extreme temperatures (below 0 °C or above 40° C) or unsettled (rainy or windy) weather, causing wrong state estimations.

The TOA algorithm had some susceptibility to overdetecting, causing some events to be detected twice, necessitating some manual removal of unrealistic TDOA values. It also couldn't detect two peaks very close in time due to the necessary detection window.

7

Conclusion and Future Work

7.1 Conclusion

This thesis has shown that even in conditions with heavy delay, a fast position estimate can be achieved. The Kalman filter worked well in achieving estimations of the clock drift and offsets in a short amount of time even with heavy measurement noise. The detection algorithm seemed reasonably robust to even heavy background noise. With the threshold used, there were few errors such as superfluous TOA detections.

Acquiring good TOA values is important to reduce error propagation when these values are used in the multilateration algorithm for the position estimation. This was apparent for the multilateration algorithm as it was very sensitive to noise and singularity errors, and also quickly decreased in accuracy at longer distances, which was to be expected due to the larger range of possible solutions combined with the limited sampling frequency. The double solutions generated by the multilateration algorithm added an element of arbitrariness.

The field tests showed that a continuous sync source is crucial to combat drift, otherwise the results might be unusable. The multilateration algorithms are not necessarily downgradable (to lower dimensions).

7.2 Future Work

A more robust multilateration algorithm that can be easily adapted to a larger/small number of nodes, along with a higher frequency clock and more discerning/adaptive detection algorithms that require less, or no manual error correction. Also, a library or protocol for easier communication between RTC-equipped nodes, and make use of the hardware originally intended by OCTECH for this thesis. For example, one could construct a setup where the detection, synchronization and multilateration is automated, the sensors continually recording and detecting events, the main unit

calculating and presenting solutions on-the-fly. The parameters of the algorithms might also be changed by the user in real-time to improve performance.

Bibliography

Bateman, H. (1918). “Mathematical theory of sound ranging”. *Monthly Weather Review* **46**:1. https://journals.ametsoc.org/mwr/article/46/1/4/85197/MATHEMATICAL-THEORY-OF-SOUND-RANGING.

Blewitt, G. (1997). *Basics of the gps technique: observation equations*. Department of Geomatics, University of Newcastle.

Bowditch, N. (2019). *The American Practical Navigator*. NSN 7642014014652. National Geospacial-Intelligence Agency.

Cramer, O. (1993). “The variation of the specific heat ratio and the speed of sound in air with temperature, pressure, humidity, and co2 concentration”. *The Journal of the Acoustical Society of America* **93**:5. Adapted from http://resource.npl.co.uk/acoustics/techguides/speedair/.

Grönroos, S., K. Nybom, and J. Björkqvist (2017). “Synchronization of low-cost distributed spectrum sensing nodes for multilateration-based geolocation”. *Analog Integrated Circuits and Signal Processing*. https://link.springer.com/article/10.1007/s10470-017-1094-0.

Kitic, S., C. Gaultier, and G. Pallone (2020). “Mathematical theory of sound ranging”. https://arxiv.org/pdf/1910.10661.pdf.

Kloot, W. V. der (2005). “Lawrence bragg’s role in the development of sound-ranging in world war i”. *Notes and Records of The Royal Society* **59**:3.

Larsson, M., V. Larsson, K. Åström, and M. Oskarsson (2019). “Optimal trilateration is an eigenvalue problem”. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE.

Misra, D. and R. Bucher (2002). “A synthesizable VHDL model of the exact solution for three-dimensional hyperbolic positioning system”. *VLSI Design* **15**:2, pp. 507–520.

Oliveira Jr., E. M., M. L. O. Souza, H. K. Kuga, and R. V. F. Lopes (2009). “Clock synchronization via kalman filtering”. DINCON’09, 8th Brazilian conference on dynamics, control and applications.

Stefanski, J. and J. Sadowski (2018). “Tdoa versus atdoa for wide area multilateration system”. *EURASIP Journal on Wireless Communications and Networking* 179. https://link.springer.com/article/10.1186/s13638-018-1191-5.

STMicroelectronics (2008). *AN2678, Application Note*.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
September 2020

Document Number
TFRT-6116

Author(s)
Tobias Samuelsson

Supervisor
Anton Cervin, Dept. of Automatic Control, Lund University, Sweden
Martina Maggio, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Sound ranging using multilateration and Kalman filter

Abstract

Sound ranging is a method to locate sources of sound waves using microphones or other types of receivers at known reference positions. Nowadays, these receivers can be made both very small and powerful, lasting for a long time on a single battery charge, making them suitable for longtime outdoor purposes such as detecting artillery fire or other loud sounds.

In this master thesis, we present a sound ranging algorithm based on threedimensional multilateration, Kalman filtering and sound detection. Results from simulations are given. Results from field experiments are presented. Furthermore, we compare the two and and evaluate their differences. Finally, suggestions on future work are provided.

The evaluation shows that while the Kalman filter and detection algorithm performs well even with high levels of measurement noise, the multilateration algorithm can provide an accurate source positioning, but it is very sensitive to errors, with performance degrading heavily due to low sampling resolution, too close node positioning, undefined sound peaks and a lack of robustness in the multilateration algorithm.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-55

Recipient's notes

Security classification

http://www.control.lth.se/publications/