

הפקולטה להנדסה ע"ש איבי ואלדר פליישמן
אוניברסיטת תל אביב



בי"ס להנדסת חשמל

פרויקט מס' 17-2-1-1488

דו"ח סיכום

שם הפרויקט: איקון מבוסס קול
מבצעים:

ת.ז. 203700505
ת.ז. 302249925

שם: רועי טולדנו
שם: ירדן אברהם

מנחה: פרופ' יוסף יובל

מקום ביצוע הפרויקט: הגן הזואולוגי – אוניברסיטת תל אביב

Table of Content:

Abstract:	4
Block Diagram:	5
Introduction:	6
Theoretical Background:	9
Simulation	14
Implementation	21
Results Analysis	35
Conclusions and suggestions for future research:	49
References:	51

List of Figures:

Figure 1 - Block Diagram	5
Figure 2 - Trilateration in 3D	10
Figure 3 - TDoA in 3D	12
Figure 4 - Correlations with an Upchirp through the three different windows	14
Figure 5 - Correlations with an Down chirp through the three different windows	15
Figure 6 - Correlations with an Up-Down chirp through the three different windows	15
Figure 7 - correlation performance of the digital string	16
Figure 8 - An example of a generated BFSK signal	17
Figure 9 - Correlations with a BFSK signal between all 4 signals	17
Figure 10 - Test results with known expected value using LuT algorithm	19
Figure 11 - Test results with known expected value using Chan's algorithm	20
Figure 12 - Simulation of our setup in the room in 2D	21
Figure 13 - ToA detector's block diagram	22
Figure 14 - All BPF used for each speaker	23
Figure 15 - LuT algorithm's block diagram	24
Figure 16 - Chan's algorithm's block diagram	25
Figure 17 - Frequency response of the speakers	26
Figure 18 - Polar diagram of the speakers	26
Figure 19 - Frequency response of the microphone	27
Figure 20 - Polar diagram of the microphone	27
Figure 21 - An 8 second record from the US microphone	35
Figure 22 - A closeup to a single transmission record	35
Figure 23 - FFT of the recorded signal before being filtered	36
Figure 24 - A closeup of the FFT of the recorded signal before being filtered	36
Figure 25 - Spectrogram of recorded signal	37
Figure 26 - Closeup of the spectrogram of recorded signal	37
Figure 27 - The recorded signal with the filtered recorded signal for each speaker	38
Figure 28 - Closeup of the recorded signal with the filtered recorded signal for each speaker	38
Figure 29 - Time correlation of each speakers filtered recorded signal	39
Figure 30 - Closeup of time correlation of each speakers filtered recorded signal	39
Figure 31 - Time correlation of all the speaker's filtered recording together	40
Figure 32 - Closeup of the time correlation of all speakers together, with the peaks	40
Figure 33 - Closeup of the time correlation of all speakers together, with the fine-tuned peaks	41
Figure 34 - TDoA results, with the outliers which we eliminated from the results	41
Figure 35 - Positioning results vs expected, in 3D with a constant z parameter, for LuT	42
Figure 36 - Euclidean error from expected for each point, in 3D with a constant z parameter, for LuT	42
Figure 37 - Average Euclidean error from expected, in 3D with a constant z parameter, for LuT	42
Figure 38 - Positioning results vs expected, in 3D, for LuT	43
Figure 39 - Euclidean error from expected for each point, in 3D, for LuT	43
Figure 40 - Positioning results vs expected, in 2D, for LuT	44
Figure 41 - Euclidean error from expected for each point, in 2D, for LuT	44
Figure 42 - Average Euclidean error from expected in 3D vs 2D, for LuT	44
Figure 43 - Euclidean error from expected for each point, in 2D with Chan's algorithm	45
Figure 44 - Euclidean error from expected for each point, in 3D with Chan's algorithm	45
Figure 45 - Average Euclidean error from expected with Chan's algorithm, in 3D vs 2D	45
Figure 46 - Positioning results vs expected, in 2D with BFSK, for LuT	46
Figure 47 - Positioning results vs expected, in 3D with BFSK, for LuT	47
Figure 48 - Average Euclidean error from expected with BFSK, in 3D vs 2D, for LuT	47
Figure 49 - Euclidean error from expected for each point with a constant Z input, with BFSK for LuT	47
Figure 50 - Average Euclidean error from expected with a constant Z, with BFSK for LuT	48
Figure 51 - Illustration of PDOP for TDoA system. The PDOP at B is greater than at A.	50

List of Tables:

Table 1 - Microphone Location	18
Table 2 - ToAs calculated in seconds	18
Table 3 - Simulation results comparison	20
Table 4 - Transmitting unit's parameters	31
Table 5 - All python's parameters	34
Table 6 - Test results comparison with Simulation	46
Table 7 - Test results with digital signal	48

Abstract:

For years, applications and devices have been using location recognition in a wide variety of different fields. Such as traffic navigation, autonomous device operations (robots) in space, proximity advertising, service search, location of military targets, and much more. All these applications work on the basis of GPS communication - satellite communications, the problem with this sort of communication is its absence and inaccuracy in closed places and places where the signal between the device and satellite is interrupted and / or indirect.

In order to overcome these problems, the high-tech industry is looking for many solutions and several different methods are being studied, such as:

- Using AoA (Angle of Arrival) and AoD (Angle of Departure) in Bluetooth Low Energy transmission between a device and beacons in closed spaces.
- Use of deep learning and computer vision technologies with autonomous devices.
- Analysis of transmission strengths of WLAN AP (Wireless LAN Access Points) and DATABASE mapping.
- Continuous positioning by analysis of sound waves transmitted from devices with a known position or analysis of the reflection of sound waves transmitted by the transmitter in space.

Our project will deal with the last subject - location using sound waves. This method is based on the bats positioning method that uses Ultrasonic waves (waves at a relatively high frequencies - not heard by the human ear) and the reception of their reflections to identify objects in their surroundings.

From a physics point of view, sound is an energy that spreads through the air, liquids, or solids in the form of periodic oscillations - waves. When sound is formed it vibrates the molecules in which it passes through. The molecules collide with each other and create a "longitudinal wave". Waves in general, and sound waves in particular, are characterized by their cycle time, amplitude and phase, and can process signals like electrical signals.

The use of TDoA (time difference of arrival) is a possible method of measurement due to the long cycle length (relative to the RF signals) that these waves are characterized and calculated by phase measurements, attenuation formulas, and location of the transmitters.

During the course of the project, we will develop a system that is able to identifies a certain location in a closed space by using at least 4 transmitters to transmit a certain kind of chirp signal and a receiver which will record the signals being transmitted. Afterwards the recorded signal will be analyzed, and the position of the receiver in the room will be calculated.

In order to optimize the study, we will examine several algorithms for calculating the location and focus on the success of filtering the resonances and reflection of the signal within the room.

Block Diagram:

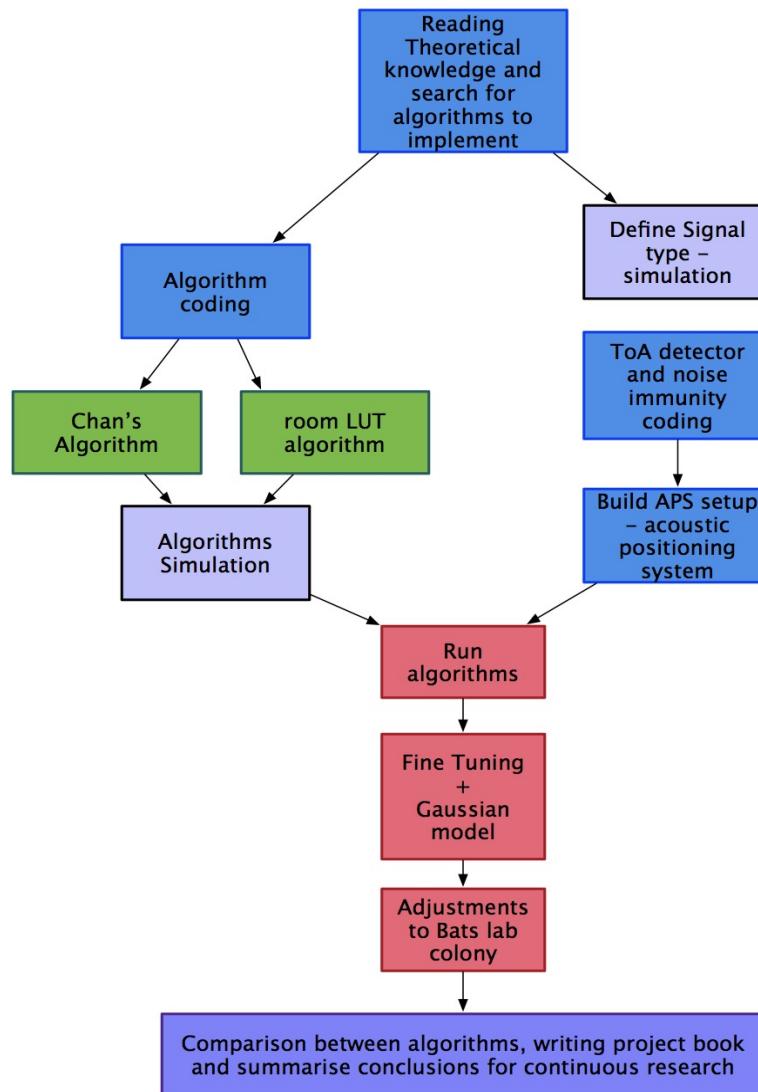


Figure 1 - Block Diagram

1 Introduction:

Purpose of the Project:

Planning and developing a localization system that mimics the way bats position themselves in space today, with the best accuracy possible. Examining different algorithms using different signals that are being transmitted.

Motivation:

Our project addresses several difficulties in today's positioning technology. First, satellite communication is not always available and is often disrupted. In these kind of places (greenhouses, malls, offices, tunnels, etc.), another way of positioning is needed in order to navigate and to be able to produce an autonomous movement.

For example, in greenhouses the GPS signals are usually of low quality due to the impact of the plastic coverings and metal structures. In addition, greater accuracy is required for navigation due to the narrow corridors between plants and the use of sound waves for system positioning is an internal solution that does not require communication of any kind with any external product. The use of optical or magnetic tracking is problematic due to the density of the plants and therefore the US signals that have a relatively long wavelength (in comparison to an RF detectors or optical waves) are more durable and are able to easily bypass the dense vegetation and are not affected by the magnetism of the metals in the greenhouse. Furthermore, acoustic waves are more durable to different lighting conditions.

There is no doubt about the importance and contribution of our project, since it is the basis for the ongoing research and implementation of autonomous machines in closed spaces or disrupted satellite communication areas. All autonomous movement is based on localization and spatial analysis to determine the next movements of the machine. Without this technology, all autonomic machines will be static and will not be able to give the expected response in a dynamic world.

Following the example from before, the greenhouses model, automating the agricultural industry will result in efficiency, lower costs for the producer - will have to employ fewer workers, and the lack of grain waste due to the control and performance of machines in real time. Finally, all these factors taking into consideration will cause market growth and lower consumer prices in the long run.

Approach to solving the problem:

Using an existing localization algorithm called Chan's Algorithm that is capable of calculating the given position of the receiver using TDoAs (Time Difference of Arrivals).

Developing an algorithm that is based upon a LuT (Lookup Table) of TDoAs of a certain room that is being used and calculating the Euclidean distance cost between the TDoA vector found to the room's LuT expected TDoA vectors with changeable quantization factor.

Examining different signals that are being transmitted and implementing them in the algorithms. Understating which signal yields better results. In addition, passing the

signals through different windows, again to be able to identify what will give us the best results, accurate wise.

Existing works and algorithms / implementations:

All available localization methods can be divided into two categories: Range-based or range-free, based on whether the distance must be ranged or not. Range-free location systems first establish a fingerprint database by collecting the received signal strength indication (RSSI) data between the Tags and Anchors to obtain a signal propagation model, then real-time RSSI parameters are measured and matched to the fingerprint database to estimate the coordinate value of Tag. Range-based localization include angle of arrival (AOA), RSSI, time of arrival (TOA), and TDoA.

The values of these properties as measured by the receiver depend on the position of the receiver relative to the transmitter. This dependency can be used for position estimation. The position can be estimated given the assumed propagation model and the functional relation between measurements and position. The different types of measurements are explained in more detail below.

Angle of Arrival (AOA)

In the absence of any discontinuities in the propagation medium, the waveform arriving at the receiver will travel along the fastest path between the transmitter and the receiver. A measurement of the direction of the incident signal can be related to the geometric angle between the transmitter and the receiver. The AOA can be measured by noting the angle at which the received signal strength at the receiver is maximum or minimum while varying the radiation pattern of either the transmitting or receiving antenna, or by using an antenna array at the receiver and noting the difference in the time of arrival (or phase) at each of the array element.

Received Signal Strength Input (RSSI)

The power of an electromagnetic signal decreases as the wave propagates further and further from the transmitter due to the inverse-square law of electromagnetic radiation. In free space, the square of the distance between a transmitter and receiver is inversely proportional to the power density of the electromagnetic wave at the receiver. Thus, the distance between the transmitter and receiver can be calculated if the transmission power and the RSS are known.

In a practical environment, the signal will travel along different paths to the receiver due to discontinuities in the propagation medium, such as reflections of the wave against obstacles in the environment. The receiver will not only measure the power of the wave that travelled along the shortest path but also the vectoral combination of time-delayed signals that travelled along different paths as well as interfering signals from other transmission sources. Subsequently, the power measured by a receiver will not decrease monotonically as the distance between the transmitter and receiver increases, and the relationship will change according to the environment. This has the effect that the position to RSS mapping is non-trivial and cannot be inverted. However,

various models exist for obtaining a position estimate from RSS measurements using a database of reference measurements that were obtained beforehand.

Time of Arrival (TOA)

Electromagnetic waves travel at a constant speed in a homogeneous medium, namely the speed of light. There is thus a linear relation between propagation time and propagation distance. The distance between a transmitter and receiver can be calculated by multiplying the time it takes a signal to travel along the shortest path from the transmitter to the receiver with the known propagation speed in the medium.

The TDoA model can be used to measure the differences in the times at which signal from the Tag directly or indirectly arrive at multiple Anchors. Because the TDoA model only requires clock synchronization between the Anchors, its hardware equipment can be more simply and easily implemented than the TOA model. The TDoA model is also lower in cost than the AOA, and has stronger anti-interference ability than RSSI (as the signal information of RSSI is vulnerable to factors like temperature, space, scene, or change in receiving terminals)

Chan, Taylor, extended Kalman filter (EKF), and particle filter (PF), et al., are frequently-used algorithms in TDoA location. The positioning precision of Chan algorithm decreases significantly in NLOS (Non-Line of Sight) environment. Taylor algorithm can obtain accurate computation when initial estimated situation approximates the actual location, otherwise, it is difficult for the algorithms to ensure the convergence.

PF is an iterative estimation method that utilizes the Bayes rule which computes the posterior distribution of the state vector based on the previous and current state observations [18]. The implementation of PF involves three steps. The first step is the initialization stage where N particles denoted as $P(s_0)$ were randomly sampled around the initial distribution of the system. The second step is the sampling stage where N samples x_k^i were drawn from $P(\tilde{x}_k | \tilde{x}_{k-1}^i)$. Lastly, the weight (likelihood) of each of the samples was computed based on Gaussian distribution model in Equation and generate the probability function of the next location:

$$[1] P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

- Where x is the estimation sample and μ is the observation value at the current time.

PF and Kalman are more complex and more suitable to RT localization, our system working offline so we decided not to implement them.

2 Theoretical Background:

The basis of both algorithms is the estimation of ToAs and TDoAs when we broadcast a certain chirp signal, analog and digital signal.

First, we will consider the usage of chirp signals.

Chirp signals are angle modulated sweeping signals. They pass in a linear or nonlinear way the whole frequency bandwidth $B[\text{Hz}]$ from one end to the other one by a sinusoidal waveform of constant amplitude within a certain time $T[\text{s}]$. If this sequence of frequencies is swept from the lowest to the highest frequency limit, we called it an Upchirp. In the opposite direction it is a Downchirp

From a theoretical point of view, chirp signals provide an astonishing number of advantages. They substantiate the following ideal features of a fundamental nature in communications engineering:

- They have a quasi-ideal rectangular spectrum to utilize the channel's capacity and to offer an optimal lowest spectral power density compared to all other existing transmission signals.
- They are programmable with respect to processing gain, which means it is possible to achieve determinable distances in ranging while at the same time suppress adaptively disturbances and noise.
- They allow a high resolution on time axis and are, therefore, best suited for ranging.
- They enable systems that prove a very short latency by asynchronously working correlative transmission systems.
- Chirp Signals prove the ability to superpose these long signals to allow the data rate and bit energy to vary adaptively or to generate multi chirps in different combinations, which achieves other advantages.

Second, we will describe the estimation of both ToAs and TDoAs.

Introduction to TOA positioning:

When a constant propagation speed is assumed, and the time of transmission as well as the time the Line-Of-Sight (LOS) signal arrives at the receiver are known, the

distance between the transmitters and the receiver can be calculated by multiplying the propagation delay of the signal with the propagation speed:

$$[2] \quad r_i = \int_{t_0}^{t_i} c \, dt = c(t_i - t_0)$$

- r_i denotes the distance between a transmitter i and the receiver.
- t_0 denotes the propagation speed.
- t_i denotes the time of arrival at receiver from transmitter i .
- c denotes the propagation speed.

For a network-position in 3D space (like the one we use) if $\langle x, y, z \rangle$ is the unknown position of the receiver and $\langle x_i, y_i, z_i \rangle$ is the known position of transmitter i , then the equation above can be written as:

$$[3] \quad r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} = c(t_i - t_0)$$

Note: The equation above denotes a sphere with a radius of $c(t_i - t_0)$ centered around $\langle x, y, z \rangle$. Which means the receive can be located at any point along the perimeter of the sphere.

If we will take measurements with multiple transmitters, we will be able to determine where all spheres intersect and that determines the position of our receives position unambiguously. This process is known as 3D-Trilateration.

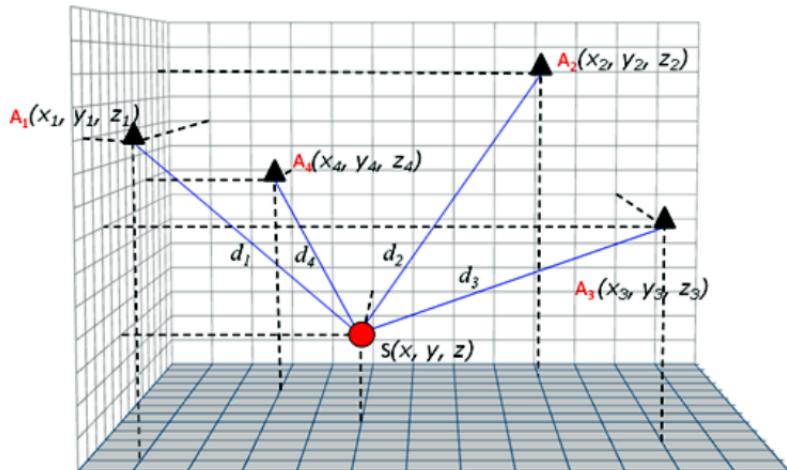


Figure 2 - Trilateration in 3D

The problem with the estimation of ToA is calculating the propagation delay, $(t_i - t_o)$. This makes sure the calculation is accurate, it requires both the transmitters and the receiver to have a clock that is synchronized with each other.

The propagation speed in air is about $3 \cdot 10^8 \left[\frac{m}{s} \right]$, which means that even a small error with the propagation delay can have a big impact on the accuracy of the receiver's position estimate.

Just to put things in perspective, an error of $1\mu[\text{sec}]$ would result in a distance measurement error of $300[m]$.

To be able to eliminate the time of transmission from the equations, we will use TDoA positioning.

TDoA is the calculation of the difference in propagation distance between pairs of transmitters instead of the absolute propagation distance.

From the equation above, if r_i and r_j denote the distance between the receiver and transmitter i and j respectively, and t_i and t_j the TOA at the respective transmitter, then:

$$[4] r_i - r_j = c(t_i - t_o) - c(t_j - t_o) = c(t_i - t_j)$$

Thus, the difference eliminated the transmission time t_0 from the equation, with the result that the difference in propagation distance can be calculated from the difference in the TOA. This technique is referred to as TDoA.

In our case, network-positioning, if $\mathbf{x} = < x, y, z >$ represents the unknown position of our receiver, and $\mathbf{x}_i = < x_i, y_i, z_i >$ and $\mathbf{x}_j = < x_j, y_j, z_j >$ the known positions of 2 base transmitters, then:

$$[5] \|\mathbf{x} - \mathbf{x}_i\| - \|\mathbf{x} - \mathbf{x}_j\| = c(t_i - t_j)$$

- $\|\mathbf{x}\|$ denotes the Euclidean length of vector \mathbf{x} .

Geometrically, in 3D space, this equation defines a locus of points of equal distance to the 2 transmitters being considered. This is the definition of one branch of a hyperboloid with the 2 transmitters as focal points.

For a single TDoA measurement, the position of the receiver is ambiguous, it can be located at any point along the hyperboloidal surface. TDoA measurements between

multiple independent pairs of transmitters define different hyperboloids of which the intersection yields the position of the receiver unit.

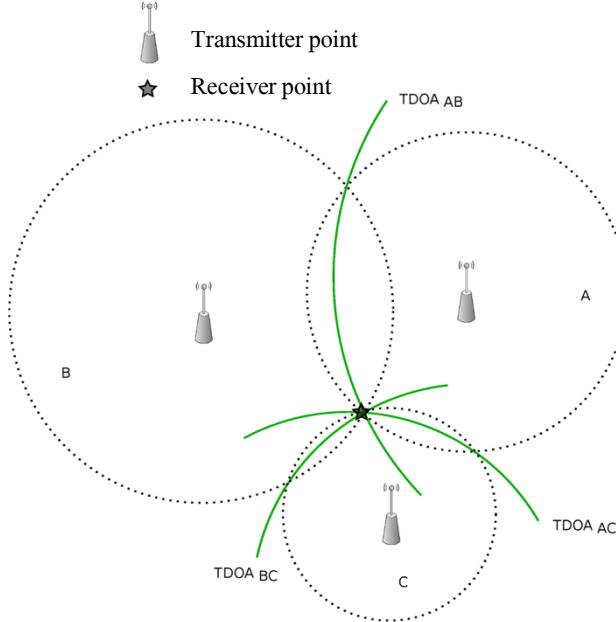


Figure 3 - TDoA in 3D

Chan's algorithm solves the hyperbolic equations generated by the TDoA procedures, explained above.

Now let's assume that: $k_i^2 = x_i^2 + y_i^2 + z_i^2$ then the equation above can be rewritten as:

$$[6] \quad r_i^2 = k_i^2 - 2x_i x - 2y_i y - 2z_i z + x^2 + y^2 + z^2$$

Chan observed that TDoA generated a non-linear hyperbolic function of (x) , (y) and (z) , which is difficult to solve. The easiest way to solve a non-linear equation is to break it down into several linear equation using the Chan's method. By using the previous equation, we get:

$$[7] \quad r_i^2 = (r_{i,1} + r_1)^2$$

$$[8] \quad r_{i,1}^2 + 2r_{i,1}r_1 + r_1^2 = k_i^2 - 2x_i x - 2y_i y - 2z_i z + x^2 + y^2 + z^2$$

Expanding equation [2] for $(i = 1)$, generates:

$$[9] \quad r_1^2 = x_1^2 + y_1^2 + z_1^2 - 2x_1 x - 2y_1 y - 2z_1 z + x^2 + y^2 + z^2$$

Now from combining the last two equation, we obtain:

$$[10] \quad r_{i,1}^2 + 2r_{i,1}r_1 = k_i^2 - k_1^2 - 2(x_i - x_1)x - 2(y_i - y_1)y - 2(z_i - z_1)z$$

By letting: $X_{i,1} = x_i - x_1$, $Y_{i,1} = y_i - y_1$ and $Z_{i,1} = z_i - z_1$, we are able to simplify equation [9] to be as:

$$[11] \quad r_{i,1}^2 + 2r_{i,1}r_1 = k_i^2 - k_1^2 - 2X_{i,1}x - 2Y_{i,1}y - 2Z_{i,1}z$$

Now we will rewrite equation [10] for our system, which includes of four transmitters, and a receiver:

$$\begin{aligned}
 [12] \quad & -2X_{2,1}x - 2Y_{2,1}y - 2Z_{2,1}z = 2r_{2,1}r_1 + r_{2,1}^2 - k_2^2 + k_1^2 \\
 [13] \quad & -2X_{3,1}x - 2Y_{3,1}y - 2Z_{3,1}z = 2r_{3,1}r_1 + r_{3,1}^2 - k_3^2 + k_1^2 \\
 [14] \quad & -2X_{4,1}x - 2Y_{4,1}y - 2Z_{4,1}z = 2r_{4,1}r_1 + r_{4,1}^2 - k_4^2 + k_1^2
 \end{aligned}$$

- Where ($i = 2$), ($i = 3$) and ($i = 4$) are for the three neighboring transmitting units (other than the home transmitter, ($i = 1$)).

We can represent the equations in the form of the following matrices:

$$[15] \quad -2 \begin{bmatrix} X_{2,1} & Y_{2,1} & Z_{2,1} \\ X_{3,1} & Y_{3,1} & Z_{3,1} \\ X_{4,1} & Y_{4,1} & Z_{4,1} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 2 \begin{bmatrix} r_{2,1} \\ r_{3,1} \\ r_{4,1} \end{bmatrix} r_1 + \begin{bmatrix} r_{2,1}^2 - k_2^2 + k_1^2 \\ r_{3,1}^2 - k_3^2 + k_1^2 \\ r_{4,1}^2 - k_4^2 + k_1^2 \end{bmatrix}$$

Or,

$$[16] \quad \begin{bmatrix} x \\ y \\ z \end{bmatrix} = - \begin{bmatrix} X_{2,1} & Y_{2,1} & Z_{2,1} \\ X_{3,1} & Y_{3,1} & Z_{3,1} \\ X_{4,1} & Y_{4,1} & Z_{4,1} \end{bmatrix}^{-1} \cdot \begin{Bmatrix} \begin{bmatrix} r_{2,1} \\ r_{3,1} \\ r_{4,1} \end{bmatrix} r_1 + \frac{1}{2} \begin{bmatrix} r_{2,1}^2 - k_2^2 + k_1^2 \\ r_{3,1}^2 - k_3^2 + k_1^2 \\ r_{4,1}^2 - k_4^2 + k_1^2 \end{bmatrix} \end{Bmatrix}$$

- Where (x, y, z) represents the location of the receiver, and (r_1) is obtained from equation [8] above and $(r_{2,1})$, $(r_{3,1})$, $(r_{4,1})$ are obtained from equation [10]. Considering that:

$$[17] \quad k_i^2 = x_i^2 + y_i^2 + z_i^2 |_{i \in \{1,2,3,4\}}$$

Our second algorithm is based on a LuT which is a matrix that replaces runtime computation with a simpler matrix indexing operation. The saving in terms of processing time can be significant, since retrieving a value from memory is often faster than undergoing an “expensive” computation. In our case the table is precalculated, we calculate all the possible TDoAs in a certain space by dividing the space into a 3rd dimension space (represented as a matrix with a changeable resolution). Afterwards we calculate the Euclidean distance cost between the TDoA vector found to the room’s LuT expected TDoA vectors. After the closest distance is found we reverse engineer the given TDoA found in the table to a point in the 3D space.

3 Simulation

During the simulation we wished to define the transmitting system that would yield the best TDoA positioning estimation and to be able to define the algorithm to collect all the relevant measurements.

First, we wanted to figure out what window we should pass the transmitted chirp through, in order to produce on one hand a good autocorrelation, but on the other a “bad” cross correlation.

In the process we also checked three types of chirps, an upchirp, downchirp and an up-down chirp, while passing them through three different windows, Blackman-Harris, Hamming and Diagonal-Linear window.

The results were as the following:

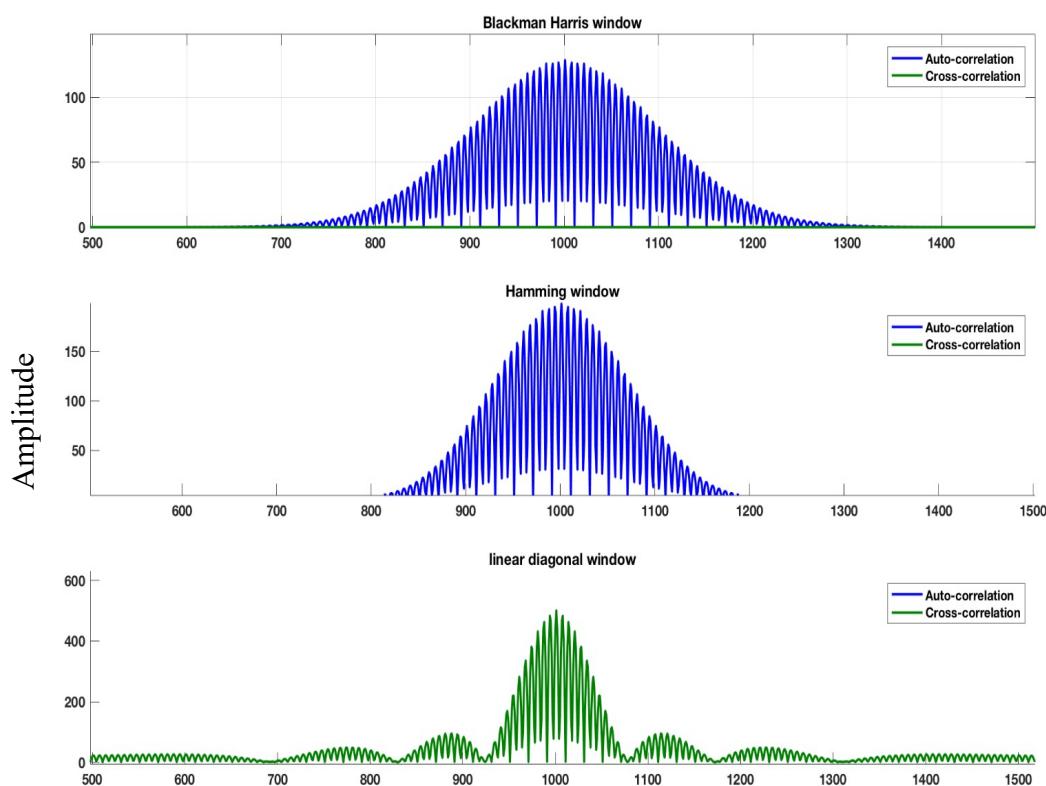


Figure 4 - Correlations with an Upchirp through the three different windows

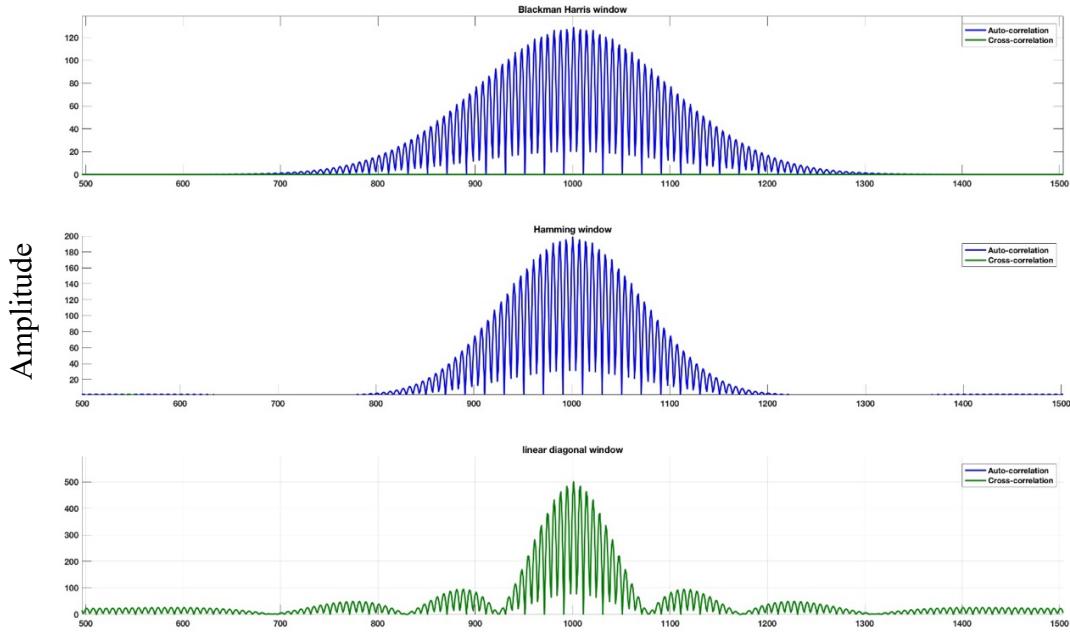


Figure 5 - Correlations with an Down chirp through the three different windows

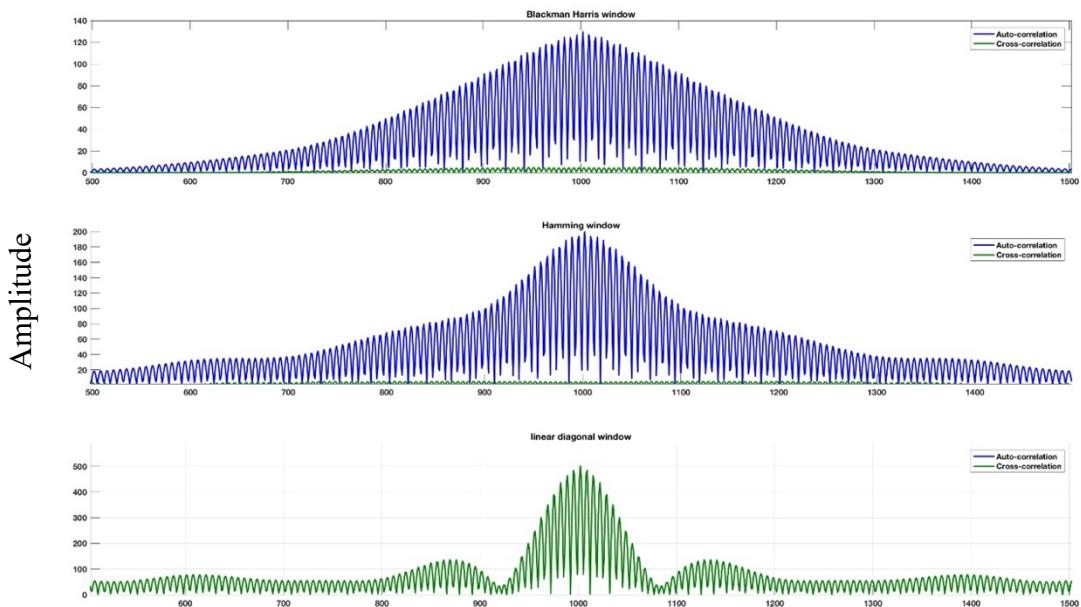


Figure 6 - Correlations with an Up-Down chirp through the three different windows

As we can see the Hamming window has a thinner lobe, but we noticed that through our equipment it generates a “click” noise while being transmitted which affected the received signal by the microphone. Regarding the Linear-Diagonal window, although the maximum peak is much higher compared to the other windows, we noticed that the autocorrelation is equal to the cross-correlation between two different speaker’s signals. Which means it would be much harder to detect two different signals from one

and other. Therefore, we chose to use in our project the Blackman-Harris window as our envelope, that has an almost zero cross correlation, and a good autocorrelation.

Regarding the three different chirps we checked, we realized that the upchirp and downchirp gave the same results. While the up-down chirp yielded a much thicker lobe, which means a less distinguishable maximum peak. Therefore, we decided to work with the upchirp in our project.

In addition, we decided to also compare the chosen signal and window with a digital signal, BFSK (Binary Frequency Shift Keying). Frequency-shift keying (FSK) is a frequency modulation scheme in which digital information is transmitted through discrete frequency changes of a carrier wave. The simplest FSK is binary FSK (BFSK). BFSK uses a pair of discrete frequencies to transmit binary (0s and 1s) information. With this scheme, the "1" is called the mark frequency and the "0" is called the space frequency.

A BFSK signal can be expressed as so,

$$[18] \quad S_{BFSK}(t) = A \cos(2\pi(f_c + m(t)f_m)t + \phi_0)$$

- Where, $m(t) = -1$ or 1 . f_m = half of the frequency distance between bits signals. A , f_c , ϕ_0 are the amplitude, frequency and phase of the sinusoidal carrier signal.

Our expected signal length is $2m[\text{sec}]$ that yields approximately 15 bits random binary string to be modulated to the BFSK signal.

First, we checked the correlation performance of the digital string (15 bits each):

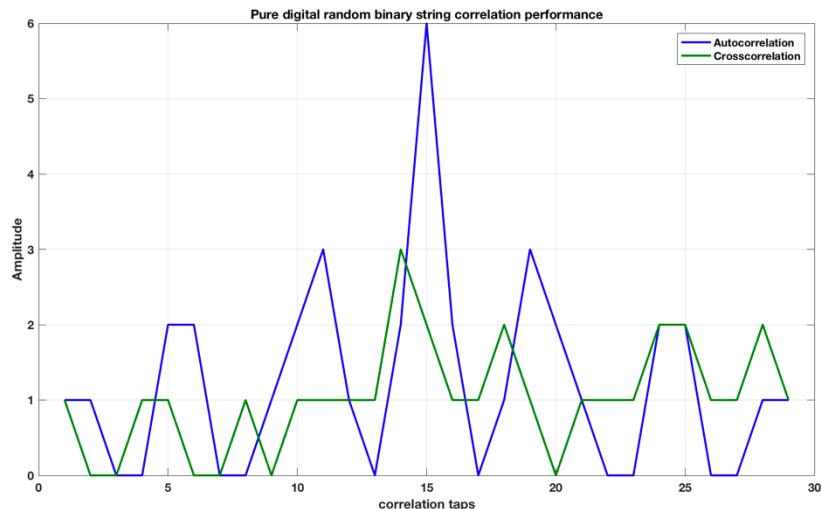


Figure 7 - correlation performance of the digital string

An example of a generated BFSK signal is the following:

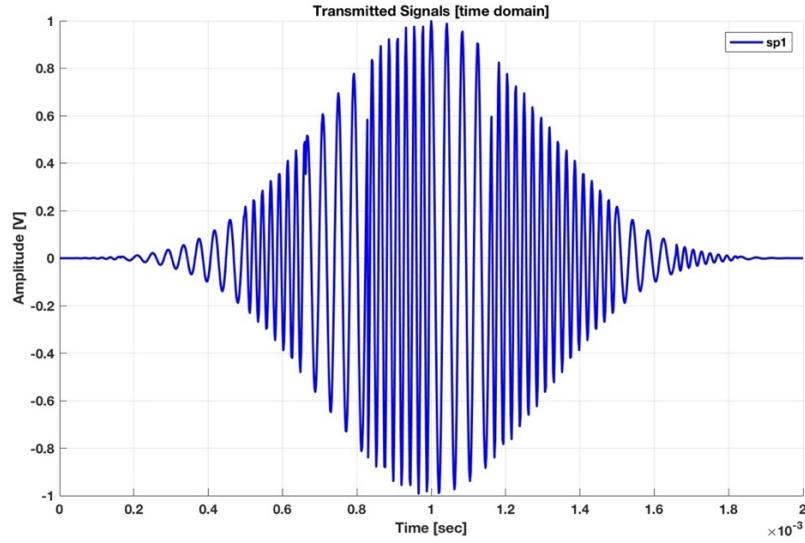


Figure 8 - An example of a generated BFSK signal

In practice, the generated signals we checked was with a $f_m = 3000\text{Hz}$ in order to have four signals with difference frequency range that don't overlap each other.

The results were as the following:

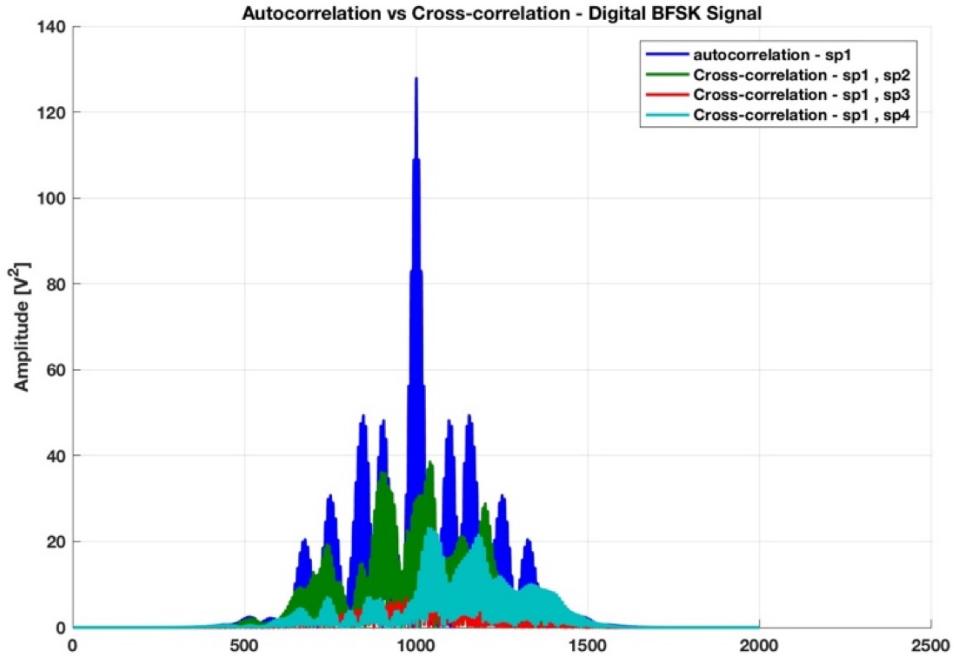


Figure 9 - Correlations with a BFSK signal between all 4 signals

As we can see, the results yield a very good autocorrelation, with a thin maximum lobe. The cross correlation we got was not as good compared to the analog signal discussed beforehand.

The second thing we wanted to verify before implementing the system was our choice as to which algorithm to use. We made a comparison between the results we would get from both algorithm when inputted a manually calculated TDoA from a known microphone location. That way we could see the algorithm's results compared to a known expected value.

We expected to see that the two algorithms will yield results that are very close to the expected.

The known parameters we inputted for both algorithms were:

Chosen Microphone Location		
X [m]	Y [m]	Z [m]
0.4947561	0.99443038	1.4784

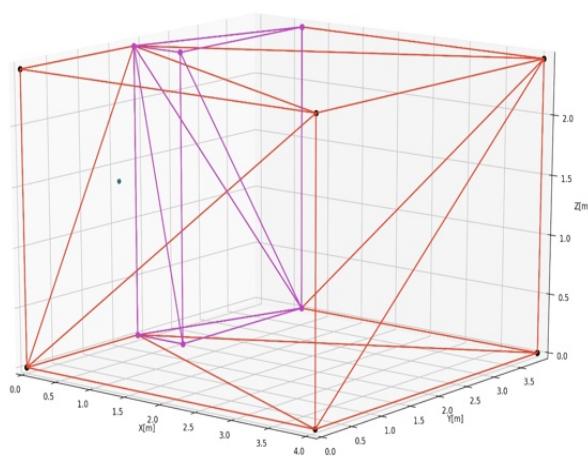
Table 1 - Microphone Location

ToAs calculated in seconds			
ToA speaker 1	ToA speaker 2	ToA speaker 3	ToA speaker 4
1	1.00482975	1.00588202	1.00940237
1.5	1.50482975	1.50588202	1.50940237
2	2.00482975	2.00588202	2.00940237
2.5	2.50482975	2.50588202	2.50940237
3	3.00482975	3.00588202	3.00940237
3.5	3.50482975	3.50588202	3.50940237
4	4.00482975	4.00588202	4.00940237
4.5	4.50482975	4.50588202	4.50940237
5	5.00482975	5.00588202	5.00940237
5.5	5.50482975	5.50588202	5.50940237
6	6.00482975	6.00588202	6.00940237

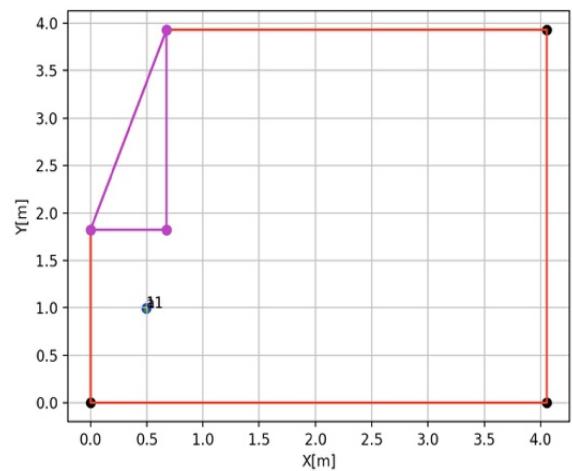
Table 2 - ToAs calculated in seconds

Regarding the LuT algorithm:

Location of microphone in 3D



Location of microphone in 2D



LuT error vs matrix resolution

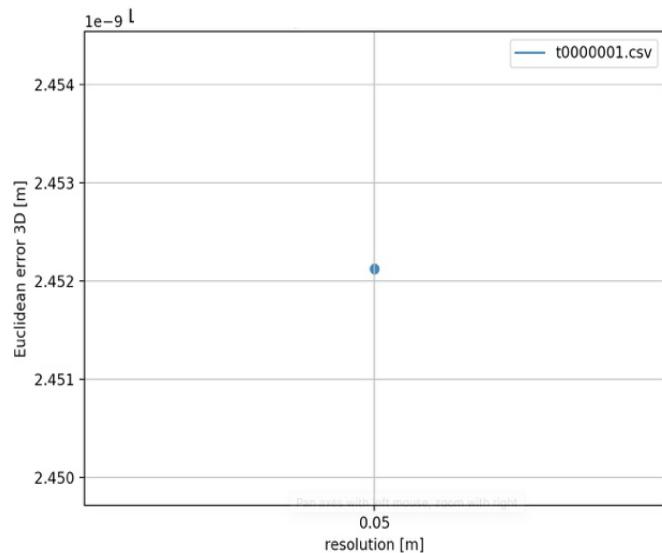


Figure 10 - Test results with known expected value using LuT algorithm

Regarding Chan's algorithm:

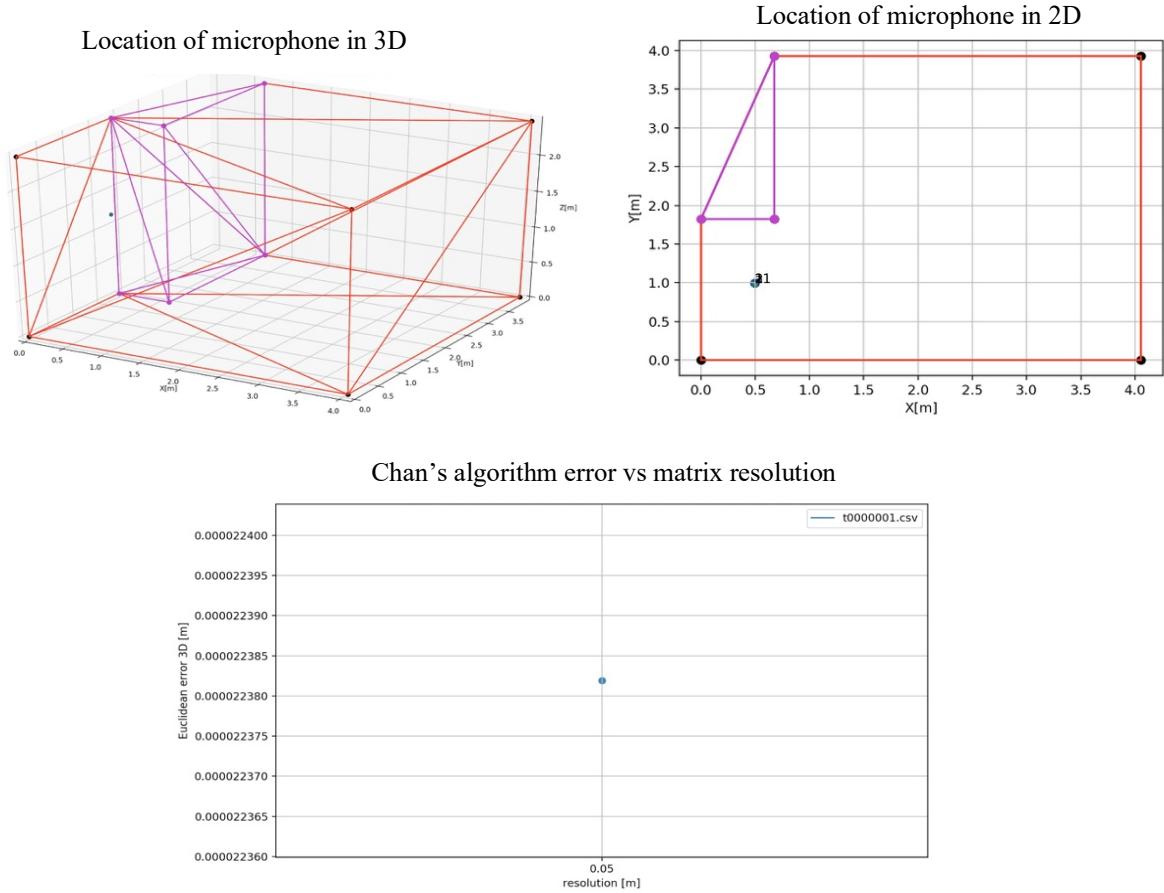


Figure 11 - Test results with known expected value using Chan's algorithm

Simulation results comparison	
Algorithm	Error
LuT	2.45E-09
Chan	2.24E-05

Table 3 - Simulation results comparison

First and foremost, we can see from the results above that in both cases we got good results, as expected. In addition, we can notice that the error we got using the LuT algorithm is much smaller compared to the error giving by Chan's algorithm. This makes sense giving the fact that Chan's algorithm is mathematically based, which means that there is a higher chance of receiving an error along the way, because a mathematical solution isn't immune to noise or error and can easily divergence.

To conclude, the test results we received are as expected so we can continue with our work, using these two algorithms.

4 Implementation

Our system includes:

- 4 speakers
- NI DAQ
- Ultrasound microphone
- A PC as a processing unit
 - MATLAB transmission code.
 - Python algorithm code.

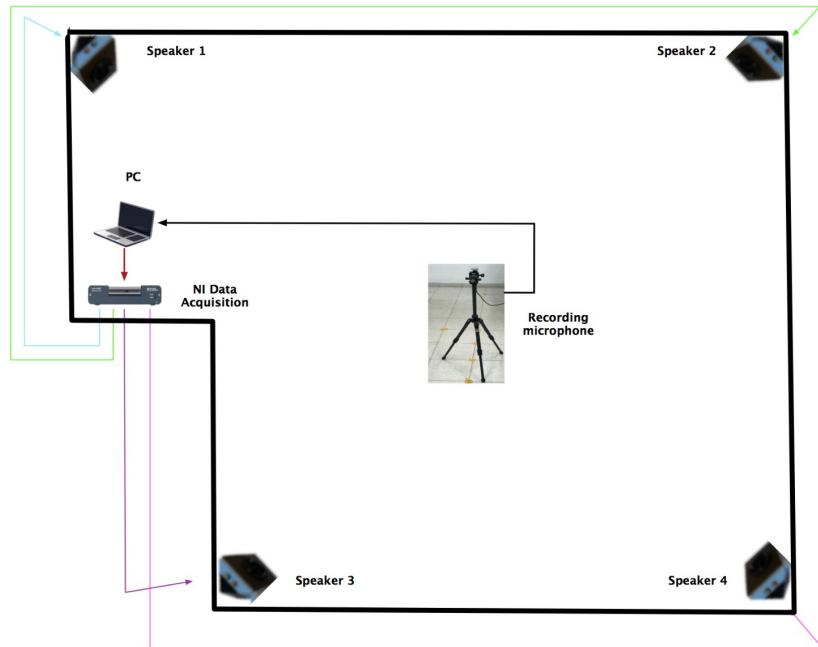


Figure 12 - Simulation of our setup in the room in 2D

As seen in the figure above, we decided to place the 4 speakers in this type of configuration (in the corners of the room) in order to minimize the ambiguous function as much as possible. In addition, we placed the speakers in different heights from one and other in order to add another dimension to the results.

After setting up our system, the next thing there was to do is to transmit the chosen chirp signal from all 4 speaker simultaneously, while each chirp had a different frequency range:

- Speaker 1: $34k[Hz] < f < 41k[Hz]$
- Speaker 2: $42k[Hz] < f < 49k[Hz]$
- Speaker 3: $20k[Hz] < f < 27k[Hz]$
- Speaker 4: $27k[Hz] < f < 34k[Hz]$

After recording the transmitted signal with the ultrasound microphone, which outputs as .wav file, we passed the .wav file into our processing unit in order to calculate the ToAs.

This process is described in the following block diagram:

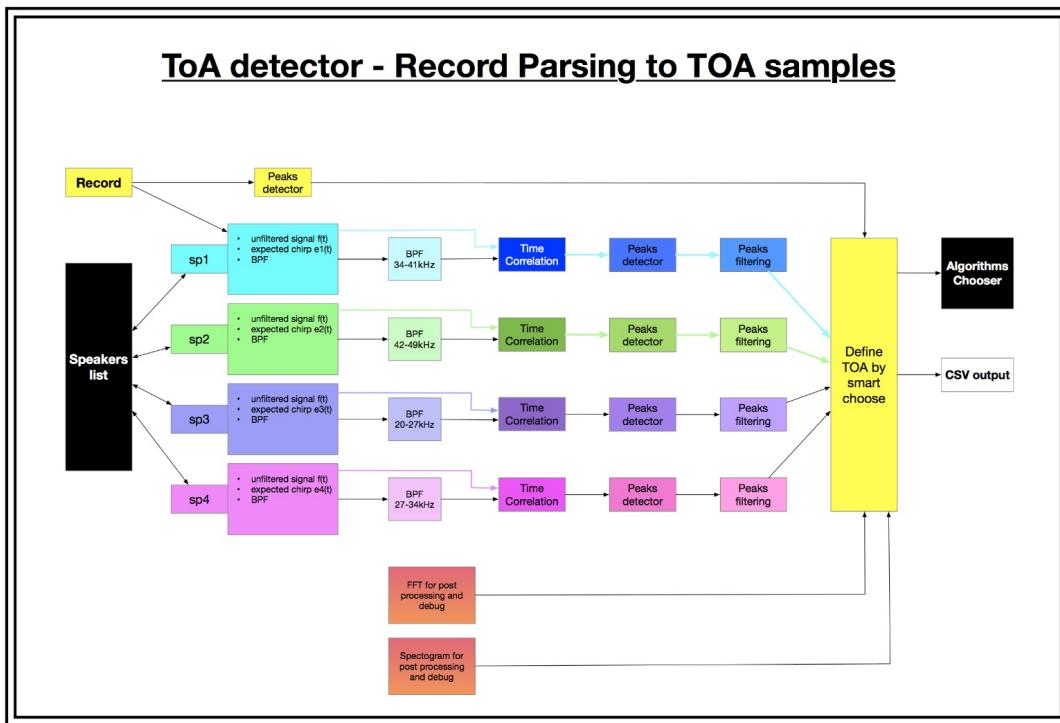


Figure 13 - ToA detector's block diagram

The block diagram above describes the flow for each speaker. We will go into detail regarding one of the speaker's flows.

The recorded signal passes through a BPF (Band Pass Filter) with a unique frequency range that is specific to that speaker. The BPF is a Linear-Phase FIR Filter type 1. Type 1 filters (odd number of taps, even symmetry) there are no constraints on the zeros at $z = 1$ and $z = -1$, the phase shift is zero (apart from the linear phase), and the group delay is an integer value. These filters are pretty universal and good for our needs, they cannot be used whenever a 90 degrees phase shift is necessary, e.g. for differentiators or Hilbert transformers.

These figures below show the BPF filters we used with their frequency and phase response:

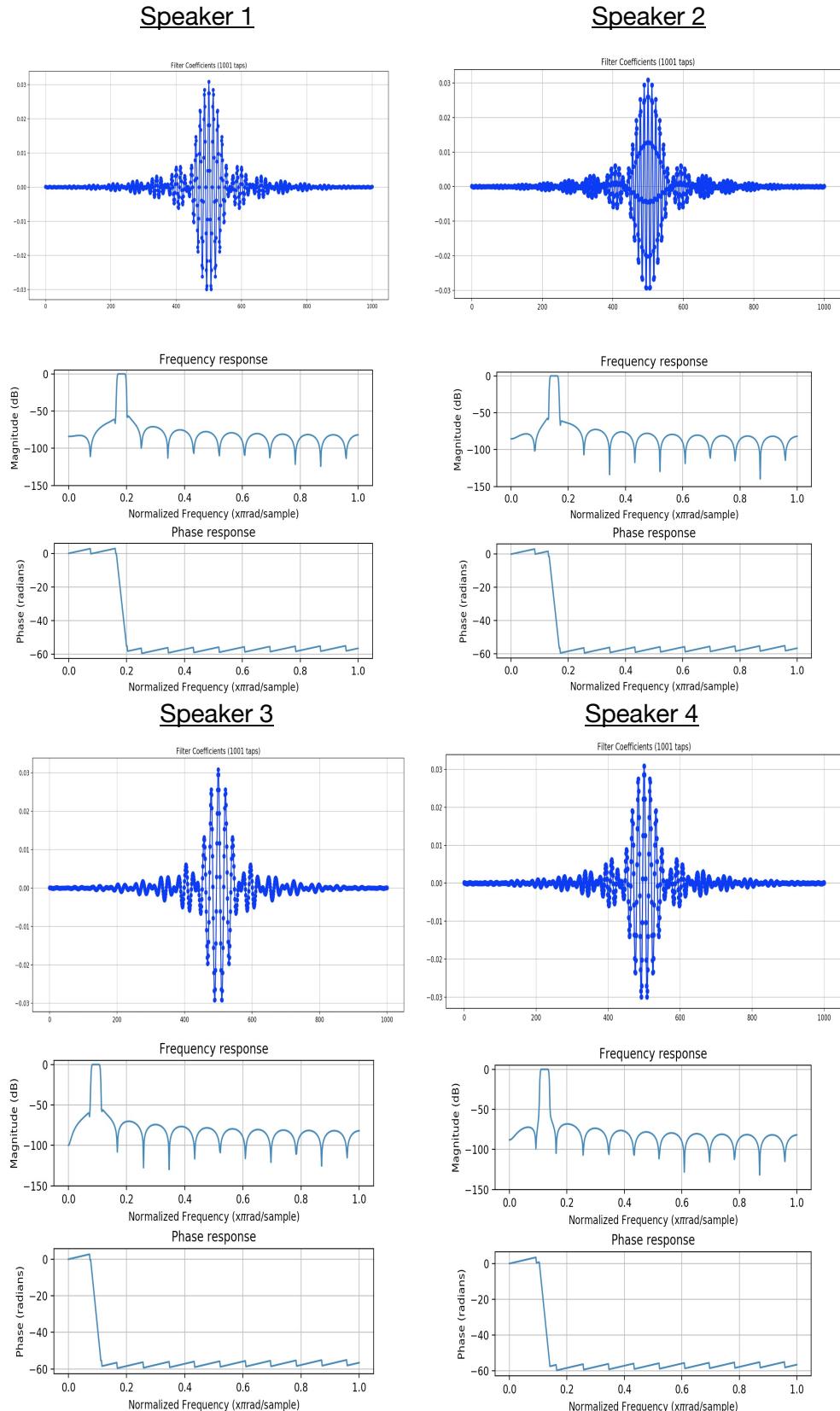


Figure 14 - All BPF used for each speaker

With the filtered signal we check the cross correlation between the expected chirp transmitted and the recorded filtered signal. With the result of the cross correlation, we aim to determine the ToA of the signal that was transmitted by the speaker.

We began by finding the maximum peaks to be able to determine the time stamps of the received signal. Afterwards, we fine tune the results in the specific time stamp:

- We find all the peaks above the noise threshold and calculate the median.
- Then we search for the first peak with an amplitude that is greater than the median calculated beforehand.
- Then we filter out certain peaks that have TDoA greater than the room's size in seconds.

Last, we define for each time period a Gaussian model in respect to ToA made TDoA samples:

$$[19] e^{-\left(\frac{(TDoA_{21}-\mu_1)^2}{\sigma_1^2} + \frac{(TDoA_{31}-\mu_2)^2}{\sigma_2^2} + \frac{(TDoA_{41}-\mu_3)^2}{\sigma_3^2}\right)}$$

After receiving the Gaussian model, we search for the μ (the expectation) and σ (the variance), then we define a criteria to determine an outlier:

$$[20] f(x) = \text{Euclidean Distance}(TDoA_{point}, \text{Gaussian})$$

$$[21] EUCD(x) = \sqrt{\frac{(TDoA_{21}-\mu_1)^2}{\sigma_1^2} + \frac{(TDoA_{31}-\mu_2)^2}{\sigma_2^2} + \frac{(TDoA_{41}-\mu_3)^2}{\sigma_3^2}}$$

And if,

$$[22] f(x) > 3 \Rightarrow \text{outlier} = \text{TRUE}$$

After all this is done, we get an output of ToA samples into a .csv file.

This .csv file is passed on to one of the two algorithms we implemented, LuT or Chan. The algorithm process is described in the following block diagram:

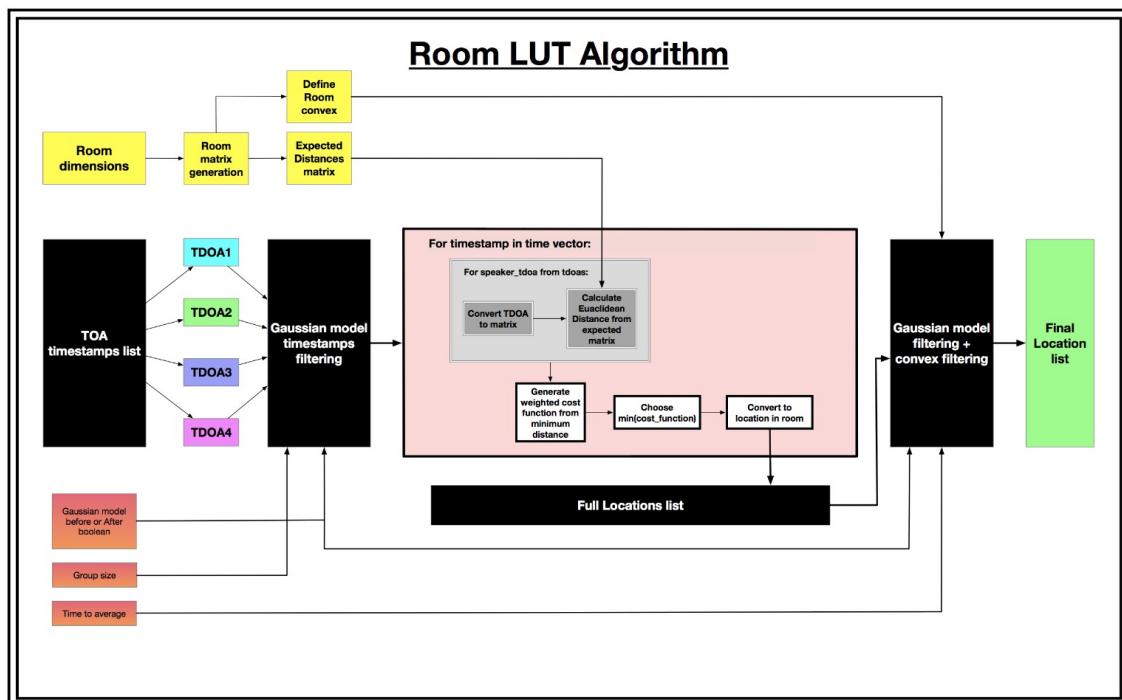


Figure 15 - LuT algorithm's block diagram

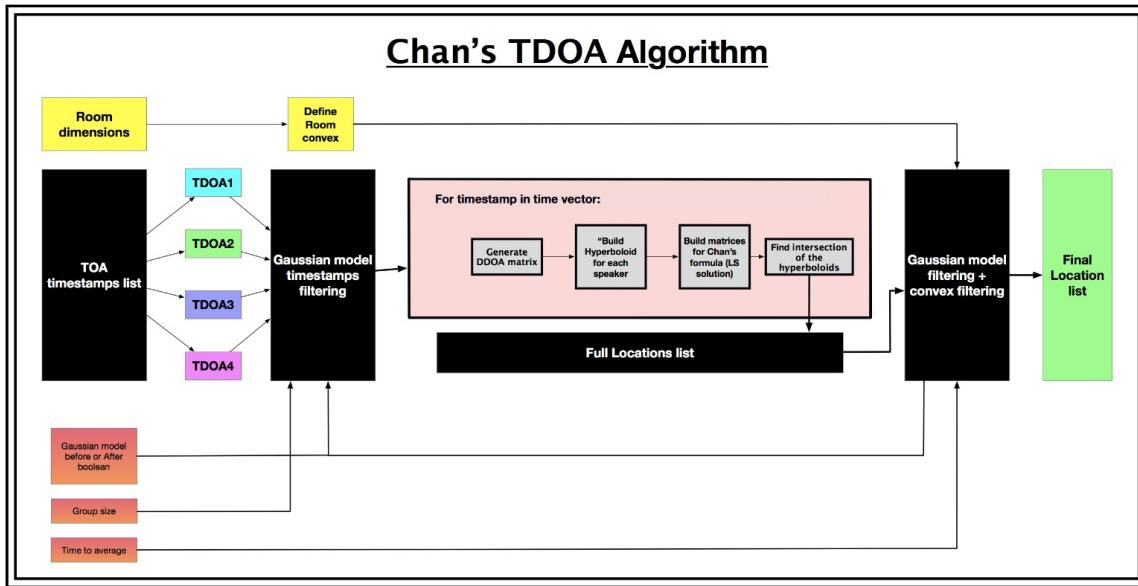


Figure 16 - Chan's algorithm's block diagram

- As you may notice the Gaussian model is shown before and after the algorithm's process, by default we did it before, but left the possibility in the code to implement it in afterwards also.

Both algorithm's process shown in the block diagrams above are implementation of what we described in the theoretical background.

Hardware Description

In our project we had limited amount of hardware used.

Ultrasonic Dynamic Speaker Vifa of Avisoft, with the following specs:

- Frequency range (+-12dB): $1 - 120\text{kHz}$
- Impedance: $4\ \Omega$
- Sensitivity at 50kHz : $92\text{ dB} / 2.83V / 1m$
- Max continuous power handling: 10 [W] (corresponds to a sine signal at 20 Vpp)
- Max output sound level in conjunction with Ultrasound Player (operated from 36V DC): $>98\text{ dB}$ at 1m distance

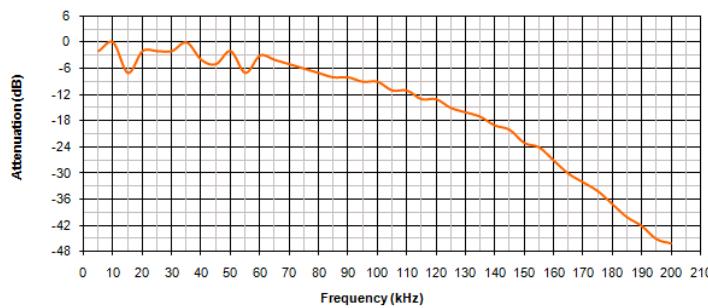


Figure 17 - Frequency response of the speakers

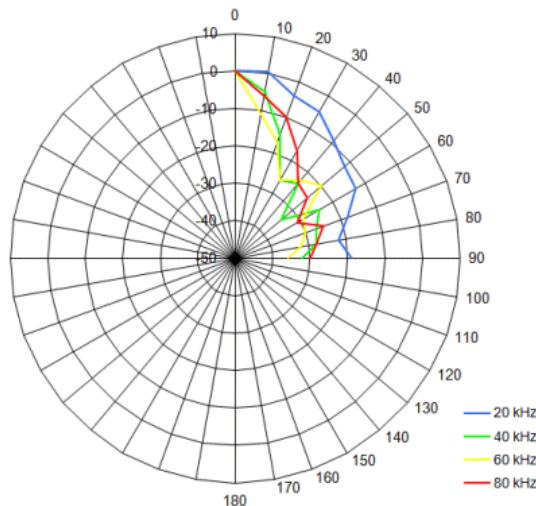


Figure 18 - Polar diagram of the speakers

UltraSoundGate CM16/CMPA model microphone of Avisoft, with the following specs:

- Microphone principle: external polarized condenser
- Frequency range: $2 - 250\text{kHz}$
- Relative flatness of the frequency response: good
- Approximate self-noise level, bandwidth: $30 - 50\text{kHz}$: 18 dB
- Reliability in high-humidity environments: good
- Power requirements: $5[\text{v}], 14\text{m}[A]$

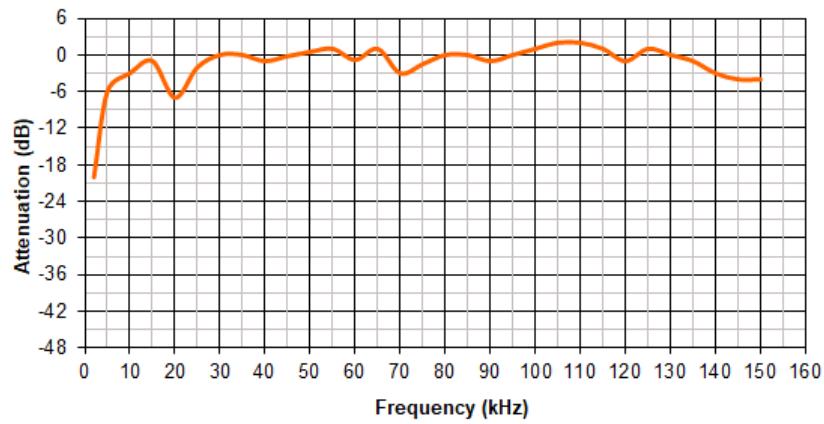


Figure 19 - Frequency response of the microphone

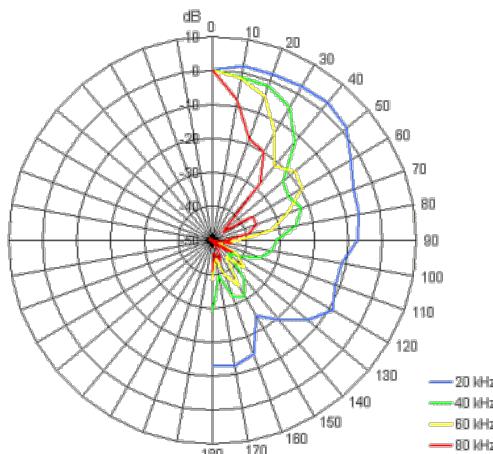


Figure 20 - Polar diagram of the microphone

NI 6343 X Series Data Acquisition, with the following specs:

We used only the analog output feature of this device, therefore the specs are mostly regarding the analog output.

- Sampling rate: $500k \left[\frac{S}{s} \right]$
 - 1 Channel analog output $900k \left[\frac{S}{s} \right]$
 - 2 Channel analog output $840k \left[\frac{S}{s} \right]$
 - 3 Channel analog output $775k \left[\frac{S}{s} \right]$
 - 4 Channel analog output $719k \left[\frac{S}{s} \right]$

- I/O: 32 analog inputs, 48 digital outputs and 4 analog outputs
- Timing resolution: $10n[s]$
- Timing accuracy: 50 ppm of sample rate
- Number of analog output channels: 4
- DAC resolution: 16 bits
- Output range $\pm 10[v]$
- Output impedance: $0.2[\Omega]$

Laser distance measure Bosch GLM 35, with the following specs:

We used a laser distance measuring device in order to be able to accurately calibrate the speaker's locations.

- Accuracy $\pm 1.5mm$

Software Description

For recording we used a built in PC application of Avisoft in order to record the signal.

This application communicates with the US microphone, and through it we were able to determine the sample rate of the recording, how to trigger the start of the recording, length of the recording and be able to see the spectrogram live.

Our written software can be divided into 4 different parts.

All the code is managed and saved in github server and need to ask for permission in order to be a developer.

Project git path:

<https://github.com/toledanoroi/TDOAchan>

MATLAB code:

Define speaker's location:

In order to calibrate the speaker's accurate location, we used a self-written code that calculates the cartesian location of a 3D point from the distances of 3 known points in space. We did this for each speaker and updated the speaker's locations in the Python code.

Transmitting unit:

To be able to use the NI DAQ certain drivers need to be installed in order for the PC to be able to connect.

The transmitting unit is operated through a MATLAB code using an NI DAQ package that is able to control the multiple output device. Through the code we can determine the different frequency range, signal type and envelope of each signal, and through which speaker the signal will output.

Transmitting unit's parameters:

Parameter	Description	Default Value
mode	Defines the mode operation of the script 0 = Welcome - transmitting welcome message through the speakers. 1 = Debug - each speaker says his number. Good to distinguish between the speakers. 2 = one by one - each speaker transmits his signal in a separate time. 3 = All together - the working mode of the system - analog signals. 4 = channel frequency response - generate from each speaker all band chirp in order to estimate the channel response. - Future use. 5 = digital signal - generating digital BFSK signal in all together mode.	3
Fs	Sampling rate of the signals	500,000
signal_time	The time of the signal in seconds. Need to be short in order to avoid reverberations.	0.002
window	The window type to envelop the transmitted signals: choose from: 'blackman_harris', 'hamming', 'linear_diagonal', 'nothing'	blackman_harris
signal_type	The signal type to be transmitted: choose one from the following: 'chirp', 'radar_chirp'	'chirp'
welcome_msg	Message to be transmitted by text to voice function in mode 0	Hi everyone I am the Transmitting Unit in the localization system of Roy Toledano and Yarden Avraham
loops	The number of loops that the DAQ will transmit the signals. Transmitting time =loops *(0.5 + pause time)/3600 [hours]	1000000
pause_time	The pause time between loops	0.001
freqs_mat	The frequency band for each speaker	[34000 41000; 41000 49000; 20000 27000; 27000 34000];
digital_distance	If mode 5 is used: the frequency distance of specific bit modulated sine wave from the carrier frequency. [Hz]	3000

gain	The gain to be added to the signals before transmitting. Max gain allowed = 6.65	5
-------------	---	---

Table 4 - Transmitting unit's parameters

Python code:

ToA Detector (TOAD):

This part includes the processing of the raw record signals into a signal that we could generate from them the Time of arrivals samples from each speaker.

This part was implemented in python 2.7 environment and uses the following site packages: Numpy, scipy, pandas, matplotlib, termcolor, time, wave, csv, os and sys.

The TOAD can run as standalone or as full system with all Receiving data software structure.

Chan's Algorithm:

One of the two algorithms that suggested to solve the problem.

A mathematic solution – lower immunity to measurement error. Needs more development for future research.

This part was implemented in python 2.7 environment and uses the following site packages: Numpy, scipy, itertools, collections, termcolor, time and os.

The Chan's algorithm can run as standalone or as full system with all Receiving data software structure.

LuT Algorithm:

One of the two algorithms that suggested to solve the problem.

A computational algorithm using expected TDOA results for each point in the room and find the closest location from the LuT.

This part was implemented in python 2.7 environment and uses the following site packages: Numpy, scipy, itertools, collections, termcolor, time and os.

The LuT algorithm can run as standalone or as full system with all Receiving data software structure.

All python's parameters:

parameter	description	default value
mode	Define the mode operation of the script 1 = Only generate TOAs from folder 2 = Run multiple records (WAV file) full - wave parsing and algorithm 3 = Use ToA csvs to run only the algorithm 4 = One shot - take one .wav file and run it full -> wave parsing and algorithm 5 = Plot results from results file. 6 = Run the localization system without expected value for multiple wav files in a folder.	6
constant_z	Changes the algorithm from 3 dimensions to 2 dimensions given a known height, (gives a better calculation) works only with Room LUT algorithm.	2.3
TOA_path	If the ToA samples already exists: path of the relevant TOA samples. Else = type: string, should be a real path of csv file	/005.m'
matlab_path	Path of the expected signals that we generated from the transmitting unit, the MATLAB code. type: string, should be a real path of mat file with variable 'allchirp'	~/inputs/digi_bh_2m.mat'
record_path	Path of the recorded signal to examine. Type: string, should be a real path of .WAV file	~/inputs/digital_bh/a0000001.wav'
speakers_frequencies	The recorded chirp frequencies that we defined in the Transmitting unit. structure: {'1': [start_freq, stop_freq], '2': [start_freq, stop_freq], ...} type: dictionary	{'1':[34000,41000], '2': [42000, 49000], '3': [20000, 27000], '4': [27000, 34000]}

speakers_locations_d	Locations dictionary for all the speakers structure: {'1': [x, y, z], '2': [x, y, z], ...} type: dictionary of int lists	{'1': [0.2973, 3.6627, 0.504], '2': [1.8253, 0.4246, 0.589], '3': [1.7077, 2.8073, 0.9979], '4': [0.1039, 0.2065, 2.21]}
room_sizes	Dictionary of the room limits with every axis structure: {'x': <maxX>, 'y': <maxY>, 'z': <maxZ>} type: dictionary	{'x': 1.966, 'y': 4.272, 'z': 3.051}
res_iteration	True if you want to examine the matrix resolution feature, else define a specific resolution type: boolean	FALSE
only_toa	Define if to run the Rxmain with algorithm calculation or only TOA measurements True -> only TOA measurements. False -> with algorithm calculation. Type: boolean	FALSE
chirp_time	The duration of the chirp signal generated by MATLAB in seconds	0.002
filter_size	The number of taps in the speakers Band Pass Filters (BPF) type: int , must be odd number for real BPF (FIR design)	1001
signal_mode	Define which reference signal to generate according to the transmitting unit. Future use.	1
expected_points	An array of expected (x,y,z) defined by human measurements.	[np.array([-1,-1,-1])]
algorithm	Which algorithm to run in the test, choose from the algorithms dictionary: algorithm_d = {'chan': 1, 'taylor': 2, 'room': 3, 'both': 4}	algorithm_d['room']
resolution	The LUT room matrix algorithm resolution	0.02
use_averaging_before_calculation	Define where to search for outliers and average results, before calculate distance or after. Type: boolean, True -> before, False -> after	TRUE
time_factor	If decided averaging after algorithm calculation, this parameter defines	2

	the time period to averaging and to throw outliers.	
avg_group_size	if decide before algorithm calculation, this parameter defines how much samples to average and throw outliers on each iteration.	5
room3D	array 3D of all edge's points in the room	room square
triangle3D	If there are edges that don't participate as edge in convex hull, define an 3D array of these points and there's neighbors	None
point_name	When run's multiple points this parameter gives number name for the point in order to distinguish between algorithm results of different plots when plotting	../output/TOA_ + str(int(time.time())) + '.csv'
frecords	Path of file that defines the path for multiple records (WAV file) to run in mode 2 --> ToA detector and algorithm calculation	~/toa_to_save/records.csv'
ftoas	Path of csv file that defines multiple ToAs file path and expected (csv file) to run in mode 3 --> run algorithm calculation only from known ToA samples file.	~/toa_to_save/toacsvs_digital.csv'
frecbase	Path of folder with multiple records (WAV file) to run in mode 1 -->ToA generator for each record file.	~/inputs/digital_bh'

Table 5 - All python's parameters

5 Results Analysis

We separated the results into 2 parts:

- Records post processing – ToA Detector and records Quality
- Algorithms performance

5.1 Records post processing – ToA Detector

Each record we loaded to the TOAD passed the following process:

- Load signal:

Using a chirp signal with a Blackman-Harris window in order to insure minimum cross-correlation between speakers and to avoid any ‘clicks’ generated.

The following signal was detected from the US microphone.

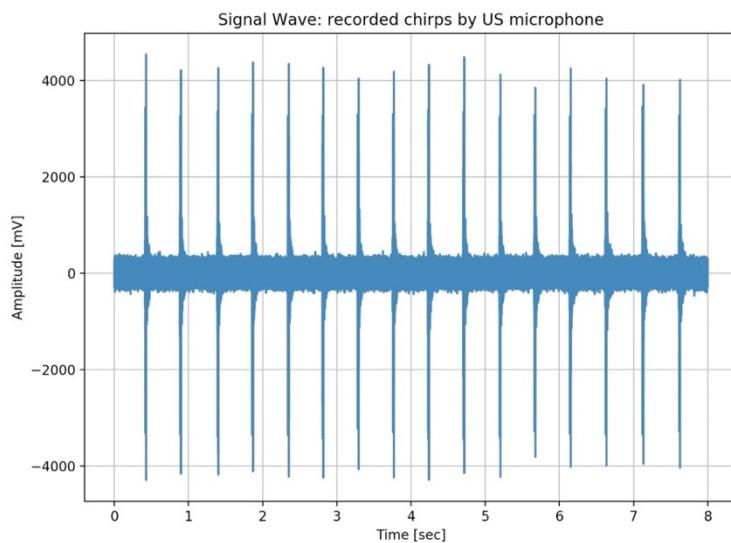


Figure 21 - An 8 second record from the US microphone

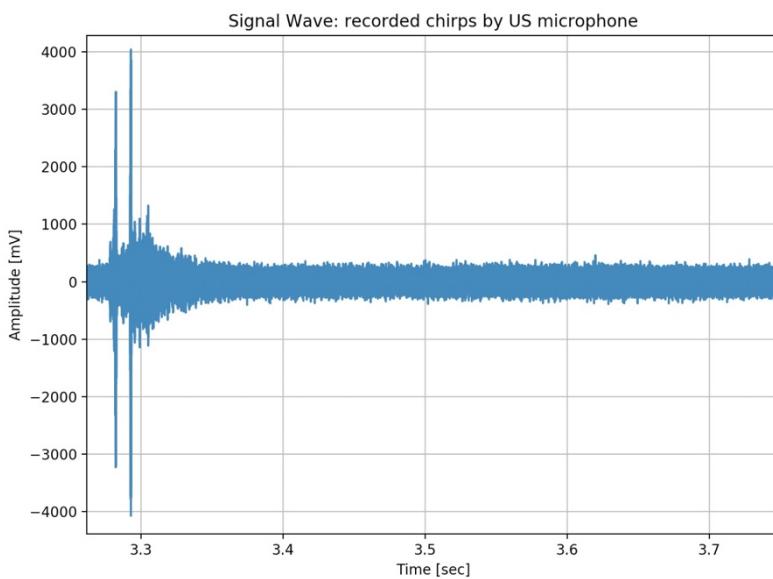


Figure 22 - A closeup to a single transmission record

- FFT and Spectrogram:

As we can see from the FFT and the spectrogram, the higher the frequency the lower the energy. Moreover, we had a continuous interference at ~50 kHz that limits the frequency bandwidth we worked with. The interference came from the DAQ.

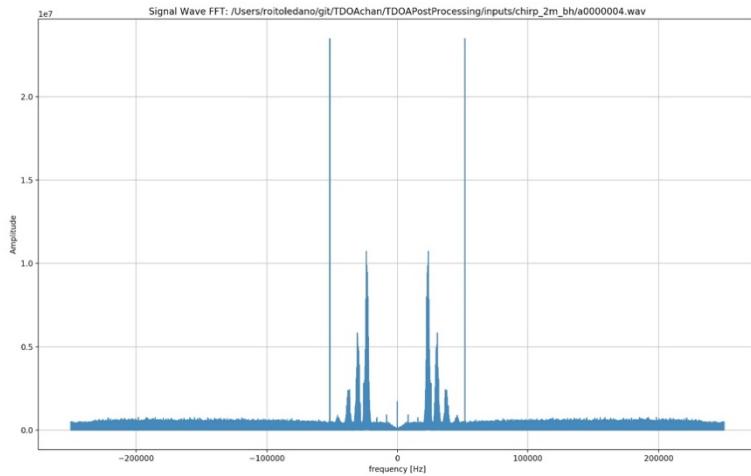


Figure 23 - FFT of the recorded signal before being filtered

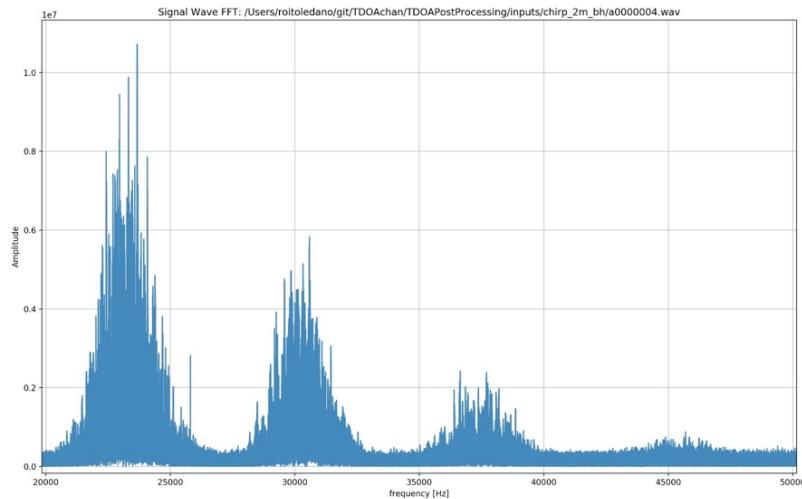


Figure 24 - A closeup of the FFT of the recorded signal before being filtered

From the spectrograms below, we can clearly see the transmitted chirps, their arrival time and their reverberation. We can see that as we thought, the highest power from a specific speaker isn't always at the first arrival of that recorded signal. This can be explained by the wave's reverberation superposition that are in the same phase and generate a greater wave amplitude.

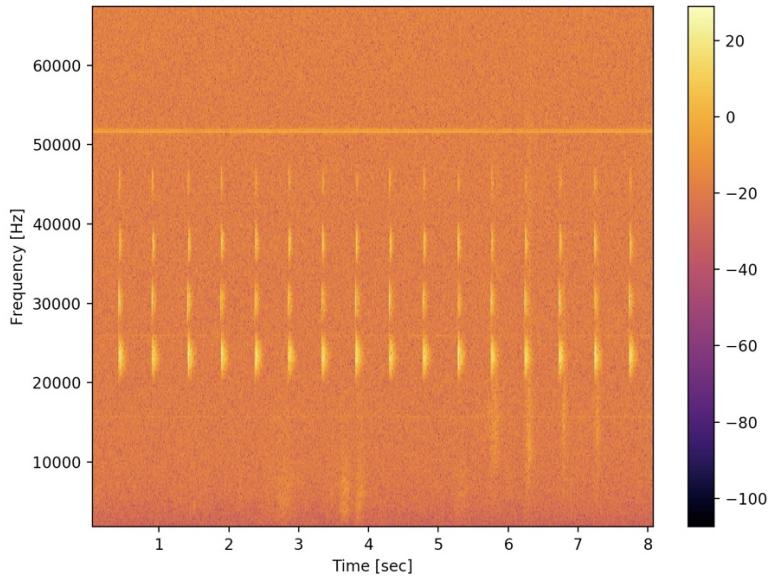


Figure 25 - Spectrogram of recorded signal

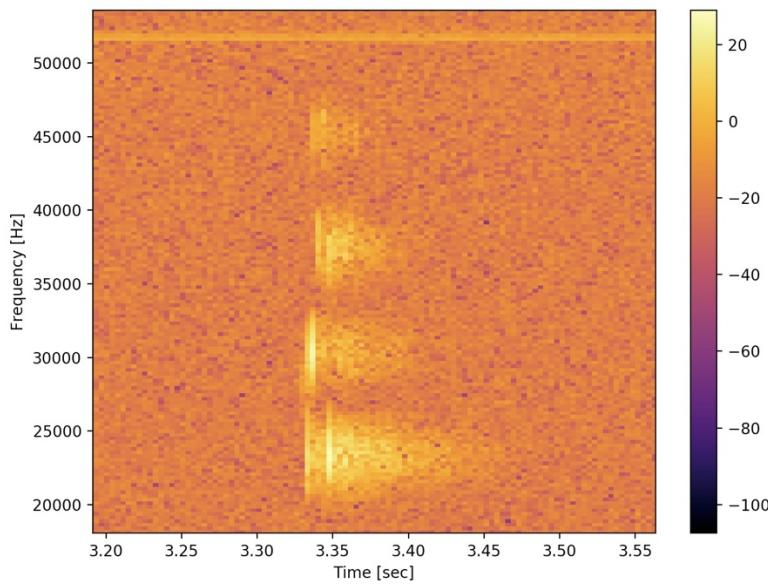


Figure 26 - Closeup of the spectrogram of recorded signal

- Filter the signal with speakers BPF:

Passing the recorded signal through a BPF as explained beforehand.

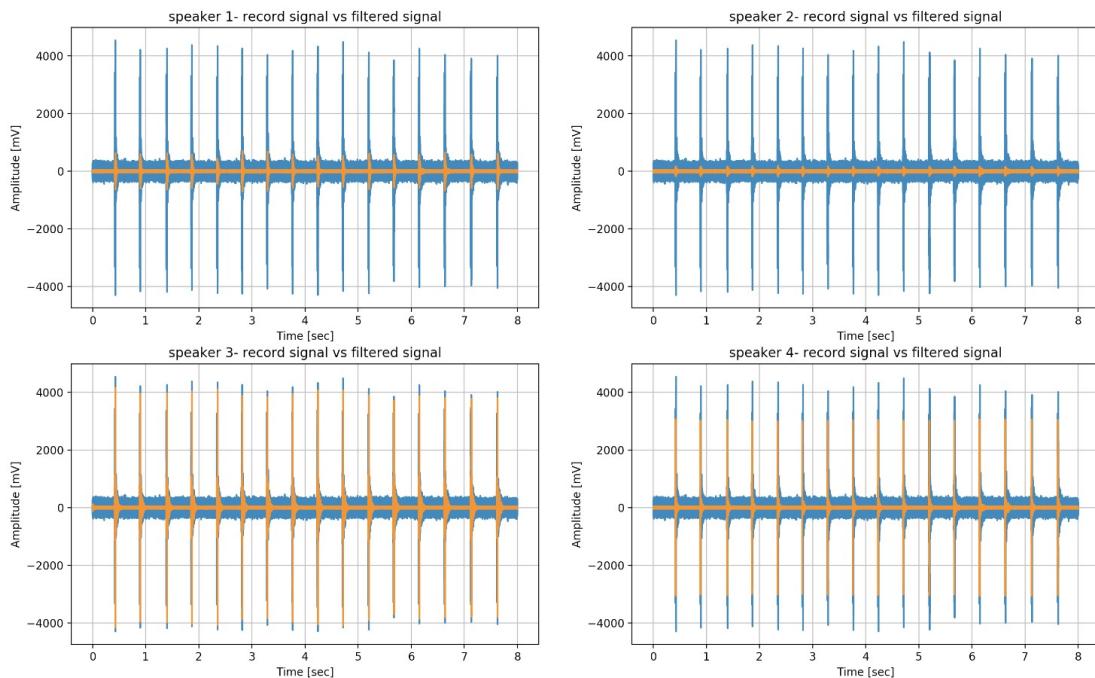


Figure 27 - The recorded signal with the filtered recorded signal for each speaker

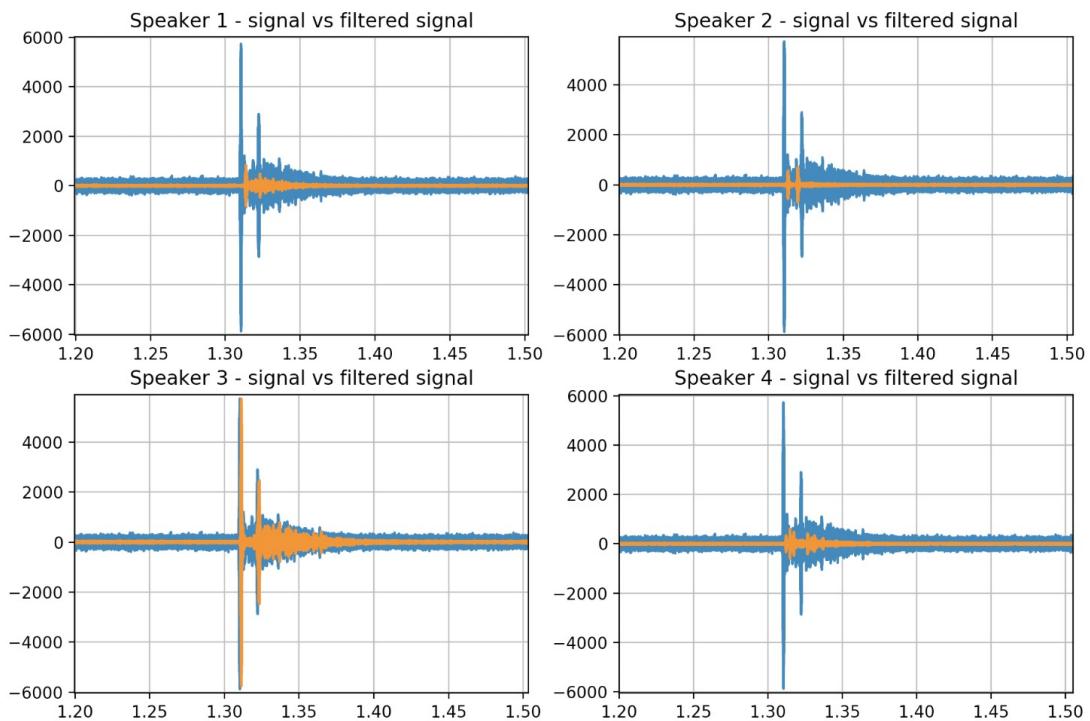


Figure 28 - Closeup of the recorded signal with the filtered recorded signal for each speaker

- Time correlation

Outputting the time correlation of the filtered recorded signal.

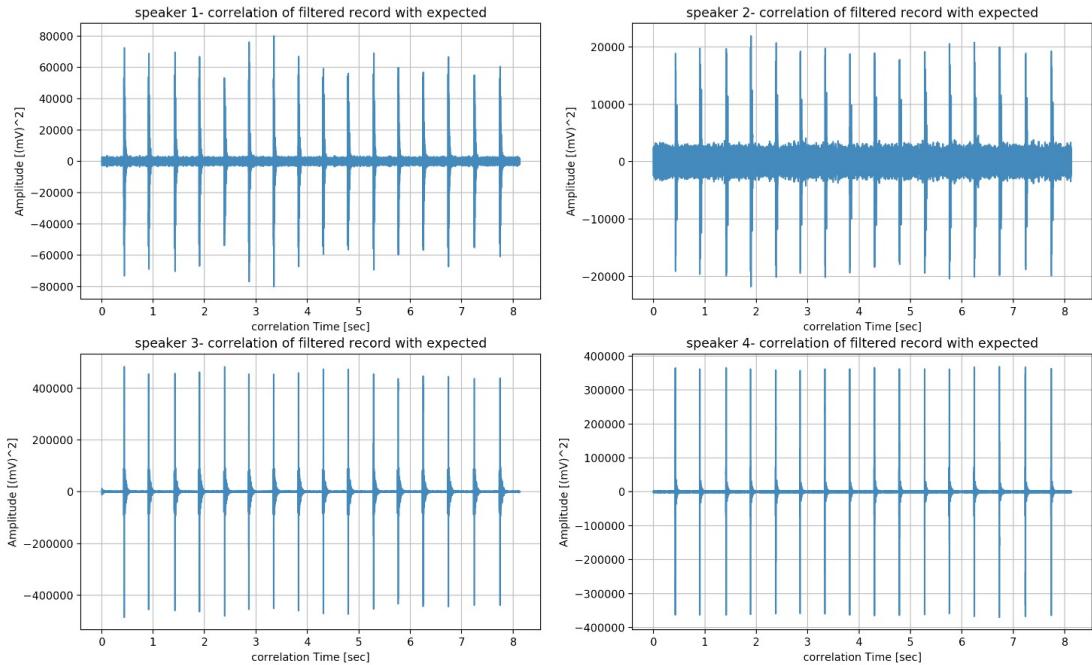


Figure 29 - Time correlation of each speakers filtered recorded signal

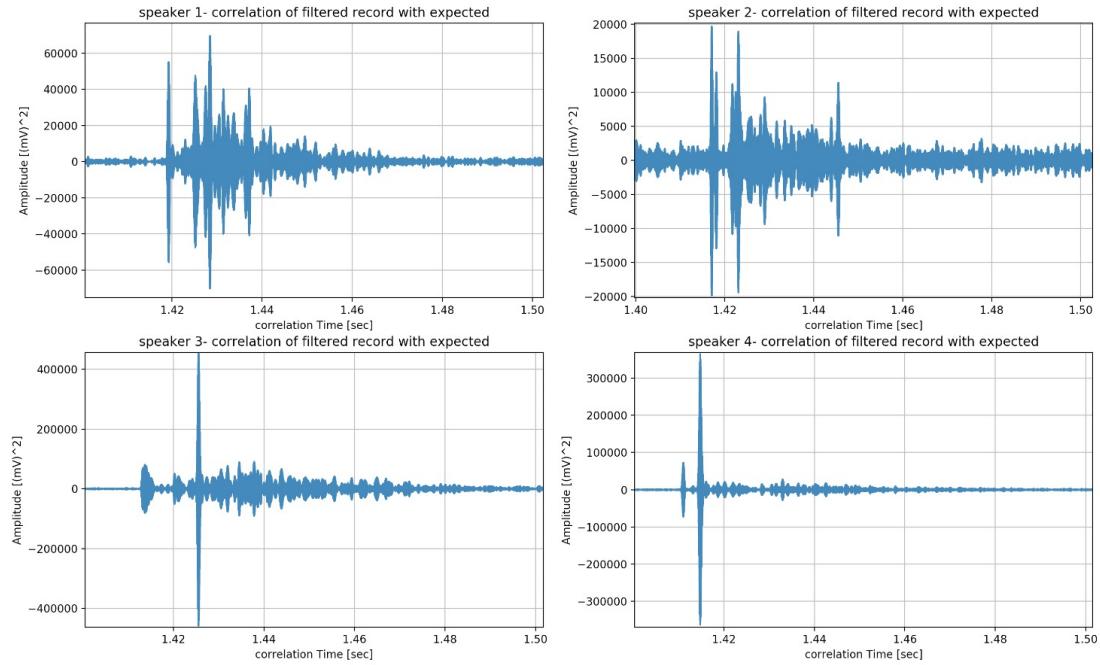


Figure 30 - Closeup of time correlation of each speakers filtered recorded signal

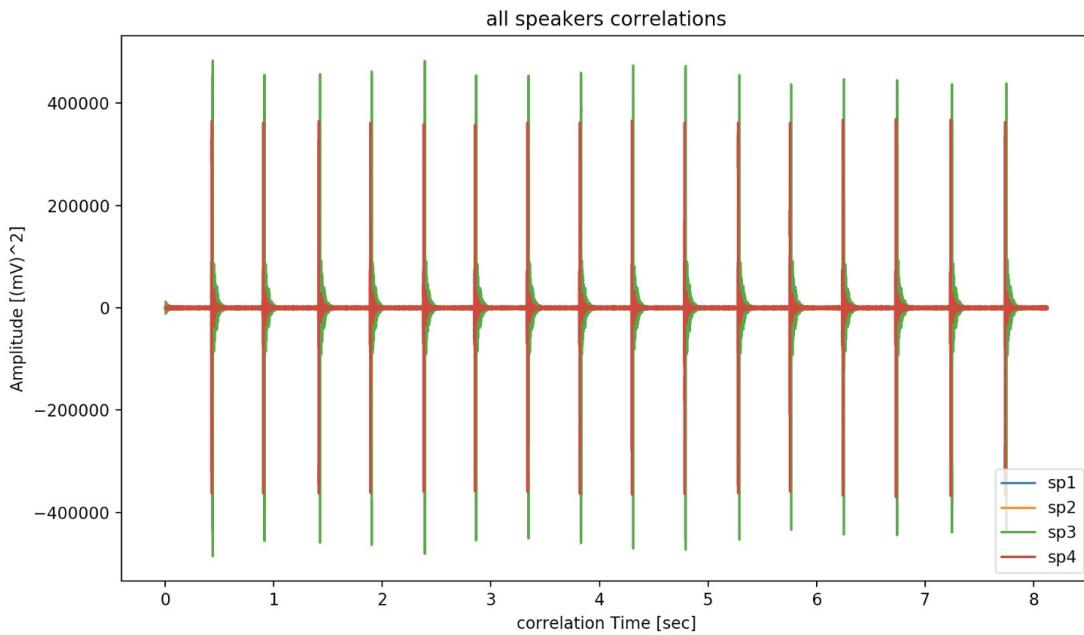


Figure 31 - Time correlation of all the speaker's filtered recording together

- Peaks detection and Fine Tuning to the first significant peak for each time stamp.

Using the time correlation of all the speakers, we detect the peaks in order to determine the ToA of the speakers, for each time stamp.

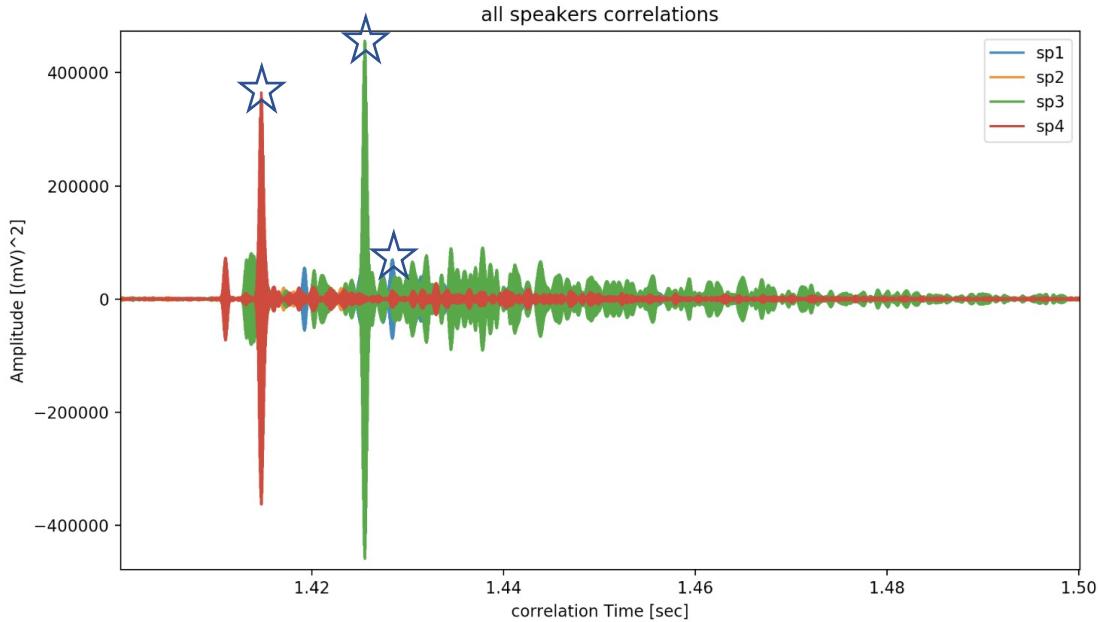


Figure 32 - Closeup of the time correlation of all speakers together, with the peaks

After fine tuning the detection of the first significant peak, we are able to get the following results.

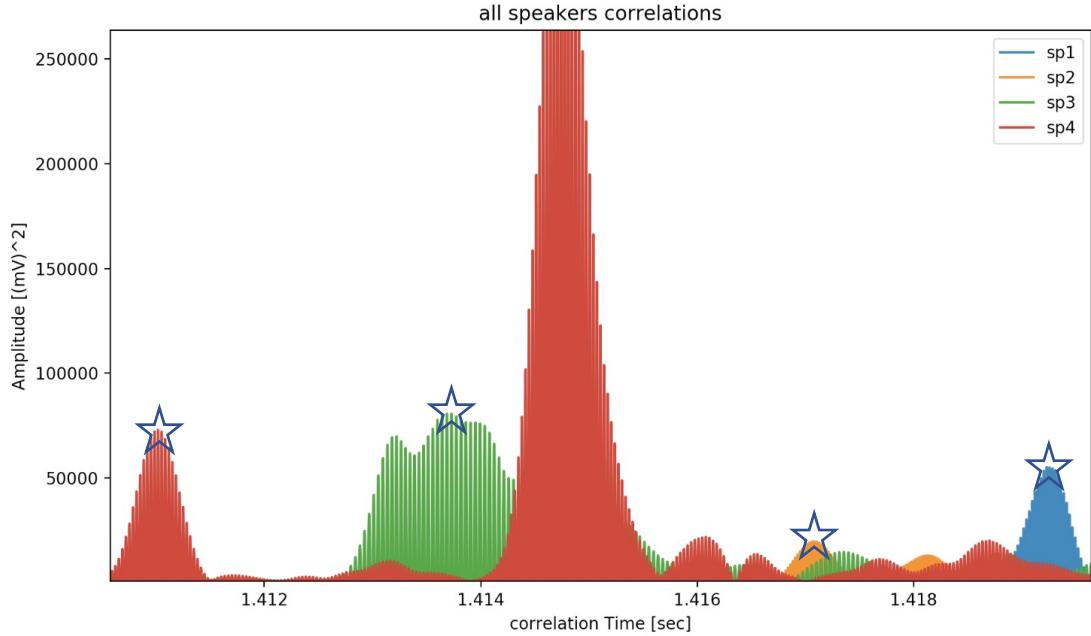


Figure 33 - Closeup of the time correlation of all speakers together, with the fine-tuned peaks

With all the fine-tuned peaks we get, we determine the ToA of all the speaker in each time stamp. With these ToA we plot all the TDoA, and eliminate the outlier points we receive.

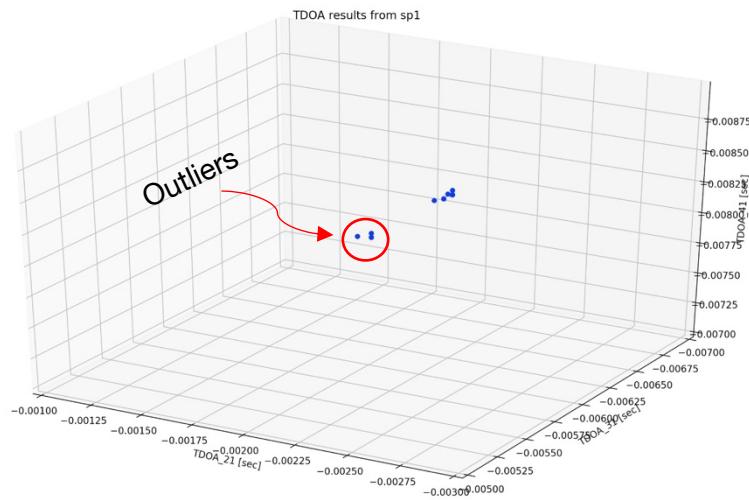


Figure 34 - TDoA results, with the outliers which we eliminated from the results

5.2 Algorithms performance

- LuT algorithm

We started by determining the height of the microphone in which the recording was made, in order to have a more accurate (x, y) position result.

The results are as the following:

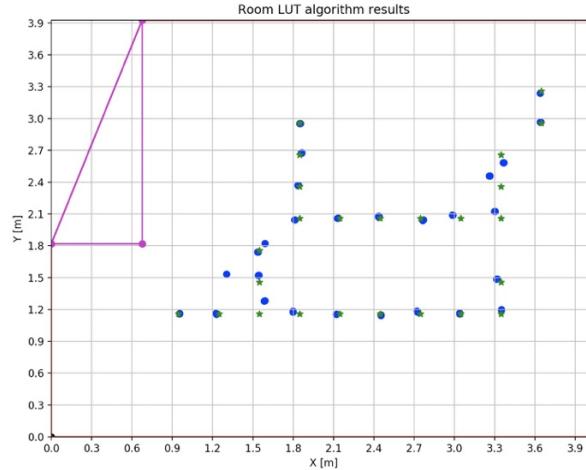


Figure 35 - Positioning results vs expected, in 3D with a constant z parameter, for LuT

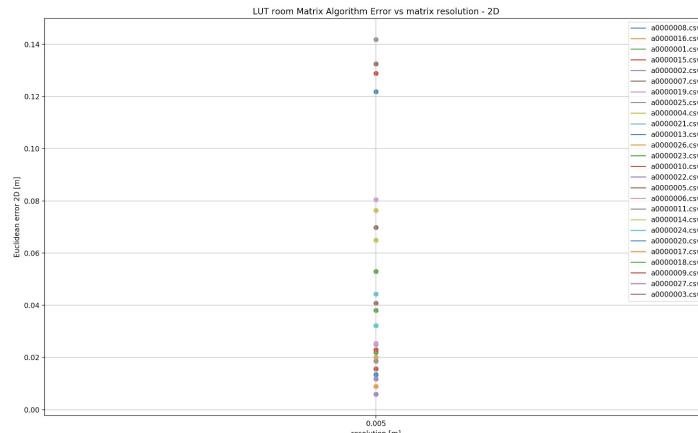


Figure 36 - Euclidean error from expected for each point, in 3D with a constant z parameter, for LuT

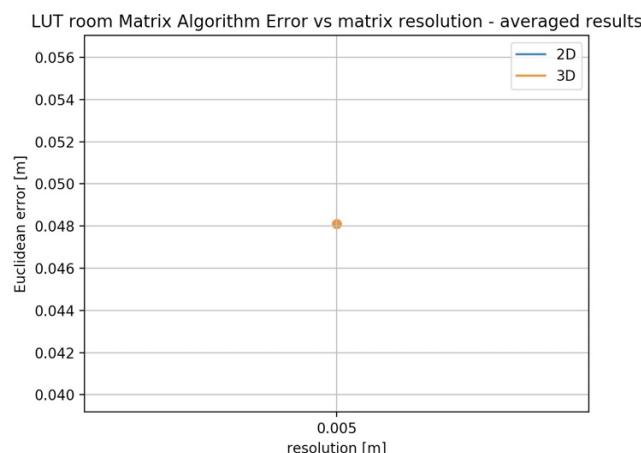


Figure 37 - Average Euclidean error from expected, in 3D with a constant z parameter, for LuT

Next, we didn't determine the microphone's known height at all. This is our 3D test.

The results are as the following:

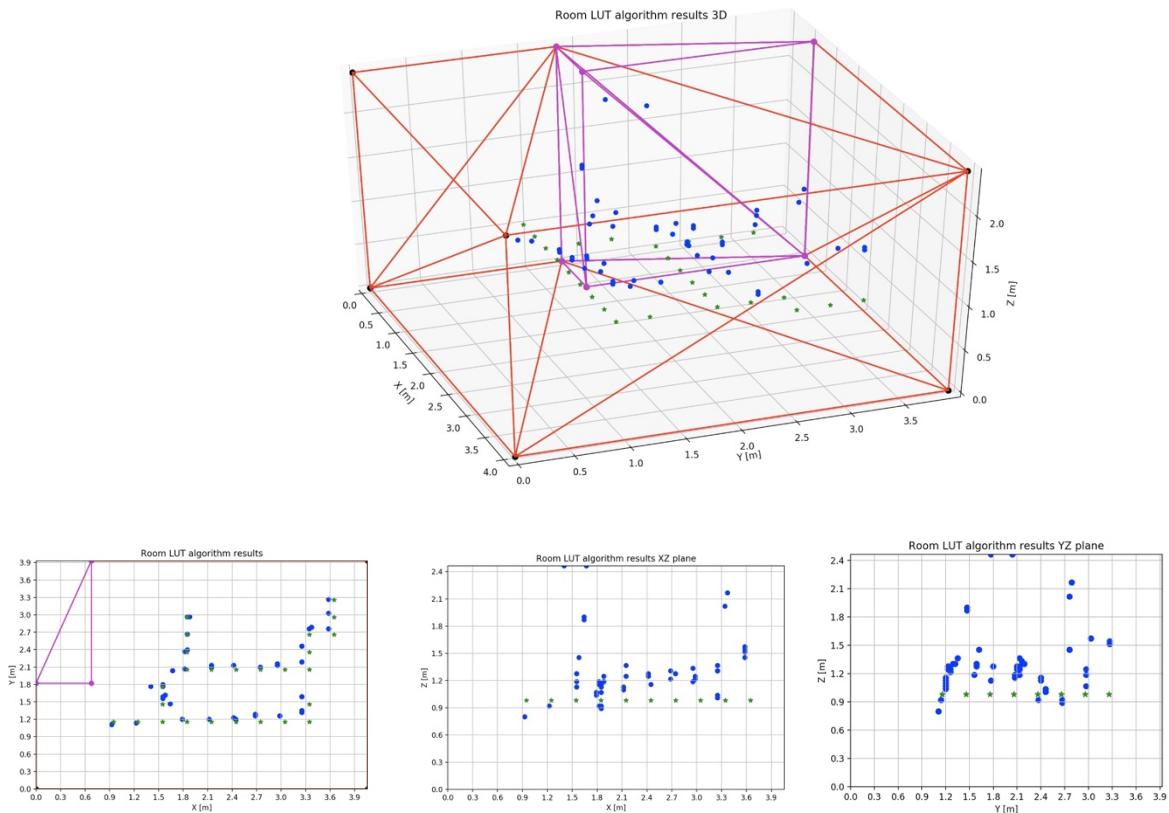


Figure 38 - Positioning results vs expected, in 3D, for LuT

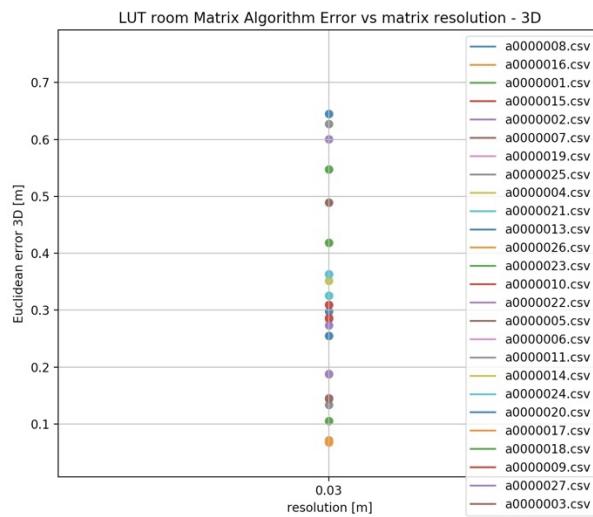


Figure 39 - Euclidean error from expected for each point, in 3D, for LuT

Finally, we didn't determine the microphone's known height, but the error results were done only by taking into consideration the (x, y) axis. This was our way of testing in 2D.

The results are as the following:

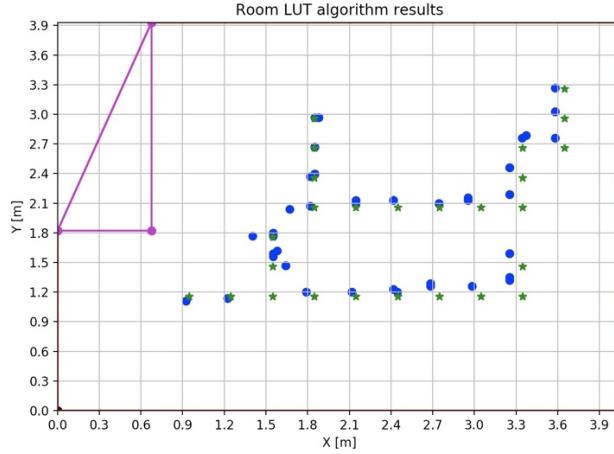


Figure 40 - Positioning results vs expected, in 2D, for LuT

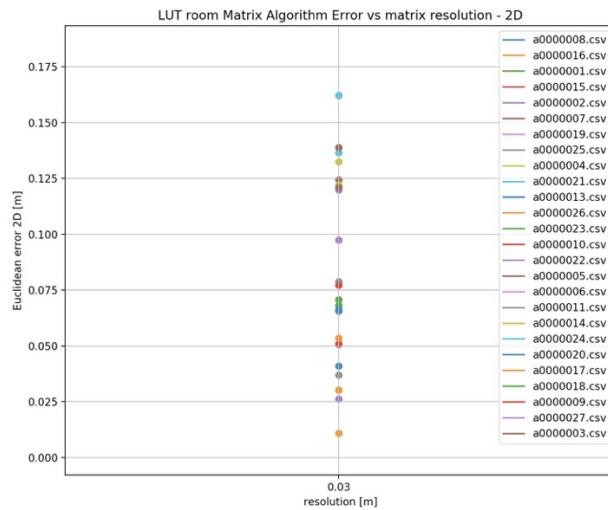


Figure 41 - Euclidean error from expected for each point, in 2D, for LuT

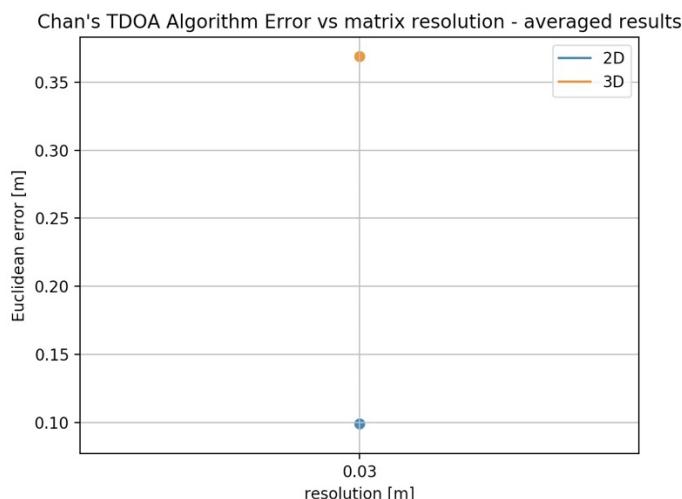


Figure 42 - Average Euclidean error from expected in 3D vs 2D, for LuT

- Chan's algorithm

We started by determining the height of the microphone in which the recording was made, in order to have a more accurate (x, y) position result.

The result was as the following:

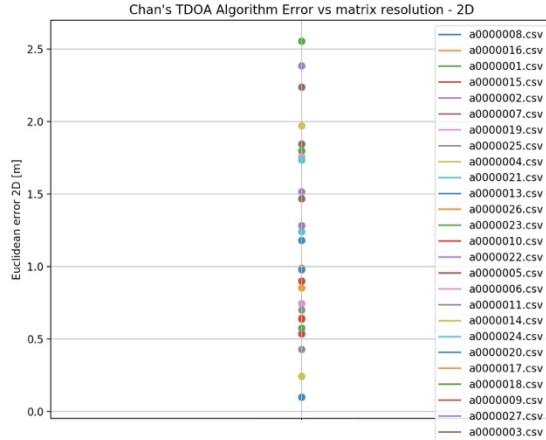


Figure 43 - Euclidean error from expected for each point, in 2D with Chan's algorithm

Next, we didn't determine the microphone's known height at all. This is our 3D test.

The result was as the following:

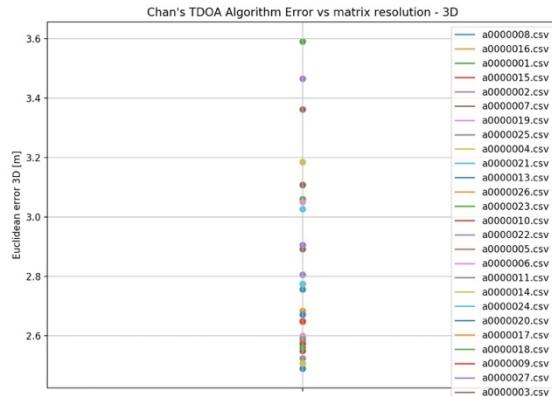


Figure 44 -0 Euclidean error from expected for each point, in 3D with Chan's algorithm

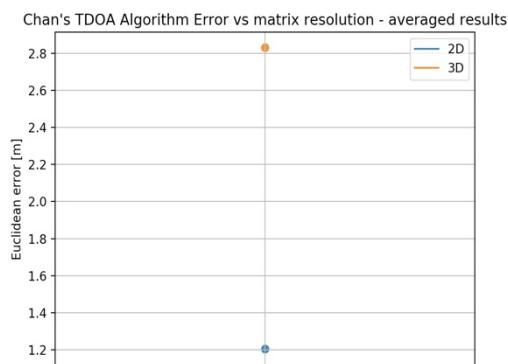


Figure 45 - Average Euclidean error from expected with Chan's algorithm, in 3D vs 2D

5.3 Summary of the test and simulation results:

The following table summarizes all the results:

Test results comparison with Simulation		
	Algorithm	Average Error [m]
LuT	2D (3D with constant Z)	0.048
	3D	0.375
	2D (without constant Z)	0.1
	simulation	2.45E-09
Chan	2D	1.2
	3D	2.81
	Simulation	2.23E-05

Table 6 - Test results comparison with Simulation

5.4 Digital signal with LuT algorithm

We can easily see from the results above that the LuT algorithm yields better positioning results. We decided to check only the LuT algorithm with a digital signal instead of the chirp.

As we saw from the simulation above with the digital signal, the correlation acts different in comparison to the chirp. Therefore, we anticipated the results wouldn't be as good.

The final results are as the following:

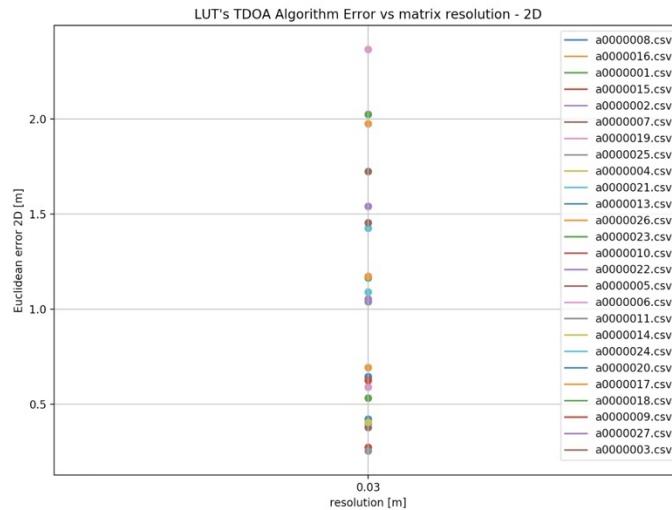


Figure 46 - Positioning results vs expected, in 2D with BFSK, for LuT

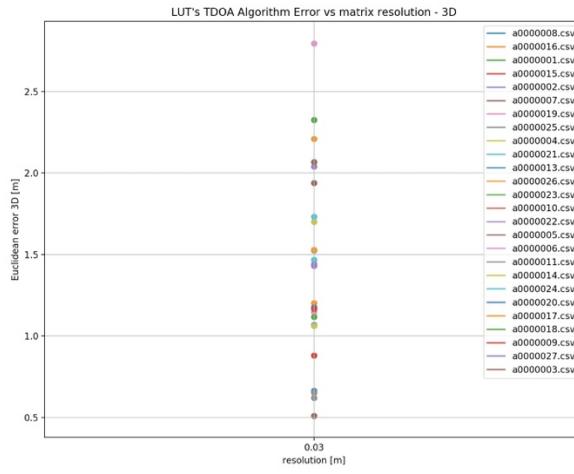


Figure 47 - Positioning results vs expected, in 3D with BFSK, for LuT

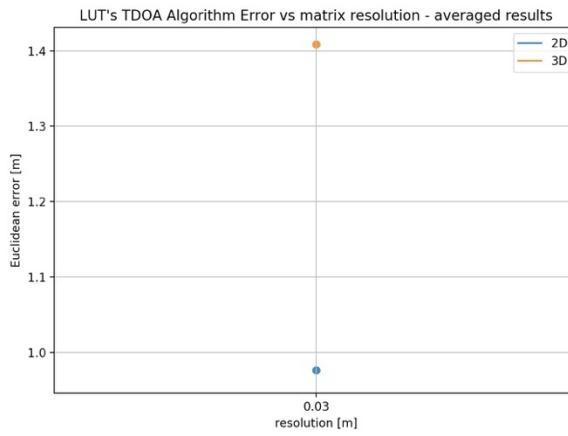


Figure 48 - Average Euclidean error from expected with BFSK, in 3D vs 2D, for LuT

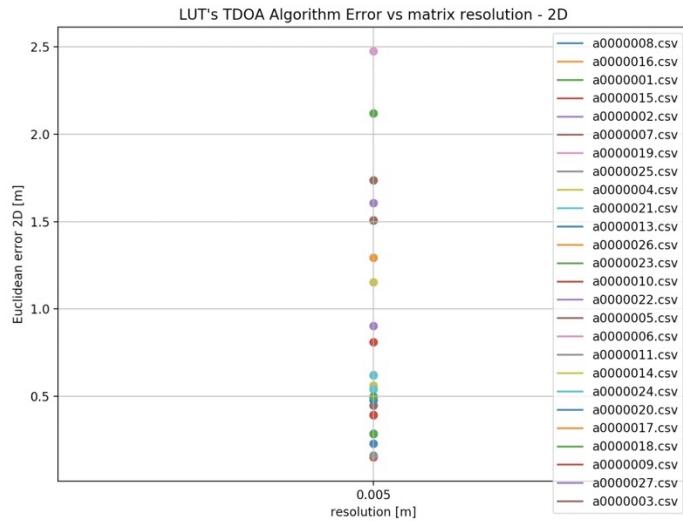


Figure 49 - Euclidean error from expected for each point with a constant Z input, with BFSK for LuT

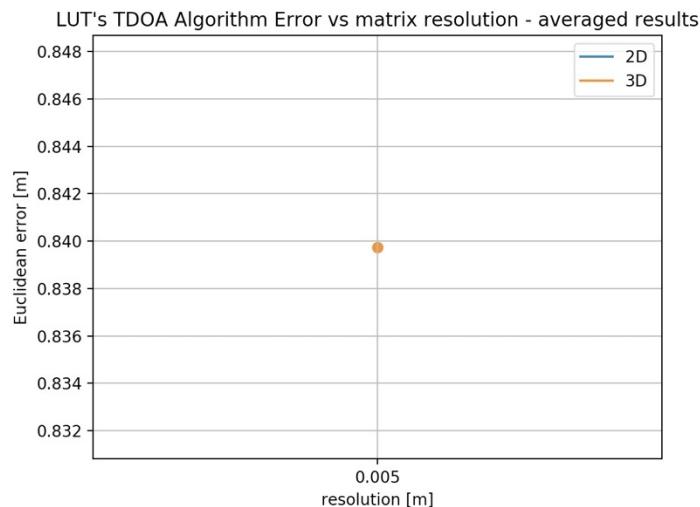


Figure 50 - Average Euclidean error from expected with a constant Z, with BFSK for LuT

Test results with digital signal		
	Algorithm	Average Error [m]
LuT	2D (3D with constant Z)	0.84
	3D	1.41
	2D (without constant Z)	0.98

Table 7 - Test results with digital signal

6 Conclusions and suggestions for future research:

- Examining the results of the project against the goals defined:

From the comparison of the positioning algorithms, our best results were created by the Room LuT algorithm.

With this algorithm we were able to achieve a 30 [cm] accuracy in 3D and a 4.8 [cm] accuracy in 2D. Because we know that most of the time the bats are located at the ceiling's area, the 2D option is relevant and with a 4.8 [cm] accuracy we can detect and distinguish between different bats in their colony.

In addition, we can see from the LuT results that even in the 3D mode without a constant z when only taking the (x, y) axis plane we were able to receive a good accuracy result, 10 [cm].

- Suggestions for improving system performance:

While implementing our project we noticed certain difficulties in performance.

- We noticed that the method in which we determine the speaker's location has too many human measurements errors which can lead to a bigger error in the positioning point.

To be able to bypass these human errors, it is necessary to build a static structure with known dimensions in order to be able to calibrate the system better.

- We saw that with each different transmitted signal the way we determine the ToA of each speaker could change drastically. For example, when implanting our code with the digital BFSK signal, we noticed that the peak detection from which we find the ToAs is wrong. Therefore, we got bigger errors with the digital signal compared to the chirp.

We think that our system has good performance while using an up-chirp signal, but for future research and to be able to have a more dynamic system, it is needed to create a more robust peak detection method, or for each signal a different peak detection method.

In addition, it is possible to build a synchronized system between the speakers and the recorder, that way the ToA will be detectable in real time without taking the peaks into consideration.

- The way the speakers are arranged in the room defines an ambiguity function.

The ambiguity function defines the Positional Dilution of Precision (PDOP) value, which can conceptually be defined as:

$$[23] \quad PDOP = \frac{\Delta(Position)}{\Delta(Measurement)}$$

The PDOP value essentially describes the sensitivity of the position estimate for errors in the measurement value. A more general measure is the Geometric Dilution of Precision (GDOP), also simply referred to as the Dilution of Precision (DOP). GDOP expresses how a change in the measured value would affect the unknown variables in the propagation model. In the case of GPS, which uses

TOA positioning, the unknown variables are the coordinates of the mobile unit and the clock bias

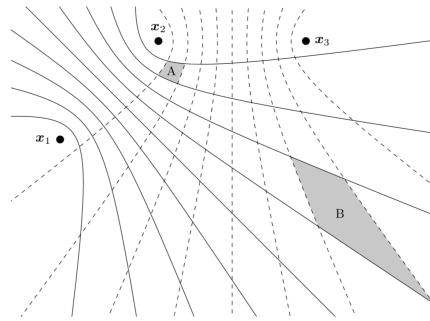


Figure 51 - Illustration of PDOP for TDoA system. The PDOP at B is greater than at A.

In order to overcome this problem, further research is needed to be able to the ambiguity function of the room we are testing.

- In our project we didn't take into consideration the change in Voice Speed as a function of the medium's temperature and air pressure. Even the smallest change in the Voice Speed can result in large errors in the positioning.

To overcome this issue, it is necessary to obtain a temperature measurement device and to add this variable to the code.

- Options for future work (development / research)

- It is needed to further research Chan's algorithm in order to avoid as much as possible any divergence and interferences. In addition, continuing looking into ways to improve Chan's algorithm or find an alternative algorithm.
- Another aspect for future research is the influence the Z axis has on the result's accuracy.
- Further investigate different signals and envelops to transmit in order to increase accuracy.
- Continue to develop and improve the digital signal to be usable for future work.
- Improving the running time of the code, to be able to implement our project in a real-time system.
- Create an easy user interface so the code will be accessible to any person that wishes to work with our system.

References:

Articles:

- [1] P. S. Y. Yovel, M. O. Franz, "Complex echo classification by echo-locating bats: a review," *J. Comp. Physiol.*, vol. 197, pp. 475–490, 2011
- [2] B. Xu, G. Sun, R. Yu, and Z. Yang, "High-accuracy TDOA-based localization without time synchronization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 8, pp. 1567–1576, 2013.
- [3] Y. T. Chm, S. Member, and K. C. Ho, "A Simple and Efficient Estimator for Hyperbolic Location," vol. 42, no. 8, pp. 1905–1915, 1994.
- [4] O. Wijk and H. I. Christensen, "Triangulation-based fusion of sonar data with application in robot pose tracking," *Ieee Trans. Robot. Autom.*, vol. 16, no. 6, pp. 740–752, 2000.
- [5] H. Balakrishnan, R. Baliga, D. Curtis, M. Goraczko, A. Miu, N. B. Priyantha, A. Smith, K. Steele, S. Teller, and K. Wang, "Lessons from developing and deploying the cricket indoor location system," *Preprint*, 2003.
- [6] Hexamite Ltd., "Hx19 flye," 2012.
- [7] S. Arrays, "A Python Implementation of Chan's TDoA algorithm for Ultrasonic Positioning and Tracking," pp. 1–31, 2008.
- [8] H. I. Ahmed, P. Wei, I. Memon, Y. Du, and W. Xie, "Estimation of Time Difference of Arrival (TDOA) for the Source Radiates BPSK Signal," *IJCSI Int. J. Computer Sci.*, vol. 10, no. 3, pp. 164–171, 2013.
- [9] Y. Y. G.Eitan, "Automatic Calibration of Micrphone Array," 2015.
- [10] H. J. Helgert and T. George, "An Improved Chan-Ho Location Algorithm for TDOA Subscriber Position Estimation," *J. Comput. Sci.*, vol. 10, no. 9, 2010.

Links to Internet Resources:

- [1] https://repository.nwu.ac.za/bitstream/handle/10394/25449/Kr%C3%BCger_SW_2017.pdf?sequence=1&isAllowed=y
- [2] <https://pdfs.semanticscholar.org/57fc/83526cdbbf20639f548d80cf4d1af839b5f2.pdf>
- [3] <http://www.sal.ufl.edu/book/RadarSignal-Tutorial2.ppt>
- [4] http://perso.ens-lyon.fr/patrick.flandrin/SPIE01_PF.pdf

Data Sheets:

- [1] NI DAQ 6343: <http://www.ni.com/pdf/manuals/374567d.pdf>
- [2] Avisoft US speaker: <https://www.avisoft.com/usg/vifa.htm>

- [3] Avisoft US recorder: <https://www.avisoft.com/RECODER.pdf>
- [4] Bosch Laser Measurer: <https://www.boschtools.com/ca/en/boschtools-ocs/laser-measuring-glm-35-125633-p/>