Lokalizacija pomoću vremenske razlike dolaska signala

Runtić, David

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Department of Mathematics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Odjel za matematiku**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:126:237586

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni diplomski studij matematike, smjer: Matematika i računarstvo

David Runtić

Localization Using the Time Difference of Arrival (TDOA)

Diplomski rad

Osijek, 2022.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni diplomski studij matematike, smjer: Matematika i računarstvo

David Runtić

Localization Using the Time Difference of Arrival (TDOA)

Diplomski rad

Mentor: izv. prof. dr. sc. Domagoj Matijević
Komentor: Jurica Maltar

Osijek, 2022.

Contents

1 Introduction **1**

2 Setting up the Problem **2**

3 Friedlander's Algorithm **4**
3.1 Basic Terms 4
3.2 How Signals are Transmitted 4
3.3 Deriving the Solution 5
3.4 A Special Case for Two-Dimensional Space using Three Sensors 7
3.5 A Special Case for Three-Dimensional Space using Four Sensors 9

4 Newton's Method **12**

5 Experimental Results **15**
5.1 Improving the Solution using more Sensors 15
5.2 Simulating the Time Difference of Arrivals 17
5.2.1 The Sampling Rate 17
5.2.2 Generating the True Time Difference of Arrival 17
5.2.3 Creating Fake Samples 18
5.2.4 Calculating the Distance between Discrete Signals 19
5.3 A Comparison between Friedlander's and Newton's Method 20
5.4 The Sampling Rate 22
5.5 Signal to Noise Ratio 24
5.6 Sensor Placement 26

6 Applying Friedlander's Algorithm in Real Life **29**
6.1 Equipment 29
6.2 Results 31

7 Conclusion **33**

List of Figures

1 S is on a unique hyperbola. 2
2 Adding two more sensors provides a unique solution. 3
3 An example showing that $R_{2s} = r_{12} + R_{1s}$. 5
4 The tangent line intersects the x axis in the point x_1. 12
5 Our possible source location is not very close to the real location. 15
6 Adding another sensor already improves the result. 16
7 Average absolute error for a different number of sensors. 16
8 The analog signal of speech audio as a function of time ([9] 10). 17
9 A discrete signal taken from the analog signal with a sample period of $T = 0.125\mu s$ ([9] 10). 17
10 An example of a usable audio. 18
11 Two similar audio signals are received at different times. 19
12 Plotting the cross correlation vector gives us a noticeable peak. 20
13 Average error in approximation between the real sound source and an estimated sound source from 100 generated sound sources. 21
14 Median error in approximation between the real sound source and the estimated sound source from 100 generated sound sources. 21
15 Average error in approximation after correction 22
16 Distance between the cross correlation generated lag nT, and the true lag t is Δt. 23
17 Increasing the sample rate decreases Δt. 23
18 Comparison between different sampling rates using nine sensors. 24
19 Our original signal with $SNR = 0$. 25
20 Distance error decreases as we increase SNR. 26
21 The six configurations we will be using. 27
22 Results of each configuration. 27
23 Using three microphones to generate two possible solutions. 29
24 Using four microphones on the same line still gives solutions S and P. 30
25 Moving M4 in the y-axis provides a unique solution S. 30
26 Using four microphones generates a bad estimated solution. 31
27 Placing a fifth microphone gives us a much better estimation. 32

1 Introduction

The aim of this paper is to calculate the location of the signal source using the time difference of arrival technique. The time difference of arrival technique is becoming increasingly more relevant because it relies on the inexpensive computing power and can be used in various fields related to geo-locating RF sources, including communication and radar technology, defense applications, and digital signal processing, e.g., in satellite navigation systems and specific applications related to the location of mobile phones for emergency and other purposes (see [4] 1-2).

The second chapter of this paper explains the time difference of arrival technique and sets up the problem – to find the source of a signal by using this technique.

Chapters 3 and 4 define two methods for calculating the position of a sound source in order to solve our problem. The first method, Friedlander's Time Difference of Arrival Algorithm, will provide an initial solution, while the second method, Newton's method, will be used in an attempt to improve the initial solution.

The fifth chapter will simulate the time difference of arrival and compare the results obtained through Friedlander's and Newton's methods. It will also try to examine the influence of the sample rate, the number of sensors, signal to noise ratio (SNR), and sensor position on the results.

Finally, in the sixth chapter, we will test our methods using real microphones to confirm that our algorithm works and to investigate whether it is applicable in real life.

2 Setting up the Problem

The time difference of arrival is one of the most popular ranging techniques due to its high accuracy, low cost of antennas, and lower complexity when compared to other techniques. In order to calculate the location of the signal source, it is enough to know the sensor positions and our signal speed. Once we have determined the time difference it took for the signal to reach two different sensors, we can calculate the difference in distances between the signal source and our two sensors in order to get a hyperbola where the signal source may be. If we add more sensors, our signal source location will be at the intersection of hyperbolas (see [8]).

Let us suppose that we have sensors S_1 and S_2 and a signal source S which sends a signal at a speed v. If we label the time it took for our signal to reach our sensors as t_1 and t_2, then if the inital sensor is S_1, the time difference of arrival of our signal will be $\tau_{21} = |t_2 - t_1|$. Multiplying this with the signal speed gives us a distance difference between the signal source and the sensors $r_{21} = |R_2 - R_1|$. This information is enough to construct a unique hyperbola with foci S_1 and S_2 on which our signal source S is located.

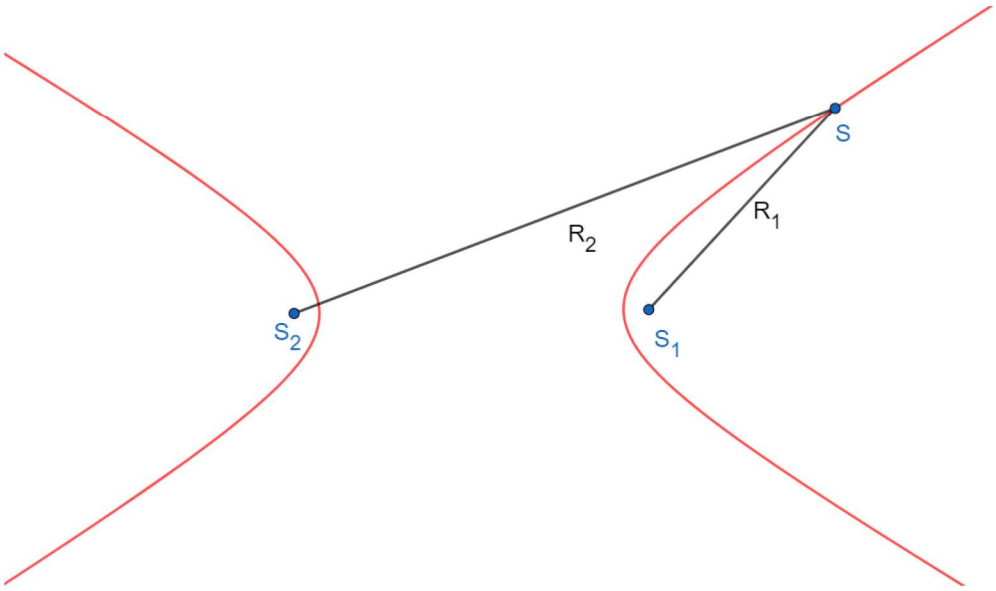

Figure 1: S is on a unique hyperbola.

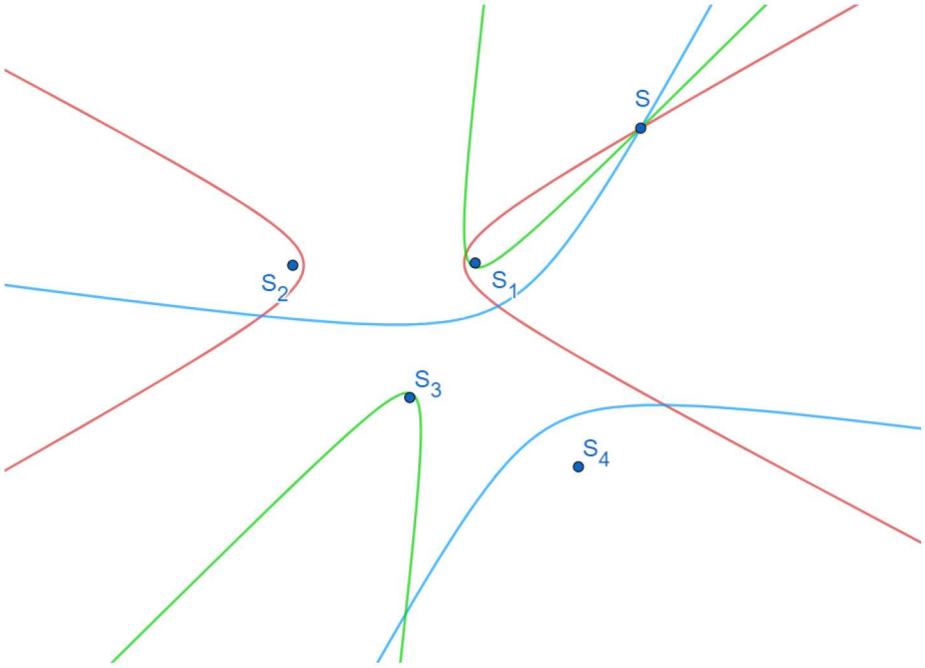

Figure 2: Adding two more sensors provides a unique solution.

Due to hardware imperfections, we will never be able to get perfect time differences of arrival. Our sample rate can be too low, clock synchronization used to find the time difference of arrival can be inaccurate, or we will simply have too much noise to find an accurate position. In order to reduce these effects, we will be using estimation methods to find our solution. Our methods should also be able to use an arbitrary number of sensors in order to mitigate the error.

3 Friedlander's Algorithm

This chapter will focus on how to calculate the position of a sound source using Friedlander's Time Difference of Arrival Algorithm. The input will be a list of length M, where the i-th element in the list will be the difference in transit time between the reference sensor and the i-th sensor.

3.1 Basic Terms

Let N be the number of sensors.
$X_i = (x_i, y_i, z_i)$ are coordinates of sensor $i, i = 1 \dots N$.
$X_s = (x_s, y_s, z_s)$ are coordinates of the signal source.
$X_j = (x_j, y_j, z_j)$ are coordinates of the initial sensor. The reference sensor j can be any of the N sensors (see [3] 235).
$R_{is} = \|X_i - X_s\|$ distance between the i-th sensor and the sound source.
$R_{so} = \|X_s\|$ distance between the sound source and the origin point.
$r_{ij} = R_{is} - R_{js} = \|X_i - X_s\| - \|X_j - X_s\|$ is the range difference between the sensors i and j.

3.2 How Signals are Transmitted

In order to derive the solution, one must first explain how signals, more specifically, audio signals, propagate through the environment. Sound vibrations, which we call sound waves, move through the environment in a wave pattern by vibrating objects. These objects then vibrate other objects, and in that way the sound is carried along. As can be seen in Fig. 1, when sensors (marked as M_1 and M_2) receive a signal at times (t_1 and t_2), multiplying t_1 and t_2 with the signal speed will give us the distance (r_{1s} and r_{2s}) between the signal source and sensors M_1 and M_2, respectively. The difference between these two distances gives us r_{12} (see [2]).

3.3 Deriving the Solution

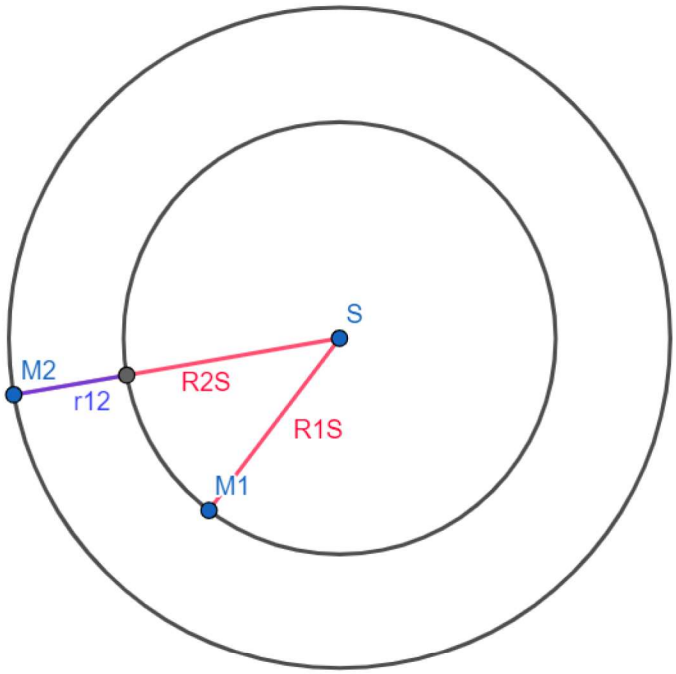

Figure 3: An example showing that $R_{2s} = r_{12} + R_{1s}$.

Notice that

$$R_{is}^2 = (r_{ij} + R_{js})^2 = r_{ij}^2 + 2R_{js}r_{rij} + R_{js}^2 \tag{1}$$
$$R_{is}^2 = \|X_i - X_s\|^2 = R_{io}^2 - 2X_i^T X_s + R_{so}^2. \tag{2}$$

Subtracting (2) from (1) gives us

$$2X_i^T X_s = R_{io}^2 - r_{ij}^2 - 2R_{js}r_{ij} + R_{so}^2 - R_{js}^2 \tag{3}$$

(see [3] 239-41).

A special case for $i = j$ in (3) gives us

$$2X_j^T X_s = R_{jo}^2 + R_{so}^2 - R_{js}^2. \tag{4}$$

Finally, by subtracting (4) from (3), we get

$$2(X_i - X_j)^T X_s = (R_{io}^2 - R_{jo}^2) - r_{ij}^2 - 2R_{js}r_{ij}.$$

This system of $N-1$ equations can be written in matrix form as

$$S_j X_s = \mu_j - R_{js}\rho_j \tag{5}$$

where

$$S_j = \begin{bmatrix} (x_1 - x_j) & (y_1 - y_j) & (z_1 - z_j) \\ (x_2 - x_j) & (y_2 - y_j) & (z_2 - z_j) \\ \vdots & \vdots & \vdots \\ (x_{j-1} - x_j) & (y_{j-1} - y_j) & (z_{j-1} - z_j) \\ (x_{j+1} - x_j) & (y_{j+1} - y_j) & (z_{j+1} - z_j) \\ \vdots & \vdots & \vdots \\ (x_N - x_j) & (y_N - y_j) & (z_N - z_j) \end{bmatrix} \in \mathbb{R}^{(N-1)\times 3}$$

$$\mu_j = \frac{1}{2}\begin{bmatrix} R_{1o}^2 & -R_{jo}^2 & -r_{1j}^2 \\ R_{2o}^2 & -R_{jo}^2 & -r_{2j}^2 \\ \vdots & \vdots & \vdots \\ R_{(j-1)o}^2 & -R_{jo}^2 & -r_{(j-1)j}^2 \\ R_{(j+1)o}^2 & -R_{jo}^2 & -r_{(j+1)j}^2 \\ \vdots & \vdots & \vdots \\ R_{No}^2 & -R_{jo}^2 & -r_{Nj}^2 \end{bmatrix} \in \mathbb{R}^{(N-1)\times 1}$$

$$\rho_j = \begin{bmatrix} r_{1j} \\ r_{2j} \\ \vdots \\ r_{(j-1)j} \\ r_{(j+1)j} \\ \vdots \\ r_{Nj} \end{bmatrix} \in \mathbb{R}^{(N-1)\times 1.}$$

We can now compute S_j, μ_j and ρ_j from the input data.

To get rid of R_{js}, which is an unknown quantity, we can multiply (5) with matrix M, where M has ρ_j in its null-space ($M\rho_j = 0$). We will define M_j^k, D_j, and Z as:

$$M_j^k = (I - Z)^k D_j,$$

where

$$D_j = diag(\rho_j)^{-1} = \begin{bmatrix} r_{1j} & & & & & & \\ & \ddots & & & & & \\ & & r_{(j-1)j} & & & & \\ & & & r_{(j+1)j} & & & \\ & & & & \ddots & & \\ & & & & & r_{Nj} & \end{bmatrix}^{-1}$$

$$Z = \begin{bmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & 0 & \ddots & 1 \\ 1 & & & 0 \end{bmatrix},$$

whereby k can be any natural number, so we will assume that $k = 1$. Z is a circular shift matrix.

Now that we have M_j, we can multiply (5) in order to eliminate $R_{js}\rho_j$. Finally, we get

$$M_j S_j X_s = M_j \mu_j \tag{6}$$

According to Friedlander, "it should be noted that M_j is a singular matrix," and, therefore, it has no inverse ([3] 236). This system can be solved using programs such as Matlab. If we cannot find the inverse, we can always find the pseudo inverse of $M_j S_j$, which exists for any matrix, and then multiply the left side of the equation (6) with the pseudo inverse. A closed form solution is given by

$$X_s = (S_j^T M_j^T M_j S_j)^{-1} S_j^T M_j^T M_j \mu_j. \tag{7}$$

3.4 A Special Case for Two-Dimensional Space using Three Sensors

We will now perform a closed-form solution for a two-dimensional space in which there are three sensors. Let n be the number of dimensions, in this case $n = 2$, and $j = 1$ be the intial sensor. Plugging these numbers in equation (8) gives us

$$[r_{31}(x_2 - x_1) - r_{21}(x_3 - x_1)]x_s + [r_{31}(y_2 - y_1) - r_{21}(y_3 - y_1)]y_s = r_{31}m_1 - r_{21}m_2 \tag{8}$$

where

$$m_i = \frac{1}{2}(R_{io}^2 - R_{1o}^2 - r_{i1}^2), i = 1, 2.$$

The equation (8) can be rewritten as

$$y_s = \alpha x_s + \beta \tag{9}$$

where

$$\alpha = \frac{r_{21}(x_3 - x_1) - r_{31}(x_2 - x_1)}{-r_{21}(y_3 - y_1) + r_{31}(y_2 - y_1)},$$
$$\beta = \frac{r_{31}m_1 - r_{21}m_2}{-r_{21}(y_3 - y_1) - r_{31}(y_2 - y_1)}.$$

According to Friedlander ([3] 240), the equation (6) establishes "a single straight LOP for the source coordinates. To get the source location along the hyperbolic lines of position (LOP) we use the first" two equations from (5) to get

$$(x_2 - x_1)x_s + (y_2 - y_1)y_s = m_1 - R_{1s}r_{21} \tag{10}$$
$$R_{1s} = \frac{(x_2 - x_1)x_s + (y_2 - y_1)y_s - m_1}{r_{21}}. \tag{11}$$

Plugging in (9) into (11) gives us

$$\begin{aligned} R_{1s} &= \frac{(x_2 - x_1)x_s + (y_2 - y_1)y_s - m_1}{r_{21}} \\ &= \frac{(x_2 - x_1)x_s + (y_2 - y_1)(\alpha x_s + \beta) - m_1}{r_{21}} \\ &= \frac{[(x_2 - x_1) + (y_2 - y_1)\alpha]x_s + (y_2 - y_1)\beta - m_1}{r_{21}}. \end{aligned} \tag{12}$$

Notice that R_{1s} can be written as

$$\begin{aligned} R_{1s} &= \sqrt{(x_1 - x_s)^2 + (y_1 - y_s)^2} \\ &= \sqrt{(x_1 - x_s)^2 + (y_1 - \alpha x_s - \beta)^2} \\ &= \sqrt{(1 + \alpha^2)x_s^2 + (y_1 - \beta)^2 + x_1^2 - 2[x_1 + (y_1 - \beta)\alpha]x_s}. \end{aligned} \tag{13}$$

Finally, we can square (12) and (13) and equate their left sides to get the quadratic equation

$$\frac{[(x_2 - x_1) + (y_2 - y_1)\alpha]^2 x_s^2 + 2[(x_2 - x_1) + (y_2 - y_1)\alpha][(y_2 - y_1)\beta - m_1]x_s + [(y_2 - y_1)\beta - m_1]^2}{r_{21}}$$
$$= (1 + \alpha^2)x_s^2 + (y_1 - \beta)^2 + x_1^2 - 2[x_1 + (y_1 - \beta)\alpha]x_s.$$

This equation can be rewritten as

$$ax_s^2 + bx_s + c = 0 \tag{14}$$

where

$$a = \frac{(1+\alpha^2) - [(x_2 - x_1) + (y_2 - y_1)\alpha]^2}{r_{21}^2},$$
$$b = -2[(x_1 + (y_1 - \beta)\alpha) - ((x_2 - x_1) + (y_2 - y_1)\alpha)((y_2 - y_1)\beta - m_1)],$$
$$c = (y_1 - \beta)^2 + x_1^2 - \left[\frac{(y_2 - y_1)\beta - m_1}{r_{21}^2}\right]^2.$$

In order to get two possible solutions, we can solve (14) in the following way:

$$x_s = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$
$$y_s = \alpha x_s + \beta.$$

3.5 A Special Case for Three-Dimensional Space using Four Sensors

Like in the case of the two-dimensional space, we will perform the case for a three-dimensional space with four sensors. Let $j = 1$ be the initial sensor and $n = 3$. According to Qu et al.([11]), plugging these numbers in equation (6) gives us

$$[r_{31}(x_2 - x_1) - r_{21}(x_3 - x_1)]x_s + [r_{31}(y_2 - y_1) - r_{21}(y_3 - y_1)]y_s +$$
$$[r_{31}(z_2 - z_1) - r_{21}(z_3 - z_1)]z_s = r_{31}m_1 - r_{21}m_2$$

and

$$[r_{41}(x_2 - x_1) - r_{31}(x_3 - x_1)]x_s + [r_{41}(y_2 - y_1) - r_{31}(y_3 - y_1)]y_s +$$
$$[r_{41}(z_2 - z_1) - r_{31}(z_3 - z_1)]z_s = r_{41}m_2 - r_{31}m_3.$$

For the sake of convenience, we will write the last two equations as

$$A_1x_s + B_1y_s + C_1z_s = D_1 \tag{15}$$
$$A_2x_s + B_2y_s + C_2z_s = D_2. \tag{16}$$

If we multiply (15) with C_2 and (16) with C_1, then subtract the new equations, we get

$$(A_1C_2 - A_2C_1)x_s + (B_1C_2 - B_2C_1)y_s + (C_1C_2 - C_2C_1)z_s = D_1C_2 - D_2C_1$$

or

$$y_s = \alpha_1 x_s + \beta_1 \tag{17}$$

where

$$\alpha_1 = -\frac{A_1C_2 - A_2C_1}{B_1C_2 - B_2C_1}$$
$$\beta_1 = \frac{D_1C_2 - D_2C_1}{B_1C_2 - B_2C_1.} \tag{18}$$

We can now substitute (17) in (15):

$$A_1 x_s + B_1(\alpha_1 x_s + \beta_1) + C_1 z_s = D_1$$
$$(A_1 + \alpha_1\beta_1)x_s + C_1 z_s = D_1 - \beta_1 B_1.$$

Similarly to (17), we will write the equation above as

$$z_s = \alpha_2 x_s + \beta_2$$

where

$$\alpha_2 = -\frac{A_1 + \alpha_1 B_1}{C_1}$$
$$\beta_2 = \frac{D_1 - B_1\beta_1}{C1.} \tag{19}$$

As in the $n = 2$ case, (6) will give us a single straight LOP for the source location. In order to get the source location along the LOP, we use the first two equations from (6) ([3] 240):

$$(x_2 - x_1)x_s + (y_2 - y_1)y_s + (z_2 - z_1)z_s = m_1 - R_{1s}r_{21.} \tag{20}$$

Now we can insert the equations (18) and (19) in (20)

$$R_{1s} = -\frac{(x_2 - x_1)x_s + \alpha_1(y_2 - y_1) + \alpha_2(z_2 - z_1)}{r_{21}}x_s$$
$$+ \frac{m_1 - (y_2 - y_1)\beta_1 - (z_2 - z_1)\beta_2}{r_{21}} = \alpha_3 x_s + \beta_3. \tag{21}$$

Note that R_{1s} can also be written as

$$R_{1s} = \sqrt{(x_1 - x_s)^2 + (y_1 - y_s)^2 + (z_1 - z_s)^2}$$
$$= \sqrt{(1 + \alpha_1^2 + \alpha_2^2)x_s^2 - 2[x_1 + (y_1 - \beta_1)\alpha_1 + (z_1 - \beta_2)\alpha_2]x_s + (y_1 - \beta_1)^2 + (z_1 - \beta_2)^2 + x_1^2}. \tag{22}$$

Finally, we can square equations (21) and (22) and equate their left sides in order to get a quadratic equation similar for $n = 2$

$$ax_s^2 + bx_s + c = 0 \tag{23}$$

where

$$a = 1 + \alpha_1^2 + \alpha_2 - \alpha_3^2$$
$$b = -2(x_1 + (y_1 - \beta_1)\alpha_1 + (z_1 - \beta_1)\alpha_2 + \alpha_3\beta_3)x_s$$
$$c = (y_1 - \beta_1)^2 + (z_1 - \beta_2)^2 + x_1^2 - \beta_3^2.$$

Like in the 2d case, we can solve (23) for x_s, y_s, and z_s as follows:

$$\begin{aligned} x_s &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \\ y_s &= \alpha_1 x_s + \beta_1, \\ z_s &= \alpha_2 x_s + \beta_2. \end{aligned}$$

4 Newton's Method

Once we have calculated an initial solution, we can begin to improve it by using an iterative method. One of the proposed methods is Newton's method. The term "Newton's method," which is also known as "Newton–Raphson method," refers to "a root-finding algorithm which produces successively better approximations to the roots (or zeroes) of a real-valued function (using fixed point iteration)" ([1]). The method is an iterative technique that can improve the initial solution. It can be used "for solving general nonlinear equations using calculus" and "solving optimization problems by setting the gradient to zero" ([1]). The original Newton's method, first published in 1685, was applied only to polynomials. It was neither connected with derivatives nor was it presented through a general formula. It was in 1740 that British mathematician and inventor Thomas Simpson defined Newton's method "as an iterative method for solving general nonlinear equations using calculus," which has led to the method's contemporary definition ([1]).

Let us show how we can derive Newton's method in two dimensional space. Suppose that we are trying to find the root of a function f. Let us pick x_0 as the first approximation of the root. We can find the value of $f(x_0)$ and calculate the slope k of the tangent line of f in the point x_0 as $f'(x_0)$.

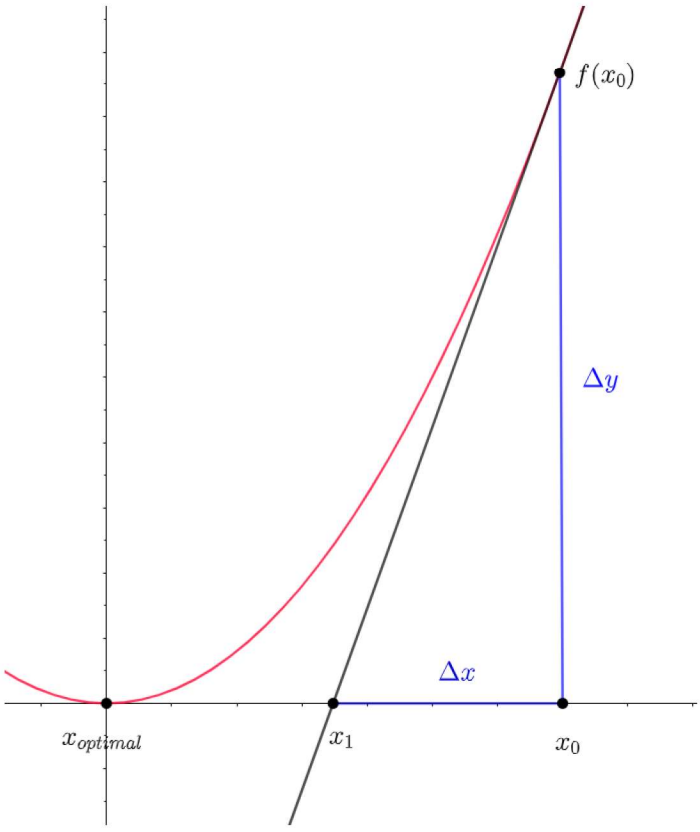

Figure 4: The tangent line intersects the x axis in the point x_1.

Notice that x_1 is closer to the optimal solution than x_0. Now we have to find a way to calculate x_1. First let us notice that $x_1 = x_0 - \Delta x$. We also know that $k = \frac{\Delta y}{\Delta x}$, which means

that Δx is equal to $\frac{\Delta y}{k}$ or $\frac{f(x_0)}{f'(x_0)}$. Finally, this gives us

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

We can repeat this process in order to get a more generalized version:

$$x_{k+1} = x_n - \frac{f(x_k)}{f'(x_k)}.$$

Newton's method can also be similarly derived for n dimensional spaces. The goal now is to define a function for which we can find the root as our solution. We will define it by applying the formula by Qu et al. ([11]):

$$f_i(x,y,z) = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} - \sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2} - c\tau_{ij}$$
$$i = 1, 2, \ldots, n, i \neq j$$

where j is the initial sensor and c is the speed of sound in our medium. The derivative of the function $f_i(x,y,z)$ with respect to x can be found in the following way:

$$\begin{aligned}\frac{\partial f_i(x,y,z)}{\partial x} =& \frac{1}{2}\frac{\frac{\partial (x-x_i)^2}{\partial x}}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}} \\ & - \frac{1}{2}\frac{\frac{\partial (x-x_i)^2}{\partial x}}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2}} \\ =& \frac{x-x_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}} \\ & - \frac{x-x_j}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2}}.\end{aligned}$$

Similarly, the derivative of the function $f_i(x,y,z)$ with respect to y and z will be:

$$\begin{aligned}\frac{\partial f_i(x,y,z)}{\partial y} =& \frac{y-y_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}} \\ & - \frac{y-y_j}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2}}.\end{aligned}$$

and

$$\begin{aligned}\frac{\partial f_i(x,y,z)}{\partial z} =& \frac{z-z_i}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}} \\ & - \frac{z-z_j}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2}}.\end{aligned}$$

We can now define the Jacobian matrix of $f(x, y, z)$ as

$$f'(x,y,z) = \begin{bmatrix} \nabla f_1(x,y,z) \\ \nabla f_2(x,y,z) \\ \vdots \\ \nabla f_{j-1}(x,y,z) \\ \nabla f_{j+1}(x,y,z) \\ \vdots \\ \nabla f_n(x,y,z) \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x} & \frac{\partial f_n}{\partial y} & \frac{\partial f_n}{\partial z} \end{bmatrix}.$$

Finally, we can calculate the pseudo inverse $f'(x, y, z)^{-1}$ of our Jacobian matrix in order to find our estimated result as follows:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} - f'(x_k, y_k, z_k)^{-1} \begin{bmatrix} f_1(x_k, y_k, z_k) \\ f_2(x_k, y_k, z_k) \\ \vdots \\ f_{j-1}(x_k, y_k, z_k) \\ f_{j+1}(x_k, y_k, z_k) \\ \vdots \\ f_n(x_k, y_k, z_k) \end{bmatrix}.$$

Newton's method is very accurate and fast, but in order for it to converge fast, the first approximation must be close to the true location. By combining our first estimated result from Friedlander's method with Newton's method, we can get an accurate and fast new method.

5 Experimental Results

In this section we will describe how results can be modified by using more sensors, by changing the sample rate and the effect of SNR, and by different sensor placement.

5.1 Improving the Solution using more Sensors

One of the main advantages of our implemented methods is that we are easily able to add more sensors in order to improve the solution. This will be able to somewhat compensate for noisy signals or inaccurate clocks in our sensors.

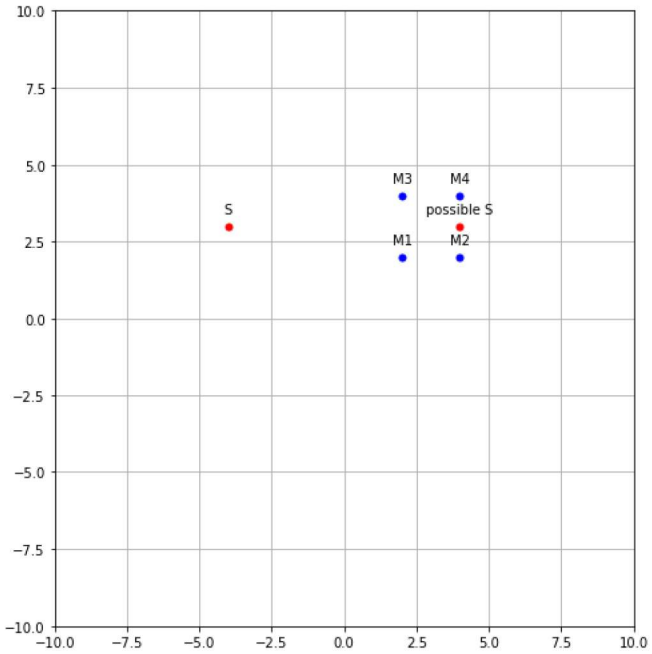

Figure 5: Our possible source location is not very close to the real location.

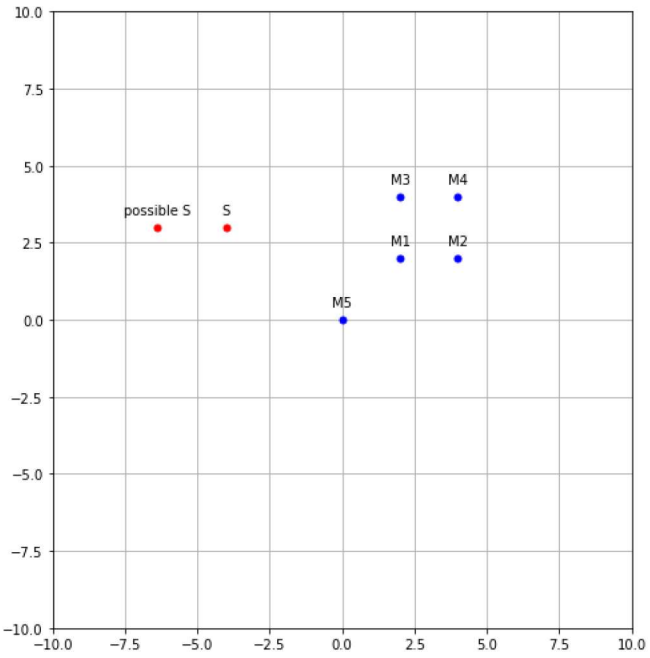

Figure 6: Adding another sensor already improves the result.

We will try to simulate the results using Friedlander's algorithm when we choose a different number of sensors. All the sensors will have a sample rate of $44kHz$. We will be generating 100 random points for each distance and taking the average error.

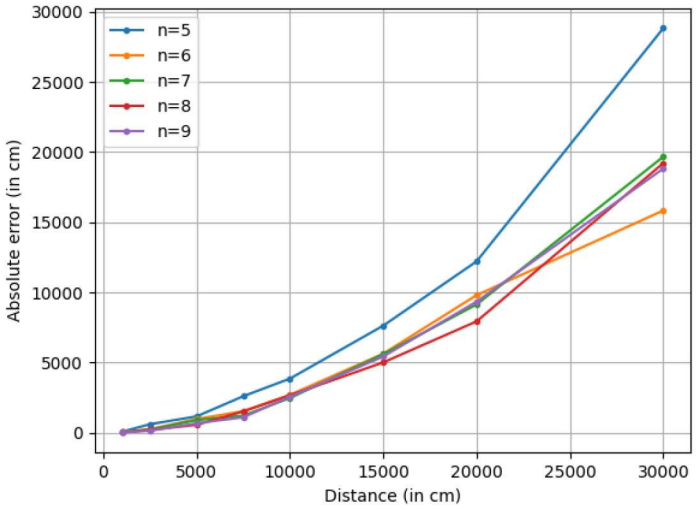

Figure 7: Average absolute error for a different number of sensors.

The results have shown that there is a significant improvement if we use more than five sensors. However, randomly using more than six sensors does not necessarily give us better results. We would need to carefully pick where the new sensors are placed if we want to take advantage of their additional information.

5.2 Simulating the Time Difference of Arrivals

5.2.1 The Sampling Rate

Before we try to simulate our time difference of arrival, let us explain the difference between discrete and analog signals and how sampling rate is defined. The term signal is generally applied to something that conveys information. An example for that would be a sound that we have heard. We can think of these signals as mathematical functions of time. These types of signals are known as analog signals. Analog signals are usually very difficult to store and difficult to replicate. Instead of trying to replicate the analog signal, what we can do is get the values of our signal at certain periods in order to get an estimation of our analog signal. This is known as sampling. The term discrete-time signal is used to describe signals whose value is known at discrete instants in time. In our simulation, this will be a vector of n values, where the i-th value in the vector corresponds to the value of our i-th sample.

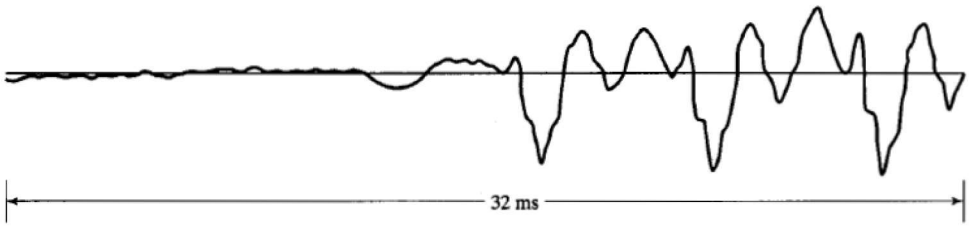

Figure 8: The analog signal of speech audio as a function of time ([9] 10).

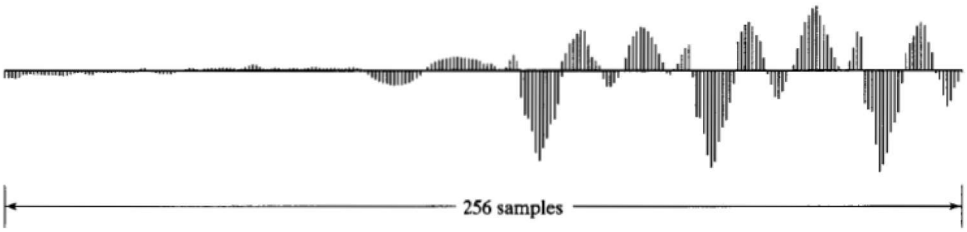

Figure 9: A discrete signal taken from the analog signal with a sample period of $T = 0.125\mu s$ ([9] 10).

Now that we have defined discrete signals, it is easy to see that the sampling rate is simply the number of samples we will take in a certain period. If we know our sampling period T, then the sampling rate F_s is equal to $\frac{1}{T}$. In the figure above, we had a sample period of $0.125\mu s$, which gives us a sampling rate of $8kHz$.

5.2.2 Generating the True Time Difference of Arrival

In order to calculate the true time difference of arrival, we will assume that our sensors' positions and signal source positions are known. By knowing the locations of each sensor

and signal source, we can also calculate the distance between the signal source and each of the sensors. Dividing our distances with the speed of our signal gives us the time it took for the signal to reach each sensor. Let us store these values for n sensors in a sequence $t = \{t_1, t_2, \ldots, t_n\}$. We can now pick an arbitrary t_j and subtract it from every value in our sequence t. This gives us our true values for the time difference of arrival:

$$\begin{aligned}\tau =& \{t_1 - t_j, t_2 - t_j, \ldots, t_j - t_j, \ldots, t_n - t_j\} \\ =& \{\tau_{1j}, \tau_{2j}, \ldots, 0, \ldots, \tau_{nj}\}.\end{aligned}$$

5.2.3 Creating Fake Samples

For our initial signal, we will be using an audio file with a sample rate of 44 kHz. It is recommended that the audio has distinct sounds, otherwise we will not be able to get a clear picture of the signal due to the added noise from the environment.

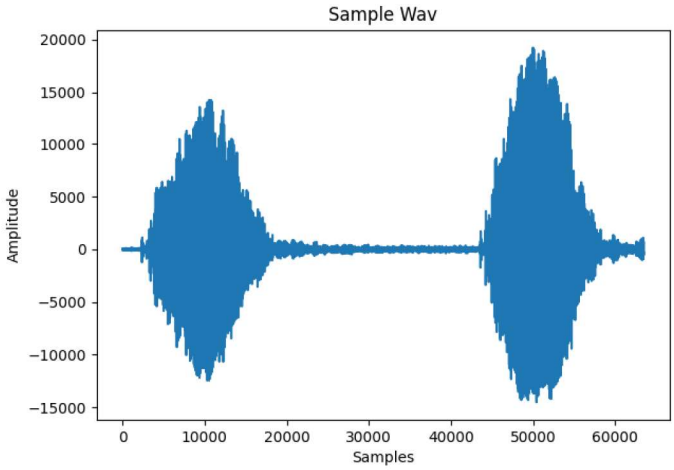

Figure 10: An example of a usable audio.

For simplicity's sake, let us assume that the j-th sensor is the first sensor which received our sound signal. Therefore, the signal as received by the j-th sensor will be the original sound file. In order to generate other samples, we will multiply τ_{ij} with our sampling rate, $\forall i \neq j$ in order to get the number of zeros with which we will pad our original audio signal in order to create a fake signal i. For now, let us also add a little bit of noise in order to make it look a bit more realistic. In a later section we will explain how exactly noise is simulated and how it affects our results.

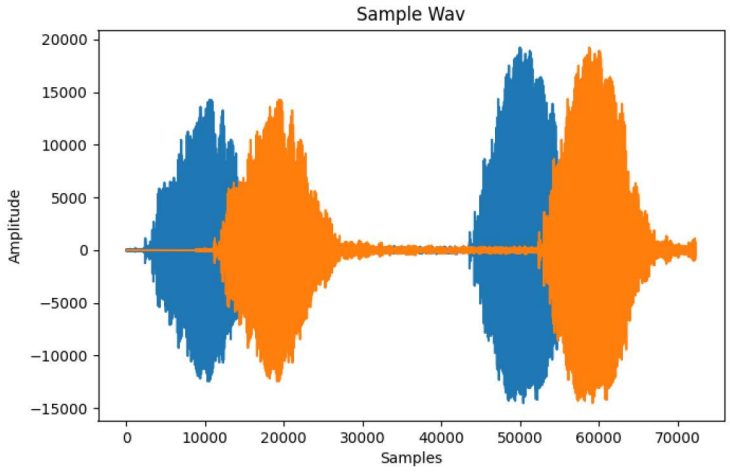

Figure 11: Two similar audio signals are received at different times.

5.2.4 Calculating the Distance between Discrete Signals

There are many ways to find the distance between two similar signals, one of which is using cross-correlation. It is defined with

$$R_{xy}(k) = \sum_{i=-\infty}^{\infty} x(i)y(i-k).$$

Cross-correlation is usually used to measure the similarity between a signal x and a lagged signal y, which is perfect for us because this is exactly what our sensors will receive. Calculating the cross-correlation between signals x and y will give us a vector of correlation values along with a corresponding vector of sample shifts. After calculating the correlation vector, we will take index i of the largest value in our correlation vector and find the i-th value in our sample shift vector. This gives us the distance in samples between the two signals. Now all we need to do is divide our sample shift i with the sampling rate in order to calculate the time difference of arrival between the two sensors which received signals x and y.

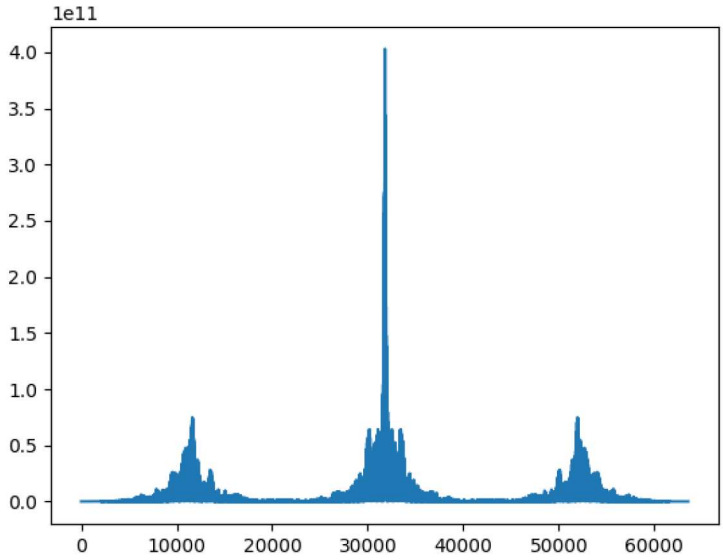

Figure 12: Plotting the cross correlation vector gives us a noticeable peak.

To summarize, we will calculate the cross correlation vector $R_{xy} = \{R_{xy}(-m), \ldots, R_{xy}(n)\}$ using vectors $x = \{x_1, \ldots, x_n\}$ and $y = \{y_1, \ldots, y_n\}$. After calculating R_{xy}, we will find the index $i = \arg\max_{i \in \{-m,\ldots,n\}} R_{xy}$ for which R_{xy} has the maximum value. Once we have found i, we can calculate τ_{ij} as $\frac{i}{F_s}$. We repeat this for all the sensors in order to get our vector τ. For the implementation of time difference of arrival in Python, see `https://gitlab.com/druntic/tdoa`.

5.3 A Comparison between Friedlander's and Newton's Method

Now that we have simulated the time difference of arrivals, we can start to evaluate the results. We will generate 100 random sound sources, where each sound source will be roughly the same distance from the origin. The points of interest will be the average and median values of these 100 sound sources. We will use nine identical sensors placed no more than $10cm$ apart from each other, with a sampling rate of 44 kHz.

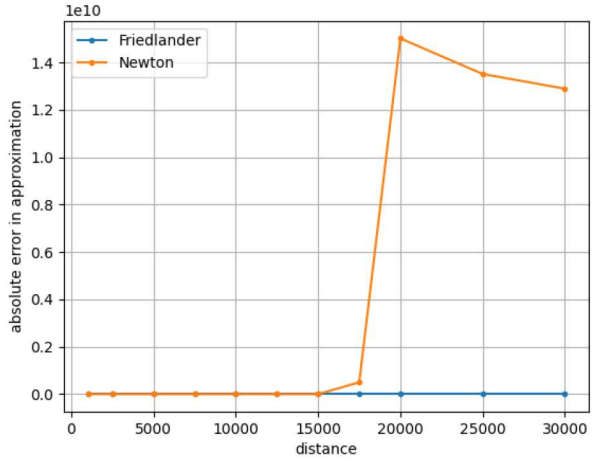

Figure 13: Average error in approximation between the real sound source and an estimated sound source from 100 generated sound sources.

Taking a look at the figure above shows us that Newton's method gives us very inaccurate results with the initial implementation. In order to understand why, let us take a look at the median values of our distances of sound sources.

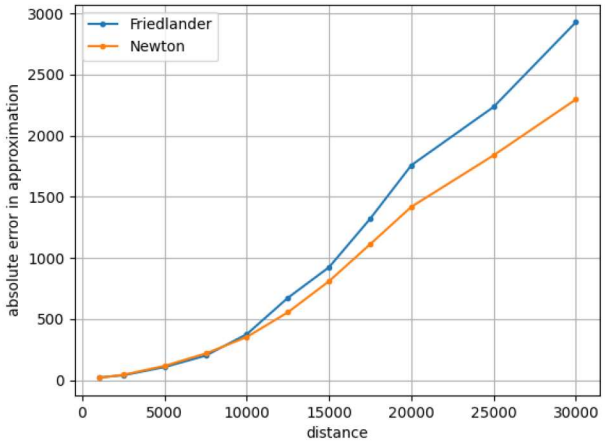

Figure 14: Median error in approximation between the real sound source and the estimated sound source from 100 generated sound sources.

Now we can see that the median distance value of our 100 points generated by Newton's method is actually similar or better than median values obtained by Friedlander's method. The issue with the proposed Newton's method lies with our initial approximation. If the initial approximation of our sound source is too far from the original source, then Newton's method will diverge from the true location. According to [10], we might be able to solve this

in the following way:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k).}$$

Suppose our solution x_k converges to some number L, taking the limit from the equation above gives us

$$L = L - \lim_{k\to\infty} \frac{f(x_k)}{f'(x_k)}.$$

Subtracting L from both sides, we get

$$\lim_{k\to\infty} \frac{f(x_k)}{f'(x_k)} = 0.$$

This means that as we approach our optimal solution, $|\frac{f(x_k)}{f'(x_k)}|$ should decrease. As long as $|\frac{f(x_k)}{f'(x_k)}| \geq |\frac{f(x_{k+1})}{f'(x_{k+1})}|$, we will know that our results are going in the right direction. Now that we have established a convergence criterion, let us once again check the average distances.

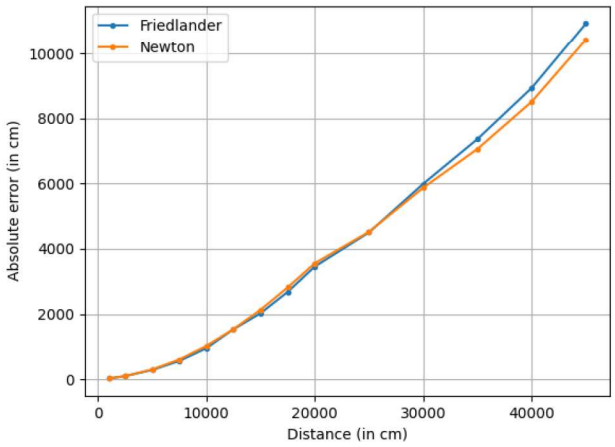

Figure 15: Average error in approximation after correction

As we can see, Newton's method improves our results after our distance between the sound source and origin point is over $300m$.

5.4 The Sampling Rate

The sampling rate is one of the most important factors in order to get an accurate time difference of arrivals. Using cross correlation, we calculated the distance in samples between two received signals. One of the problems when we are sampling signals is that we can only sample at certain periods T. This means that the distance in samples, when comparing the two signals, will always be a multiple of T. In reality this number is never an integer but a real number. Let us have a look at the signal with the sample lag S_n, which we got using cross correlation.

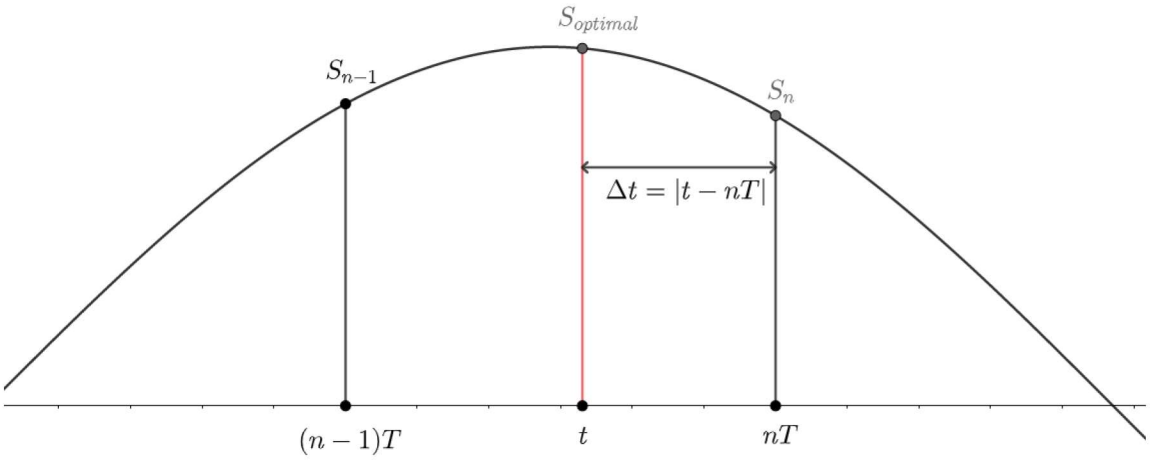

Figure 16: Distance between the cross correlation generated lag nT, and the true lag t is Δt.

We can see that the distance between S_{n-1} and S_n is exactly one sample. Our point of interest is the optimal sample. Δt is also known as fractional delay. The closer our estimation is to the optimal location, the better our time difference of arrival results will be. Now let us have a look at what will happen if we double our sampling rate.

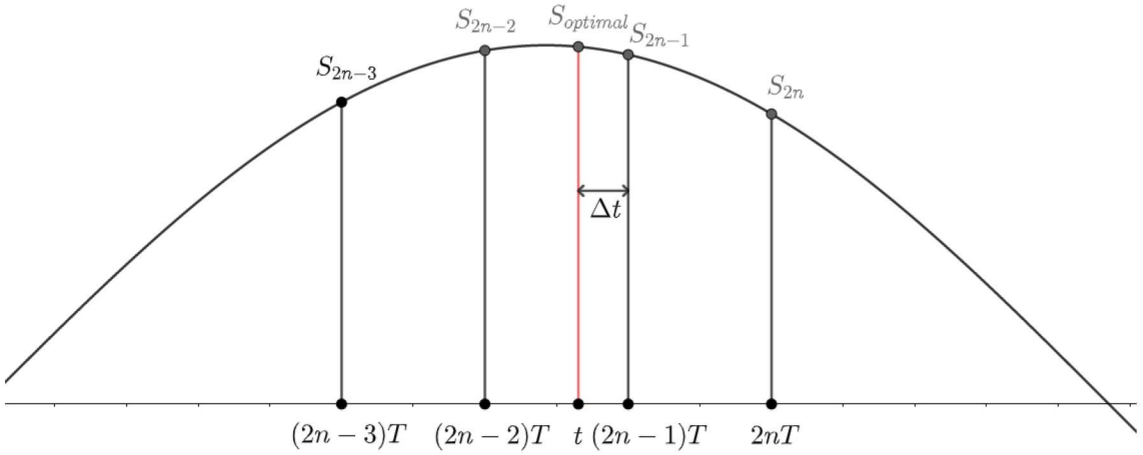

Figure 17: Increasing the sample rate decreases Δt.

We can see that our new sample S_{2n-1} is closer to the optimal sample $S_{optimal}$ than the sample S_{2n}, which we got using the original sampling rate. Increasing the sampling rate does not always guarantee a decreased Δt, but at the very least, it will never give us a worse result. These days most sound cards use a sampling rate of 44 kHz due to the human range of hearing being around 20 kHz, which corresponds to Nyquist's theorem [7], which

states that if a signal has a maximum frequency f, then the sampling rate has to be at least $2f$ in order to get an accurate representation of the signal. Professional microphones for detecting the time difference of arrivals can have a sampling rate of over 250 kHz. We will now simulate our TDOAs with different sampling rates and simulate the results using Friendlander's algorithm.

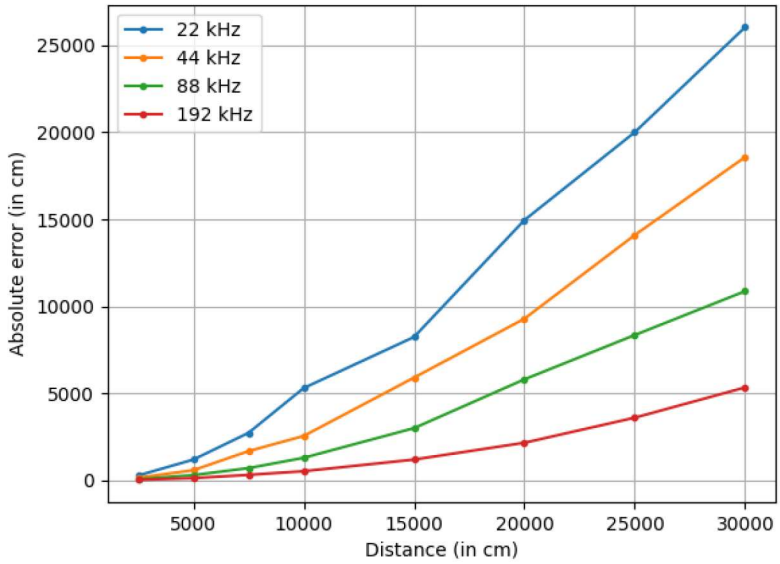

Figure 18: Comparison between different sampling rates using nine sensors.

As we can see in Figure 18, by changing our sampling rate, we can greatly improve our estimated location accuracy.

5.5 Signal to Noise Ratio

One of the problems when receiving a signal is the added noise from the environment. If the signal is too noisy, the cross correlation will not be able to provide accurate results. Almost all sensors have information about signal to noise ratio, otherwise known as SNR. This is simply the ratio between the average power of the signal and the average power of the noise:

$$SNR = \frac{P_{signal}}{P_{noise}}.$$

We would like to simulate the results when the SNR changes; thus our goal will be to add Additive White Gaussian Noise (AWGN) to the signal according to the SNR. Generally, most manufacturers give the SNR in decibels

$$SNR_{dB} = 10 \log_{10} \frac{P_{signal}}{P_{noise}}$$

where the average power of our discrete time domain signal is

$$P_{signal} = \frac{1}{n} \sum_{t=0}^{n} x(t)^2.$$

This means that if our SNR is negative, then the average noise power is larger than the average signal power:

$$\begin{aligned} SNR_{dB} &= 10\log_{10} P_{signal} - 10\log_{10} P_{noise} \\ &= P_{signal,dB} - P_{noise,dB}. \end{aligned}$$

Now we can clearly see that the average noise power is simply the average signal power minus the SNR. Converting this from decibels gives us

$$P_{noise} = 10^{\frac{P_{noise,dB}}{10}}.$$

Now that we have calculated the average noise power, we will create a Gaussian random variable with a mean $\mu = 0$ and variance $\sigma^2 = P_{noise}$ and add it do our signal.

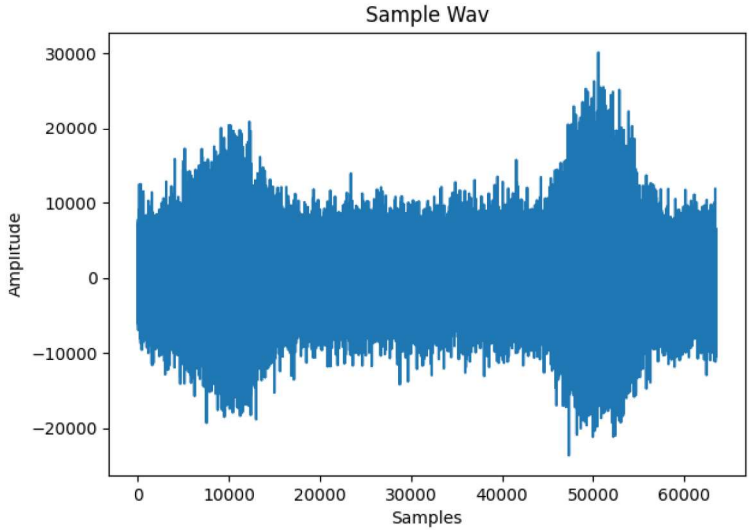

Figure 19: Our original signal with $SNR = 0$.

As we can see, even with the average noise power being equal to the average signal power, it will still resemble the original signal. Cross correlation is very robust to noise. We will be able to see this as long as the standard deviation of the noise is not larger than the peak in the signal which we will transmit ([4] 3).

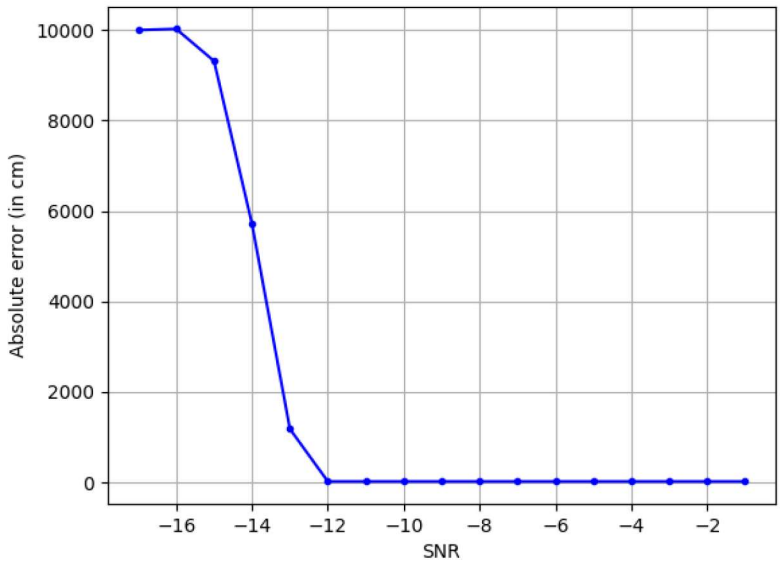

Figure 20: Distance error decreases as we increase SNR.

For this specific audio signal we were able to get the correct time differences of arrival starting from $SNR = -12$. The best case scenario for our audio signals would be complete silence followed by short bursts of sound (e.g. gunshots), while the worst case would be white noise.

5.6 Sensor Placement

For the next experiment we will try to use a few different sensor configurations in order to see how they affect our results. Every configuration will be placed on a $20cm \times 10cm \times 10cm$ box and will use exactly six sensors. The last two configurations will have one sensor placed $10cm$ away from the box. This should have a significant impact on our results.

	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6
Configuration 1	(-10,-5,-5)	(10,-5,-5)	(-10,5,-5)	(-10,-5,5)	(10,5,5)	(10,-5,5)
Configuration 2	(-10,-5,-5)	(10,-5,-5)	(-10,5,-5)	(-10,-5,5)	(10,5,5)	(10,5,-5)
Configuration 3	(-10,-5,-5)	(10,-5,-5)	(-10,5,-5)	(-10,-5,5)	(10,5,-5)	(10,-5,5)
Configuration 4	(-10,-5,-5)	(10,-5,-5)	(-10,5,-5)	(-10,-5,5)	(10,5,5)	(0,5,0)
Configuration 5	(0,-5,5)	(10,-5,-5)	(-10,5,-5)	(-10,-5,15)	(10,5,5)	(0,5,0)
Configuration 6	(0,-5,5)	(10,5,5)	(-10,5,-5)	(-10,5,15)	(10,5,-5)	(0,5,2.5)

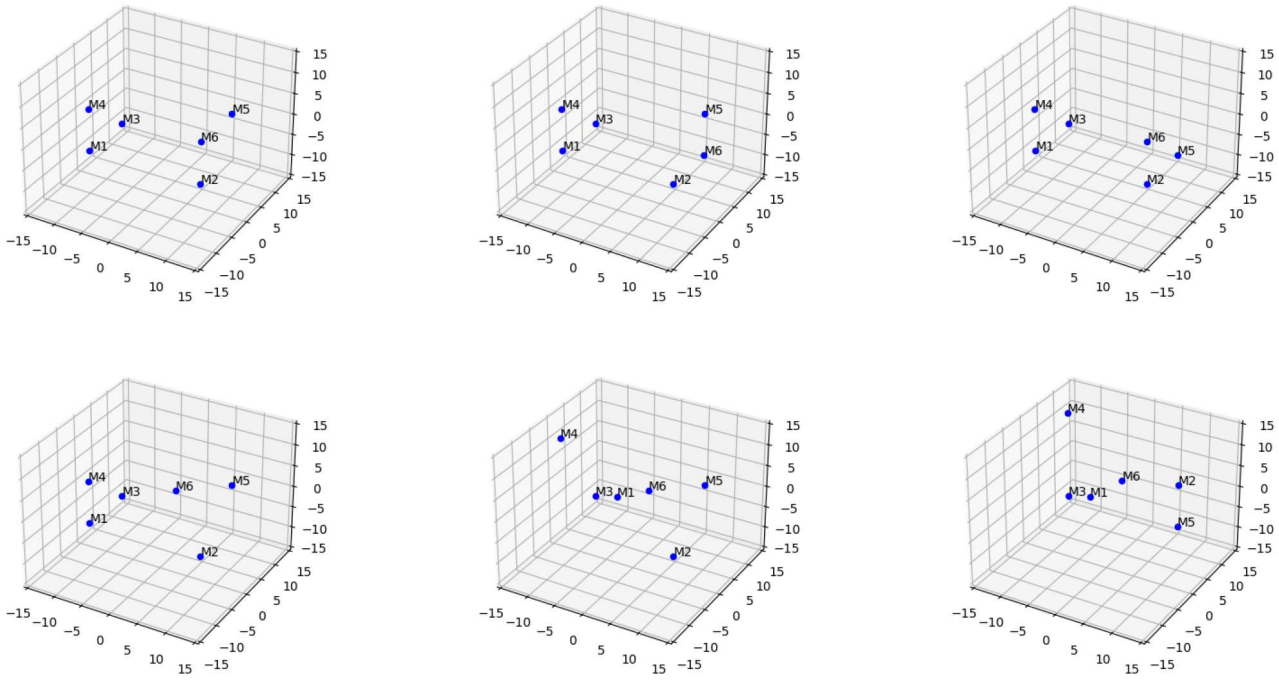

Figure 21: The six configurations we will be using.

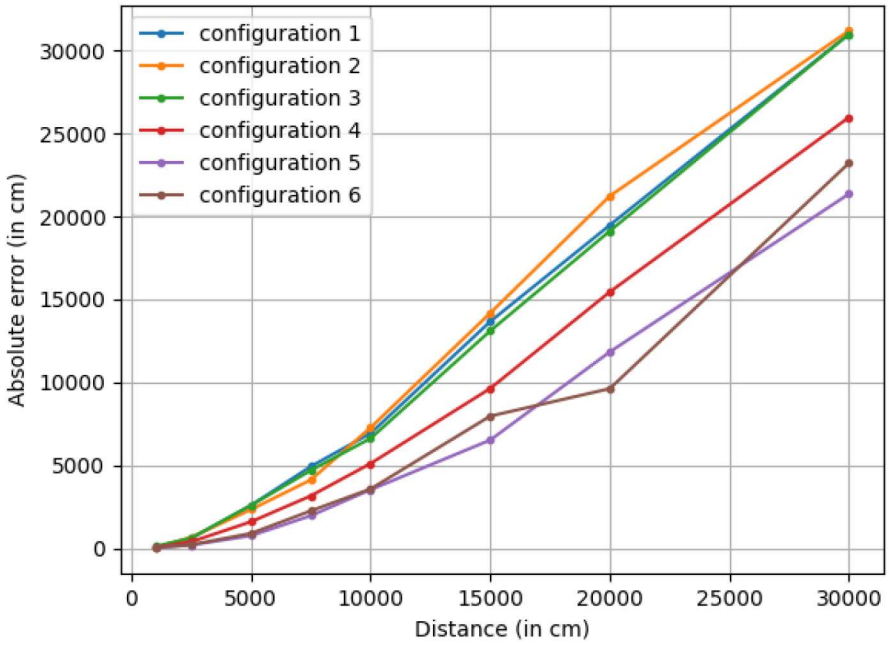

Figure 22: Results of each configuration.

We can see that the fourth configuration has a massive increase in accuracy when compared to the previous three configurations. Configurations five and six have also had an increase due to one sensor not being inside of our box. We can compare this to placing an antenna on our box in order to increase the distance.

6 Applying Friedlander's Algorithm in Real Life

6.1 Equipment

For this experiment we will be using four identical microphones with sample rates of 44 kHz with coordinates

$$(-0.04875, 0, 0), (0.04875, 0, 0), (-0.0403, -0.00425, 0.02852), (0.0403, -0.00425, 0.02852)$$

in meters. Since our algorithms required at least five microphones in order to work in 3D, and we had only four microphones, we tried to find a solution in 2D instead of in 3D. The microphones were placed in such a way that all of them were on the same plane; this means that even if our algorithms had been able to calculate the position using four microphones, it would still have given us at least one coordinate which was wrong. An explanation of the issue will be given in 2D.

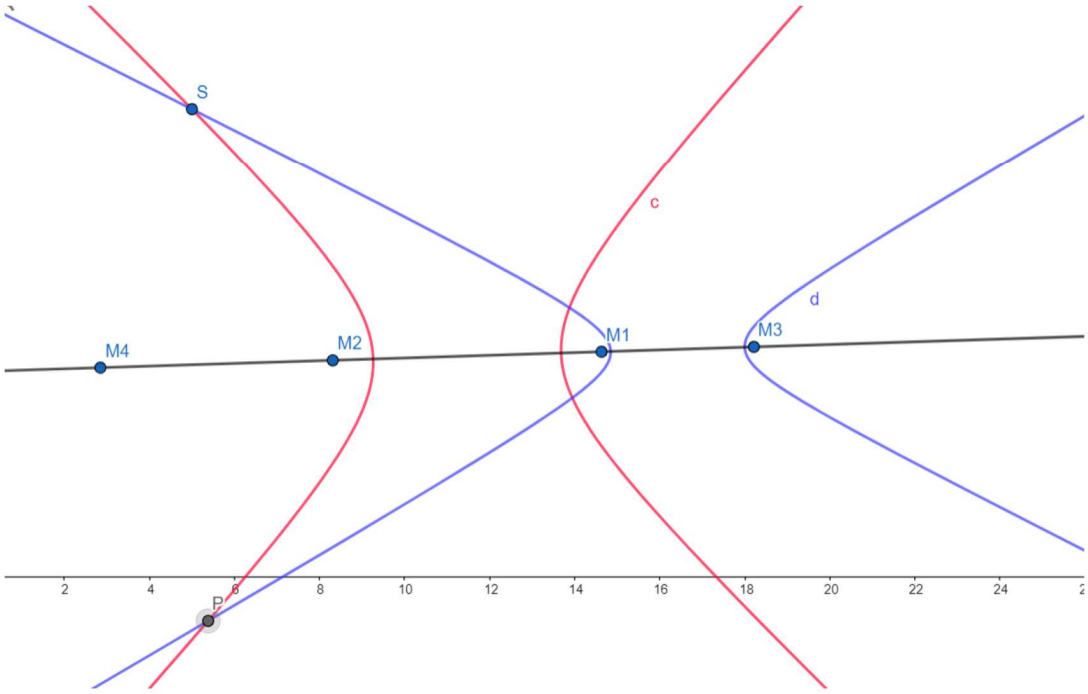

Figure 23: Using three microphones to generate two possible solutions.

Let us assume that our initial microphone is M_1 and our sound source is S. Using three microphones, we can generate two hyperbolas; their intersection will give us two possible solutions. If we place the $4th$ microphone on the same line as the rest of them, we will get no additional information and Friedlander's algorithm will not be able to figure out in which direction to look.

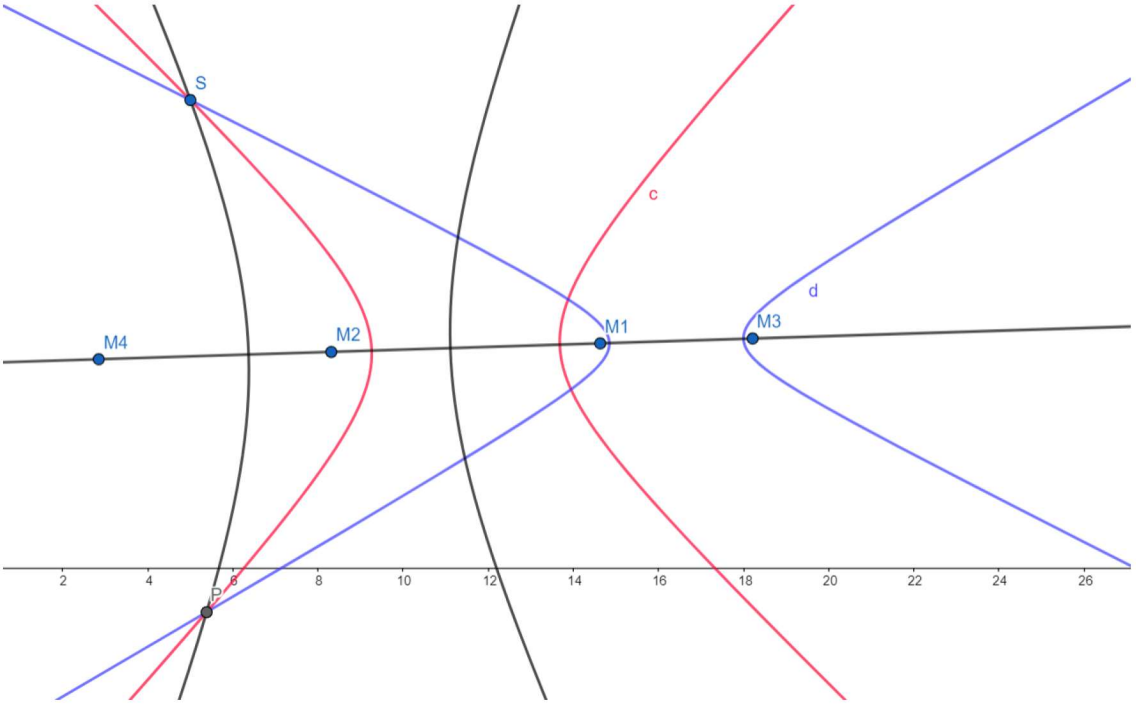

Figure 24: Using four microphones on the same line still gives solutions S and P.

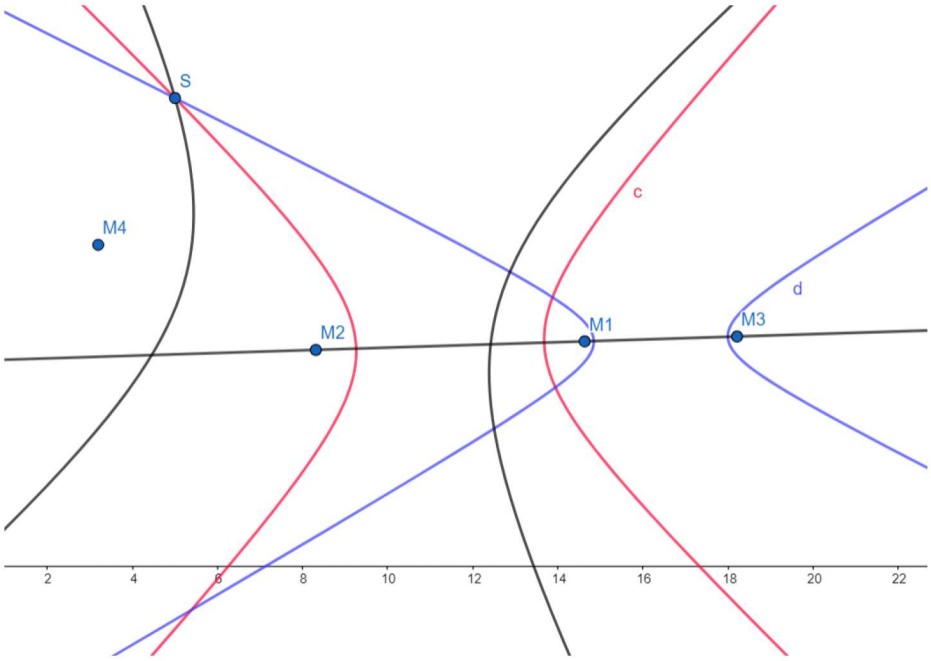

Figure 25: Moving M4 in the y-axis provides a unique solution S.

6.2 Results

Due to the problem mentioned above, we will ignore the second coordinate in all our results. Our results will be measured in meters, and our initial microphone will be the third microphone.

True position	Estimated position	True sample lag	Estimated sample lag
[-0.108, -0.09, -0.018]	[-0.0122, 0.0003, -0.0022]	[-1.23, 8.15, 0.0, 7.61]	[1, 11, 0, 11]
[-0.128, 0.091, -0.018]	[0.0322, -0.0018, 0.0121]	[-2.01, 8.03, 0.0, 7.94]	[-1, 8, 0, 10]
[-0.062, 0.123, -0.018]	[0.0108, -0.0015, 0.0103]	[-1.58, 3.79, 0.0, 4.21]	[-1, 3, 0, 5]
[0.051, 0.091, -0.018]	[-0.0422, -0.0066, 0.0444]	[-0.47, -6.10, 0.0, -4.32]	[-2, -8, 0, -6]
[0.059, -0.052, -0.018]	[0, 0.0014, -0.0098]	[0.17, -8.24, 0.0, -6.52]	[-1, -11, 0, -12]
[-0.124, 0.07, -0.018]	[0.0145, 0.0005, -0.0035]	[-2.18, 8.56, 0.0, 8.42]	[0, 12, 0, 13]
[0.136, 0.013, -0.018]	[-0.0463, -0.0017, 0.0120]	[0.37, -12.06, 0.0, -9.76]	[-2, -14, 0, -13]

Due to our microphones being too close to each other, we were not able to get satisfactory results. At these distances, a difference in samples of 1 to 4 will generate completely different locations compared to the original locations. One of the solutions is to use better microphones or to increase the distances between them. We will also have to care about their positioning so that at least one microphone is not on the same plane as the others.

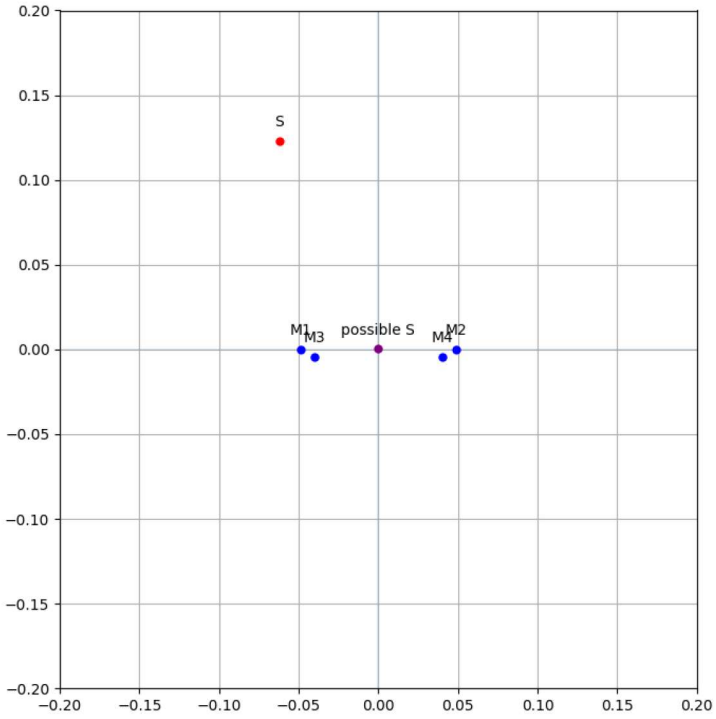

Figure 26: Using four microphones generates a bad estimated solution.

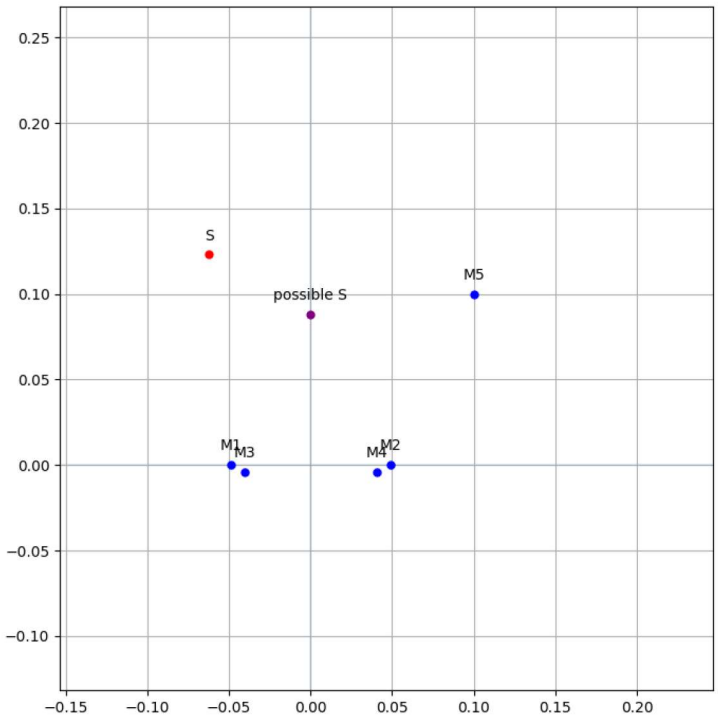

Figure 27: Placing a fifth microphone gives us a much better estimation.

7 Conclusion

Based on the simulation of time difference of arrival conducted in this analysis, it can be concluded that time difference of arrival can produce precise results, even when the sensors are extremely close.

Friendlander's algorithm has been used to derive the first approximation due to its simple implementation and ability to add any number of sensors. We have also devised a way to improve the initial solution using Newton's method.

Depending on the signal, cross correlation is able to handle signals with a negative SNR. Increasing the sample rate can drastically improve our results. Adding more sensors does not necessarily improve the accuracy. Because of that, in order to use the new information obtained by the added sensors, the sensors must placed in optimal locations.

We were not able to get satisfactory results in real life due to bad sensor placement, not enough sensors for 3D, and bad clock synchronization between sensors, which resulted in imprecise calculation of time difference of arrival.

References

[1] Newton's Method https://en.wikipedia.org/wiki/Newton%27s_method

[2] Circular and Spherical Waves. Britannica https://www.britannica.com/science/sound-physics/Beats

[3] Friedlander, Benjamin, *A Passive Localization Algorithm and Its Accuracy Analysis*, IEEE Journal of Oceanic Engineering, Vol. 12, No. 1 (1987), 234-245, doi: https://ieeexplore.ieee.org/document/1145216

[4] International Telecommunication Union, *Comparison of Time-Difference-of-Arrival and Angle-of-Arrival Methods of Signal Geolocation – Report ITU-R SM.2211-2 (06/2018)*, ITU, 2018. www.itu.int/dms_pub/itu-r/opb/rep/R-REP-SM.2211-2-2018-PDF-E.pdf

[5] Li, Zan, Desislava C. Dimitrova, David Hawes Raluy, and Torsten Braun, *TDOA for Narrow-Band Signal with Low Sampling Rate and Imperfect Synchronization*, 7th IFIP Wireless and Mobile Networking Conference (WMNC) (2014), 1-8.

[6] Lyons, Richard G., *Understanding Digital Signal Processing. Pearson Education*, https://www.mikrocontroller.net/attachment/341426/Understanding_digital_signal_processing.pdf

[7] Nyquist's Theorem., *Science Direct*, https://www.sciencedirect.com/topics/engineering/nyquist-theorem

[8] O'Keefe, Brian, *Finding Location with Time of Arrival and Time Difference of Arrival Techniques*, 2017 Tech Notes, sites.tufts.edu/eeseniordesignhandbook/files/2017/05/FireBrick_OKeefe_F1.pdf

[9] Oppenheim, Alan V., Ronald. W. Schafer, and John R. Buck, *Discrete-time Signal Processing*, Prentice Hall, 1999. https://research.iaun.ac.ir/pd/naghsh/pdfs/UploadFile_2230.pdf

[10] *Prove Newton's Iteration Will Diverge for These Functions, No Matter What Real Starting Point Is Selected*, StackExchange – Mathematic, https://math.stackexchange.com/questions/1472515/prove-newtons-iteration-will-diverge-for-these-functions-no-matter-what-real-s

[11] Qu, Junsuo, Shi, Haonan, Qiao, Ning et al. *New Three-dimensional Positioning Algorithm through Integrating TDOA and Newton's Method*, J Wireless Com Network 2020, no. 77 (2020), doi.org/10.1186/s13638-020-01684-7

[12] Sanders, John Nick, Benjamin Noble, and Jeremy Hopfinger, *KAcoustic Triangulation Device*, https://www.ece.ucf.edu/seniordesign/fa2009sp2010/g13/files/Conference%20Paper.Group%2013.pdf

Abstract

The aim of this paper is to find the location of a signal source using time difference of arrival. The paper first defines time difference of arrival and how it can give us a unique solution. It proceeds to present two estimation methods – Friedlander's Time Difference of Arrival Algorithm and Newton's Method. The first method provides an initial solution, while the second method attempts to improve on the initial solution. After deriving the algorithm, we attempted a simulation of time difference of arrival using an audio file, an arbitrary number of sensors whose locations are known, and a signal source whose location is also known. An attempt at simulating the effects of different number of sensors, the distance between sensors and source location, the sample rate, the signal to noise ratio, and, finally, different sensor placements on our accuracy has also been conducted. In order to confirm the accuracy, the algorithm has been tested using real microphones. The analysis of time difference of arrival has produced the following results. The results obtained through a simulation of time difference of arrival can be precise, even when the sensors are close to each other. Cross correlation can handle signals with a negative SNR. Whereas increasing the sample rate can drastically improve our results, adding more sensors does not necessarily improve the accuracy. For the accuracy to be improved, the sensors should be placed in optimal locations.

Keywords: time difference of arrival, Python TDOA implementation, TDOA sample rate, TDOA noise, TDOA Newton, Friedlander's Algorithm

Bio-Note

David Runtić was born in Osijek, Croatia in 1998. He received his bachelor's degree in Mathematics from the Department of Mathematics, J. J. Strossmayer University of Osijek in 2020 with the thesis titled *The Chromatic Polynomial*, completed under the supervision of Professor Snježana Majstorović. This master's thesis is the result of an experimental study that was conducted from March to April 2022 during his internship at Orqa d.o.o., Osijek. On completion of his internship, he has worked at Orqa d.o.o. as a collaborator on the project *Implementation of the OFDM Algorithm on WiFi and 4G Protocols for Drone Data Transmission*. His research interests include signal processing, software defined radio, and machine learning.