UNIVERSITY OF CAPE TOWN

EEE3097S

ECE DESIGN

Milestone 2

Author:
Murray Inglis
Tinashe Timba
Dylan Trowsdale

Student Number:
INGMUR002
TMBTIN004
TRWDYL001

September 15, 2023

Murray Inglis
Tinashe Timba
Dylan Trowsdale

Contents

1 ADMIN DOCUMENTS

1.1 Contributions

Name	Student Number	Contributions
Dylan Trowsdale	TRWDYL001	Evaluation, Conclusion, Timeline Update, Algorithm Research
Murray Inglis	INGMUR002	Algorithm Research, Coding, Simulation Results and Analysis, Github
Tinashe Timba	TMBTIN004	Simulation Setup, Trello Page setup, System Design and Implementation, GUI, Algorithm Research

1.2 Trello

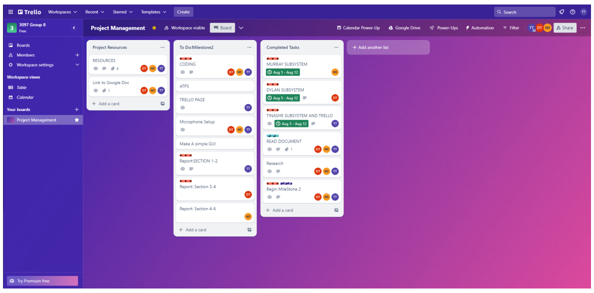

Figure 1: Project Management Tool

1.3 Link To GitHub

https://github.com/murrayinglis/3097-Group-8-2023

1.4 Timeline

As of this timeline, the project is up to date and currently slightly ahead of progress. The simulations are successful, confirming the algorithm works and Milestone 2 is completed. The Raspberry Pis have also already been configured with the microphones, confirming they can record and sample audio and cross-correlate the microphone inputs. The next step is to synchronize the 2 pis and finalize the system.

Weeks	Tasks
1	Requirements gathering
2	Subsystem design
3	Subsystem design
4	Report for assignment 1: Milestone 1
5	Algorithm design
6	Coding: Milestone 2
7	Project assembly
8	Testing and debugging
9	Testing and debugging
10	Demonstration: Milestone 4

Figure 2: Project Management Tool

2 SIMULATION SETUP

Matlab was chosen to run the simulation due to the ease of access to various libraries. The signal processing toolbox in particular is used for cross-correlation to calculate the time difference of arrivals.
A Matlab script, *localize.m*, was written and can be found on the group Github repository. This script can be called in Matlab using *localize(x,y)*. The signals with time delay received by each microphone will be plotted, the hyperbolas used for triangulation will be plotted and the estimated source position will be outputted.

2.1 Rationale Behind Simulation Approach

The simulation aims to estimate the position of a sound source using 4 microphones. The rationale behind this is as follows:

TDoA Based Localisation: By measuring the time differences at which the sound of the source arrives at different locations i.e. the microphones it is possible to estimate the location of origin, This is based on the fact the sound waves travel at a constant speed and the time delay provided by the GCC function that utilizes the signal processing toolbox provides information about the sources distance from each microphone and the distances of the microphones from each other.

Hyperbola Intersections: The use of hyperbolas for intersection is a method used in TDOA localization. It uses the fact that the difference in TDoA values from different microphones can be used to find the difference in distance between the source and each of the microphones compared to a reference microphone. By using the definition of hyperbolas: The locus of points whose difference in distance between two fixed foci is a constant value. Using this definition we can estimate the source position by solving for the intersection of these hyperbolas.

2.2 Simplification-Assumptions Made

- The simulation uses the ideal sinusoidal signal from the source and received by the microphones. In reality, the signals can be affected by noise which can impact the accuracy of the results.
- It is assumed that there are no obstacles or surfaces in the way that could cause reflection or cause further delays in the reception of the signals.
- Speed of sound is said to be constant at 343m/s
- White noise is added to each signal to create a more realistic signal received.
- The simulation does not consider any attenuation of the signal
- The TDOA measurements are made under the assumption that the sound signal received travels on the shortest path to the target microphones. If this path is blocked then the microphones will not receive the signal and the TDOA measurements cannot be made, thus, we have made the assumption that the 'line-of-sight' path from the source to the microphones will not be blocked.

3 SYSTEM DESIGN AND IMPLEMENTATION

The simulation deals with signal acquisition, time delay estimation, and triangulation.

3.1 Signal Acquisition

A sinusoidal signal of a chosen frequency is defined as the source signal. The microphones are set to some specific coordinates as one would do in a real-life run. Four separate sinusoidal signals of the same frequency are defined, each with an appropriate time delay. The ideal time delay is calculated using the distance of the microphone from the source and the ideal speed of sound. Microphone 1 is set to be the reference microphone. White Gaussian noise is added to each signal.

3.2 Time Delay Estimation

A generalized cross-correlation function is defined as taking in the reference signal (the signal acquired by the reference microphone) and the signal received by another microphone. It then uses the xcorr() function to cross-correlate the two signals. The result is divided by the sampling frequency to acquire an estimated time delay. This gives the time delay on arrival from the reference mics to the other three mics.

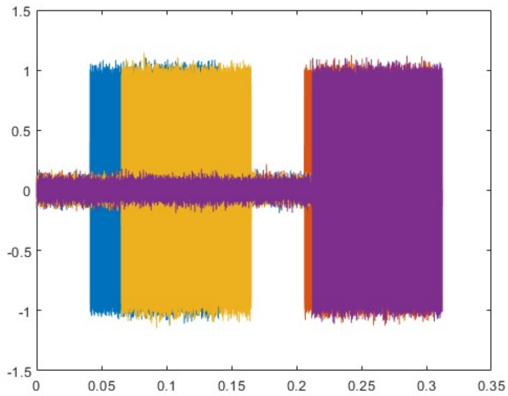

Figure 3: Signals received

3.3 Triangulation

By determining the time difference of arrival between each microphone we can find the difference in distance between the source and each subsequent microphone compared to a reference microphone. Using these differences in distances, we can form 3 equations:

$$t_{d12} \times 343 = \sqrt{(x-x_1)^2 + (y-y_1)^2} - \sqrt{(x-x_2)^2 + (y-y_1)^2} \tag{1}$$

$$t_{d13} \times 343 = \sqrt{(x-x_1)^2 + (y-y_1)^2} - \sqrt{(x-x_3)^2 + (y-y_3)^2} \tag{2}$$

$$t_{d14} \times 343 = \sqrt{(x-x_1)^2 + (y-y_1)^2} - \sqrt{(x-x_4)^2 + (y-y_4)^2} \tag{3}$$

These equations can be transformed into a hyperbolic form and will have three intersection points, forming a triangle. The estimated source position is taken as the center of this triangle.
The equations are solved using matlab's numerical solver, *fsolve()* to give the estimated position of the source.

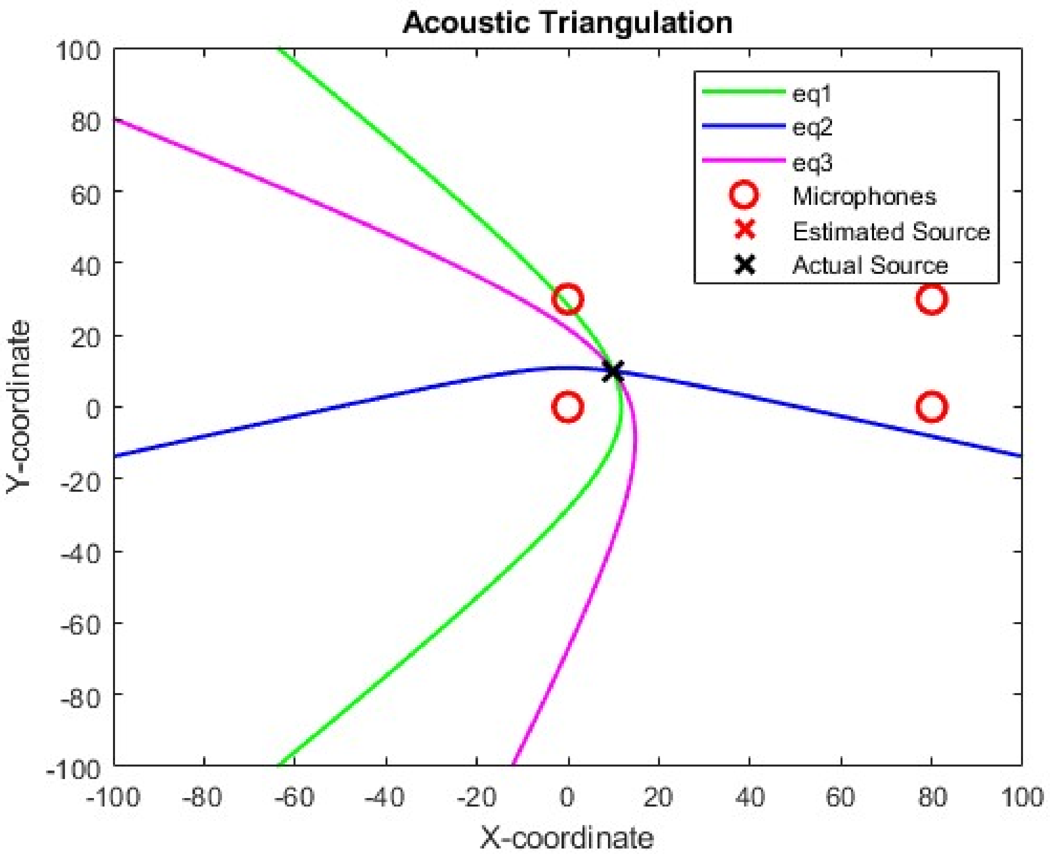

Figure 4: Acoustic localisation using TDoA. The source position (10,10) was estimated by finding the intersection of the hyperbolas.

3.4 System Architecture

Two microphones are connected in Stereo to the Raspberry Pis. The microphones acquire the sound and the signals are processed by the pis: Each Pi and a pair of microphones are connected as shown below:

Connections are as follows:

- Left Mic SEL to Pi GND
- Right Mic SEL to Pi 3.3V
- Mic BCLK to BCM 18 (pin 12)
- Mic DOUT to BCM 20 (pin 38)
- Mic LRCL to BCM 19 (pin 35)

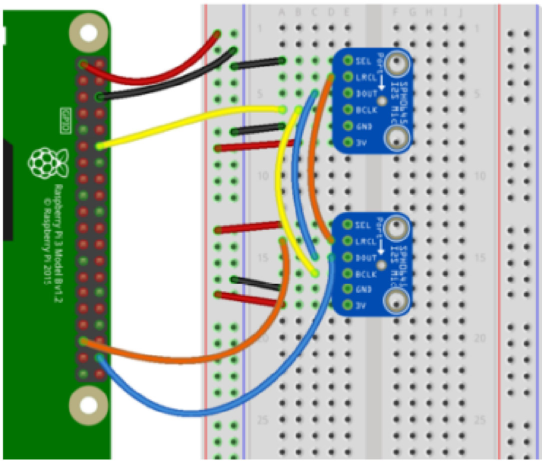

Figure 5: Stereo connection to Raspberry Pi

3.5 Sensor Network

The microphones are connected in Stereo to the Raspberry Pis. Each Pi therefore records 2 audio channels. The microphones are set up in the corners of the grid.
One Pi is designated to do all the signal processing, time delay calculations, and triangulation calculations as well as recording audio for 2 of the microphones. The other Pi will just record audio for 2 of the microphones and send this data to the other Pi to be processed.

4 SIMULATION RESULTS AND ANALYSIS

5 EVALUATION

ATP	Stated Acceptable Performance	Discussion
Position Accuracy Test	• Mean position deviation within ±2cm. • 80 % of positions within ±5cm.	Most estimated positions are within less than 1 cm from the specified source position
Sound Detection and SNR	• The microphones are able to pick up sound from 100 - 10 000 Hz. • The SNR is high. • Minimal effect of noise on position accuracy.	The maximum SNR for the microphones is 50dB. The microphones have been simulated so as to pick up any frequency however this is not telling of the limitations of the microphones.
Usability Testing of GUI	• Users should be able to easily navigate and perform tasks using the GUI without confusion.	The current GUI from MATLAB allows the user to vary the source position and run the program. It is therefore simple and easy to understand
Unit tests for the Raspberry Pi code	• The code runs without errors	Since we are at the simulation stage there was no direct testing of the code on the Raspberry Pis.
Synchronization Stability and Accuracy and synchronization	• Error should remain within the specified range (e.g., within a few milliseconds) consistently.	The simulation stage does not involve the Raspberry Pi hence this ATP could not be tested for.

The next step for this project is to test if the Matlab code works on the Raspberry Pis. Following this, we can begin to test a physical implementation of the scenario by setting up the microphones in the specified grid positions with a sound source inside the grid. We can then test our implementation of the signal acquisition, signal processing, TDoA calculations, and Triangulation algorithms in order to test and verify the accuracy of our design.

6 CONCLUSION

From our research, we determined that we could determine the location of the source position using TDoA by calculating the difference in distance between each microphone and the source compared to a reference microphone, thus forming 3 hyperbolic functions whose intersections can be solved to determine the source position. From our simulations, our algorithm estimates the source's position with very good accuracy. To account for more realistic signals, we added Gaussian noise to the signals and observed that there was no noticeable effect on the source's position estimation.

We have also realized that because we are controlling the signal that we are sampling from, we might experience some non-ideal behaviors in real life, the sound from the speakers will not have a perfect 360° wave propagation and this might affect how the microphones will receive the sound from the source. This insight has given us insight into the importance of the physical testing of the system.

References

[1] How to locate an unknown point using the tdoa method of trilateration, mathematics stack exchange. https://math.stackexchange.com/questions/3367330/how-to-locate-an-unknown-point-using-the-tdoa-method-of-trilateration. Accessed: 2023-09-10.

[2] i2s datasheet. https://cdn-shop.adafruit.com/product-files/3421/i2S+Datasheet.PDF. Accessed: 2023-08-17.

[3] i2s microphone breakout pdf. https://cdn-learn.adafruit.com/downloads/pdf/adafruit-i2s-mems-microphone-breakout.pdf. Accessed: 2023-08-17.