University of Cape Town
Department of Electrical Engineering
Cape Town, South Africa

EEE3097S 2023 ASSIGNMENT 3: SECOND PROGRESS REPORT

ACOTRIANGULATOR

Travimadox Webb
WBBTRA001@myuct.ac.za

Rumbidzai Mashumba
MSHRUM006@myuct.ac.za

Best Nkhumeleni
NKHBES001@myuct.ac.za

15/10/2023

Contents

List of Figures

List of Tables

1. Introduction

Sound triangulation, the process of determining the precise location of a sound source based on its acoustic signals, has been a subject of significant interest and innovation in the fields of acoustics, engineering, and technology. In our pursuit of harnessing the potential of sound triangulation, this report outlines our journey in designing and constructing a robust and accurate sound triangulator.

This scientific endeavor involved a multifaceted approach, combining elements of hardware, software, and innovative problem-solving. Our methodology included the strategic use of Raspberry Pis, microphones, and SSH technology, to enable remote data collection and triangulation. Throughout this process, we encountered a series of challenges that required creative solutions and adjustments.

In this report, we present a comprehensive overview of the methods employed in building the sound triangulator, delve into the specific challenges we faced during the development process, and provide detailed insights into the solutions we devised to address these obstacles. Furthermore, we shed light on the reasons that led us to adopt the particular design we ultimately selected, emphasizing its advantages and the impact it has on the accuracy and reliability of sound triangulation.

This report serves as a testament to the convergence of technology and scientific innovation, highlighting the potential of sound triangulation systems in various real-world applications. Our research journey encapsulates the determination and ingenuity required to overcome obstacles in pursuit of technological advancement and precision in sound source localization.

2. Admin Documents

2.1 Individual Contributions

Table 1: Contributors Table

Name	Contribution	Percentage
Best Nkhumeleni	Building GUI	100%
Rumbidzai Mashumba	Report writing	100%
TraviMadox Webb	System Setup	100%

2.2 Link to Github

Our git repo can be found at: https://github.com/Travimadox/AcoTriangulator/tree/main

2.3 Project Management Tool

We are using the Monday.com Project Management Tool as shown below:

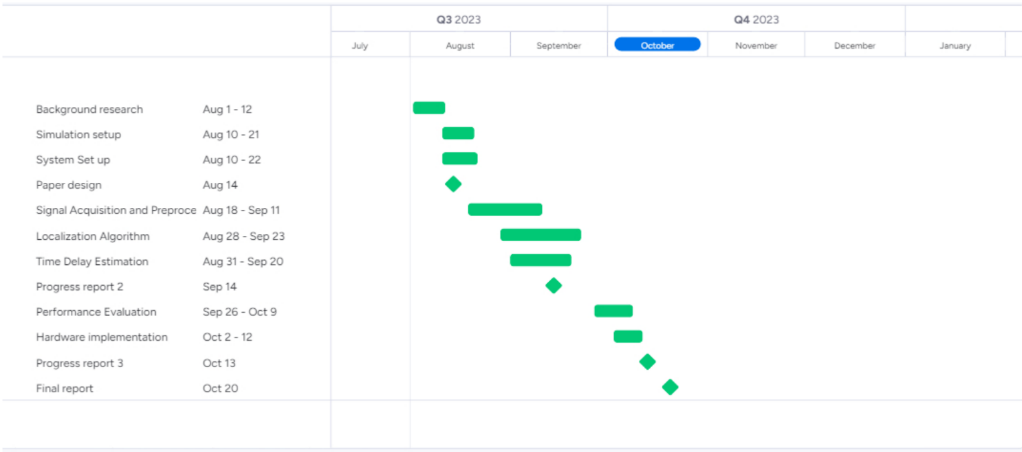

Figure 1: Project Management Tool

2.4 Timeline and Progress

We report that progress has been made on multiple fronts within the project. The Time Delay Estimation and Performance Evaluation tasks are all complete as shown below

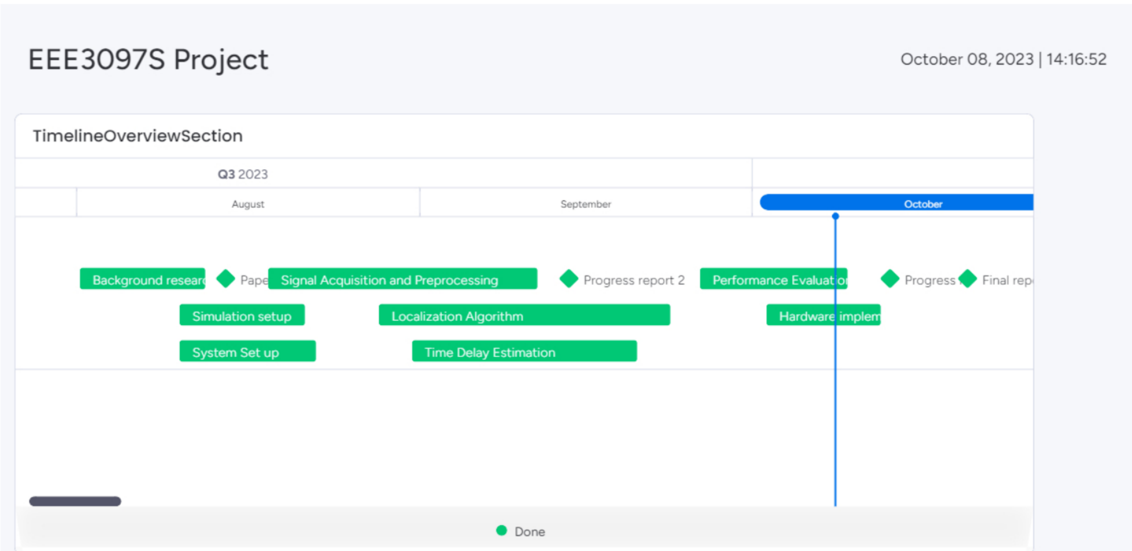

Figure 2: Timeline and Progress

3. System Implementation

AcoTriangulation is an acoustic triangulation system engineered to precisely determine the coordinates of a sound source within a defined rectangular grid. It employs the Time Difference of Arrival (TDoA) method by using strategically placed microphones to achieve its primary objective. Implementing the AcoTriangulator system involves successfully integrating systems in harmony to achieve precise results. The following text dissects and explores each individual subsystem.

3.1 Data acquisition

Data acquisition is the process of collecting, measuring, and recording data from various sources, such as sensors, instruments, devices, or computer systems. It involves the conversion of real-world physical phenomena or analog signals into digital data that can be stored, processed, and analyzed by computers. Data acquisition systems typically consist of hardware and software components designed to capture, process, and store data.

In this report, microphones will be used to acquire data from the surrounding environment which will then be processed by the Raspberry Pi

3.1.1 Objective 1: Accurate Audio Data Capture

The primary objective is to capture audio data from the four microphones accurately, ensuring that the data remains free from corruption or any quality degradation. This approach will enable us to obtain highly precise position readings.

3.1.2 Implementation 1:

We utilize SSH to access the Raspberry Pi, where we execute scripts that command the Raspberry Pi to record audio signals. These recordings are carried out with precise settings, specifying parameters such as duration, sample rate, and frequency. The audio signals are subsequently saved as WAV files.

3.1.3 Challenges and modifications

We observed that audio data captured sometimes suffered from noise. To address this issue used a software filter, that isolates our sound of interest.

It was challenging to ensure consistent audio quality across all microphones. We carefully calibrated each microphone to address this by using a calibration sound. Thorough testing was also conducted to identify hardware-related issues, like loss of connections.

3.1.4 Objective 2: Continuous Data Acquisition

Our intention is to continuously record data using the microphones. This continuous data collection serves two purposes: first, it ensures the acquisition of accurate readings, and second, it allows us to update the system promptly as changes occur.

3.1.5 Implementation 2:

The 'FullAcotriangulator.py' script employs SSH to access the Raspberry Pi and conducts a brief sound check. If all microphones are active and prepared, it proceeds to issue a command to record the surrounding sounds. This 'FullAcotriangulator.py' script is invoked within our graphical user interface (GUI) each time we refresh it, allowing us to obtain new environmental readings and subsequently a new approximate location reading.

3.1.6 Challenges and modifications

- Data synchronization: Using multiple acquisition devices proved to be a challenge when ensuring that data was synchronized. To solve this a calibration sound was introduced to the system which allowed us to sync the audio data.

- Data security: Ensuring data is transferred securely over networks was a bit of a challenge but this was easily resolved by using SSH for data transfer.

3.1.7 Objective 3: Remote Data Collection

Our original plan was to use SSH to remotely access and retrieve data from the Raspberry Pi without the need for a physical, wired connection to the device.

3.1.8 Implementation 3:

Using SSH commands in the pi1() and pi2() functions audio recording on the Raspberry Pis could be started remotely. This allowed for audio data acquisition to be triggered and controlled from a central location, enabling remote data collection without the need for physical access to the Pis.

3.2 Time Delay Estimation (TDoA) Subsystem

This critical subsystem plays a pivotal role in achieving accurate sound localization within the designated rectangular grid.

3.2.1 Objective 1: Accurate Sound Source Localization

To accurately calculate the time differences between sound signals received by multiple microphones. This is a very important step for the localization of the sound source.

3.2.2 Implementation 1:

The Acotriangulator.m script is responsible for TDOA calculation

Signal Extraction: Multiple microphones record audio signals which are then read by the code. Due to the microphones' spatial separation, they have a different time delay.

Time Delay Estimation: GCC-PHAT is employed to compute the time delays based on the phase information of the audio signals.

Conversion to Spatial Delays: The calculated time delays are then converted to spatial delays by multiplying them by the speed of sound (c). This is a necessary step for mapping the time-based information to physical distances within the grid.

Noise Reduction and Signal Processing: A bandpass filter is applied to each of the signals from the mics to isolate the desired frequency range. The signals are zero-padded to ensure consistent length.

Validation of Estimated Position: The estimated position is validated to ensure that it falls within the predefined grid boundaries. If not an error message is displayed boundaries

This implementation effectively combines signal processing, time delay estimation, and spatial conversion techniques to provide accurate and reliable sound source localization within the specified area.

3.2.3 Challenges and Modifications

Grid Boundary Handling: Making sure that estimated positions fell within the set boundaries was important. We refined our validation logic to handle boundary cases more effectively.

3.2.4 Objective 2: Visualization

To provide data on sound source location within the defined rectangular grid.

3.2.5 Implementation 2:

Visualization: AcoTriangulator$_{t}isresponsibleforthevisualizationofthesoundsource.Thecodegeneratesalookuptablean$

The visualization is done by reading the estimated points and plotting them using Matplotlib.

3.2.6 Challenges and Modifications:

Data Synchronization: Ensuring that audio data acquisition and real-time processing remained synchronized was crucial. We improved data synchronization by adding a calibration sound to minimize delays and discrepancies.

User interaction: We identified that the user might have difficulty navigating the system and interpreting the results. To address this issue a UI was implemented making it easier for users to control the system and understand the outcome.

3.3 Triangulation Subsystem

The aim of the simulation of this subsystem is to assess the accuracy of source position estimation using triangulation techniques. The environment is a 594 mm $\times$ 841 mm A1 grid, with microphones located at each corner.

Triangulation, as employed by this subsystem, leverages the TDoA of sound signals at multiple microphones to calculate the source's position.

3.3.1 Objective 1: Precise Sound Source Localization

The primary objective of this sub-system is to calculate the precise location of a sound source within a defined space accurately. This objective is centred on determining the source's coordinates with a high level of accuracy.

3.3.2 Implementation 1: Non-Linear Squares Estimation

Non-Linear Squares Estimation: The code formulated a non-linear least squares optimization to estimate the sound source within the grid. This is done by the optimization stating an initial guess. The algorithm then iteratively refines the estimated source position. This process minimizes the the differences between the expected (based on estimated position) and observed data (time delays). The refinement continues until convergence criteria are met.

3.3.3 Challenges and Modifications:

Initial Guess Enhancement: The accuracy of the initial guess plays a crucial role in the convergence process. To address this challenge, the algorithm has been enhanced to provide more informed and accurate initial guesses. This improvement contributes to faster convergence and more precise sound source localization.

3.4 User Interface

The User Interface (UI) plays a pivotal role within the AcoTriangulator system, serving as the primary interaction point between users and the system's functionalities. The UI in the AcoTriangulator system serves as a crucial medium for users to interact with and control various aspects of the system's operation. It encapsulates the complexity of sound source localization, data processing, and visualization within a user-friendly and comprehensible interface:

3.4.1 Tools for UI Development

Python Flask Framework Matplotlib HTML templates

3.4.2 Objective 1: Enhanced Usability

To simplify user interactions and enhance the overall usability of the system ensuring that users can easily initiate tasks regardless of their expertise.

3.4.3 Implementation 1: Graphical Interface

Graphical Interface: The UI provides users with buttons and menus that minimize ambiguity and provide clear indications of their functions.

Task Initiation: Tasks can be initiated with the simple click of the start button which when clicked triggers the "FullAcotriangulation.py" script.

3.4.4 Objective 2:User-Centered Design

User-centred UI design that ensures a straightforward and visually appealing interface.

3.4.5 Implementation 2: User-Centered Design

User-centred design: The UI follows user-centred design principles, prioritizing the needs and preferences of the system's users aiming to provide a seamless and intuitive experience.

Simplicity and Clarity: The UI features a clean layout, clear labels and minimalistic elements making sure users are not overwhelmed with unnecessary information.

Graphical Feedback: The UI provides graphical feedback in the form of a dynamic plot. It displays the acoustic location grid, enhancing the visual representation of the system's operation.

4. Experimental Setup

4.1 Overview

The experimental setup for the AcoTriangulator system involves validating its various subsystems, including Data Acquisition, TDoA, Triangulation, and User Interface. The primary objective is to ensure that each subsystem functions as expected and contributes to accurate sound source localization.

4.2 Equipment and Components

4.2.1 Microphones:

Four microphones positioned at each corner of the A1 grid, are used to capture sound signals.

4.2.2 Raspberry Pis :

Two Raspberry Pi computers were deployed to interface with the microphones. They play a crucial role in recording and communicating with the central system.

4.2.3 Sound Source:

A controlled sound source emitting a known acoustic signal is employed to simulate sound events within the environment.

4.2.4 MATLAB:

MATLAB is used for signal processing and nonlinear optimization.

4.2.5 Flask Web Application:

This forms the UI component of the systems allowing users to initiate system tasks.

5. Data Collection And Analysis

5.1 Procedure for Collecting Acoustic Signals and TDOA Measurements:

5.1.1 Initialization:

The experimental setup is configured, including the positioning of microphones (M1, M2, M3, M4) within the A1 grid. The system is prepared for data collection.

5.1.2 Sound Source Activation:

The controlled sound source is activated to emit an audio signal. This serves as the sound source of the system.

5.1.3 Data Acquisition:

This process is facilitated by the two Raspberry Pi devices connected to the microphones. The audio is recorded by each microphone.

5.1.4 TDoA Calculation:

The TDoA, calculated by the triangulation subsystem using GCC-PHAT, is determined by comparing the arrival times of the same sound event at different microphones.

5.1.5 Spatial Delay Conversion:

The TDOA values are converted into spatial delays (delta-t values) by multiplying them by the speed of sound (c). Each delta-t represents the difference in the distance travelled by the sound wave to reach each microphone.

5.1.6 Nonlinear Optimization:

The Nonlinear Optimization subsystem uses the delta_t values to estimate the precise location of the sound source within the A1 grid. It formulates an optimization problem, which is solved to obtain the estimated source position.

5.2 Recording and Storing Data for Analysis:

5.2.1 Audio Data:

Recorded audio data is saved as digital audio files in WAV format.

5.2.2 TDoA and Spatila Delay

The calculated TDOA values and their corresponding spatial delay (delta-t) values are logged for each microphone pair.

5.2.3 Estimated Position:

Using Non-linear Optimization the position of the sound source is estimated.

6. Results

6.0.1 Introduction

This section of the report provides a thorough analysis of a Time Difference of Arrival (TDOA) system designed for locating acoustic signals within a given grid. The primary objective is to evaluate the system's accuracy and precision in estimating the position of acoustic sources.

The analysis was organized into the following four steps:

- Error Computation
- Statistical Analysis
- Visual Insights
- Correlation with Source Position

6.1 Time Difference Of Arrival Of the System

6.1.1 Error Computation

This step aims to quantize the error between the estimated TDoA and the measured TDoA. This is a crucial step because it helps us understand the experiment's success. It also helps us visualize the extent of deviation between the measured value and the estimated theoretical value.

Formula:

The absolute error for each sensor pair was computed using the following formula:

Absolute Error for TDOA pair = Theoretical TDOA -Measured TDOA

6.1.2 Statistical Analysis

This analysis involved finding the:

- Mean Error: average error for all sensor pairs.
- Physical implementation Standard Deviation: a measure of the dispersion of the error values.
- Physical Implementation Range Error: maximum and minimum error variations

Mean error:

The average error for the sensors in the Physical implementation is as follows:

- Mean Error: 0.219 m

  The average error for the sensors in the simulation is as follows:

- Mean Position Estimation Error: 4.9×10^{-8} mm

The presented values represent the average error in the measurements made by different microphone pairs when determining the time difference of arrival. From the values stated above it is clear to see that the average errors produced by the Physical implementation are not negligible. This suggests moderate accuracy in the system's performance. Whereas the mean error of the simulation is only 4.9×10^{-8} mm. The simulation has a much higher accuracy than the physical implementation.

Standard Deviation

Physical implementation measure of the dispersion of the error values :

- Standard Deviation of Error in X: (0.123) meters
- Standard Deviation of Error in Y: (0.058) meters.

Simulation measure of the dispersion of the error value :

- Standard Deviation of Error : (1.23×10^{-9}) meters

Standard deviation here is a measure of how much the individual error values deviate from the mean error values. In the context of TDOA the smaller the standard deviation the higher the precision. Looking at the physical implementation, the X coordinate standard deviation of the error is 0.123 meters, indicating significant variability and the Y coordinate standard deviation is (0.058) meters, suggesting relatively consistent estimations. On the other hand for the simulation, the Standard Deviation of Error is only 1.23×10^{-9} meters. This shows that the simulation is much more precise than the physical implementation.

Reasons for Discrepancies:

In the simulation, signal processing was implemented with high precision, while real-world implementations were subject to limitations in processing power and computational accuracy. This is backed up by the fact that real-world environments are subject to various environmental conditions, which can affect the speed of sound and lead to errors in TDOA calculations. This led to the differences in Standard deviation and mean error values.

6.1.3 Visual Insights

Histograms

The histograms illustrate how the errors are distributed for each sensor pair.

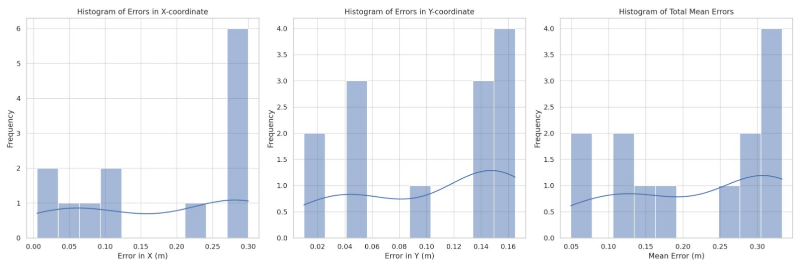

Figure 3: Error distribution of the physical implementation

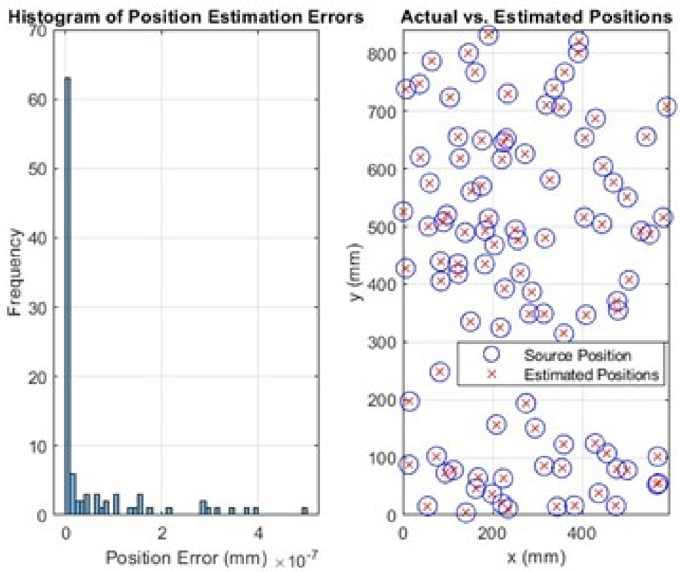

Figure 4: Error distribution of the simulation

Histogram Analysis:

1. Histogram of Errors in X-coordinate:

   Most of the errors in the X-coordinate seem to be around 0.1 to 0.3 meters.

2. Histogram of Errors in Y-coordinate:

   Most of the errors in the Y-coordinate are concentrated around 0.05 to 0.15 meters.

3. Histogram of Total Mean Errors:

The total mean errors appear to be primarily within the range of 0.1 to 0.35 meters
Using Figure 5.2 above we observe that there overall error of the simulation was very minimal as there is a greater frequency of readings where there were no errors entirely unlike that of the Physical implementation where a majority of the mean errors lie beyond 0.3m.

6.1.4 Correlation with Source Position

Scatter Plots

Scatter plots were generated to investigate if the errors correlate with the source positions ((Source-X) and (Source-Y)).

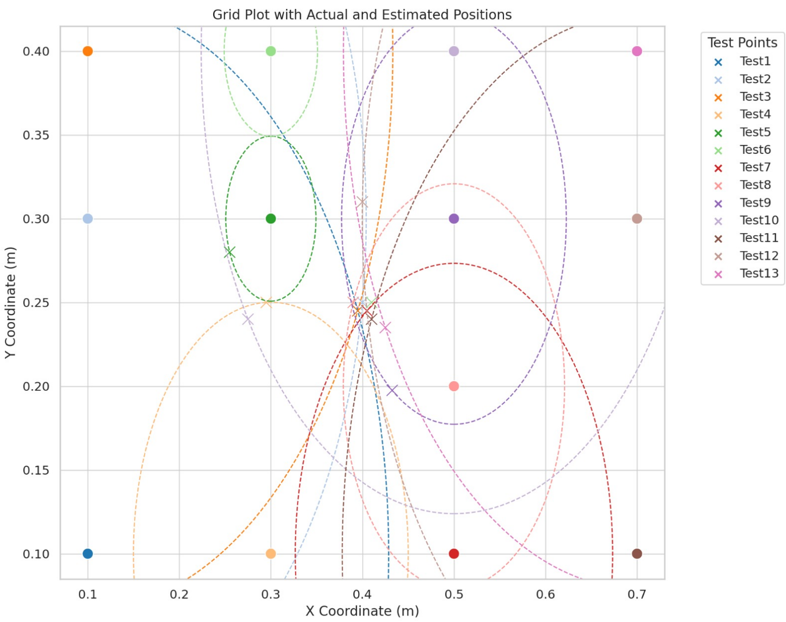

Figure 5: Scatter plots

The grid plot visualizes the actual and estimated positions. Dashed circles centered at the source positions have a radius equal to the mean error for each test. This representation provides a spatial understanding of the system's accuracy and precision.

6.1.5 Discussion

Performance in Different Scenarios

The system's performance might be influenced by several factors:

1. Signal Strength: Stronger signals usually allow for more accurate position estimations.
2. Noise Level: Elevated levels of noise can negatively affect the accuracy of position estimations.
3. Sensor Configuration: The geometry and arrangement of sensors can also significantly impact performance.

6.1.6 Accuracy and Precision Assessment

- Accuracy: The mean error of (0.219) meters indicates moderate accuracy but this accuracy is much poorer than that of the simulation.
- Precision: The standard deviations of errors in X and Y coordinates suggest decent precision but this precision is much poorer than that of the simulation.

1. 7. Evaluation Against ATPs

7.1 Recreated ATPs

The experiment primarily focused on the Time Delay of Arrival (TDOA) and the Positions calculation of the Acotriangulator System. as outlined in the original ATP.

7.2 Table: Evaluation of Results

7.2.1 Sound Emission SubSystem

Objective:

Validate the audible signal generation capability across specific frequencies.

Test Procedure:

1. Use a calibrated sound frequency metre to validate the emitted sound frequency. Test at multiple points within the range of 20 Hz to 20,000 Hz.
2. Measure the sound volume at a 1-metre distance using a decibel metre.

Test no	Pass/fail
1	Pass
2	Pass

Table 2: Sound Emission.

Expected Results:

1. The emitted sound is within the specified frequency range.
2. The volume at 1 metre is at least 70 dB.

7.2.2 Hardware Setup and Management Subsystem

Objective:

Ensure the correct installation and optimal layout configuration.

Test Procedure:

1. Perform a visual inspection to check the installation of Raspberry Pis and microphone breakout boards.
2. Confirm the layout formation of microphones.

Test No.	Inspection Checkpoint	Pass/Fail
1	Raspberry Pi installation	Pass
2	Microphone breakout board installation	Pass
3	Microphone layout confirmation(Square format)	Pass

Table 3: Hardware Setup and Management

Expected Results:

1. All hardware components are installed and functioning correctly.
2. Microphones are arranged in a square formation.

7.2.3 Audio Acquisition and Processing Subsystem

Objective:

Validate the system's ability to capture, digitise, and filter audio signals.

Test Procedure:

1. Record audio signals and analyse the digital output.
2. Introduce background noise and test the system's noise-filtering capability.

Test No.	Pass/Fail
1	Pass
2	Pass

Table 4: Audio Acquisition and Processing

Expected Results:

1. Accurate digital representation of the captured audio.
2. An SNR of more than 60 dB is maintained even in noisy conditions.

7.2.4 Time Synchronisation Subsystem

Objective:

Test the synchronisation of audio signals between Raspberry Pis.

Test Procedure:

1. Play audio signals simultaneously on different Raspberry Pis and measure synchronisation errors.
2. Introduce a known drift in one Raspberry Pi clock and observe system compensation.

Test No.	Pass/Fail
1	Pass
2	Pass

Table 5: Time Synchronization

Expected Results:

1. The synchronisation error is less than 1 ms.
2. The system compensates for the known clock drift correctly.

7.2.5 TDoA Calculation Subsystem

Objective:

Validate the accuracy of TDoA calculations.

Test Procedure:

1. Introduce test audio data and measure calculated TDoA.
2. Vary the position of sound sources and test TDoA calculations.

Test No.	Pass/Fail
1	Fail
2	Fail

Table 6: TDoA

Expected Results:

1. TDoA calculation error is less than 1 ms.
2. Accurate TDoA calculations for varying sound source positions.

7.2.6 GUI (Graphical User Interface) Subsystem

Objective:
Ensure an intuitive and responsive GUI display.
Test Procedure:

1. Feed the GUI with test position data and observe the display accuracy.
2. Simulate system failures and assess GUI error message displays.

Test No.	Pass/Fail
1	Pass

Table 7: GUI

Expected Results:

1. Sound source positions are displayed correctly in real time.
2. System failures are accurately reported via GUI error messages.

7.2.7 System Testing and Validation Subsystem

Objective: Comprehensive testing and validation of the system.
Test Procedure:

1. Conduct end-to-end system tests to validate all components.

Test No.	Subsystem	Pass/Fail
1	Hardware Setup and Sound Emission Subsystem	Pass
2	Data Acquisition and Pre-processing Subsystem	Pass
3	Data Transmission and Reception Subsystem from the Raspberry Pi Zero	Pass
4	Data Analysis and Localization Subsystem	Fail
5	User Interface Subsystem	Pass

Table 8: System Testing and Validation

Expected Results:

1. All components pass their respective ATPs.

7.3 Analysis

7.3.1 ATPs Met:

- Data Acquisition accurate (5A): This subsystem was successfully carried out. All the data collected was accurate and properly stored for processing.
- Software and Hardware integration (5b): Ensuring seamless integration of all subsystems was successful. All subsystems worked together harmoniously
- Ease of Use(5c): Successful incorporation of a user interface allowing the system to be used with ease.
-

7.3.2 ATPs Not Met:

- TDoA Calculation (5A): The simulation did not successfully validate the TDoA calculation accuracy. The mean error and standard deviation of the results are too significant to ignore.
- Position Calculation (5B): Unfortunately the Position Calculation subsystem did not show exceptional accuracy as the mean error was quite significant.

8. Conclusion

In conclusion, the physical implementation did not yield the results that were hoped for. While the simulation demonstrated great promise, the real-world deployment faced numerous challenges. Several factors contributed to this discrepancy, which were not fully accounted for in the simulation. However, while the physical implementation faced significant challenges, it offers valuable insights for further refinement. The promising results obtained in the simulation provide a clear path forward, emphasizing the need for continuous improvement to bridge the gap between theory and practice. By addressing the identified issues and implementing modifications, the system can ultimately achieve the precise sound source localization that is envisioned.