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

EEE3097S 2023 ASSIGNMENT 4: FINAL REPORT

ACOTRIANGULATOR

Travimadox Webb
WBBTRA001@myuct.ac.za

Rumbidzai Mashumba
MSHRUM006@myuct.ac.za

Best Nkhumeleni
NKHBES001@myuct.ac.za

23/10/2023

Contents

1. Introduction 1

2. Admin Documents 2
2.1 Individual Contributions 2
2.2 Link to Github 2
2.3 Project Management Tool 2
2.4 Timeline and Progress 3

3. Requirement Analysis 4
3.1 Requirement Discussion 4
3.2 Analysis of Requirements 4
3.2.1 Sound Emission 4
3.2.2 Hardware Installation 4
3.2.3 Audio Acquisition and Processing 4
3.2.4 Data Transmission 4
3.2.5 Data Analysis and Localization 4
3.2.6 Graphical User Interface 5
3.2.7 System Validation 5
3.3 Final List of Requirements 5
3.4 Specifications 6
3.5 Acceptance Test Procedures(ATP) 6
3.5.1 ATP001 & ATP002 :Sound Emission 6
3.5.2 ATP003: Hardware Setup and Management Subsystem 6
3.5.3 ATP004: Audio Acquisition and Processing 7
3.5.4 ATP005:Data Transmission and Reception 7
3.5.5 ATP006: TDoA Calculation 7
3.5.6 ATP007: Position Calculation 8
3.6 ATP008:GUI (Graphical User Interface) 8
3.7 ATP009: System Integration 9

4. Paper Design 10
4.1 System Design 10
4.2 Data Acquisition and Pre-processing Subsystem 10
4.2.1 Sound Emission 10
4.2.2 Hardware Setup and Management 11
4.2.3 Time Synchronisation 11
4.2.4 Audio Acquisition on Raspberry Pi Zero 11
4.2.5 Noise Handling 12
4.2.6 Data Transmission 12
4.2.7 Data Reception 12
4.3 Time Delay Estimation Subsytem(TDOA) 12
4.4 Triangulation Susbsystem 13
4.5 User Interface 13

5. Validation Using Simulations 14
5.1 Importance of Simulation 14
5.2 Simulation Setup 14
5.2.1 Simulation Environment and tools used 14
5.2.2 Rationale Behind the Chosen Simulation Approach 15
5.2.3 Simplifications or Assumptions 15
5.3 System Design and Implementation 16
5.3.1 Time Delay Estimation (TDoA) Subsystem 16
5.3.2 Triangulation Subsystem 16
5.3.3 Sound Localization Simulation 17
5.4 System Architecture 18

5.5 Distributed Sensor Network Structure . . . 18
5.6 Algorithms/Techniques for TDOA Calculation . . . 18
5.6.1 Time Delay Calculation with `gccphat` . . . 18
5.6.2 Conversion to Spatial Delays . . . 18
5.6.3 Nonlinear Least-Squares Estimation with `lsqnonlin` . . . 19
5.7 Results . . . 20
5.7.1 Time Delay Estimation . . . 20
5.7.2 Triangulation . . . 21
5.7.3 System as a Whole . . . 21
5.7.4 SNR Sensitivity Analysis Using `SNRAwareLocalization.m` . . . 21
5.7.5 Targeted SNR Performance Evaluation Using `AdvancedAcousticLocalization.m` 22
5.7.6 Key Performance Indicators for Acoustic Source Localization . . . 23

6. Validation Using Final Implementation 24
6.1 Importance of Hardware-Based Implementation . . . 24
6.2 System Set up and Implementation . . . 24
6.2.1 Sound Emission . . . 24
6.2.2 Hardware Set-Up . . . 24
6.2.3 Time Synchronisation . . . 25
6.2.4 Audio Acquisition . . . 25
6.2.5 Noise Handling . . . 25
6.2.6 Time Delay Estimation (TDOA) . . . 25
6.2.7 Triangulation . . . 26
6.2.8 Error Calculation and Validation . . . 26
6.2.9 User-Interface . . . 26
6.3 Experimental Setup . . . 27
6.4 Results . . . 28
6.4.1 Descriptive Statistics . . . 29
6.4.2 Accuracy and Precision Analysis . . . 29

7. Consolidation of ATPs and Future Plan 30
7.1 ATP Validation . . . 30
7.2 Analysis of ATPs . . . 30
7.2.1 Sound Emission: ATP001 & ATP002 . . . 30
7.2.2 Hardware Setup and Management: ATP003 . . . 30
7.2.3 Audio Acquisition and Processing: ATP004 . . . 30
7.2.4 Data Transmission and Reception: ATP005 . . . 30
7.2.5 TDoA Calculation: ATP006 . . . 30
7.2.6 Position Calculation: ATP007 . . . 30
7.2.7 GUI: ATP008 . . . 31
7.2.8 System Integration: ATP009 . . . 31
7.2.9 Summary: . . . 31
7.3 Consolidation of ATPs and Future Plan . . . 31
7.4 Future Work . . . 31

8. Conclusion 33

9. Acknowledgment 34

References 34

Appendix A Appendix 35
1.1 Code . . . 35

Appendix B ATP001 & ATP002 :Sound Emission 1

Appendix C ATP003: Hardware Setup and Management Subsystem 1

Appendix D ATP004: Audio Acquisition and Processing **1**

Appendix E ATP005:Data Transmission and Reception **2**

Appendix F ATP006: TDoA Calculation **2**

Appendix G ATP007: Position Calculation **2**

Appendix H ATP008:GUI (Graphical User Interface) **3**

Appendix I ATP009: System Integration **4**

List of Figures

1 Project Management Tool 2
2 Timeline and Progress 3
3 System Overview Block Diagram 10
4 Hardware Block Diagram 10
5 Sound Emission 11
6 Microphone Layout 11
7 Histogram 20
8 Histogram and Position Plot 21
9 SNR Plot 22
10 5 dB Performance 22
11 60 dB Performance 23
12 Hardware Set-Up 25
13 Acotriangulator GUI 27
14 AcoTriangulator Test Data 28

List of Tables

1 Contributors Table . . . 2
2 AcoTriangulator Specifications . . . 6
3 ATP001 & ATP002 . . . 6
4 ATP003 . . . 6
5 ATP004 . . . 7
6 ATP005 . . . 7
7 ATP006 . . . 7
8 ATP007 . . . 8
9 ATP008 . . . 8
10 ATP009 . . . 9
11 AcoTriangular test results . . . 28
12 ATP Summary . . . 30
13 Consolidation of ATP Results . . . 31
14 ATP001 & ATP002 . . . 1
15 ATP003 . . . 1
16 ATP004 . . . 2
17 ATP005 . . . 2
18 ATP006 . . . 2
19 ATP007 . . . 3
20 ATP008 . . . 3
21 ATP009 . . . 4

1. Introduction

This document provides an in-depth report on the development of the Acotriangulator System, offering a comprehensive understanding of the project's intricacies. The report serves multiple critical purposes, each of which contributes to the holistic view of the system's development:

- **Examination of System Requirements**: In this section, we will dissect the specific requirements that drove the creation of the Acotriangulator System. This includes identifying the precise needs and constraints, ensuring that our design aligns seamlessly with real-world demands.
- **Specification Details**: We will delve into the meticulous specification of system components and functionalities. This includes defining the technical specifications of the microphones, the triangulation algorithm, and the user interface, ensuring a clear and efficient system architecture.
- **Creation of Acceptance Test Procedures (ATPs)**: Developing ATPs is vital to guarantee the system's functionality and performance. We will outline the rigorous testing procedures that the Acotriangulator underwent to meet and exceed predetermined standards.
- **Validation through Simulations and Real-World Testing**: The report will detail the validation process, which encompasses both simulated testing and real-world experiments. This dual approach allows us to evaluate the system's capabilities in controlled environments as well as under practical conditions.
- **Rationale Behind Design Choices**: This section will provide insights into the thought process behind selecting the specific microphone arrangement and triangulation algorithm. We will explain why these choices were made and how they contribute to the system's overall effectiveness.
- **Interconnection of Components**: We will elucidate the intricacies of how the system's components, including microphones, algorithms, and interface, are interconnected to ensure seamless communication and functionality.
- **Development of the User Interface**: The report will offer a comprehensive overview of the user interface's design and development process, detailing how it was tailored to enhance user experience and system usability.
- **Results and Path to Improvement**: The document will conclude with an analysis of the project's outcomes, including successes and challenges. We will explore potential avenues for improvement and how these results align with our initial project objectives.

This report is a testament to our commitment to transparency and innovation, shedding light on the journey of creating the Acotriangulator System and providing valuable insights into its design, testing, and potential enhancements.

2. Admin Documents

2.1 Individual Contributions

Table 1: Contributors Table

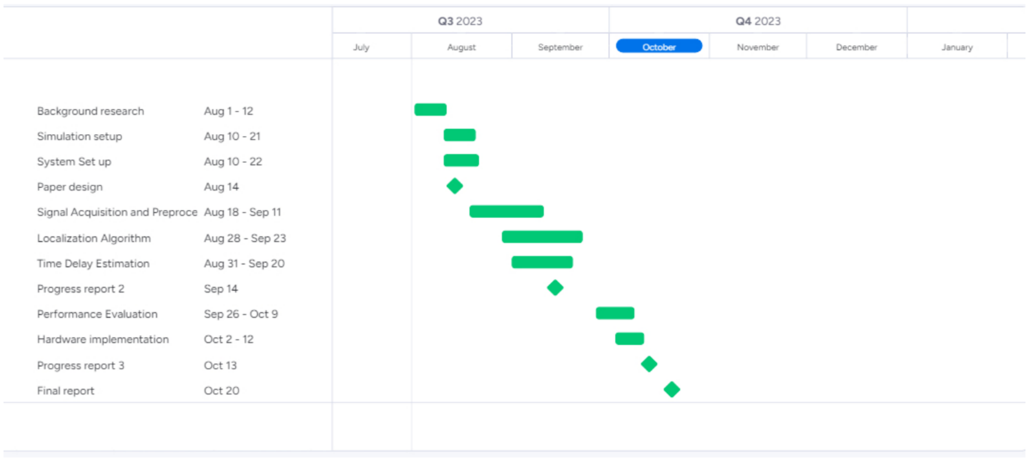
Name	Contribution	Percentage
Best Nkhumeleni	Report Writing	33%
Rumbidzai Mashumba	Results Analysis	33%
TraviMadox Webb	Simulation Setup and ATPs	33%

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:



Background research	Aug 1 - 12
Simulation setup	Aug 10 - 21
System Set up	Aug 10 - 22
Paper design	Aug 14
Signal Acquisition and Preproce	Aug 18 - Sep 11
Localization Algorithm	Aug 28 - Sep 23
Time Delay Estimation	Aug 31 - Sep 20
Progress report 2	Sep 14
Performance Evaluation	Sep 26 - Oct 9
Hardware implementation	Oct 2 - 12
Progress report 3	Oct 13
Final report	Oct 20

Figure 1: Project Management Tool

2.4 Timeline and Progress

We report that progress is being made on multiple fronts within the project. The Time Delay Estimation and Performance Evaluation tasks are ahead of schedule. Meanwhile, the Localization Algorithm is proceeding according to the established timeline as shown below:

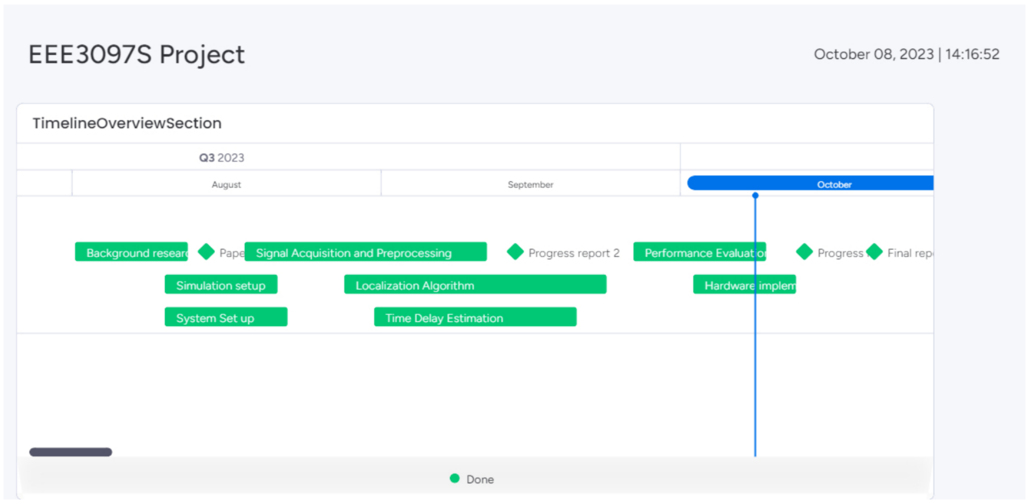


Figure 2: Timeline and Progress

3. Requirement Analysis

3.1 Requirement Discussion

1. **Sound Emission:**The system must be capable of emitting sound across a specified frequency range
2. **Hardware Installation:** Raspberry Pis and microphone breakout boards must be installed correctly and configured for optimal data acquisition.
3. **Audio Acquisition and Processing:** The system should be capable of capturing, digitizing, and filtering audio signals with a high Signal-to-Noise Ratio (SNR).
4. **Data Transmission:** The system must efficiently transfer audio data to a PC using without data loss or corruption.
5. **Data Analysis and Localization:** The system must accurately compute the Time Difference of Arrival (TDoA) and use it for sound source localization within the defined grid.
6. **Graphical User Interface:** The system must have a GUI that displays the sound source location in real-time and shows error messages in case of system failures.
7. **System Validation:** The entire system must undergo rigorous end-to-end testing to ensure all components and subsystems meet their respective ATPs.

3.2 Analysis of Requirements

3.2.1 Sound Emission

Acoustic localization systems rely on the emission of sound waves across a defined frequency spectrum to enable precise detection and localization of acoustic events. The effectiveness of these systems can be significantly impacted by the frequency range over which they operate. A paper discussing Time Difference of Arrival (TDoA)-based Acoustic Source Localization mentions an intelligent system for locating and classifying audio source signals, implying the importance of sound emission in such systems[1].

3.2.2 Hardware Installation

The installation and configuration of Raspberry Pis and microphone breakout boards are fundamental for optimal data acquisition in acoustic localization setups. Proper hardware installation ensures that the system can capture, digitize, and process audio signals effectively, thereby achieving a high Signal-to-Noise Ratio (SNR)[2].

3.2.3 Audio Acquisition and Processing

High-quality audio acquisition and processing are pivotal for the accurate localization of sound sources. A paper on TDoA-based acoustic emission source localization highlights the significance of capturing audio signals and processing them to obtain the location of the sound source using algorithms like the Firefly Algorithm and Gradient Descent Algorithm[3][4].

3.2.4 Data Transmission

Efficient data transmission to a PC without data loss or corruption is essential to ensure that the audio data captured and processed is accurately analyzed for sound source localization[3].

3.2.5 Data Analysis and Localization

Accurate computation of the Time Difference of Arrival (TDoA) and its utilization for sound source localization within a defined grid is crucial for the effectiveness of acoustic localization systems. Papers on TDoA-based Acoustic Source Localization emphasize the importance of accurate data analysis for sound source localization[5]**simon2022** [7].

3.2.6 Graphical User Interface

A user-friendly GUI that displays the sound source location in real time and provides error messages in case of system failures enhances the usability and troubleshooting capabilities of the system.

3.2.7 System Validation

Rigorous end-to-end testing to ensure that all components and subsystems meet their respective Acceptance Test Procedures (ATPs) is vital to validate the performance and reliability of the entire system.

3.3 Final List of Requirements

1. **Frequency Range:** 20 Hz to 20,000 Hz
2. **Volume:** At least 70 dB at a 1-meter distance
3. **Installation:** Hardware components must be correctly installed.
4. **Microphone Layout:** Square formation
5. **Signal to Noise Ratio (SNR):** More than 60 dB
6. **File Integrity:** No data loss or corruption during the transfer
7. **TDoA Accuracy:** Error less than 1 ms
8. **Position Accuracy:** Error less than 0.5 meters
9. **GUI Responsiveness:** Real-time display with error messages
10. **System Validation:** Pass all Acceptance Test Plans (ATPs)

3.4 Specifications

Requirement No.	**Specification**	**ATP Reference**
1	Frequency range for sound emission: 20 Hz - 20,000 Hz	ATP001
2	Minimum sound volume at 1-meter distance: 70 dB	ATP002
3,4	Hardware setup must include Raspberry Pis and microphone breakout boards in a square formation	ATP003
5	SNR for digitized audio must be greater than 60 dB	ATP004
6	Data transmission must occur without errors	ATP005
7	TDoA calculation error must be less than 1 ms	ATP006
8	Position calculation error must be less than 0.05 meters	ATP007
9	GUI must display sound source positions and system failures in real-time	ATP008
10	All subsystems must pass end-to-end tests	ATP009

Table 2: AcoTriangulator Specifications

3.5 Acceptance Test Procedures(ATP)

3.5.1 ATP001 & ATP002 :Sound Emission

To validate the audible signal generation capability across specified frequencies.

Procedure:

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

Test	Frequency	Measured Volume(dB)	Pass/Fail

Table 3: ATP001 & ATP002

Expected Results:

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

3.5.2 ATP003: Hardware Setup and Management Subsystem

To ensure 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 Check	Pass/Fail

Table 4: ATP003

Expected Results:

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

3.5.3 ATP004: Audio Acquisition and Processing

To 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	Signal Type	SNR(dB)	Pass/Fail

Table 5: ATP004

Expected Results:

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

3.5.4 ATP005:Data Transmission and Reception

To validate the efficient and error-free transfer of audio data from the Raspberry Pi to the PC using the scp command.
Test Procedure:

1. Use the scp command to transfer a sample audio file from the Raspberry Pi Zero to the PC.
2. Verify the reception and integrity of the transferred file

Test No	Test Action	Expected File Size	Received File Size	Pass/Fail

Table 6: ATP005

Expected Results: The transferred file is intact without any data loss or corruption.

3.5.5 ATP006: TDoA Calculation

To 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	Source Position	Estimated Position	Error	Pass/Fail

Table 7: ATP006

Expected Results:

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

3.5.6 ATP007: Position Calculation

To test the accuracy of sound source position calculations.

Test Procedure:

1. Use test TDoA data to calculate sound source position.
2. Introduce sound sources outside the microphone grid and evaluate position estimates.

Test No	Source Position	Estimated TDOA	Calculated TDOA	Error	Pass/Fail

Table 8: ATP007

Expected Results:

1. Position calculation error is less than 0.05 meters.
2. The system provides reasonable position estimates even for out-of-grid sound sources

3.6 ATP008:GUI (Graphical User Interface)

To ensure an intuitive, accurate, 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	Test Input Data	Gui Display(x,y)	Display Lag(ms)	Error Message	Pass/Fail

Table 9: ATP008

Expected Results:

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

3.7 ATP009: System Integration

To ensure comprehensive testing and validation of the system.

Test Procedure:

1. Conduct end-to-end system tests to validate all components.
2. Use built-in diagnostic tools to simulate various failures.

Test No	SubSystem	Identified Problem	Pass/Fail

Table 10: ATP009

Expected Results:

1. All components pass their respective ATPs.
2. Diagnostic tools accurately identify and report simulated problems.

4. Paper Design

4.1 System Design

The overarching aim of the Acoutriangulator system is to offer a precise, reliable mechanism for sound localization. This multifaceted system encompasses various sub-components, such as sound emission, hardware configuration, data acquisition, pre-processing, transmission, and sound source localization. Subsequent sections elucidate the functionalities of each subsystem and their interdependencies. Figure **??** and Figure 4 illustrate the System Overview and Hardware Block Diagrams, respectively.

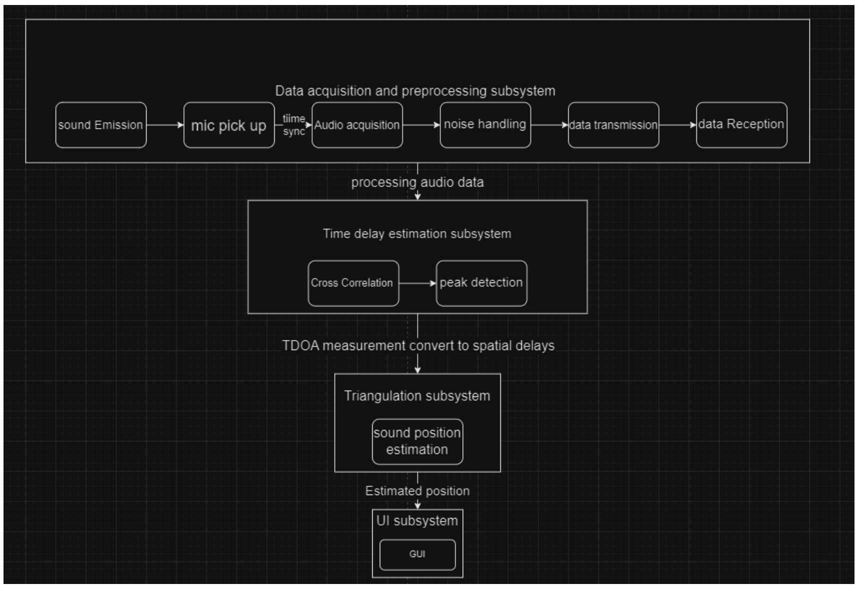


Figure 3: System Overview Block Diagram

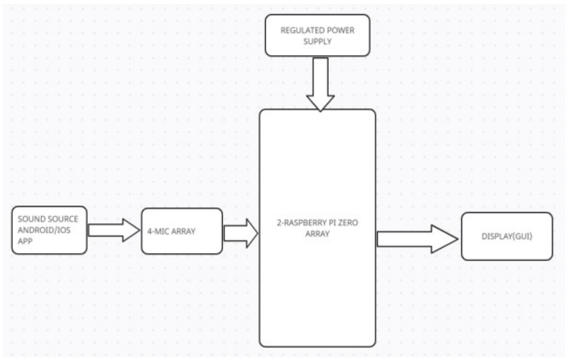


Figure 4: Hardware Block Diagram

4.2 Data Acquisition and Pre-processing Subsystem

4.2.1 Sound Emission

The mobile application of the Acoutriangulator system functions as the primary interface for controlling sound emission. Designed for ease-of-use, the application centralizes essential features required for effective system operation. A representative screenshot of the application is depicted in Figure 5.

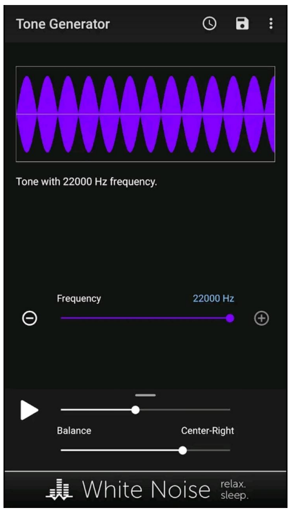


Figure 5: Sound Emission

4.2.2 Hardware Setup and Management

This aspect involves the strategic positioning of Raspberry Pi Zero devices and the appropriate arrangement of microphones. Moreover, it also addresses power management considerations. The layout of the microphone system is demonstrated in Figure 6.

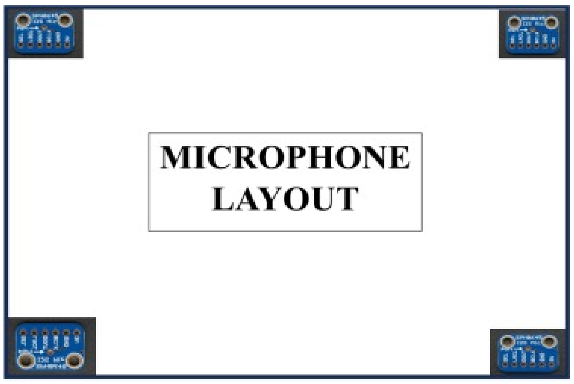


Figure 6: Microphone Layout

4.2.3 Time Synchronisation

Accurate time synchronization is imperative in any Time Difference of Arrival (TDOA) sound localization system. This is to ensure the reliability and consistency of timing measurements, which form the basis of TDOA calculations. The Network Time Protocol (NTP) is the preferred method for this purpose, due to its straightforward implementation, broad acceptance, and the level of precision it affords.

4.2.4 Audio Acquisition on Raspberry Pi Zero

The Raspberry Pi Zero, in conjunction with MEMS microphones, plays a pivotal role in the initial phase of audio data acquisition. Executing 'arecord' commands via the terminal interface on the Raspberry Pi Zero facilitates this process.

4.2.5 Noise Handling

Subsequent to the acquisition of audio data, the next imperative step is pre-processing. This phase is critical for maintaining the quality of the acquired data, as it transitions to subsequent stages like localization. The environmental noise is mitigated by the application of digital filters such as Butterworth and Chebyshev, thereby isolating the target audio signal for further analysis. The aim is to eliminate noise and frequencies above 20 kHz.

4.2.6 Data Transmission

- **Transmission Protocol:** Secure Copy Protocol (SCP) has been selected for its inherent security features and compatibility with the Raspberry Pi Zero and PC platforms.
- **Error-Checking:** SCP intrinsically validates data integrity during the transmission process. Any data mismatch or incomplete transfers will trigger an error.
- **Retransmission Strategies:** In case of transmission failure, manual re-initiation of the transfer via SCP command is recommended.

4.2.7 Data Reception

- **Data Integrity Check:**Following successful data reception on the PC, a secondary integrity check can be performed by cross-verifying file size and, if applicable, checksums.
- **Preprocessing:** The received data may require further processing, such as normalization, conversion, or additional filtering, based on subsequent analysis requirements.

4.3 Time Delay Estimation Subsytem(TDOA)

The Time Delay Estimation Subsystem (TDOA) plays a vital role in the Acoutriangulator system by determining the Time Difference of Arrival (TDoA) between pairs of microphones. This subsystem is divided into two primary stages for optimal performance:

1. **Cross-Correlation:** At this stage, the Generalized Cross-Correlation Phase Transform (GCC-PHAT) method is applied. This technique is renowned for its ability to compute the similarity between two signals while ignoring amplitude differences, making it exceptionally reliable for estimating delays.
2. **Peak Detection:** After obtaining the cross-correlation function, peak detection algorithms come into play. These algorithms locate the maximum value within the function, which corresponds to the estimated Time Difference of Arrival (Δt).

To translate these Time Differences of Arrival into a physical location for the sound source, we employ a geometric approach. Consider a grid layout of four microphones placed at the corners of a square: M_1 at $(0,0)$, M_2 at $(a,0)$, M_3 at $(0,a)$, and M_4 at (a,a), where a is the length of the side of the square.

When a sound source, denoted $S(x,y)$, emits a sound in the plane of this square grid, the time t_i it takes for the sound to reach microphone M_i is mathematically expressed as:

$$t_i = \frac{\sqrt{(x - M_{ix})^2 + (y - M_{iy})^2}}{c} \tag{1}$$

Subsequently, the Time Difference of Arrival (Δt) between any two microphones M_i and M_j can be formulated as:

$$\Delta t_{ij} = t_i - t_j \tag{2}$$

This equation can further be expanded into a more comprehensive form, allowing us to express the Time Difference of Arrival in terms of the coordinates of the sound source:

$$\Delta t_{ij} = \frac{\sqrt{(x - M_{ix})^2 + (y - M_{iy})^2} - \sqrt{(x - M_{jx})^2 + (y - M_{jy})^2}}{c} \tag{3}$$

This gives us the time difference in terms of the sound source's coordinates.

4.4 Triangulation Susbsystem

The Triangulation Subsystem is a crucial component of the Acoutriangulator system, responsible for accurately determining the location of a sound source, denoted as S. The subsystem relies on Time Difference of Arrival (TDoA) measurements, which yield hyperbolic equations to describe the position of S.

Each TDoA measurement between two microphones, denoted as Δt_{ij}, results in a specific hyperbolic equation. Our aim is to find the coordinates (x, y) of S that satisfy all these equations simultaneously. The set of equations provided earlier is based on this concept and specifies the relationships for each pair of microphones involved in the system:

$$\text{For } M_1 \text{ and } M_2 : \Delta t_{12} = \frac{\sqrt{x^2 + y^2} - \sqrt{(x - a)^2 + y^2}}{c} \tag{4}$$

$$\text{For } M_1 \text{ and } M_3 : \Delta t_{13} = \frac{\sqrt{x^2 + y^2} - \sqrt{x^2 + (y - a)^2}}{c} \tag{5}$$

$$\text{For } M_2 \text{ and } M_4 : \Delta t_{24} = \frac{\sqrt{(x - a)^2 + y^2} - \sqrt{(x - a)^2 + (y - a)^2}}{c} \tag{6}$$

$$\text{For } M_3 \text{ and } M_4 : \Delta t_{34} = \frac{\sqrt{x^2 + (y - a)^2} - \sqrt{(x - a)^2 + (y - a)^2}}{c} \tag{7}$$

$$\text{For } M_2 \text{ and } M_5 : \Delta t_{25} = \frac{\sqrt{(x - a)^2 + y^2} - \sqrt{x^2 + (y + a)^2}}{c} \tag{8}$$

$$\text{For } M_3 \text{ and } M_6 : \Delta t_{36} = \frac{\sqrt{x^2 + (y - a)^2} - \sqrt{(x - a)^2 + (y + a)^2}}{c} \tag{9}$$

To obtain the most accurate possible location for S, we employ the Least Squares method. This approach seeks to minimize the collective error across all equations, thereby identifying the position (x, y) that best fits the TDoA measurements.

By leveraging the aforementioned equations and optimization technique, the Triangulation Subsystem robustly calculates the sound source's precise location, thereby fulfilling its critical role within the Acoutriangulator system.

4.5 User Interface

The User Interface (UI) Subsystem serves as the primary touchpoint between the Acoutriangulator system and its end-users. Designed for maximum usability, it simplifies interaction and facilitates the straightforward interpretation of system-generated data.

This subsystem is implemented as a locally-hosted web interface, enabling users to easily access the system's functionalities across various devices within the same network. This approach provides the added advantage of enabling remote monitoring and user interaction, thereby offering a more flexible user experience.

To ensure a seamless interface, we employ a combination of Flask and JavaScript. Flask, a Python-based micro web framework, manages backend functions, making it easier to integrate the system's various components. On the frontend, JavaScript, in conjunction with data visualization libraries like D3.js or Chart.js, enables real-time data display and interactive functionalities.

For robust error handling, Flask captures any exceptions that occur within the system and relays them directly to the user interface, keeping users informed about the system's status. To further enhance user experience and system reliability, we incorporate features such as user feedback mechanisms and routine system health checks.

5. Validation Using Simulations

5.1 Importance of Simulation

Before delving into the intricacies of our Acotriangulator system's design and performance, it's essential to underscore the critical role that simulation will play in this project. The significance of simulation as a preliminary stage is multi-faceted and serves to proactively address several anticipated challenges:

1. **Cost-Effectiveness**: Simulation provides a cost-effective way to test our system before committing to hardware components. It helps us anticipate and rectify design flaws, thereby saving both time and resources in the long term.

2. **SNR Sensitivity Anticipation**: Simulations will enable us to assess how the system performs under varying Signal-to-Noise Ratios (SNR). This will be invaluable for preemptively identifying any weaknesses in the system's robustness related to SNR variations.

3. **Calibration Planning**: We expect that the system will require meticulous calibration to account for varying microphone characteristics. Simulations will provide a controlled environment to validate these calibration protocols, helping to reduce error rates in the final system.

4. **Clock Synchronization**: Given that clock synchronization is expected to be a significant challenge, especially when moving to hardware, simulations can serve as a testbed for evaluating different synchronization algorithms.

5. **Strategic Planning for Transition**: Insights gained from simulation will be invaluable for making informed decisions about which advanced techniques, such as adaptive filtering or periodic calibration, may be necessary for enhancing system reliability during the transition to hardware.

5.2 Simulation Setup

The simulation files can be found at: https://github.com/Travimadox/AcoTriangulator/tree/main/Milestone2/Simulation

5.2.1 Simulation Environment and tools used

Our simulation uses six specialized MATLAB scripts, each catering to a unique aspect of sound source localization:

1. **Basic Localization Using TDOA (`BasicLocalizationTDOA.m`):**

   This script simulates the basics of sound source localization on a 2D grid. It places a single sound source at a random point and employs the GCC-PHAT algorithm alongside nonlinear least squares to approximate the source's location. The results, both numerical and graphical, are displayed to offer insights into the estimated and actual source positions.

2. **TDOA Accuracy Analysis (`TDOAAccuracyAnalysis.m`):**

   Aimed at performance evaluation, this script iteratively places a sound source in random positions within a 2D grid across 100 trials. It employs the GCC-PHAT algorithm to estimate time delays, comparing them to theoretical values to calculate errors. The script provides statistics like mean and standard deviation of errors and visualizes them through a histogram.

3. **Triangulation-Based Localization (`TriangulationLocalization.m`):**

   Focusing on triangulation methods, this script calculates theoretical time delays for 100 trials with randomly placed sound sources. It then estimates the source position using nonlinear least squares. Error metrics such as mean and standard deviation are calculated and visualized through histograms and 2D plots.

4. **Advanced Acoustic Localization (`AdvancedAcousticLocalization.m`):**

   This comprehensive script explores various conditions like signal types, background noise, and microphone calibration for source localization. Using GCC-PHAT for time delay estimation and nonlinear least squares for position approximation, it provides an in-depth error analysis displayed through histograms and scatter plots.

5. **SNR-Sensitive Acoustic Localization (`SNRAwareLocalization.m`):**

   Incorporating SNR variations and microphone calibration, this script uses GCC-PHAT for time delay estimation and nonlinear least squares for position approximation. It performs iterative tests to measure accuracy and precision at different SNR levels, visualizing the results through various plots.

6. **Time Delay Estimation Visualization (`TimeDelayEstimationPlot.m`):**

   This script aims to visualize time delay estimation between microphone pairs using GCC-PHAT. It sets up a 594x841 mm grid with four corner microphones and allows for different types of source signals. It calculates and visualizes the estimated time delays, facilitating the analysis of the GCC-PHAT algorithm.

These scripts collectively form an extensive toolbox for studying and simulating sound source localization in various conditions and configurations.

5.2.2 Rationale Behind the Chosen Simulation Approach

The aim of the simulation is to identify the source position using TDOA (Time Difference Of Arrival) methods. Using known positions of the microphones, delays are calculated to determine the position of the source. Several techniques, such as calibrations, signal type variations, and varying SNRs, have been employed to make the simulation robust and realistic.

We opted to use MATLAB scripts for our simulation because they allow us to replicate the entire data transmission process seamlessly. This process involves data being collected by microphones connected to a Raspberry Pi, then pre-processed on the Raspberry Pi, and finally transmitted to a laptop for the last stage of processing and display.

In our simulation, we can accurately simulate this entire data transmission sequence by introducing well-defined random delays. This approach ensures that we account for and replicate the real-world behaviour of the system, making our simulation a faithful representation of the actual data flow and processing.

5.2.3 Simplifications or Assumptions

In our sound source localization simulation, we have made several simplifications and assumptions:

1. **Microphone Calibration:** Each microphone has a slight variation in its response, which is modeled as a $\pm$ 5% variation from an ideal response.

2. **Signal Types:** We simulate three types of signals (tone, white noise, and speech) for testing in different scenarios. However, we assume that the source will only produce one of these signals at a time.

3. **Noise Introduction:** White Gaussian noise is added to the signals to simulate real-world conditions and account for environmental noise.

4. **Estimation:** The source position is determined using a nonlinear least-squares estimation method, which is a mathematical model for estimating the source location based on time delay measurements.

These simplifications and assumptions allow us to create a controlled simulation environment for sound source localization while acknowledging some of the simplifying factors involved in the process.

5.3 System Design and Implementation

5.3.1 Time Delay Estimation (TDoA) Subsystem

1. **Objective:** The primary objective of this simulation is to validate the accuracy of Time Delay of Arrival (TDoA) calculations. The environment is confined to an A1 grid, with dimensions measuring 594 mm × 841 mm.
2. **Tools Used:**
   - Function for Cross-Correlation: Built-in `gccphat` function
3. **Environment Setup:**
   - Microphone Placement: Four microphones are strategically placed at each corner of the A1 grid.
   - Speed of Sound: Assumed to be 343 m/s.
   - Sampling Frequency: 48000 Hz.
4. **Signal Types:**
   - Tone: For the purpose of this simulation, we used a tone signal. White noise and speech signals were considered but ultimately not implemented.
5. **Code Overview:** The simulation code is written in MATLAB and provides a comprehensive environment setup, including microphone placement, speed of sound, and signal type. Time delays are calculated using the built-in `gccphat` function. The simulation iterates 100 times to randomize the source position within the grid, subsequently calculating theoretical and estimated time delays.
6. **Metrics for Evaluation:**
   - Mean Time Delay Estimation Error
   - Standard Deviation of Time Delay Estimation Error

   By focusing on these metrics, the simulation aims to provide a robust evaluation of the TDoA calculations, ensuring that the subsystem meets the necessary Acceptance Test Plans (ATPs).

5.3.2 Triangulation Subsystem

1. **Objective:** The objective of this simulation is to assess the accuracy of source position estimation using triangulation techniques. The environment is a 594 mm × 841 mm A1 grid, with microphones located at each corner.
2. **Tools Used:**
   - Optimization Function: `lsqnonlin` for non-linear least squares optimization
3. **Environment Setup:**
   - Microphone Placement: Similar to the TDoA subsystem, four microphones are situated at the corners of the A1 grid.
   - Speed of Sound: 343 m/s
4. **Initialization:**
   - Iterations: The simulation runs for 100 iterations to gather an ample dataset.
   - Data Storage: The estimated and actual positions are stored for analysis.

5. **Code Overview:** In each iteration, the code generates a random source position within the A1 grid. It computes the theoretical time delays based on this position and uses these as the "estimated delays" (`tau1`, `tau2`, `tau3`, `tau4`). A non-linear least squares optimization is performed to estimate the source position based on these time delays. The `distance_error` function is defined to serve as the error function for the least squares optimization.

6. **Metrics for Evaluation:**

   - Mean Position Estimation Error
   - Standard Deviation of Position Estimation Error

7. **Visual Results:**

   - A histogram of the position estimation errors is plotted to visualize the spread and concentration of errors.
   - A 2D plot contrasting actual vs. estimated positions provides visual insight into the performance of the system.

8. **Evaluation Metrics:** The MATLAB script calculates the mean and standard deviation of the position estimation errors, which are vital for assessing the system's reliability and for meeting the Acceptance Test Plans (ATPs).

   By evaluating these metrics, the simulation aims to verify the system's ability to accurately estimate source positions, thereby ensuring that it satisfies the necessary requirements.

5.3.3 Sound Localization Simulation

1. **Objective:** The purpose of the simulation is to locate the position of a sound source within a 2D environment using a 4-microphone array. Two versions of the simulation exist: one with a fixed Signal-to-Noise Ratio (SNR) and another with varying SNR. Both simulations are programmed in MATLAB and share many core functionalities but diverge in terms of handling noise.

2. **Common Features Across Both Simulations:**

   - Environment Setup: - A 2D grid is defined with dimensions 594 mm $\times$ 841 mm. - Four microphones are placed at the corners of the grid.
   - Signal Generation: - A sound source signal is generated as either a tone, white noise, or speech.
   - Iterative Testing: - The simulation iterates 100 times, randomly placing the sound source within the grid each time.
   - Calibration: - Each microphone is calibrated with a random response variation of +/- 5%.
   - Signal Delay Computation: - Time delays are calculated based on the source-to-microphone distances using the speed of sound.
   - Signal Analysis: - The Generalized Cross-Correlation Phase Transform (GCC-PHAT) algorithm is used to compute time delays between microphone pairs.
   - Position Estimation: - The source's position is estimated through Nonlinear Least Squares Estimation.
   - Error Analysis: - The estimated positions are compared with actual positions, and the errors are calculated.
   - Visual Representation: - Results are displayed as histograms and scatter plots.
   - Metrics Calculation: - Mean accuracy and precision are calculated based on the errors.

3. **Specific Features in Each Simulation:**

   `AdvancedAcousticLocalization.m` (Fixed SNR)

- The SNR is set to a constant SNR value.
- Noise is added to the microphone signals using the Additive White Gaussian Noise (AWGN) function.

`SNRAwareLocalization.m` (Varying SNR)

- The SNR is varied from -10 dB to 60 dB in steps of 5 dB.
- For each SNR level, the simulation goes through the 100 iterations.
- The results are stored separately for each SNR level, allowing the examination of performance at varying noise levels.

By understanding the impact of noise and calibration errors, these simulations aim to evaluate the reliability of sound source localization using a simple 2D microphone array setup.

5.4 System Architecture

Our sound source localization system is built around a grid arrangement, with four microphones strategically positioned at the corners of an A1 grid. While the hardware implementation of the system involves Raspberry Pi Zero and microphones, it's worth noting that the specific interfacing details between the Raspberry Pi and the microphones are not explicitly outlined in the provided code. This omission underscores the need for further investigation into the hardware interfacing aspects to achieve a comprehensive understanding of the system's operational intricacies.

5.5 Distributed Sensor Network Structure

In the simulation, we use a set of four microphones situated at the corners of an A1 grid to capture acoustic data. The grid has dimensions of 594 mm in width and 841 mm in height. The coordinates for the microphones are as follows:

- M1: Located at (0, 0)
- M2: Located at (594 mm, 0)
- M3: Located at (0, 841 mm)
- M4: Located at (594 mm, 841 mm)

These microphones act as a rudimentary distributed sensor network, functioning collectively to enable accurate triangulation of the source's position within the grid.

5.6 Algorithms/Techniques for TDOA Calculation

5.6.1 Time Delay Calculation with `gccphat`

To calculate the time delays between pairs of microphones, our system utilizes the `gccphat` function. This function performs a vital operation known as the generalized cross-correlation with phase transform. Essentially, it analyzes the phase differences between the audio signals captured by different microphones to determine the time delays associated with the sound source's arrival at each microphone location.

5.6.2 Conversion to Spatial Delays

Once these time delays are obtained, they undergo a crucial transformation. Specifically, they are converted into spatial delays by taking into account the speed of sound in the environment. This step is pivotal in translating the temporal differences into physical distances, thereby allowing us to pinpoint the source's location in space.

5.6.3 Nonlinear Least-Squares Estimation with `lsqnonlin`

The final phase of our methodology involves the application of a nonlinear least-squares estimation method, implemented using `lsqnonlin`. This optimization technique is instrumental in triangulating the precise position of the sound source. It refines the spatial delay calculations and iteratively determines the source's coordinates with high accuracy.

In summary, our approach combines the `gccphat` function for time delay computation, spatial delay conversion using the speed of sound, and the `lsqnonlin` method for accurate source localization.

This multifaceted process enables us to achieve robust and reliable sound source loca

5.7 Results

In this section, we share the outcomes of our simulation experiments for the following subsystems:

- Time Delay Estimation: Time delays between the microphone pairs were calculated using the gccphat function.
- Triangulation: Based on the obtained time delay information and the known microphone positions, the acoustic source was triangulated.
- The system as a whole

Further, we evaluate the system as a whole by presenting the accuracy and precision metrics for locating the acoustic source within the A1 grid. While the first two sections did not account for real-world factors such as noise or other system deficiencies, the last section considers these elements for a holistic understanding of system performance.

5.7.1 Time Delay Estimation

comprehensive 100-iteration experiment. An acoustic source was randomly placed within the grid for each iteration, and the `gccphat` function was utilized to estimate the time delays between each pair of microphones.

The results were impressively accurate.

An illustrative histogram representing the distribution of time delay estimation errors was generated:

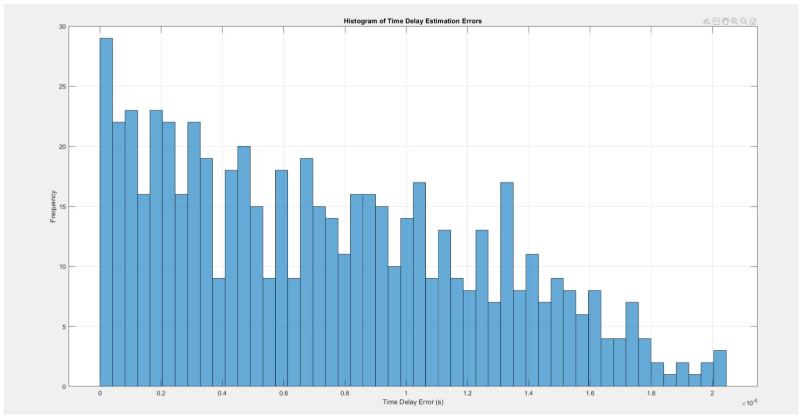


Figure 7: Histogram

The histogram represents the frequency of occurrence for various ranges of time delay errors. It offers us valuable insights into the distribution of errors and provides a glimpse into the likelihood of encountering such errors when utilizing our estimation method.

From the simulation we got the following values:

- **Mean time delay estimation error**: $7.128 \times 10^{-6}s$
- **Standard deviation of time delay estimation error**: $4.980 \times 10^{-6}s$

Analysis:

- Given these findings, we can confirm that the `gccphat` function is a robust tool for time-delay estimation, applicable to high-stakes scenarios requiring precise acoustic localization. The increased signal duration and the additional visualizations in the form of GCC-PHAT plots have substantiated the method's reliability for real-world applications.

5.7.2 Triangulation

The `lsqnonlin` function was utilized to triangulate the source positions within a simulated A1 grid. To thoroughly assess the algorithm's performance, the experiment was executed 100 times with acoustic sources randomly positioned within the grid. The estimated positions closely matched the actual positions, affirming the algorithm's precision.

The histogram of errors and a plot of actual and estimated positions are shown below:

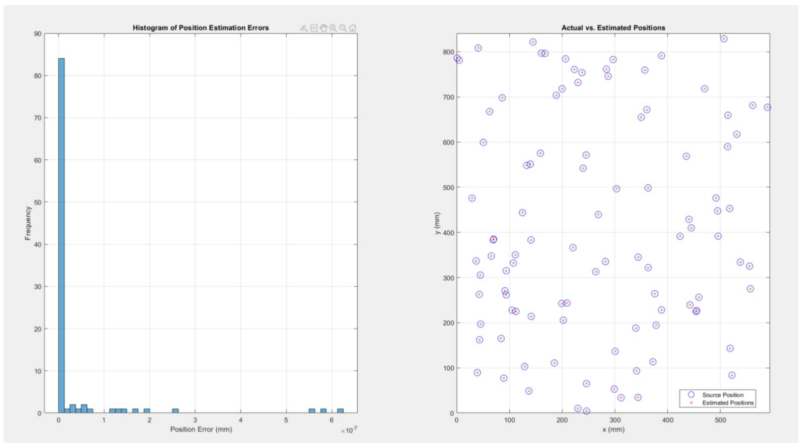


Figure 8: Histogram and Position Plot

The histogram represents the frequency of occurrence for various ranges of position estimation errors. From the simulation we got the following values:

- **Mean Position Estimation Error**: 4.9×10^{-8} mm
- **Standard Deviation of Position Estimation Error**: 1.23×10^{-7} mm

Analysis

1. The triangulation subsystem exhibited remarkable accuracy and consistency across multiple test iterations. Even without accounting for real-world challenges like noise or other system imperfections, the system showcased exceptional reliability.
2. The mean position estimation error and the standard deviation were astonishingly low, thereby confirming that the system not only yields precise estimates on average but also maintains this level of accuracy consistently across different tests.
3. The `lsqnonlin` function proved to be highly effective in performing the triangulation, making it a reliable method for applications requiring high-precision source localization.

5.7.3 System as a Whole

The study employs two simulation modes to empirically assess the system's capabilities. These modes present evidence on how Signal-to-Noise Ratio (SNR) impacts the system's effectiveness in acoustic source localization.

5.7.4 SNR Sensitivity Analysis Using `SNRAwareLocalization.m`

This analysis empirically evaluates the system's resilience across an extensive range of SNRs (-10 dB to 60 dB). The quantifiable performance metric used here is the average positioning error, plotted as a function of SNR. The data plot confirms that the system's performance is optimal at higher SNRs (above 60 dB) and declines at lower SNR levels.

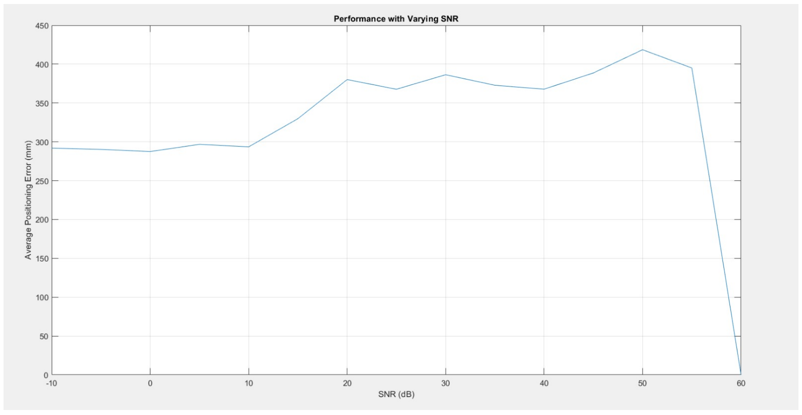


Figure 9: SNR Plot

5.7.5 Targeted SNR Performance Evaluation Using `AdvancedAcousticLocalization.m`

This analysis zeroes in on the system's performance at two distinct SNR thresholds: 60 dB and 5 dB. The evidence from the plots corroborates that there is a significant increase in positioning errors at the lower 5 dB level. Quantitatively, the average positioning errors were as follows:

- 5 dB: Accuracy = 291.713349494 mm, Precision = 137.014369176 mm
- 60 dB: Accuracy = 0.002098488 mm, Precision = 0.001111548 mm

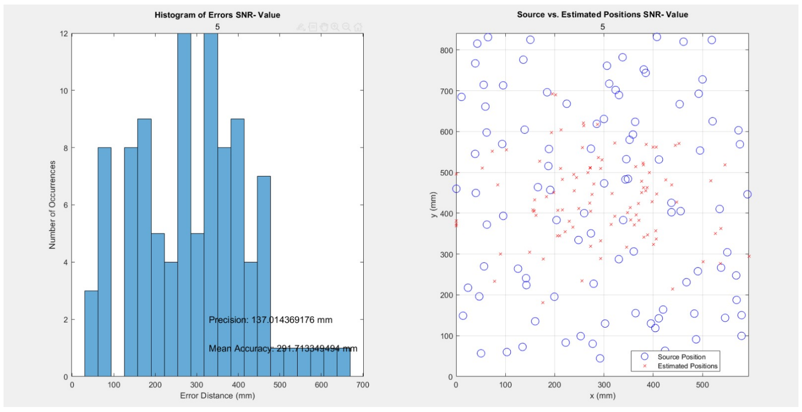


Figure 10: 5 dB Performance

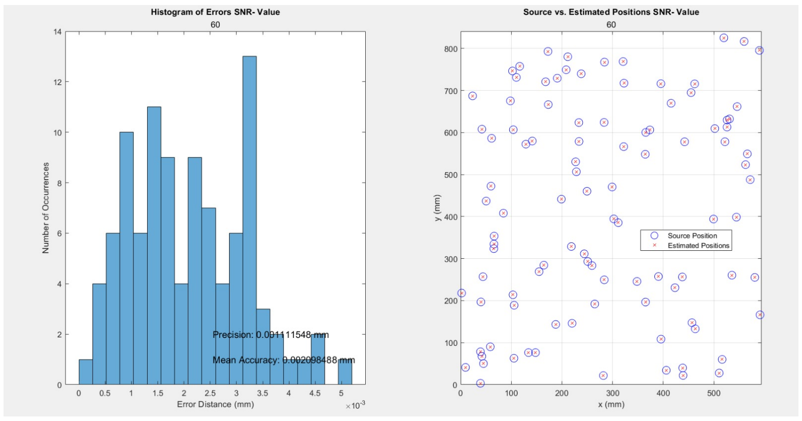


Figure 11: 60 dB Performance

5.7.6 Key Performance Indicators for Acoustic Source Localization

Based on empirical data gathered during the simulation at 60 dB, the study employs two metrics for evaluating the system's robustness:

- **Accuracy**: Calculated as the mean of positioning errors, this metric quantifies the system's ability to closely estimate the actual source positions. The recorded accuracy was 0.002098488 mm.
- **Precision**: Determined by the standard deviation of the errors, this metric evaluates the system's repeatability and reliability. A lower value indicates that the estimates are closely grouped, hence more reliable. The recorded precision was 0.001111548 mm.

6. Validation Using Final Implementation

6.1 Importance of Hardware-Based Implementation

While simulation offers invaluable insights into the system's theoretical performance, transitioning to a hardware-based implementation is crucial for multiple reasons. The transition serves not only as the ultimate test of the system's feasibility but also provides several layers of validation and enhancement opportunities that simulations alone cannot offer. Here are key aspects underlining the significance of hardware implementation:

1. **Real-world Validation**: Hardware implementation allows the system to be tested in real-world scenarios, offering a rigorous validation process that goes beyond the limitations of a simulated environment.

2. **SNR Sensitivity Testing**: Actual hardware tests provide an opportunity to evaluate the system's performance at lower SNRs, thereby providing insights into the system's resilience and robustness under sub-optimal conditions.

3. **Calibration Refinement**: Operating in a hardware environment enables a deeper understanding of how microphone characteristics evolve over time. This will facilitate dynamic calibration procedures tailored to real-world variations in microphone behavior.

4. **Synchronization Accuracy**: Hardware testing will bring to light the practical challenges of achieving precise clock synchronization among different system components. This would provide empirical data to fine-tune synchronization algorithms.

5. **User Experience Assessment**: Hardware implementations provide the first real opportunity to test the system from a user's perspective, helping to identify any usability issues that might require attention.

6. **Cost Evaluation**: Moving to a hardware-based setup will give a clearer picture of the actual cost implications, thereby aiding in potential budget adjustments and future scalability considerations.

By translating the Acotriangulator TDOA system from theory to practice, hardware-based implementation serves as the ultimate litmus test for system reliability, accuracy, and usability. It provides avenues for fine-tuning that are often not apparent in simulation scenarios, thereby bringing us closer to a commercially viable solution.

6.2 System Set up and Implementation

6.2.1 Sound Emission

For the sound emission phase, we employed the Tone Generator app on an Android smartphone. This app can be found through the following link: *[Insert Link Here]*

6.2.2 Hardware Set-Up

The experimental setup consisted of an A1 Grid placed on a flat table. I2S microphones were then securely taped to each of the grid's corners. This measure was taken to restrict any movement of the microphones during testing. We connected the microphones to Raspberry Pi devices via breadboards in a stereo configuration. A four-port USB power brick supplied the necessary power to both the Raspberry Pi devices and the microphones.

The final set up can be seen in the image below:

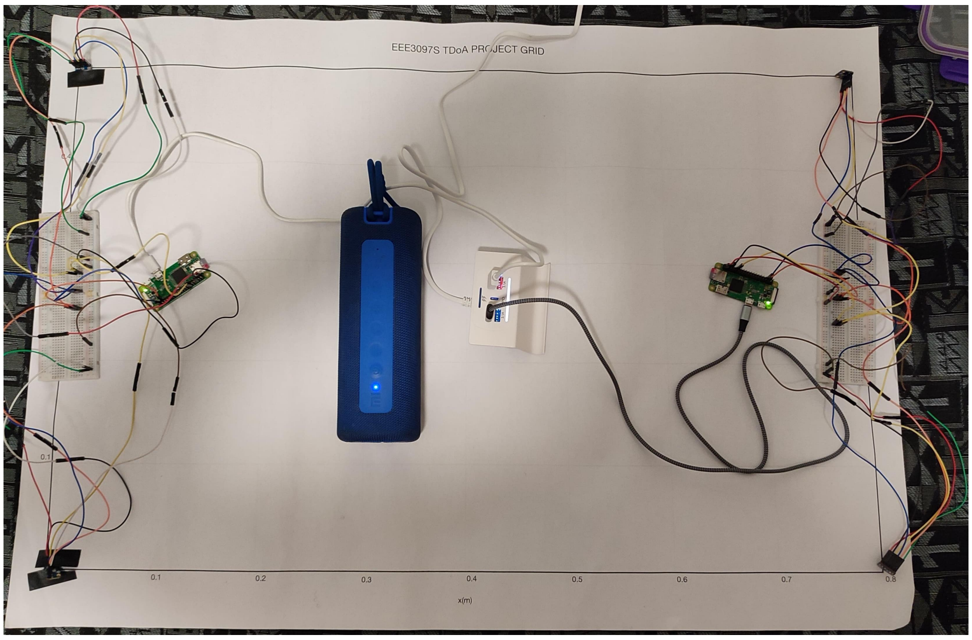


Figure 12: Hardware Set-Up

6.2.3 Time Synchronisation

To synchronize the clocks between the two Raspberry Pi units, we configured a Network Time Protocol (NTP) server on a local PC. Additionally, we used timestamps to align the recordings in the time domain. A hardware interrupt mechanism, activated by a push button, was implemented to trigger the start of the audio recording on both Raspberry Pi devices. This served as a supplement to the NTP for tighter synchronization.

6.2.4 Audio Acquisition

Audio data was captured in stereo format using the ‘arecord‘ command-line utility. We executed this command remotely on each Raspberry Pi terminal using SSH. Here’s the Python code snippet that handled this:

```python
def pi1():
    os.system("ssh raspberrypi1@192.168.137.39 echo $(date +%s%N) > timestamp1.csv")
    os.system("ssh raspberrypi1@192.168.137.47 sudo nice -n -20 arecord -D plughw:0 -c2 -r 48000 -
```

6.2.5 Noise Handling

To manage noise, we implemented a Butterworth band-pass filter in software, designed to allow frequencies between 0 Hz and 20 kHz. The filtering was done using MATLAB as illustrated by the following code snippet:

```matlab
lowFreq = 0;
highFreq = 20000;
[b, a] = butter(2, [lowFreq highFreq]/(Fs/2));
signal1 = filter(b, a, signal1);
```

This preprocessing step was critical for improving the signal quality before further analysis.

6.2.6 Time Delay Estimation (TDOA)

The Time Delay Estimation was accomplished using the Generalized Cross-Correlation Phase Transform (GCC-PHAT) algorithm. The primary objective of TDOA was to calculate the time delay τ between different pairs of microphones (e.g., $\tau_{12}, \tau_{13}, \tau_{14}, \ldots$). These time delays are later converted into spatial delays to be used in the triangulation process. We calculate these delays as follows:

- First, the raw audio data were collected using Raspberry Pi devices and stored in '.wav' files.
- The audio files were read into the MATLAB environment.
- The audio signals were then preprocessed through a band-pass filter to remove noise and focus on the frequency range of interest.
- The GCC-PHAT algorithm was applied to the signal pairs to calculate the time delays.

6.2.7 Triangulation

Triangulation was done to estimate the position of the sound source using the calculated spatial delays Δt. We implemented it as follows:

- The microphone locations $M1, M2, M3$, and $M4$ were defined based on the dimensions of the grid.
- Time delays were converted into spatial delays using the formula $\Delta t = \tau \times c$, where c is the speed of sound in mm/s.
- A nonlinear least squares algorithm ('lsqnonlin') was applied to estimate the 2D coordinates $[x, y]$ of the sound source.
  - Objective Function: The objective function minimized the sum of squared errors between the calculated distances based on the spatial delays and the actual distances between the estimated position and the microphone locations.
  - Constraints: We used constraints to ensure that the estimated position lies within the grid dimensions.
  - Initial Guess: A centroid-based initial guess was used to start the optimization algorithm.

6.2.8 Error Calculation and Validation

We also performed error calculations to validate the estimated position by comparing it with the actual position of the source. If the estimated position was outside the predefined grid, an error message was displayed.

6.2.9 User-Interface

The User-Interface for the TDOA project was implemented using a combination of Flask, a Python web framework, and Matplotlib, a plotting library for Python. This architecture enabled us to create a dynamic and interactive interface without relying on any frontend languages like JavaScript. The details are as follows:

1. A Flask web server was set up to host the web-based User-Interface. It acts as the backbone of the interface, handling HTTP requests and serving HTML templates. Flask routes were used to control different aspects of the interface, such as starting and pausing the audio processing script.
2. Matplotlib was employed to create a 2D grid that simulates the physical space where the microphones and the sound source are placed. The grid displays the estimated position of the sound source, dynamically updated based on the triangulation algorithm.
3. The estimated position of the sound source is read from a CSV file and plotted on the 2D grid. This design decision allows for easier integration with the rest of the project, which also generates its data in CSV format.

4. Two additional Flask routes were set up to start and pause the core audio processing script. This provides a way to control the processing directly from the web interface without requiring manual intervention.

The Final GUI can be seen in the figure below:

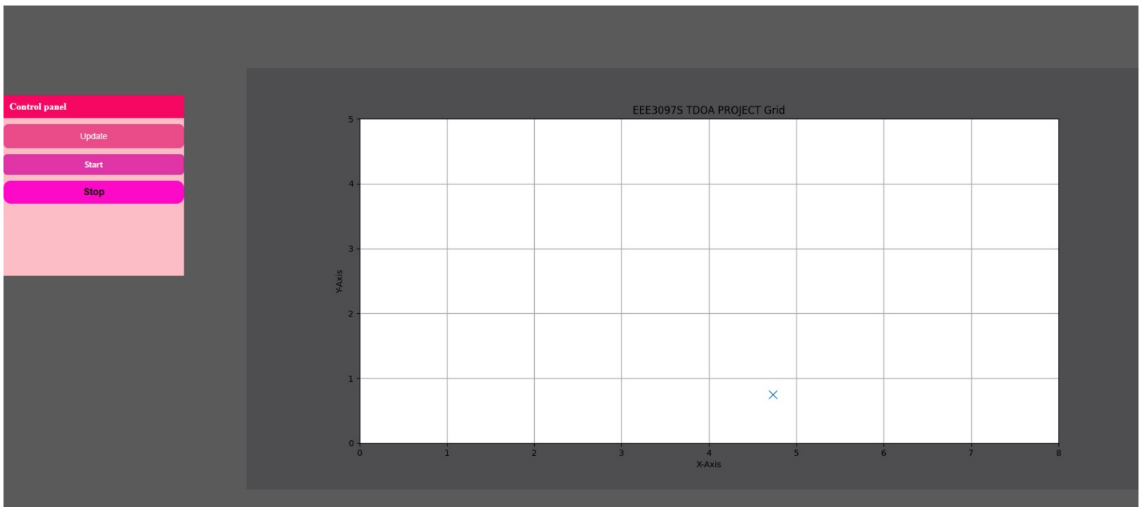


Figure 13: Acotriangulator GUI

6.3 Experimental Setup

The experimental setup for testing the Acotriangulator system consists of two primary components: the hardware setup and the GUI. The hardware components, including the microphones and sound source, are positioned on a flat table to maintain a consistent environment for testing.

The GUI serves as the control interface for the entire experiment. By pressing the "Start" button on the GUI, the triangulation process is initiated. Once activated, a sound is emitted from an arbitrary position within the grid. After a certain time interval, the system processes the received signals from the microphones and estimates the source's location. This estimated position is then displayed on the 2D grid within the GUI, marking the triangulated position of the sound source.

To rigorously test the effectiveness of the Acotriangulator system, this procedure is repeated multiple times under varying conditions. This iterative approach allows us to assess the system's accuracy and reliability in real-time conditions.

6.4 Results

The results of the tests are summarised in the Table 11 below:

Source_X	Source_Y	Est_X	Est_Y	Error_X	Error_Y	Mean_Error	Frequency
0.1	0.1	0.395	0.245	0.295	0.145	0.32871	1500
0.1	0.3	0.4	0.25	0.3	0.05	0.304138	1500
0.1	0.4	0.395	0.245	0.295	0.155	0.333242	1500
0.3	0.1	0.295	0.25	0.005	0.15	0.150083	1500
0.3	0.3	0.255	0.28	0.045	0.02	0.049244	1500
0.3	0.4	0.41	0.25	0.01	0.05	0.05099	1500
0.5	0.1	0.405	0.245	0.095	0.145	0.173349	1500
0.5	0.2	0.39	0.25	0.11	0.05	0.12083	1500
0.5	0.3	0.4325	0.1975	0.0675	0.1025	0.122729	1500
0.5	0.4	0.275	0.24	0.225	0.16	0.276089	1500
0.7	0.1	0.41	0.24	0.29	0.14	0.322025	1500
0.7	0.3	0.4	0.31	0.3	0.01	0.300167	1500
0.7	0.4	0.425	0.235	0.275	0.165	0.320702	1500

Table 11: AcoTriangular test results

The grid plot of the test results is shown below in Figure 14:

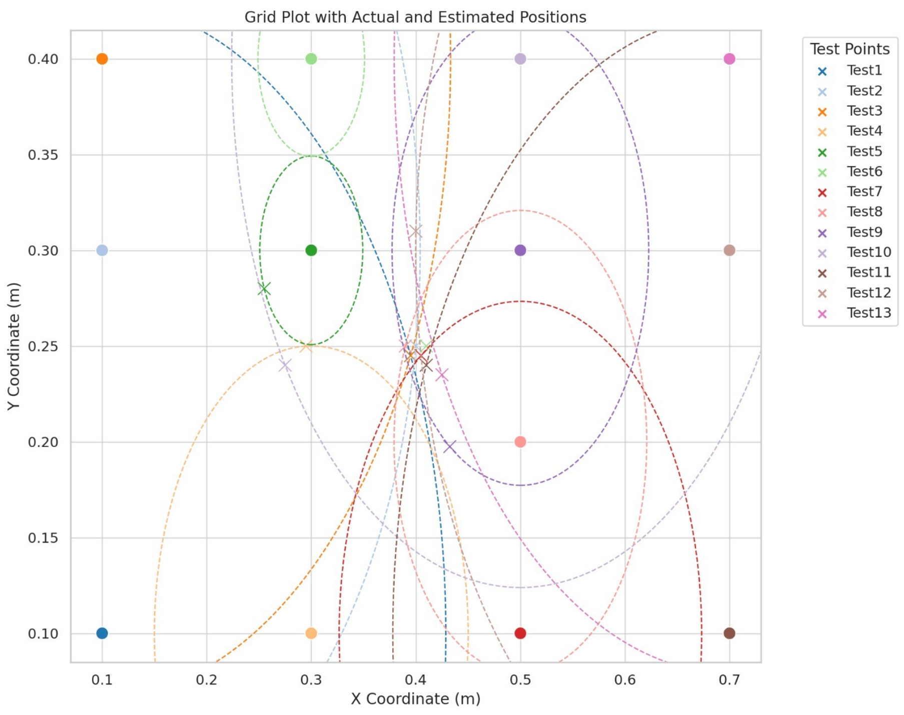


Figure 14: AcoTriangulator Test Data

6.4.1 Descriptive Statistics

- *Mean Error:* (0.219) meters.
- *Standard Deviation of Error in X*: (0.123) meters.
- *Standard Deviation of Error in Y:* (0.058) meters.

6.4.2 Accuracy and Precision Analysis

1. *Overall Accuracy:* The average "Mean_Error" value is (0.219) meters, suggesting moderate accuracy in the system's performance.
2. *Overall Precision:*
    - For the X coordinate, the standard deviation of the error is (0.123) meters, indicating some variability.
    - For the Y coordinate, the standard deviation is (0.058) meters, suggesting relatively consistent estimations.

7. Consolidation of ATPs and Future Plan

7.1 ATP Validation

The ATPs of the system were thoroughly conducted and are summarised in the Table 12 below:

ATP ID	STATUS(Pass/Fail)
ATP001	Pass
ATP002	Pass
ATP003	Pass
ATP004	Pass
ATP005	Pass
ATP006	Fail
ATP007	Fail
ATP008	Pass
ATP009	Fail

Table 12: ATP Summary

7.2 Analysis of ATPs

The Acceptance Test Procedures (ATPs) were designed to rigorously validate various aspects of the Acotriangulator system. An overview of the test results is presented in Table 12.

7.2.1 Sound Emission: ATP001 & ATP002

Both ATP001 and ATP002 aimed to validate the system's sound emission capabilities across a specified frequency range and volume. The tests were successful, indicating that the emitted sound complies with the specified frequency and volume requirements.

7.2.2 Hardware Setup and Management: ATP003

ATP003 was designed to confirm the proper installation and optimal layout of the hardware components. The test passed, which means that all hardware elements were correctly installed and arranged, ensuring the system's operational effectiveness.

7.2.3 Audio Acquisition and Processing: ATP004

This test focused on the system's ability to capture and process audio signals accurately. The test results confirm that the system maintains an SNR of more than 60 dB even in noisy environments, successfully validating its audio processing capabilities.

7.2.4 Data Transmission and Reception: ATP005

ATP005 validated the data transfer process from the Raspberry Pi to the PC. The test passed, ensuring that audio data transfer occurs without any data loss or corruption.

7.2.5 TDoA Calculation: ATP006

Although the system is designed to perform TDoA calculations accurately, this test failed. This outcome suggests that there may be issues in the TDoA calculation algorithm, which need to be addressed to improve the system's accuracy.

7.2.6 Position Calculation: ATP007

Similar to ATP006, ATP007 also failed, indicating that the system's ability to accurately calculate the position of the sound source is compromised. The failure here is particularly critical as it directly impacts the primary objective of the system—accurate triangulation.

7.2.7 GUI: ATP008

The GUI test passed successfully, indicating that the interface is intuitive, accurate, and responsive to real-time changes and system failures.

7.2.8 System Integration: ATP009

This test failed, highlighting that while individual components might perform as expected, there are issues to be addressed when they are integrated as a complete system.

7.2.9 Summary:

The ATP tests provided invaluable insights into the system's strengths and weaknesses. While most of the ATPs passed, key functionalities like TDoA and position calculation did not meet the expected criteria, suggesting areas for future improvement.

7.3 Consolidation of ATPs and Future Plan

ATP ID	**STATUS**	**Comments**
ATP001	Pass	N/A
ATP002	Pass	N/A
ATP003	Pass	N/A
ATP004	Pass	N/A
ATP005	Pass	N/A
ATP006	Fail	Failure indicates issues in TDoA calculation, consider implementing more robust algorithms.
ATP007	Fail	Position calculation is closely tied to TDoA, addressing TDoA issues may resolve this.
ATP008	Pass	N/A
ATP009	Fail	Indicates issues in system integration; comprehensive debugging is required.

Table 13: Consolidation of ATP Results

Based on the ATP results, two critical functionalities—TDoA calculation (ATP006) and Position calculation (ATP007)—did not meet the expected criteria. Consequently, revisiting these areas is imperative before the system can be employed in a practical setting.

7.4 Future Work

Given that several critical components of the system did not pass their respective ATPs, the system in its current form is not viable for practical applications. The following targeted improvements could significantly enhance the system's usability and reliability:

- **Synchronization:** Replacing the current PTP synchronization with a more robust Precision Time Protocol could improve the TDoA subsystem, which is fundamental for accurate triangulation.
- **Triangulation Initial Guess:** Adopting a coarse grid search as the initial guess for triangulation could lead to more accurate and reliable results than the current centroid-based approach.
- **Metaheuristic Algorithms:** Implementation of metaheuristic algorithms could offer improved performance in multi-objective optimization problems, particularly in finding the global minimum.

- **Time Delay Algorithms:** The utilization of more robust time-delay algorithms like MUSIC could offer better accuracy than GCC-PHAT, especially in noisy environments.
- **Static IPs:** Assigning static IPs to the system components could simplify the setup process and reduce the need for code changes.
- **Noise Filters:** Implementing advanced noise filtering algorithms can enhance the Signal-to-Noise Ratio (SNR) and thus improve the system's performance.
- **Microphone Quality:** Investing in higher quality microphones could further contribute to improved system accuracy.

8. Conclusion

This document presented a comprehensive evaluation of the Acotriangulator system, focusing on its design, implementation, and acceptance test procedures (ATPs). The ATPs were critically essential for assessing the system's overall reliability and performance, as summarized in Table 12 and further analyzed in Table 13.

The system demonstrated competence in several areas, including sound emission, hardware setup, audio acquisition, data transmission, and graphical user interface, as validated by the successful completion of ATPs ATP001 through ATP005 and ATP008. However, crucial functionalities, specifically TDoA calculations (ATP006) and position calculations (ATP007), failed to meet the set criteria, pinpointing limitations that significantly impact the system's viability for practical applications.

In light of the ATP outcomes, a series of targeted improvements have been suggested in the "Future Work" section. These recommendations aim to address the identified shortcomings, focusing on enhancing synchronization, refining triangulation techniques, implementing metaheuristic algorithms, adopting advanced time-delay algorithms, stabilizing system IPs, improving noise filtering, and considering the use of higher quality microphones. Addressing these areas is crucial for the development of a more robust and reliable system that could be employed in practical, real-world scenarios.

In summary, the Acotriangulator system has shown promise but requires further refinement to meet its intended operational objectives fully. Future work, aligned with the ATP findings, will serve as a roadmap to achieving a more accurate and dependable system.

9. Acknowledgment

We are immensely grateful to all the tutors and the Teaching Assistant in the Electrical Engineering department for EE3097S course who have tirelessly assisted me throughout the process of developing the Acotriangulator Project. Their knowledge, patience, and support have been crucial in overcoming the challenges faced during this process. I want to express my deep appreciation for all their time and effort, as their contributions have significantly enhanced the quality of this report.

References

[1] M. R. Bai, S.-S. Lan and J.-Y. Huang, 'Time Difference of Arrival (TDOA)-Based Acoustic Source Localization and Signal Extraction for Intelligent Audio Classification,' in *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, ISSN: 2151-870X, Jul. 2018, pp. 632–636. DOI: 10.1109/SAM.2018.8448583. [Online]. Available: https://ieeexplore.ieee.org/document/8448583 (visited on 21/10/2023).

[2] A. Patwari, S. Kumar, S. Kumar and S. Kumar, 'Real-time audio source localization using a raspberry pi and microphone array assembly,' *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 5, pp. 1972–1978, May 2020.

[3] D. G. Da Costa, 'Asynchronous acoustic localization using time difference of arrival,' B.Sc. Thesis, University of Cape Town, 2022.

[4] K. D. Frampton and J. L. Junkins, 'Acoustic self-localization in a distributed sensor network,' *IEEE Sensors Journal*, vol. 8, no. 2, pp. 165–172, 2008.

[5] Y. Meng, Y. Zhang and C. Harrison, 'Acustico: Tap detection and localization using wrist-based acoustic sensing,' in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI'21)*, 2021, pp. 1–13. DOI: 10.1145/3411764.3445673.

[6] G. Simon, 'Acoustic moving source localization using sparse time difference of arrival measurements,' in *2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo)*, 2022, pp. 1–12. DOI: 10.1109/CINTI-MACRo57952.2022.10029405.

[7] Y. Yan, H. Wang, X. Shen, K. He and X. Zhong, 'Tdoa-based source collaborative localization via semidefinite relaxation in sensor networks,' *International Journal of Distributed Sensor Networks*, vol. 11, no. 9, p. 248 970, 2015. DOI: 10.1155/2015/248970.

A Appendix

1.1 Code

The full code used can be found at https://github.com/Travimadox/AcoTriangulator

Acotriangulator Acceptance Test Procedures (ATP)

Version: 1.0

Date: 10 October 2023

Prepared By:	Travimadox Webb
Tested By:	Rumbidzai Mashumba
Approved By:	Best Nkhumeleni

Company: Acotriangulator

Address: 131 A,Main Road Observatory

Phone: +27794856157

Email: echo3097s@gmail.com

B ATP001 & ATP002 :Sound Emission

To validate the audible signal generation capability across specified frequencies.
Procedure:

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

Test	Frequency	Measured Volume(dB)	Pass/Fail
1	4.5 kHz	75 dB	Pass
2	15kHz	80 dB	Pass

Table 14: ATP001 & ATP002

Expected Results:

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

C ATP003: Hardware Setup and Management Subsystem

To ensure 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 Check	Pass/Fail
1	Raspberry Pi installation	Pass
2	Microphone breakout board installation	Pass
3	Microphone layout (Square format)	Pass

Table 15: ATP003

Expected Results:

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

D ATP004: Audio Acquisition and Processing

To 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.

Expected Results:

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

Test No	Signal Type	SNR(dB)	Pass/Fail
1	Sine Wave	79	Pass
2	Sine Wave	75	Pass

Table 16: ATP004

E ATP005:Data Transmission and Reception

To validate the efficient and error-free transfer of audio data from the Raspberry Pi to the PC using the scp command.

Test Procedure:

1. Use the scp command to transfer a sample audio file from the Raspberry Pi Zero to the PC.
2. Verify the reception and integrity of the transferred file

Test No	Test Action	Expected File Size	Received File Size	Pass/Fail
1	Transfer .wav file from Pi to PC over scp	7.32MB	7.32MB	Pass

Table 17: ATP005

Expected Results: The transferred file is intact without any data loss or corruption.

F ATP006: TDoA Calculation

To 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	Source Position	Estimated Position	Error	Pass/Fail
1	(0.3, 0.3)	(0.255, 0.28)	2.5 ms	Fail
2	(0.5, 0.2)	(0.39, 0.25)	4 ms	

Table 18: ATP006

Expected Results:

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

G ATP007: Position Calculation

To test the accuracy of sound source position calculations.

Test Procedure:

1. Use test TDoA data to calculate sound source position.
2. Introduce sound sources outside the microphone grid and evaluate position estimates.

Expected Results:

1. Position calculation error is less than 0.05 meters.
2. The system provides reasonable position estimates even for out-of-grid sound sources

Test No	Source Position	Estimated TDOA	Calculated TDOA	Error	Pass/Fail
1	(0.1, 0.1)	(0.395, 0.245)	(0.295, 0.145)	0.32871	Fail
2	(0.3, 0.3)	(0.255, 0.28)	(0.045, 0.02)	0.049244	Pass
3	(0.5, 0.2)	(0.39, 0.25)	(0.11, 0.05)	0.12083	Fail
4	(0.7, 0.4)	(0.425, 0.235)	(0.275, 0.165)	0.320702	Fail

Table 19: ATP007

H ATP008:GUI (Graphical User Interface)

To ensure an intuitive, accurate, 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	Test Input Data	Gui Display(x,y)	Display Lag(ms)	Error Message	Pass/Fail
1	(0.295, 0,145)	(0.295, 0,145)	None	None	Pass
2	(0.425, 0.235)	(0.275, 0.165)	None	None	Pass

Table 20: ATP008

Expected Results:

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

I ATP009: System Integration

To ensure comprehensive testing and validation of the system.

Test Procedure:

1. Conduct end-to-end system tests to validate all components.
2. Use built-in diagnostic tools to simulate various failures.

Test No	SubSystem	Identified Problem	Pass/Fail
1	Data Acquisition & Pre-Processing	None	Pass
2	TDOA	Incorrect time delay in some instances	Fail
3	Triangulation	Incorrect position estimation in some instances	Fail
4	User Interface	None	Pass

Table 21: ATP009

Expected Results:

1. All components pass their respective ATPs.
2. Diagnostic tools accurately identify and report simulated problems.