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

EEE3097S 2023 ASSIGNMENT 2: FIRST PROGRESS REPORT

ACOTRIANGULATOR

Travimadox Webb
WBBTRA001@myuct.ac.za

Rumbidzai Mashumba
MSHRUM006@myuct.ac.za

Best Nkhumeleni
NKHBES001@myuct.ac.za

12/09/2023

Contents

List of Figures

List of Tables

1. Introduction

The Time-Difference-of-Arrival (TDoA) algorithm is a fundamental tool in sound source localisation, allowing us to discern the origin of sound in our environment. By harnessing the capabilities of multiple microphones and meticulously measuring the temporal disparities in sound arrival at each sensor, TDoA provides a means to approximate the directional source of acoustic emissions. This milestone endeavours to construct a comprehensive simulation to advance our understanding and application of sound source localisation. The objective is to craft a simulation environment that emulates our real-world system with the utmost fidelity. We aspire to render this simulation as faithful and precise as possible, replicating the intricacies and nuances that typify real-life scenarios. In doing so, we embark on a journey to uncover insights and enhance the robustness of sound source localisation in practical contexts.

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: Acotriangulator

2.3 Project Management Tool

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

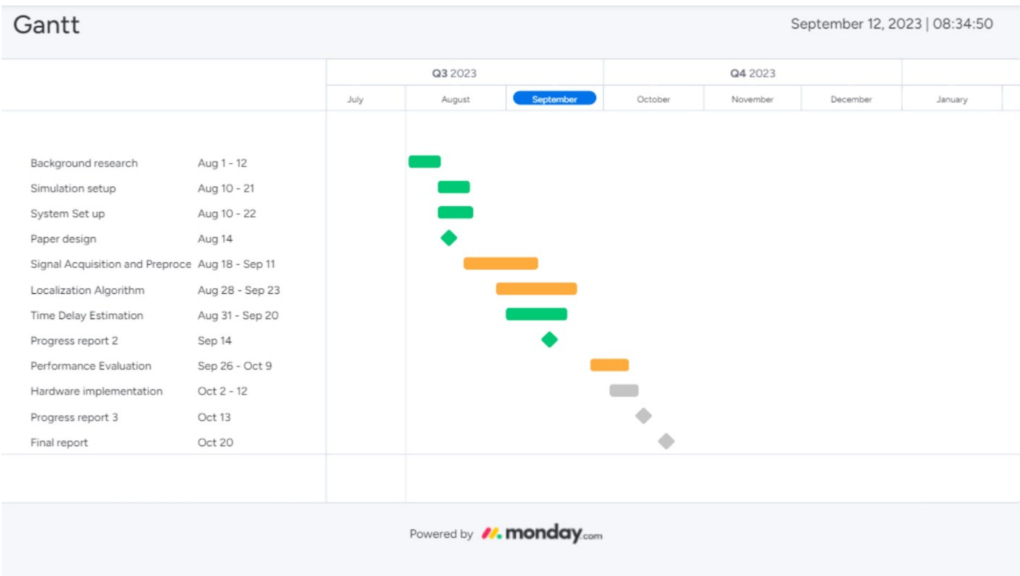

Figure 1: Project Management Tool

2.4 Tutor Comment

Demo not required refer to the TAs email dated 11th September 2023.

2.5 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:

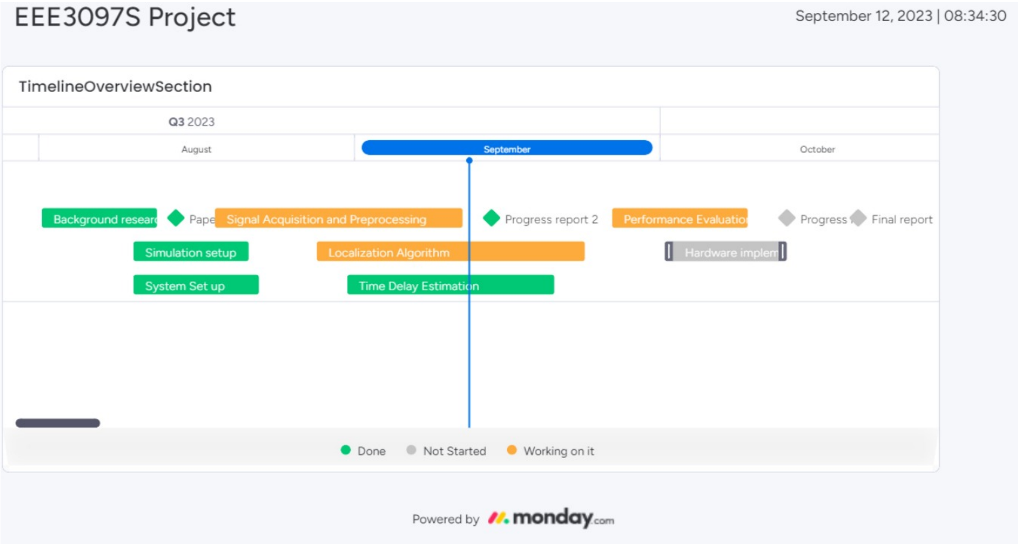

Figure 2: Timeline and Progress

3. Simulation Setup

The simualtion files can be found at: Simulation Files

3.1 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.

3.2 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.

3.3 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.

4. System Design and Implementation

4.1 Subsystems to be Simulated

4.1.1 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 $\times$ 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).

4.1.2 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 $\times$ 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.

4.1.3 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 × 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 30 dB.
- 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 30 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.

4.2 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.

4.3 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.

4.4 Algorithms/Techniques for TDOA Calculation

4.4.1 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.

4.4.2 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.

4.4.3 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 localization in our system.

5. Simulation Results and Analysis

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.

5.1 Time Delay Estimation

5.1.1 Preliminary Observations

Before advancing to the exhaustive 100-iteration simulation phase, it is imperative to perform preliminary tests. These initial investigations aim to evaluate the efficacy of the `gccphat` function in time-delay estimation between various pairs of microphones.

The estimated time delays, extracted via the GCC-PHAT algorithm for each microphone pair, are as follows:

- Time Delay between M1 and M2: -0.000013 s
- Time Delay between M1 and M3: 0.000002 s
- Time Delay between M1 and M4: -0.000012 s
- Time Delay between M2 and M3: 0.000014 s
- Time Delay between M2 and M4: 0.000001 s
- Time Delay between M3 and M4: -0.000013 s

These results serve as preliminary indicators of the system's capability to estimate time delays accurately and will form the basis for subsequent, more complex analyses.

The following figure presents these plots:

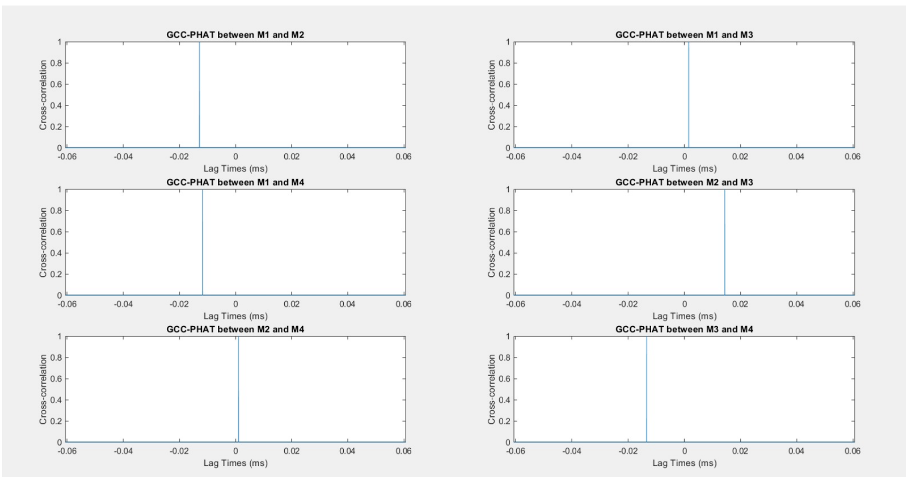

Figure 3: GCC-PHAT Plots

Each plot's peak serves as a graphical representation of the estimated time delay for the corresponding microphone pair. These peaks signify the effectiveness of the `gccphat` function, thereby substantiating its reliability for time-delay estimation.

5.1.2 Comprehensive Results

After this preliminary step, we moved on to the 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:

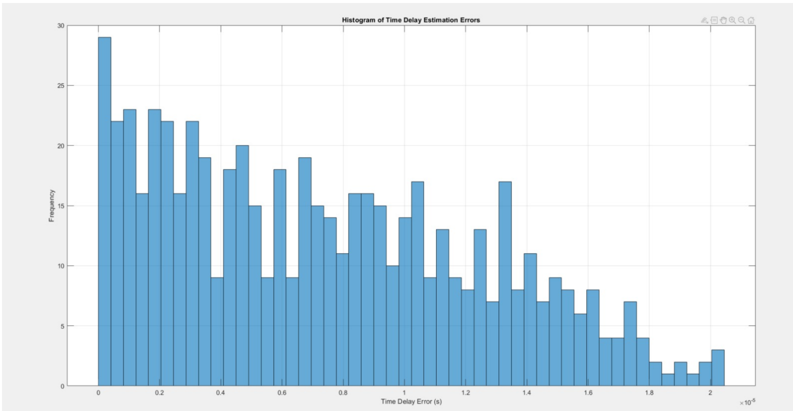

Figure 4: Histogram

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**: $4.980 \times 10^{-6} s$

5.1.3 Analysis

With the addition of the GCC-PHAT cross-correlation plots, we've gained another layer of verification for the efficacy of the `gccphat` function in our system. These plots serve as a qualitative confirmation of the function's accuracy, which is quantitatively supported by the low mean and standard deviation in the time-delay estimates.

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.

5.2 Triangulation

5.2.1 Results

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:

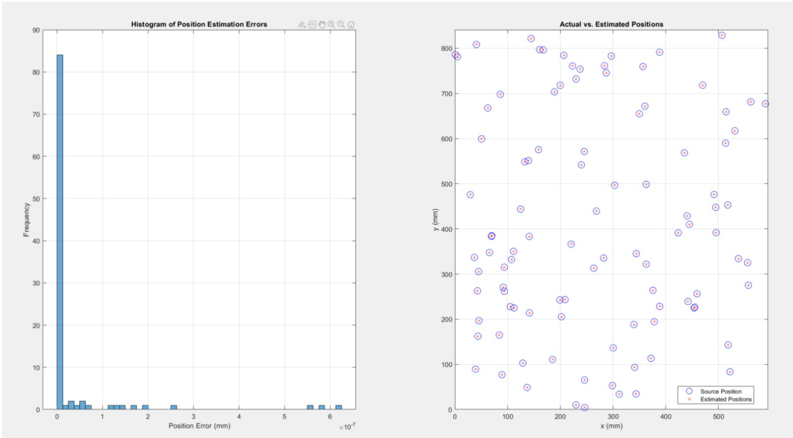

Figure 5: Histogram and Position Plot

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

5.2.2 Analysis

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.

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.

The `lsqnonlin` function proved to be highly effective in performing the triangulation, making it a reliable method for applications requiring high-precision source localization.

5.3 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.

5.3.1 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.

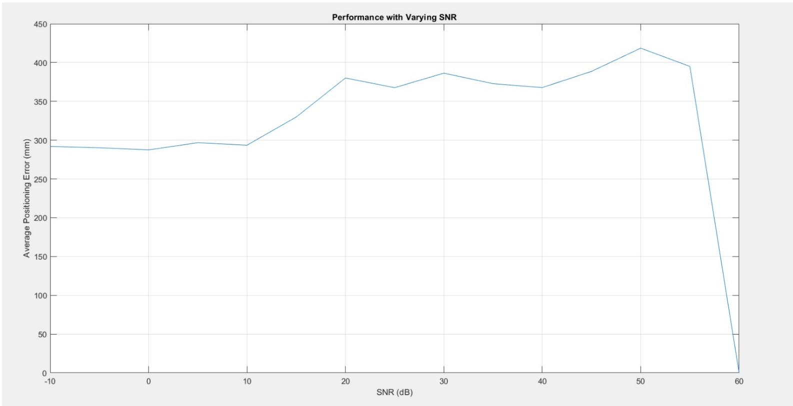

Figure 6: SNR Plot

5.3.2 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

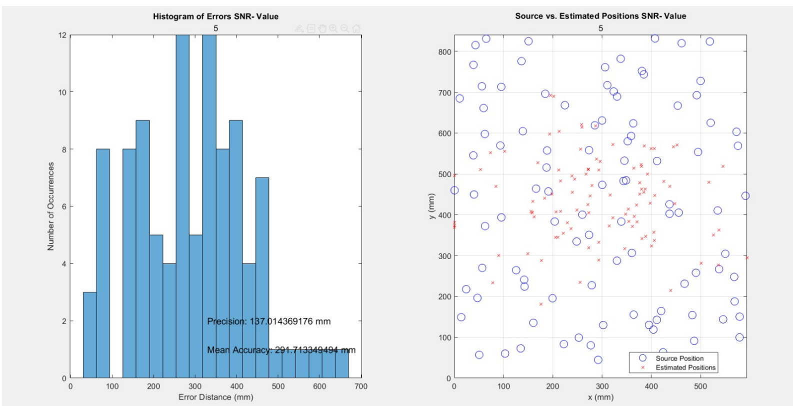

Figure 7: 5 dB Performance

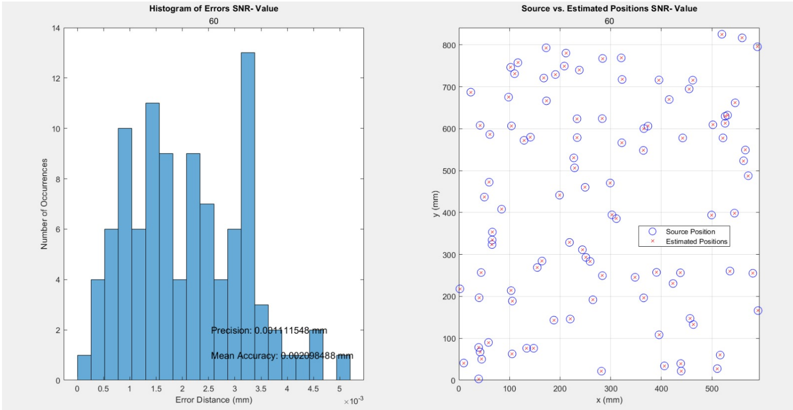

Figure 8: 60 dB Performance

5.3.3 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.

5.3.4 Encountered Challenges and Mitigations

1. **SNR Sensitivity**: The average positioning errors tend to rise at lower SNRs, undermining the system's reliability.
2. **Microphone Calibration**: While existing calibration accommodates some variations, evolving microphone characteristics across time or different setups could introduce errors.
3. **Synchronization Errors**: We faced challenges in accurately simulating clock synchronization errors among the system components. This would be critical when shifting to the hardware implementation.

5.3.5 Suggestions for Overcoming Obstacles in Transition to Hardware

1. **Periodic Calibration**: Implementing regular calibration protocols can help adapt to changing microphone conditions, thus reducing error rates.
2. **Advanced Signal Processing Techniques**: Utilizing sophisticated methods like adaptive filtering can enhance source localization, especially in lower SNR settings.

By methodically addressing these challenges, we aim to further refine our system's robustness and reliability in diverse scenarios.

6. Evaluation of Simulation Against ATPs

6.1 Recreated ATPs

The simulation primarily focused on the Time Delay of Arrival (TDoA) and Position Calculation subsystems of the Acotriangulator System, as outlined in the original Acceptance Test Plan (ATP).

6.2 Table: Evaluation of Simulation Results

Table 2: Simulation Test Results for Sound Source Localization

ATP No.	Subsystem	Objective	Expected Results	Results	Status
5A	TDoA Calculation	Validate the accuracy of TDoA calculations	TDoA error < 1ms	TDoA error: 7.128×10^{-6} s	Pass
5B	Position Calculation	Validate sound source position calculations	Position error < 0.5 meters	Position Error: 0.000000049 mm	Pass

6.3 Analysis

6.3.1 ATPs Met:

- **TDoA Calculation (5A)**: The simulation successfully validated the TDoA calculation accuracy, with a mean time delay estimation error of 7.128×10^{-6} s and a standard deviation of 4.980×10^{-6} s. These results surpassed the expected performance, implying no immediate need for improvement.
- **Position Calculation (5B)**: The Position Calculation subsystem showed exceptional accuracy with a mean position estimation error of 0.000000049 mm and a standard deviation of 0.000000123 mm. This also surpasses the ATP requirement, indicating no changes needed.

6.3.2 ATPs Not Simulatable:

Reasons for Non-Simulatable Subsystems:

- **Sound Emission Subsystem**: This subsystem requires physical hardware to emit sound at different frequencies and a calibrated meter to measure those frequencies. As it is a hardware-dependent process, it cannot be simulated.
- **Hardware Setup and Management Subsystem**: The proper layout and installation of Raspberry Pis and microphones are physical activities that need actual hardware. They cannot be represented or validated purely in a simulation environment.
- **Audio Acquisition and Processing Subsystem**: This involves capturing real-world audio, including background noise, and processing it. It requires actual microphones and ambient conditions, thus cannot be simulated.
- **Data Transmission and Reception Subsystem**: Actual hardware and a network setup are required to validate the scp command for data transfer. The integrity of the transferred file also needs to be checked on real equipment.
- **GUI Subsystem**: While certain GUI features could potentially be simulated, the full integration and response time of the GUI depend on the real-time data and hardware interaction, which is beyond the capabilities of the current simulation setup.
- **System Testing and Validation Subsystem**: This involves end-to-end testing of all integrated components, including hardware and software, which cannot be fully represented in a simulation environment.

By focusing on the simulatable aspects, the table offers a streamlined view of what has been validated so far in the simulation, paving the way for subsequent physical tests.

6.3.3 Potential Improvements

While the simulated subsystems exceeded the expectations set in the ATPs, the next phase involving physical implementation may introduce new challenges. Possible improvements might include:

- **Robustness to Noise**: Although not a concern in the simulation, adding algorithms to filter out background noise can improve performance in real-world conditions.
- **Optimization for Computation**: Implementing faster algorithms for real-time processing could be advantageous when transitioning to hardware.

6.3.4 Next Steps

The next steps in the project involve:

- Transitioning from simulation to physical implementation.
- Conducting field tests to validate the TDoA and Position Calculation subsystems with real-world data.
- Implementing the remaining subsystems like Sound Emission, Audio Acquisition, and GUI to make the system fully functional.

By methodically following these steps, the project aims to transition seamlessly from the simulation phase to a complete, functional system.

7. Conclusion

In conclusion, our simulation underscores two fundamental principles for optimising sound source localisation systems. First and foremost, we have established the importance of minimising noise interference. Noise, in all its forms, challenges the accuracy and reliability of sound source localisation. Our findings from the MATLAB scripts emphasise the need for comprehensive noise mitigation strategies to enhance the efficacy of these systems.

By utilising sophisticated algorithms like Generalised Cross-Correlation Phase Transform (GCC-PHAT) and Nonlinear Least Squares Estimation, the simulation provides insights into various influencing factors such as signal type, calibration errors, and noise levels. Key metrics like mean accuracy and precision were calculated to measure the system's performance quantitatively. Results were visualised through histograms and scatter plots for easy interpretation.

In summary, this simulation is a comprehensive tool for analysing the efficacy of sound localisation using a 4-microphone array. It offers valuable data that can guide the development of the actual physical system.