Acoustic Source Localization

ECE4873 Senior Design Project

Team Name (Spot)
Team Number (sd22p11)
Project Faculty Advisor (Dr. Ma)

Tiffany Ho – Major: EE, GT Account: tho65, email: tiffanyho@gatech.edu
Ajeetpal Dhillon – Major: EE, GT Account: adhillon30, email: adhillon30@gatech.edu,
Harry Nguyen – Major: CmpE, GT Account: hnguyen363, email: hnguyen363@gatech.edu
Sidong Guo – Major: EE, GT Account: sguo93, email: sguo93@gatech.edu
Andrew Dulaney – Major: EE, GT Account: adulaney, email: adulaney7@gatech.edu
Daniel Scarborough – Major: EE, GT Account: dscarborough3, email: dscarborough3@gatech.edu

Submitted

2022 May 2

Table of Contents

Executive Summary

RF and Wi-Fi localization has been the center of research and development for many decades, but acoustic localization remains an active area of research with algorithms like ODAS (Open Embedded Acoustic System). The objective of this project is to explore the concept of source localization for acoustic signals; evaluate, research, and devise an inexpensive, accurate algorithm that incorporates tracking and can have estimation results displayed on a flexible GUI (Graphical User Interface) This software can be applied in a classroom setting for automatic camera angle adjustment, can help strengthen various surveillance technologies, and be used in various search-and-rescue operations when visibility is limited. The project addresses the high complexity of other algorithms by utilizing a novel, less computationally expensive localization approach that utilizes TDOA (Time Difference of Arrival) and triangulation to feed angle information to be displayed on a GUI. The angle information is established through averaging multiple microphone pairs through far-field assumption and outlier filters. The core product of the project is a software algorithm that does not cost anything. To demonstrate and evaluate the project, we used a Raspberry Pi3 and its MATRIX VOICE module, which costs around 200 dollars, depending on the exact model and manufacturer and there were no other sources of expenditure. The project achieves at least 8 degrees angular accuracy in location estimation in a classroom setting and an operation range of at least 20 meters. The algorithm should also support sampling frequency no greater than industry standard 96 kHz and real time display on GUI. For future improvements, the algorithm can incorporate RF signal estimation and neural network for more sophisticated tracking algorithms.

Index:

1. **TDOA (Time Difference of Arrival): A location estimation algorithm that uses the difference in arrival time of receive signal to extract position information.**

2. **GCC (Generalized Cross-Correlation): A time-delay estimation method that uses cross-correlation with certain weighting to estimate the time difference between two signals.**
3. **PHAT (Phase Transform): A GCC weighting that helps to heighten the peak.**
4. **MATRIX: A microphone array module compatible with Raspberry Pi.**
5. **LLS (Linear Least Square): An estimation theory method that approximates a set of non-linear equations into equivalent matrix representations for closest estimation.**

Acoustic Source Localization

1. Introduction

Acoustic Source Localization using microphone array is an active research topic for the past decades. With the development of communication systems, outdoor localization is dominated by GPS (Global Positioning System) and leaves little room for acoustic signals. As the focus of research shifts from outdoor to indoor, there is a rekindle of interest in acoustic signals. TDOA is one of the main methods for indoor source localization. TDOA acoustic source localization is composed of two main steps: Time difference estimation and triangulation. The use of cross-correlation for estimating time delay between pairs of microphones was well documented in C. Knapp and G. Carter's paper. Different geometries of microphone and time estimation methods were continuously developed. The measurement errors in timing information mean that a closed-form calculation of position is not possible and therefore requires linear transformation through estimation theory or parallel assumptions. Methods like Gillette-Silverman algorithm and other linear algorithms were developed and proposed to solve this problem. But after extensive testing and simulation, we decided that they were either inaccurate or very geometry specific and therefore lacking versatility. The other methods proposed use far-field simplifications, which is unattainable in an indoor environment and will always produce an estimation error. This project, however, uses multiple trigonometry calculations by assuming far-field. Therefore, by averaging over multiple angle calculations, a reasonable value can be attained between upper and lower bounds. By far the most established algorithm is ODAS, which utilizes beamforming and Hierarchy Search Matching. Although ODAS is accurate, it is extremely hard to translate to other geometries.

Objective

Our team created a software algorithm that can detect the direction of a sound that is generated in an area with desirable accuracy and simplicity. The software tracks the incoming sound and the positions on a GUI. The algorithm compensates for the inaccuracy and restrictiveness of some other models.

Motivation

While a lot of detection is done using cameras, infrared technology and Wi-Fi, our group designed an algorithm that can determine location based on incoming acoustic signals. We hope this software is more accessible than infrared and more flexible than cameras. We would like owners to be able to place the device down in a room and determine where an acoustic signal is coming from. This could be used in classroom settings, tracking hyperactive kids, or places that care for elderly people who might wander off.

Background

While there are multiple ways that a location can be determined, our project will focus on TDOA. TDOA is derived from a different method called time or arrival (TOA). TOA calculates the position of an acoustic signal based on the time it takes to reach a receiver. It is easy to implement, but not very practical because the transmitter and receiver are not synchronized. Because the timing information is unknown in locating speech signal, our device focuses on TDOA, which can determine the direction of a source without knowing its starting time. Because sounds do not move instantaneously, we use the difference between the arrivals to determine the location of the source. This is a good method, but it heavily depends on the location of the project's microphones and estimation precision. Because we are using microphones that are spaced closely together, the distance cannot be reliably calculated. Another method we investigated was receive signal strength. We theorized it could be used determine the distance of the source. This method is not perfect because the original amplitude of the signal is not known and without it, distance would be hard to calculate.

2. Project Description, Customer Requirements, and Goals

The team designed a real time localization software that uses a Matrix Voice microphone array to capture acoustic signals. This design includes a GUI to display the direction of the source relative to our device. This GUI is viewable on an external monitor or laptop via VNC connection.

This design is intended to be accessible to all users who need to track acoustic or speech signals within a limited radius. To ensure the ease of accessibility, the goal is to make the UI easy to read and adjust for different circumstances. Furthermore, the circular array of LEDs on the Matrix Voice is utilized to display the direction of the source. The LEDs visualize the same data as the GUI in a simple format for ease of interpretation.

The goal of this team is to create an algorithm with adjustable parameters to conduct point source localization in varying environments. These parameters include sensitivity, LED brightness, and degrees of freedom (accuracy). This product could be used in a variety of fields: emergency response, surveillance systems, and speech/target tracking.

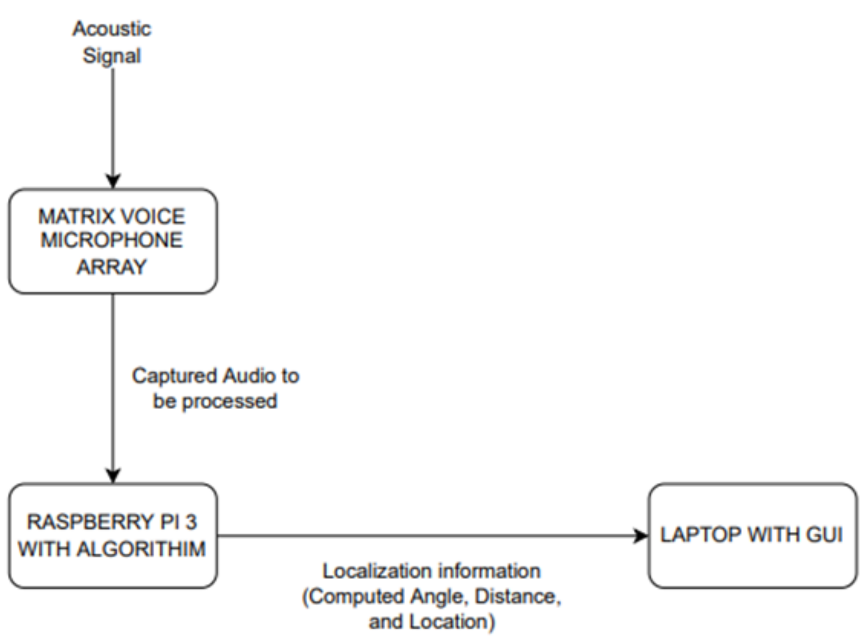


Figure 1. System Block Diagram

Our metric of evaluation in terms of customer needs are Accuracy, Convenience, Durability, Responsiveness, Flexibility and Cost. Each metric is quantified in the next section.

The project has several hardware and software constraints:

1. The software we designed interfaced mainly with Raspberry PI and its companion MATRIX module, however, the microphone parameters can be redefined and support other microprocessors if they are in C++.
2. The software does not support sampling rates higher than 96 kHz, in other words, the software is unresponsive to signals with frequency higher than 48 kHz, which is usually outside the range of speech signal.
3. Raspberry Pi has a limited memory and processing ability, increasing the sampling rate will visibly lower the responsiveness of the algorithm.

3. Technical Specifications & Verification

Table 1. Design Parameters for the project

Parameters	Specifications
Audio Specification	8-96 kHz
Bit Depth	Signed 16 bits
Operation Range	20 meters
Cost	Below $200
Angle Accuracy	Average difference within 3 degrees
Processing Delay	Average processing delay within 1 second

Table 2. House of Quality for the project

Customer and Engineering Requirements	Customer importance (Most Important =5)	Design Criterion A	Design Criterion B	Design Criterion C	Design Criterion D
Easy to Use	4	X	Y	Z	W
Reliability	5	Y	Y	X	W
Durability	3	W	X	W	W
Low Price	5	W	W	W	X
Responsiveness	2	W	Y	Z	W
Flexibility	3	W	X	Z	W

X: Strong Positive Correlation Y: Positive Correlation Z: Negative Correlation W: No Correlation

- Design Criterion A: The product needs to have an intuitive, clean user interface that supports several quality-of-life options and parameter definitions.
- Design Criterion B: The product needs to meet microphone industrial standards sampling frequency and operate well within design parameters posed by MATRIX module.
- Design Criterion C: The product needs to have reliable localization and tracking algorithms that meet the accuracy specifications outlined in table 1.
- Design Criterion D: The product needs to minimize costs and must not exceed 1500$.

4. Design Approach and Details

The end product of the project uses Time Difference of Arrival (TDOA) scheme on a circular MATRIX VOICE microphone array and performs timing information acquisition and multi-lateration on incoming waves to tell the direction of the acoustic source.

Compared to the initial proposal, the project completed the following design aspects:

1. The algorithm provides directional information on random speech signal
2. The algorithm has an operation range of 20 meters (provided the incoming sound is audible at given distance)
3. The algorithm incorporates a tracking algorithm and dynamically locates the source when the source is moving
4. The algorithm updates and provides real-time display on GUI
5. Users can dynamically change certain design parameters in the algorithm via GUI
6. The Algorithm can combat background noise under certain average power floor

The project did not resolve the following design aspects:

1. Provides distance information on acoustic waves with known characteristics
2. Provides multi-source localization

A. Provide directional information on random speech signal

For any random speech signal or any other signal that can be characterized as a random acoustic wave, the algorithm goes through the following steps to provide directional information:

1. The main Python file runs a C++ executable, that records the received acoustic information as 9 separate raw files (8 microphone channel with a beam-forming channel) with two limiting parameters: Length and Sampling rate. Tentatively, they are assumed to be 1 second and 16 kHz for the rest of the document.
2. The main Python file runs a series of terminal commands on Raspberry Pi terminal that transforms raw files into standard wav files. The Python file then imports wav files and stores them as 9 separate vectors for further signal processing.

3. Generalized Cross-Correlation (GCC) was used between 7 microphones on the external circle and the middle reference microphone.
4. Peak detections are performed on correlation functions and from index values, by dividing the sampling rate and multiplying speed of propagation, produces 7 DDOA (Distance Difference of Arrival) information with respect to the acoustic source.
5. 7 microphones on the external circle partition the XY plane into 7 angular domains. The algorithm first determines which angular direction the acoustic signal is coming from.
6. For each angular domain, a different combination of 4 DDOAs are picked to produce 4 DOA (Direction of Arrival) estimation based on far-field assumptions.
7. 4 DOAs (Direction of Arrival) are passed through a set of error detection and outlier detection filters to reach a final DOA information with respect to the relative X-axis on the XY cartesian coordinates.

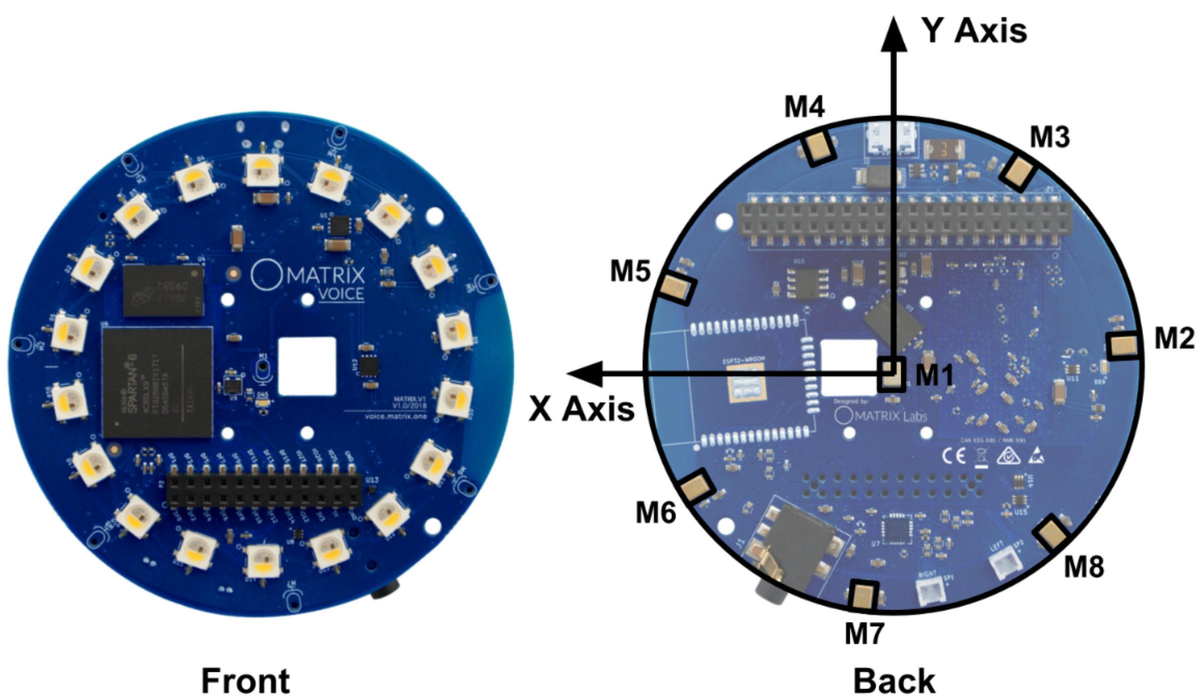


Figure 1. MATRIX VOICE microphone distribution and relative XY coordinate

The entire algorithm is characterized by the following flow graph:

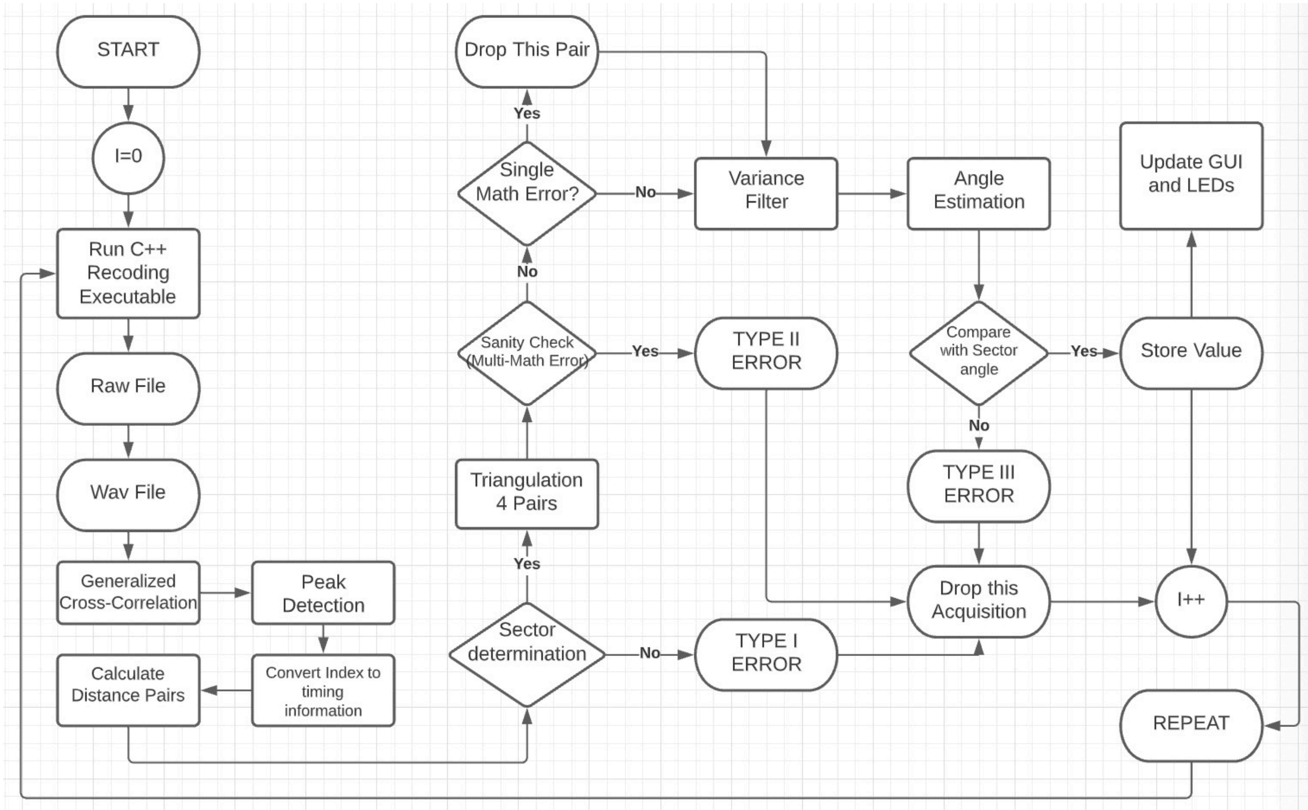


Figure 2. Algorithm software flowchart

Elaboration on GCC:

Generalized Cross-Correlation is when a weighting factor is applied before Cross-Correlation with an attempt to combat background noise and/or provide other desirable properties. Some of the more notable weighting factors include ROTH, PHAT, Smoothed Coherence Transform. For our algorithm, GCC-PHAT is selected for its ability to heighten the correlation peak and therefore reduce measurement error. The proof is provided in Appendix A.

Elaboration on Angular domain determination:

For any acquisition, the algorithm runs a set of nested if statements that are mutually exclusive to determine the angular domain of acoustic wave. Each if statement selects a pair of adjacent microphones along with 2 other microphones that are 1 microphone apart from the pair. For instance, one of the if condition entails that sound captured on M3 and M4 must arrive earlier than M1 and that

M6 and M8 arrive later than M1. If this condition is met, then we know the source has a direction between M3 and M4. Similarly, another condition uses M2, M3, M5 and M7.

Elaboration on far-field assumption and triangulation:

To illustrate the triangulation algorithm, consider the following scenario:

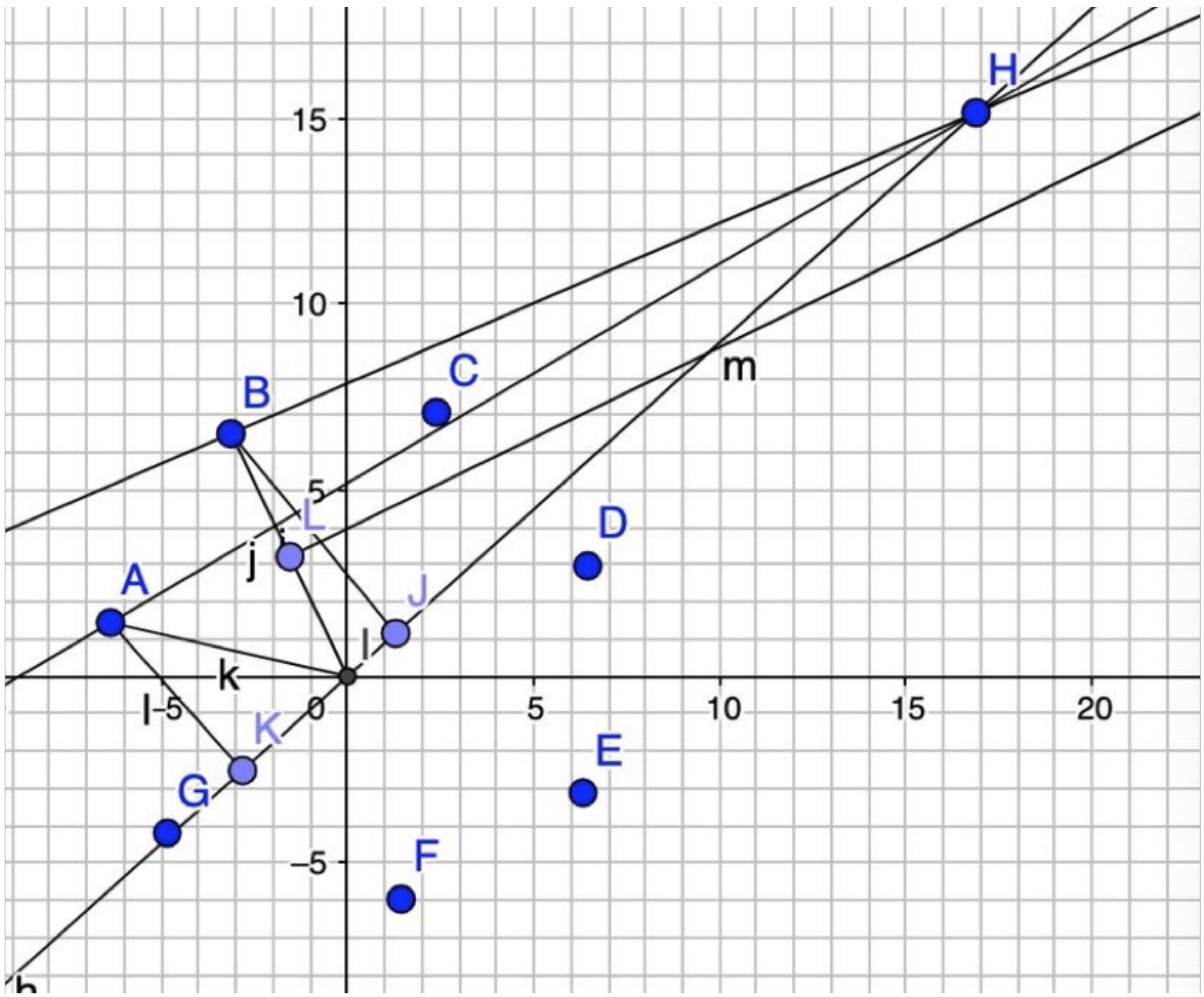


Figure 3. Triangulation

A sound source H where distance CH is far greater than CD, is radiating towards the microphone array. Microphone A receives a wave that has a delay and an independent noise than the wave received at microphone I. From previous processing, the distance difference is already obtained. Here, we assume radiating lines AH and IH are parallel and therefore the distance difference measured is KI (due to low frequency of carrier wave, this does not attain the far-field criterion in EM wave, so it will produce some errors, but we consider this measure superior to other methods discussed in following sections). Since AKI is a right triangle, the angle AIK can be obtained through triangulation, and therefore the angle H makes with the x-axis can also be obtained. Notice that due to the parallel assumption KI being shorter than the measured distance difference, the set of measurements produces an underestimation of H's AOA (Angle of Arrival). Vice versa, using microphone B to perform triangulation will also give an AOA, but this time with an overestimation. The algorithm therefore selects microphone A, C, D, F in this case for 4 sets of triangulation measurements.

Elaboration on error control:

It can be clearly seen that no distance difference between any microphone on the exterior and reference microphone should be greater than radius. However, due to measurement errors on hardware caused by thermal noise and multipath (Multipath where a reflected component with higher strength will cause the correlation peak to shift towards the right and produce a distance difference larger than reality), that value can sometimes be greater than radius, which in Python, will cause math error during triangulation. The first step of the error control algorithm is to disregard any measurement that produces math error. If 2 out of 4 measurements are corrupted, this acquisition is dropped, and algorithm moves on to next acquisition. After first filtering, the remaining measurements are passed through a variance filter where the smallest variance combination of (N-1) measurements are regarded as the final set of measurements, and the remaining measurements are averaged to produce the angle Estimation. Lastly, the Estimation will be compared to initial angular domain. If the Estimation is outside of the angular by a large margin (determined by Degrees of Forgiveness in following section), this acquisition is also dropped.

B. The algorithm has an operation range of 20 meters

To dynamically adjust the operation range of the device, we employ a method of thresholding. Each channel (media between microphone and source with which the acoustic wave propagates) has a separate gain and we statistically averaged across all background noise and constructed an average noise floor (mostly due to thermal noise). The baseline threshold is therefore the noise floor, where it is extremely sensitive to any sound that increases the sound floor above the background noise floor. During testing, a human with moderate clapping sound can be detected from 10 meters away.

C. The algorithm incorporates a tracking algorithm and dynamically locates the source when the source is moving

The process outlined in A is repeated every 1 second, and therefore can follow the sound source. The acquisition time can be changed in CPP file and generates a new executable. However, there is a visible tradeoff between sampling rate and processing time which we will discuss below.

D. The algorithm updates and provides real-time display on GUI and users can dynamically change certain design parameters in the algorithm via GUI

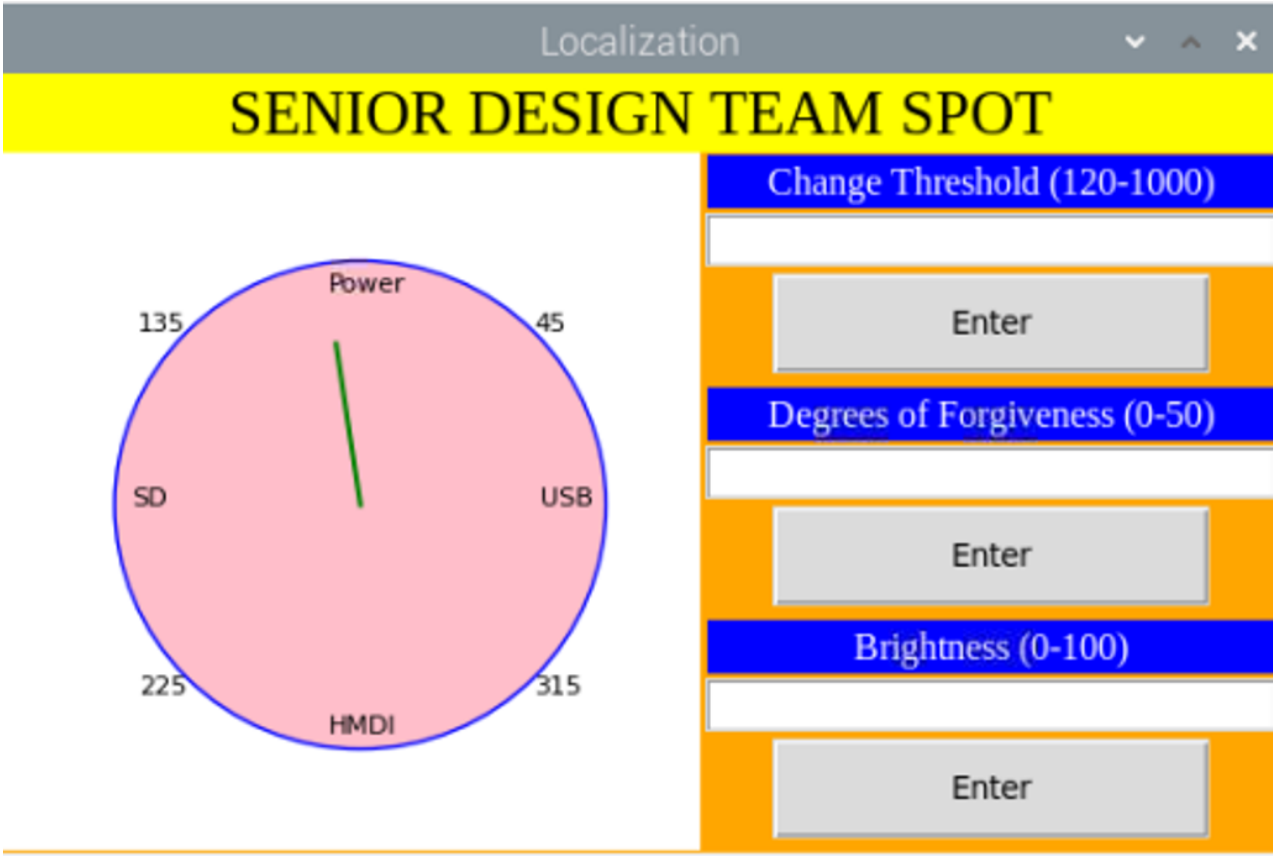


Figure 4. GUI display

Every second, the GUI (made with Python Tkinter) will plot out the direction of the source with respect to the relative Cartesian coordinates. Labels are also included on this plot to indicate the orientation of the GUI with respect to the Matrix and Pi. The GUI provides three User interactions, Threshold is inversely related to operation range; Degrees of Forgiveness will decrease the acquisition accuracy but reduced the number of times an acquisition is classified as an error. Brightness will change the LED brightness during demonstration.

Recommendations for future improvements

1. To provide distance information, one has two options: First is to utilize a linear multi-lateration approach that jointly estimates x and y coordinates, we will discuss its disadvantages and advantages in the next section. Second is to use receive signal strength (RSS) to establish a

pathloss model. However, during testing, we have noticed that due to random Gaussian noise, the minimum difference in signal strength will produce a large distance error in the near field. Substantial further research is needed in this area.

2. We have not resolved the problem with multiple acoustic sources, and it is a promising direction of research. The realization requires successive cancellation of one user from overall received signal and further signal processing research needs to be conducted.
3. The algorithm would benefit from a more systematic tracking method such as the Kalman filter, where the acoustic source is modelled as a gaussian process.
4. Due to hardware constraints, the acquisition window cannot go below 1 second, which makes it impossible to conduct ensemble average across acquisitions. Future teams can investigate ways to improve hardware and processing speed of the micro controller.

4.1 Design Concept Ideation, Constraints, Alternatives, and Tradeoffs

We made the following decisions regarding the overarching direction of the projects:

- We decided to use 2-D instead of 3-D, because of the limitation of MATRIX VOICE module with no z-axis microphone component. In addition, during simulation in MATLAB, the elevation of the human will in fact only minimally affect the angle estimation.
- We decided in favor of acoustic signal localization over RF or Wi-Fi signal localization. The reasons are acoustic signal provides more utility in indoor environments acoustic and support passive localization as in the user does not carry any device.

Software

A. Localization Scheme

We then came down to deciding the exact overarching algorithm to go for based on a decision matrix. The performance is measured on a scale of 1-5 with 5 being the best. TOA, TDOA, RSS's relative difficulty for Wi-Fi has been established in *Liu-Survey on Wi-Fi,* and we adjusted the metric for acoustic signals.

Table 3. Decision Matrix of all Active Localization method considered

	TOA	TDOA	RSS	Interferometric
Easiness to implement	4	3	2	2
Accuracy	4	3	2	5
Novelty	2	2	3	4
Adaptability	1	4	3	1

We decided TDOA is the better approach based on the following grounds:

1. Time of Arrival requires synchronized clocks and for random acoustic signal, it is impossible to sync on any level.
2. RSS suffers from changes in topology and background noise, considering the design expo will have a very different environment than the lab where the algorithm is tested, it is very risky and undesirable.

B. Timing information Acquisition

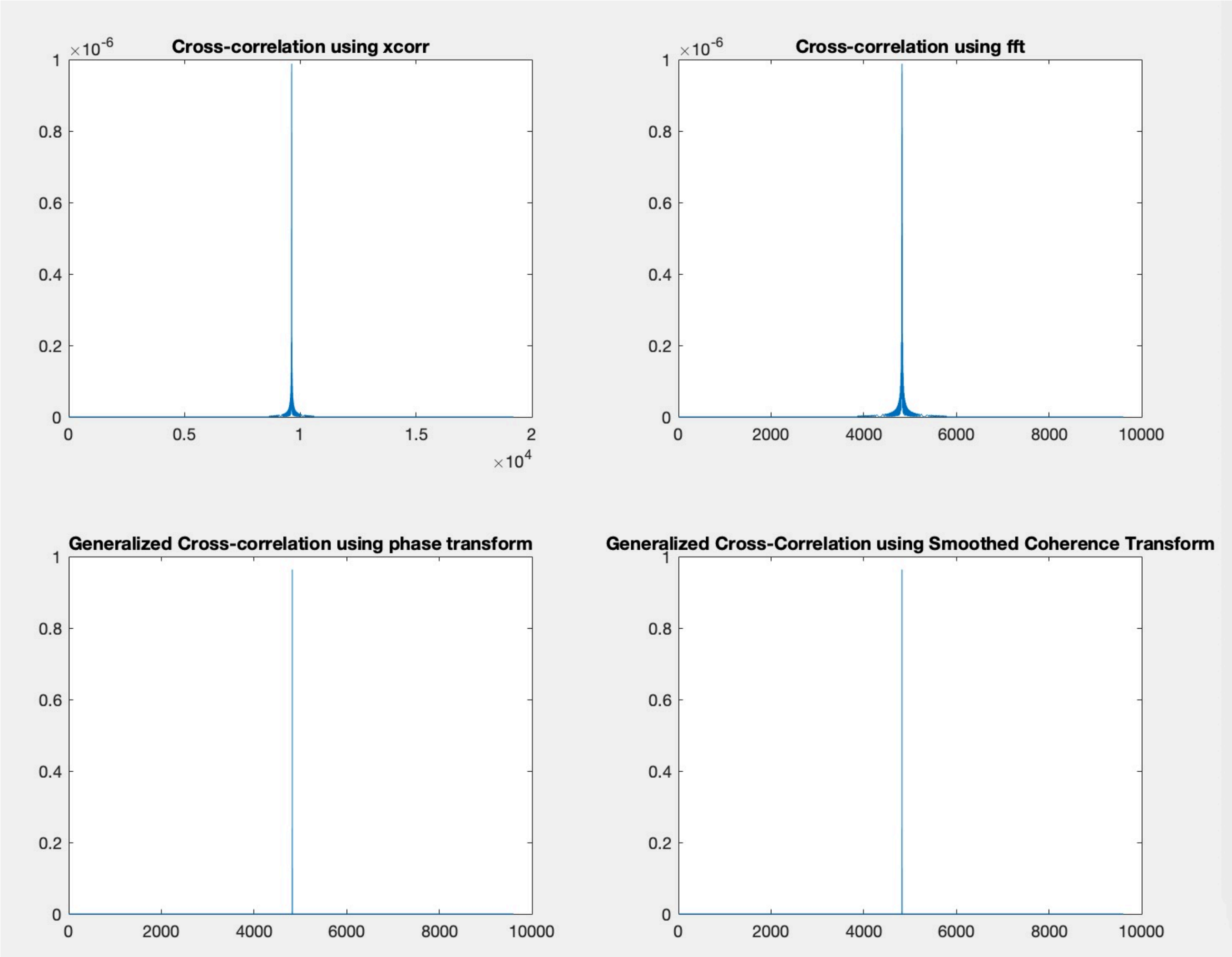


Figure 5. Cross-Correlation Effect Comparison

Compared to no weighting, Phase Transform and Smoothed Coherence Transform provides a significant peak heightening effect and therefore selected in algorithm. Compared to PHAT, SCOT has a widening effect under noisy conditions that makes it less desirable.

C. Multi-lateration Technique

Instead of far-field assumption, the most popular competing method falls under the category of linear approaches. Linear approaches attempt to convert a set of non-linear equations to a linear matrix equivalent form and use estimation theory (Maximum Likelihood Estimator (MLE) and Least Square (LS)) to jointly solve for X, Y and distance. A complete derivation for TDOA Linear approach is provided in Appendix B, where we have considered Linear Least Square (LLS), Weighted Linear Least Square (WLLS) and Constrained Weighted Linear Least Square (CWLLS), Grid search.

The theoretical performances of the linear estimators are compared to CRLB and CWLLS attains CRLB in most cases. However, we noticed several disadvantages:

1. Computations involving the inversion of square matrix will sometimes break Python because the matrix is ill-conditioned (singular or close to singular).
2. Matrix inversion and five root equation in CWLLS, has extremely high computation complexity and while efficient enough on a regular computer, is extremely slow on Raspberry Pi and therefore defeats the purpose of a real-time algorithm.

Hardware

A. Microphone array structure

1. We decided to use MATRIX VOICE instead of designing our own PCB or using several anchor points each with one antenna for active localization. The rationale is that for centimeter level localization, any small difference between microphone pairs will contribute to a large error margin. Producing PCB is unrealistic because through hole precisions are very demanding and any change in design parameters or algorithm means the PCB needs to be remade. The advantage of using a microphone array over individual microphone as anchor point is that it is more portable and does not require active measurements of distances between microphones. (If someone moves the anchor point, the entire algorithm will be off).
2. We have decided to use MATRIX VOICE over MATRIX CREATOR because 7 microphones on the exterior is more beneficial in selecting 2 pairs of symmetric microphones for triangulation.

B. Matrix Library and Dependencies

1. Initial setup steps were followed for the Matrix Voice using the official Matrix documentation. Through trial and error, it was discovered that only a certain raspberry pi OS would be compatible with the Matrix libraries. Specifically, we found that the Buster OS was the least resistant to the number of dependencies we needed to be installed.
2. Several Matrix libraries are available to interface with the Matrix Voice microphone array. Initially we considered using the Python library strictly, but this produced errors in data acquisitions shorter than 5 seconds. The alternative option was to make use of the Matrix HAL (hardware abstraction layer) library. This library written in C++ needed to be interfaced with our algorithm written in Python. Our work around this was to make the 1 second data acquisition a C++ executable to be called via our Python script.
3. When installing the Matrix HAL library from package a few issues came in the form of dated documentation. Research and troubleshooting became a large part of setting up the Matrix Voice to be used with our raspberry pi. The links provided to add the Matrix HAL repository and key were incorrect and had no information on which OS had a release. This troubleshooting contributed significantly to our time constraints.

GUI

A. Design Approach

1. The design approach taken for our GUI had gone under several revisions before the end of the semester. The GUI sub team considered an interface implemented in C++, C#, and Python. After some consideration of the team's experience and the decision to implement the algorithm in Python, Tkinter was chosen as the main package for the GUI. Tkinter is a Python package used for

implementing GUIs. The object-oriented layer of this package was easiest to use for the GUI sub team. Initially, many customizable settings and options were envisioned for the GUI, however,

4.2 Engineering Analyses and Experiment

The analysis is done with MATLAB simulation using the phased array toolbox in a free space propagation environment with FM wave.

Prior to actual implementation, the simulation has demonstrated the viability of Generalized Cross-Correlation with Phase transform. This method achieves high accuracy in a free-space simulation environment and 96K sampling frequency, the widely regarded industry standard microphone sampling rate, and is supported by the MATRIX VOICE module.

Attached is a simulation result of 8 pairs of distance difference of arrival of microphones. The microphones are labelled 1 to 6 and are positioned in terms of relative positioning axis on (0,0,0), (d,0,0), (0, d,0), (-d,0,0), (0, -d,0), (0,0, d), where d is the microphone spacing with a value of 20 cm.

Table 3. Estimated and actual difference of distance of arrival

	D12	D13	D14	D15	D16	D23	D24	D25
Estimated	0.1842	0.2833	-0.2054	-0.2975	0.1842	0.0992	-0.3896	-0.4817
Actual	0.1864	0.2863	-0.2013	-0.2955	0.1864	0.0999	-0.3877	-0.4819

Theoretical analysis with MATLAB demonstrated that an average angle error of 1.697 degrees is produced with the algorithm outlined. Random sources are chosen scattered on the XY plane. The following is the estimated and actual position in one such acquisition.

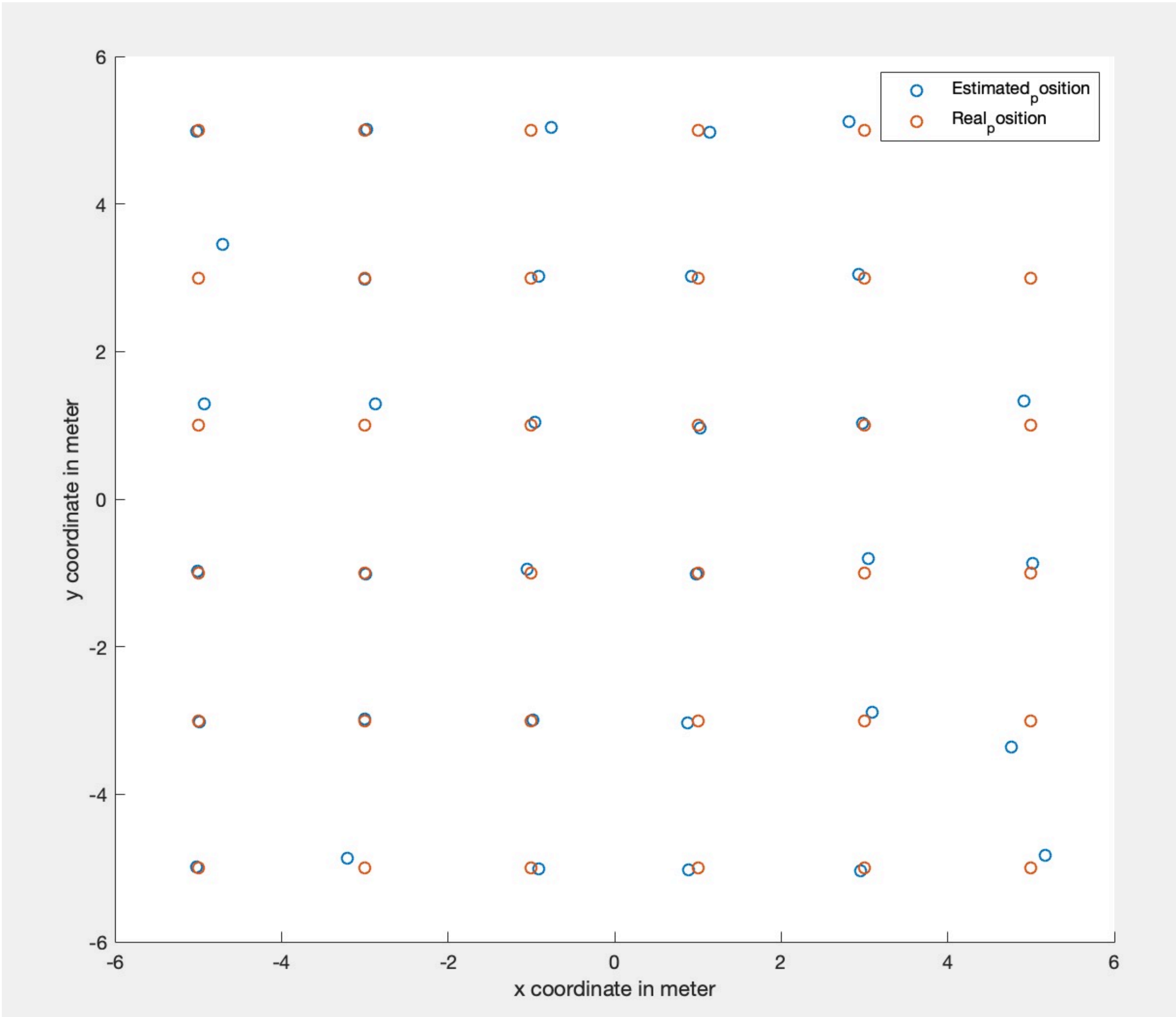


Figure 6. Estimated and Actual position in simulation

We also conducted indoor error analysis. The speaker randomly produces sound from any direction and with 16 kHz the average angle error is 7.6375 degrees, significantly higher than theoretical value. The result is easily explainable: there is a tradeoff between sampling rate and processing speed. The simulation uses 96 kHz sampling rate and attains a higher accuracy. This is because with limited sampling rate, the timing offset in terms of index between a pair of microphones can be as low as one index, when dequantized to distance information and used to calculate AOA, the arriving angles only take a finite number of values due to a lack of resolution from limited sampling rate. Experimentally we have shown that increasing the sampling rate to higher values will reduce the error and provide mor angular resolution to the algorithm. However, this will slow down the algorithm significantly because convolution scales with O(nlog(n)) and a higher sampling rate will produce a larger vector wav file within the same acquisition time. This has the undesirable effect of making the algorithm not real time. For a localization scheme we consider real time to be more important than 5 degrees of accuracy.

4.3 Codes and Standards

The project will have two coding components. The software simulation will be done in MATLAB and will be translated into Python for MATRIX LITE library. The GUI will also be made using Python. In addition, Python runs a C++ executable. The final project references will follow IEEE standard.

5. Project Demonstration

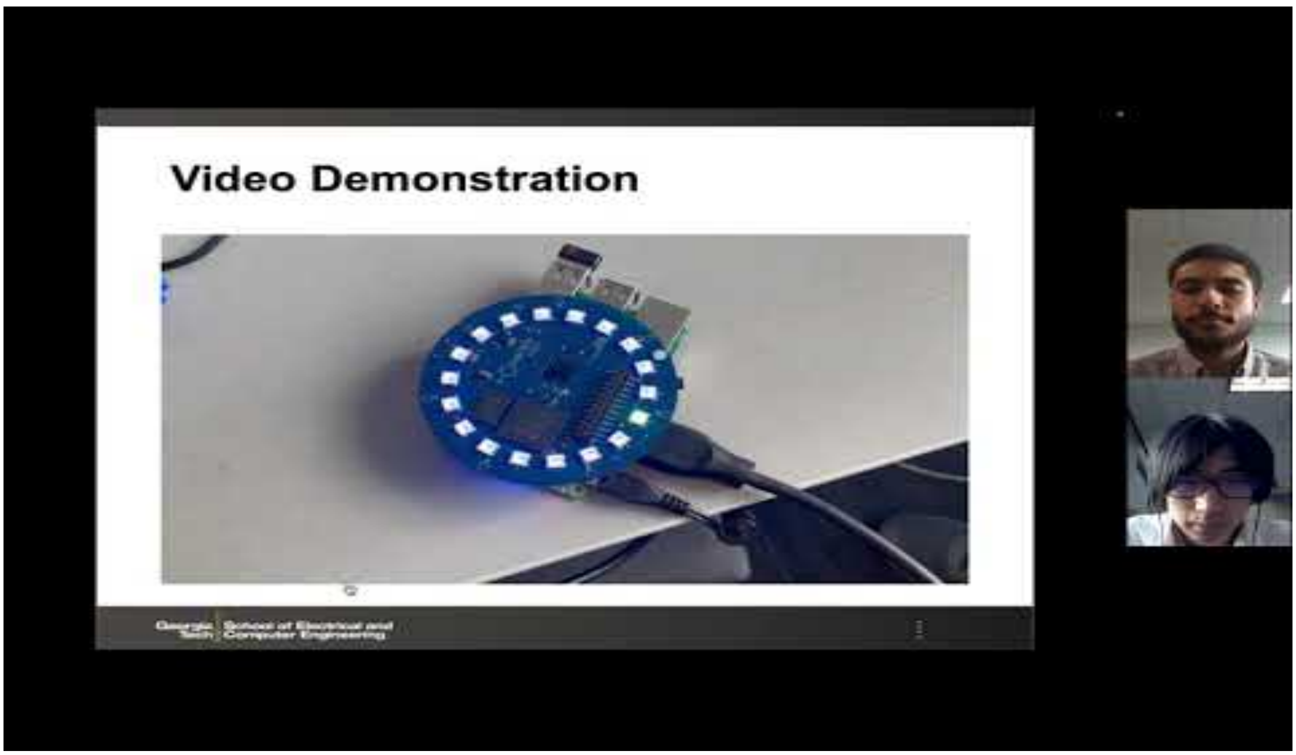
For our project demonstration, we set-up our microphone array at the senior design expo, and we increased the threshold so that it would pick up acoustic signals that were very close first, so that the large amount of background noise at the senior design expo would not negatively affect our results. We also set up a computer monitor so that the GUI could be displayed. We demonstrated:

- how to run our algorithm

- how to interact with the GUI to change the brightness of the LED display, the degrees of freedom for the angle measurements, and the operating threshold of our algorithm
- that the algorithm could work from a distance by sending acoustic signals from far away (confirmed to be functional from at least ten meters away)
- That the angle measurements are accurate based on the LED and GUI display correctly showing the angle

As a failsafe, we recorded a video of our working project showing our project working properly:

[Localization Senior Design Team SPOT (2022)]



6. Schedule, Tasks, and Milestones:

SENIOR DESIGN SCHEDULE	Week	5	6	7	8	9	10	11	12	13	14	15	16
Task													
PROPOSAL		ALL											
PROJECT SUMMARY		ALL											
GUI DESIGN			ALL (GUI)										
ORDER 2nd MATRIX			HARRY										
Design Review Presentation					ALL	ALL							
Update Project Summary					ALL	ALL							
Microphone Test			AJ										
Far-Field Test				AJ + SIDONG									
Decide on Plan A or B				AJ + SIDONG									
Build GUI (2D PLANE)			Daniel	Andrew	Andrew	Andrew							
Build GUI (Display Data)			Tiffany	Harry	Tiffany	Daniel							
Build GUI (QoL Features)				Andrew	Harry	Harry							
Test GUI							Harry+Andrew	Harry+Andrew	ALL GUI	ALL GUI			
SINGLE SOURCE TEST					AJ+SIDONG	AJ+SIDONG							
MULTI SOURCE TEST						AJ+SIDONG	AJ+SIDONG						
TRACKING					AJ+SIDONG	AJ+SIDONG	AJ+SIDONG						
CREATE/WORK ON POSTER							Tiffany+Daniel	Tiffany+Daniel	Tiffany+Daniel	Tiffany+Daniel			
GET MATERIALS FOR POSTER									Tiffany+Daniel	Tiffany+Daniel			
PRINT POSTER										Tiffany+Daniel			
INTERFACE GUI WITH ALGO								Harry+Andrew	Harry+Andrew				
CAPSTONE DESIGN EXPO								ALL	ALL	ALL	ALL		
FINAL DEMO								ALL	ALL	ALL	ALL		
FINAL REPORT								ALL	ALL	ALL	ALL		
Update Project Summary								ALL	ALL	ALL	ALL		

7. Cost Analysis

The project required a microphone array. We reached out to a few professors on campus, and Dr. Anderson had a Matrix Voice (Xilinx Spartan 6 XC6SLX9 FPGA and 8MEMSMP34DB02 audio sensor digital microphones) available for us to use. With the initial request of a $90 budget, we obtained another Matrix Voice with the assumption that it was needed for the TOA localization algorithm. As we progressed with the project, the team realized that a single microphone array was enough using the TDOA algorithm instead. In the end, the second Matrix Voice was not required and utilized in the final product. A poster and easel were printed and purchased for $40 to be displayed at the Capstone Exposition.

The team has put in 15 hours per week for the research of the algorithm, which was spearheaded by Sidong. The hardware of the Matrix Voice and Raspberry Pi was worked with for 5 hours per week with Ajeetpal leading this part of the team. The GUI sub-team started from learning Python as beginner to help develop a GUI for the Matrix to creating functioning GUI averaged to 5 hours per week. The cost of labor and research for an approximate 25 hours a week—for a rate of $40 per hour—is $1,000 a week.

8. Conclusion & Current Status

The project accomplished single source soft localization (direction information) and tracking of single source with the following final design specifications:

Table 2. Final Design Specifications

Parameters	Specifications
Audio Specification	16-96 kHz
Bit Depth	Signed 16 bits
Operation Range	20 meters
Cost	Below $200
Angle Accuracy	Average difference below 8 degrees
Processing Delay	Average processing delay within 1 second

As shown in the table, there are several design specifications differences:

1. The audio specification and bit depth closely follow the industry standard; however, it is important to note that audio frequency range is not discrete. The C++ executive only takes on the following values: 16, 32, 48, 96 kHz.
2. We did not achieve the 3-degree angle accuracy under all circumstances. As mentioned in the previous sections, the maximum average angle error is 7.7 degrees with 16 kHz, which reduces with the increase of sampling rate.

There are several design choices that turned out to be bad and should be avoided in the future:

1. Matrix and Raspberry Pi have extremely limited memory and in general slow processing speed, we should have gone with a more robust processor.
2. We should have looked more into the MATRIX library and tried to reduce the acquisition speed.

9. Leadership Roles

1. **Tiffany Ho:**
    - Group leader – Collaborates with others to create a schedule.
    - Documentation Coordinator – Documents attendance for every meeting along with detailed notes for project recommendations from Dr. Ma for others to reference to.
        - Documents specific notes for every meeting in general (with or without Dr. Ma present) – whether in-person or virtual.
    - Capstone Poster Lead: In charge of creating the expo poster with details on the project as well as getting it printed before capstone.
2. **Daniel Scarborough:**
    - Webmaster – Documents the webmaster portion of our project.
    - Capstone Poster and Parts Lead – Works with Tiffany Ho on the expo poster, responsible for obtaining necessary resources off campus required for capstone.
3. **Ajeetpal Dhillon:**
    - Expo Coordinator – Ensured that the expo ran smoothly. Spoke with judges about specific project details during the event and demonstrated our project accordingly during capstone.
    - Hardware Lead – Collaborates with the software lead on creating the algorithm and worked closely with the matrix for this project.
4. **Harry Nguyen:**
    - Financial Manager – Responsible for keeping track of all finances along with order forms to be submitted to the ECE department for parts that are necessary for the project.
    - GUI Lead – Modified the python code with Andrew Dulaney in order to make the GUI look presentable with the matrix for capstone.
5. **Sidong Guo:**
    - Software/Algorithms Lead – Modifies the MATLAB algorithm and raspberry pi along with the MATRIX VOICE module and research methods of algorithms for our project. Collaborates with Ajeetpal Dhillon for this sub team.
    - Expo Coordinator – Worked along with Ajeetpal Dhillon that the expo went according to planned. Spoke with judges about specific details of the design on the algorithm of our project.
6. **Andrew Dulaney:**
    - GUI Main Lead – Wrote most of the GUI python code, collaborated with Harry Nguyen in order to ensure that the GUI worked accordingly with the matrix for capstone. Also worked with LED lights on the matrix.

10. References

1. M. D. Gillette and H. F. Silverman, "A Linear Closed-Form Algorithm for Source Localization from Time-Differences of Arrival," in IEEE Signal Processing Letters, vol. 15, pp. 1-4, 2008, doi: 10.1109/LSP.2007.910324.

2. L. Li, Y. Wang, X. Ma, C. Chen and X. Guan, "Dual-tone radio interferometric positioning systems for multi-target localization using a single mobile anchor," in China Communications, vol. 12, no. 1, pp. 25-35, Jan. 2015, doi: 10.1109/CC.2015.7084381.

3. M. Shinotsuka, Y. Wang, X. Ma and G. T. Zhou, "Designing radio interferometric positioning systems for indoor localizations in millimeter wave bands," 2014 48th Asilomar Conference on Signals, Systems and Computers, 2014, pp. 1184-1188, doi: 10.1109/ACSSC.2014.7094645.

4. C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 24, no. 4, pp. 320-327, August 1976, doi: 10.1109/TASSP.1976.1162830.

5. J. Hu, C. Tsai, C. Chan and Y. Chang, "Geometrical arrangement of microphone array for accuracy enhancement in sound source localization," 2011 8th Asian Control Conference (ASCC), 2011, pp. 299-304.

6. G. Xiong, X. Li, X. Wu and Y. Ou, "A closed-form approach to acoustic source localization," 2012 IEEE International Conference on Information and Automation, 2012, pp. 739-744, doi: 10.1109/ICInfA.2012.6246916.

7. M. Shinotsuka, Lin Cheng, X. Ma, G. K. Chang and G. T. Zhou, "Hardware implementation of the space-time radio interferometric positioning system," 2015 IEEE International Wireless Symposium (IWS 2015), 2015, pp. 1-4, doi: 10.1109/IEEE-IWS.2015.7164550.

8. Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," in IEEE Transactions on Signal Processing, vol. 42, no. 8, pp. 1905-1915, Aug. 1994, doi: 10.1109/78.301830.

9. K. W. Cheung, H. C. So, W. -. Ma and Y. T. Chan, "Least squares algorithms for time-of-arrival-based mobile location," in IEEE Transactions on Signal Processing, vol. 52, no. 4, pp. 1121-1130, April 2004, doi: 10.1109/TSP.2004.823465.

Appendix A. GCC-PHAT

Consider a pair of microphones, receiving the same acoustic waveform, but with a delay apart, the time domain equations are characterized as:

$$x_1(t) = s(t) + n_1(t)$$

$$x_2(t) = as(t - \tau_0) + n_2(t)$$

With Corresponding frequency domain response characterized as:

$$X_1(w) = S(w) + N_1(w)$$

$$X_2(w) = aS(w)e^{-jw\tau_0} + N_2(w)$$

Then by definition of discrete cross-correlation function, we have:

$$R_{x_1x_2}(\tau) = E[x_1(t)x_2(t - \tau)]$$

The FFT of cross-correlation is the cross-power spectrum defined as:

$$R_{x_1x_2}(w) = X_1(w)X_2^*(w)$$

The GCC-PHAT is by definition:

$$\overline{\overline{R}}_{x_1x_2}(\tau) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{R_{x_1x_2}(w)}{\left|R_{x_1x_2}(w)\right|} e^{jw\tau}\, dw$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{jw(\tau+\tau_0)}\, dw$$

$$= \delta(\tau - \tau_0)$$

This result explains why GCC-PHAT provides an impulse peak at the index of the delay.

Appendix B. Multi-lateration Techniques

Assume a 2-D system model (elevation of antennas usually have very minimal influences on localization accuracy for the sake of convenience Z-axis is omitted) where there are N receivers and 1 transmitter whose location information we are interested in.

$$\tau_{1n}c = \sqrt{(y-y_1)^2 + (x-x_1)^2} - \sqrt{(y-y_n)^2 + (x-x_n)^2}$$

Where $\tau_{1n}c$ equals d_{1n}, is the distance difference between receiver n to transmitter and between receiver 1 to transmitter, where receiver 1 is used as reference receiver. This is a positive

value when receiver 1 is further away.

However, there will be a measurement error modelled as Gaussian random variable and 7 pairs of microphones (each with respect to the middle microphone) can construct 7 non-linear equations as follows:

$$\tau_{1n}c = \sqrt{(y-y_1)^2 + (x-x_1)^2} - \sqrt{(y-y_n)^2 + (x-x_n)^2} + n_n$$

By letting $d_n = \sqrt{(x-x_n)^2 + (y-y_n)^2}$

And subtract from d_1, d_n We have:

$$d_1^2 - d_n^2 = x_1^2 + y_1^2 - x_n^2 - y_n^2 - 2(xx_1 + yy_1) + 2(xx_n + yy_n)$$

Using identity $d_n = d_1 - d_{1n}$

And with linear approximation in ignoring the error term and rearrange the equation:

$2d_1 d_{1n} - d_{1n}^2 = k_1 - k_n + 2x(x_n - x_1) + 2y(y_n - y_1)$

Where $k = x^2 + y^2$ and $k_n = x_n^2 + y_n^2$

Now the set of non-linear equations can be written in matrix form as follows:

$H_1 = (x_2 - x_1\ y_2 - y_1\ \ - d_{12})$ is the first row of the N by 3 matrices, where N is the number of microphones minus 1 (the reference microphone).

$x_1 = (-d_{12}^2 - k_1 + k_2)$is the first row of the N by 1 matrix.

The estimation matrix is defined as $\theta = [x \quad y \quad d_1]^T$

And by the law of Least Square it can be easily shown that the optimum estimator for LLS is:

$$\overline{\theta} = \frac{1}{2}(H^T H)^{-1} H^T x$$