MICROPHONE ARRAY BASED SOUND SOURCE DETECTION AND

LOCALIZATION WITH BEAMFORMING, GCC-PHAT AND

TRIANGULATION USING ARDUINO AND MATLAB

by

Tarun Kadukuntla, B.Tech

A thesis submitted to the Graduate Council of
Texas State University in partial fulfillment
of the requirements for the degree of
Master of Science
with a Major in Engineering
December 2019

Committee Members:

Semih Aslan, Chair

George Koutitas, Co-Chair

Ravi Droopad

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis committee chair and advisor Dr. Semih Aslan for giving me the opportunity to work with him. Under his guidance.

I would also like to express my gratitude to all the faculty members of the Ingram School of Engineering at Texas State University for their excellent advice, encouragement, critical reviews and suggestions throughout my MS program.

Special thanks to Dr. George Koutitas and Dr. Ravi Droopad who kindly agreed to serve in my thesis defense committee. Their valuable comments and suggestions have helped me to achieve a stable research path and motivation towards this thesis.

I would also like to acknowledge and thank all my friends for creating a wonderful family-like atmosphere and giving me a warm memory during my master's life.

Finally, I am truly indebted to my beloved parents for their support and encouragement. Without this, the successful accomplishment of this advanced degree would not have been possible.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Description
2D	Two-Dimensional
3D	Three-Dimensional
AC	Analog Current
ADC	Analog to Digital Converter
AO	Analog Output
AOA	Angle of Arrival
AVR	Automatic Voltage Regulator
BCC	Basic Cross Correlation
CCDF	Cross Correlation Derivative Function
CCF	Cross Correlation Function
COM	Communication Port
DC	Digital Current
DO	Digital Output
DOA	Direction of Arrival
GBM	Grid Based Method
GCC	Generalized Cross Correlation
GPS	Global Positioning System
HRTF	Head Related Transfer Function

IDE	Integrated Development Environment
JFET	Junction Field Effect Transistor
LED	Light Emitting Diode
MEMS	Micro-Electromechanical System
ML	Maximum Likelihood
MUSIC	Multiple Signal Classification
MVDR	Minimum Variance Distortionless Response
PDOA	Phase Difference of Arrival
PHAT	Phased Transform
PWM	Pulse Width Modulation
SGA	Spatial Gradient Approach
SSL	Sound Source Localization
TDOA	Time Difference of Arrival
TOA	Time of Arrival
UCC	Unfiltered Cross Correlation
ULA	Uniform Linear Array
URA	Uniform Rectangular Array
USB	Universal Serial Bus

ABSTRACT

Microphone arrays are becoming popular for sound source localization (SSL). Keeping real-time applications in mind, such as search and rescue missions, military operations for gunshot detection, scream based localization, etc., it will help us understand how important sound localization is for people's safety. This research adopts a simple and powerful, Arduino based MATLAB sound source detection and localization system, with a reduction in localization error. Sound source localization rely mostly on an estimation of time of arrival (TOA), direction of arrival (DOA) and time difference of arrival (TDOA) methods at microphone sound sensors. The signals are then analyzed by cross-correlation functions and spatial gradient approach or time-delay based methods are applied, which will be described in our methodology. A tap-based sound source is used to detect and analyze the sound locations data. Testing of the whole system is embedded with MATLAB to increase the performance and locate the source point on MATLAB-GUI. The data form microphone sound sensors are collected by Arduino Uno and analyzed in MATLAB. The number of actual tap locations to that of estimated tap locations using GCC-PHAT and Sound triangulation methods are collected. The results are compared between software and hardware data tap localizations. In this approach of microphone array-based sound source detection and localization system, the accuracy is checked by calculating the localization errors, which is a viable path in this research work.

1 INTRODUCTION

In this thesis, the study of sound source localization in done in two-dimensional (2D) space using both uniform and partially non-uniform microphone arrays with different configurations. The tap-based sound system is used to create a two-dimensional (2D) audio effect in real-time which can analyze the data and localize the sound within a 2D space. Six-microphone sound sensors in our work are used to estimate sound source position as a point on the table that best fits a set of TDOA (time difference of arrival) measurements with sound triangulation algorithm. My area of interest is in determining the position of sound source using hardware. Research in this area has been increasing rapidly over the last few years. This thesis aims at contributing some of its findings to the scientific community.

1.1 Research Motivation

Microphone arrays have been implemented in many applications over the years in different parts of the world for audio localization. It was first designed by Professor Mayer for navigation improvement in foggy weather conditions, it is called Mayer’s topophone[1] shown in Figure 1. The start of the biggest interest in audio-based localization systems was between world war 1 and 2 to measure the location of enemy aircrafts shown in Figure 2. This helped them in aiming at the target before visual contact. It was based on Mayer’s topophone but an improved version. Hence, the Direction of Arrival (DOA) estimation of acoustic signals using microphones that are spatially separated has applications in everyday life such as gunshots detection, avoiding oncoming traffic, etc.

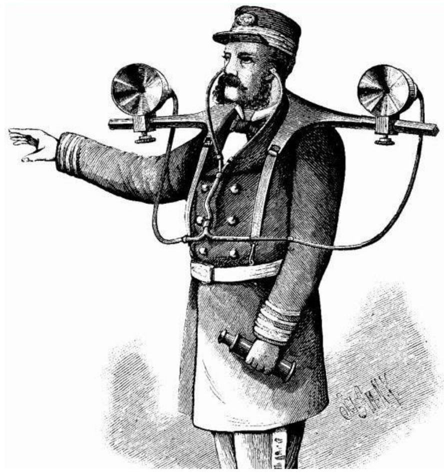

Figure 1: Professor mayer's topophone[2].

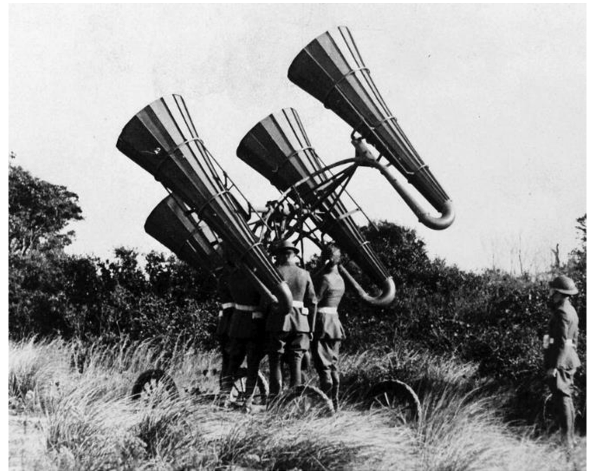

Figure 2: Improved version of mayer's topophone for aircraft localization[3].

Microphone arrays are becoming popular for sound source detection and localization, keeping real-time applications in mind, such as search and rescue missions and military operations. This research would be more realistic if implemented on hardware, which is our main goal.

The use of audio sensors in video monitoring applications is becoming widespread in detecting or locating objects. Recently, microphone arrays are used for speech detection for use in clinical diagnoses in measuring speech articulator motions and audio-based sensors are used for other medical applications. Hence audio-based detection of sound is useful, especially when camera-based applications like videos fail to detect the event.

Our focus here is to adapt a simple, low-cost, flexible, reconfigurable, powerful Arduino based sound detection and localization.

Techniques such as basic cross-correlation (BCC) and generalized cross-correlation (GCC) methods are widely used for DOA estimation. With the addition of phase transform (PHAT) as a weighted function to Generalized cross-correlation function

(GCC-PHAT), an improvement to the DOA estimation can be made. The time delay for a pair of microphones is assumed to be the delay that maximizes the GCC-PHAT function for that pair[4], [5]. This function is also used to determine the near-field and far-field applications. However, since it works better with near-field application GCC-PHAT is considered in this work.

In this thesis, the results are simulated in MATLAB and testing is done in the lab with a microphone array system using Arduino, which will be explained in chapter 4 and chapter 5. The combination of both MATLAB and Arduino are presented, which would result in a better localization point. Multiple positions are being tested, one after the other by tapping at different points. It is envisioned that this research would lead to a better understanding of sound source localization and how accurately it can be mapped in real-time with the data received from multiple microphones at a similar instance of time.

1.2 Problem Description

Many audio processing applications can obtain substantial benefits from the spatial positioning of the sound source [4]. Also, many efforts have been devoted to investigation in this research area and several alternative approaches have been proposed over the years. Sound detection in everyday life is usually done through us, humans. The human ears can detect different kinds of sound frequencies i.e. from 20Hz up to 20,000Hz in different kinds of noisy environments.

Consider future applications such as a hardware system, a machine or humanoid robots that need to detect a sound and find its location. Here microphone array is used to detect sound. Localizing sound was never an easy task with hardware until microphone arrays came into existence [5]. When localizing the sound, the ability to tell the direction of the sound source in a 2-D/3-D space provides a more natural and comfortable listening experience in the area of robotics. Sound detection is also important for some safety reasons in events like gunshots, screams, avoiding oncoming traffic, and approaching cyclists on a running path or a falling object. Being able to localize helps us determine the direction of sound, as sound signals are used to transmit or receive digital data it helps us in wireless communication systems as well. Also, use additional necessary communication network tools to help improve MIMO (multiple input, multiple output) systems. The problem of locating the source of the sound in space has received a growing interest from the scientific community, during the last decade.

There is always a growing interest in the problem of locating the sound source direction and location in space crafts to detect events that occur. It takes a lot of time to figure where exactly a sound happened in a space station, if it is hit by some tiny space particles. Using this microphone array system can help us resolve such scenarios. Algorithms for sound source localization rely on the estimation of Time Difference of Arrival (TDOA) at microphone sound sensor pairs from which one can derive information about the spatial position of a sound source[8]. Another such method is the Direction of Arrival (DOA), which plays a very important role in many applications. The most prominent techniques for estimating the DOAs are beamforming and beamscanning [9].

Also, localization depends on cross-correlation functions of the signals; the purpose of this correlation module is to compare two input signals and determine how much one signal lags from the other (in terms of several samples, which corresponds to time).

1.3 Current Research Gaps and Proposal Importance

Many studies have been conducted on Time Difference of Arrival (TDOA) techniques for spectrum monitoring in radio receiver technology and compact computing power to calculate source point-based timing and wave comparisons.

The detection of single acoustic event sources is usually performed with relatively high accuracy in event detection. But the problem in sound detection of an event is acoustic signal overlapping. In fact, the overlapping problem has recently gained a strong interest in speech processing as well. Due to overlapping, the TDOA methods are becoming more and more popular.

A concept that attracted my research area is to ensure the systems TDOA technique is functional with hardware and the determination of the exact location of the sound. To know where the sound is coming from, it is needed to know the Direction of Arrival (DOA) and the Time Difference of arrival (TDOA).

The DOA estimation is usually done by a group of acoustic microphone sensors. The signals are captured by a set of microphone arrays, and the signals are then analyzed by cross-correlation functions and spatial gradient approach with time-delay based methods. To estimate the DOA of a signal sound source when arranged uniformly in a linear array, all the microphone sensors will receive the same signal data but slightly at different instants of time, which is skilled by spatially separating these microphone sensors. By using beamforming, some trails are performed on the sound localization. This task of improving the efficiency in locating the sound with more accuracy, will be an enlightening way for future studies. This can be applied in real-world scenarios such as gunshot detection to alert police within seconds of shots being fired or in an instance of a blast by activating surveillance cameras near their location and sending a push notification to a nearest police station. Also, in military applications to localize enemies' gun firing location, teleconference, future humanoid robots, etc.

1.4 Literature survey

There exists a huge amount of work in past research experiments that focus on audio-based detection systems and localization systems using microphone arrays [5]. Few of such ideas from previous research are considered for sound source detection and localization. In our research, the work is implemented on a hardware-based platform with some integrations in algorithms and the main difference is that, the localization system is centralized. i.e. the tap is done in between two linear microphone arrays.

One wants to build a much more effective way of detecting the target sound by eliminating the noise from the background and only focusing on a range of desired frequencies using audio signal analysis.

Shazam, which uses an audio search algorithm, is one of the most common audio recognition systems [10]. This app uses mobile devices to capture a piece of an audio album. That is then sent to a server that generates an audio spectrum. Larger frequencies are picked up, the distance and time relations between them is noted, along with different surrounding frequencies for audio song detection. Noted frequency points are placed in a hash table and then the resulting values are compared with other songs' fingerprints by searching for similar frequency/intensity vs time relationships [11].

A 2007 paper ("Scream and Gunshot Detection in Noisy Environments") presented an audio event detection system for gunshot and scream detections. The model mentioned in the paper, uses a combination of 47 temporal, perceptual features, spectral distribution, and Cepstral Coefficients. As well as a zero-crossing rate in one of the perceptual features [4]. There is also no use of devices that may be overly sensitive to the signal to noise ratio (SNR) such as short time and loudness. The process starts by taking autocorrelations of each frame, 23 ms long and are sampled at 22.05kHz, to determine the energy distribution over different time lags. Much of the energy is found in the first few time-lags when a gunshot is fired, which is extremely impulsive.

There are two other gunshot detection systems currently used by the military and police departments, although there is little information on the methods used to detect them. The first is called a shot spotter [5]. It is marketed to commercial shopping buildings, university campuses, and police departments. Shot spotter is expensive but a wide range of sharp acoustic events like explosions, supersonic and subsonic gunfire can be detected. The system uses spatial filtering which depends on positioning of the sensing nodes at farther distances. This will help eliminate surrounding construction or traffic noise; therefore, the detection system will not be activated, and false results are eliminated. This will also help the users to be alerted to the location of an activity in real time.

Human sound localization experiments are already being done and they are focused more on replicating the auditory scene using loudspeakers and different audio sources, despite errors along the path, shown in Figure 3. A regression tree was used to classify localization errors which have similar standard deviations with respect to the azimuth and elevation angles of the sound source that produced the error [14].

Various distributions were then selected to represent each group of localization errors in algorithms for use in the simulation. To eliminate these kinds of errors, the following approaches can be used:

1) Cross-correlation functions (CCF) and Cross-correlation derivative functions (CCDF)
2) Beamforming techniques
3) Spatial Gradient approach (SGA)
4) HRTF (Head Related Transfer functions)

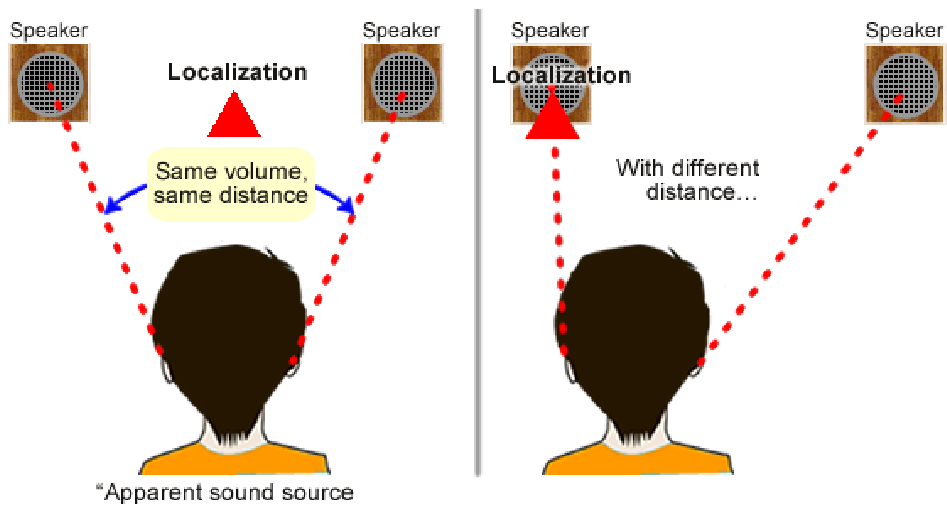

Figure 3: Speaker localization principle by humans [15].

The use of microphone arrays for sound source localization is a well- researched problem. Many microphone arrays are designed for a specific type of sound-source. Another example of a military application would be a counter-sniper system based on the sound source profile and triangulation, which is able to locate the sniper's position. The time response is crucial for those vital situations.

A scream/non-scream test is done in the case of a human voice to detect screaming in an abnormal situation such as screaming for help during a bank robbery or campsite failure. In the case of non-human sound, an emergency sound classifier can detect unusual sounds like a gunshot, glass breaking, and explosion.

In some special situations such as darkness or enclosed spaces where camera confidentiality is a public concern, the benefit of acoustic surveillance over video surveillance is an advantage. Hence, acoustic source localization is one of the most essential solutions.

The use of microphone arrays is a well-researched problem, but many microphone arrays are designed for improving the quality of the sound captured[12]. To be specific the signals sometimes can be distorted by additional noise and reverberation. If sound-source characteristics are known in advance, researchers can avoid this problem and deal with it.

Due to the recent miniaturization and price drop of good quality microphones based on microelectromechanical systems (MEMS), microphone arrays are rapidly adopting such types of microphones. MEMS microphones are much cheaper than traditional microphones and have a relatively good signal-to-noise ratio and frequency response. And their small size allows miniature arrays with high-level integration.

For general-purpose sound processors like microphones, most of the signal processing required by such array is traditionally calculated using cross correlation methods [8]. Nonetheless, the computational demand is directly linked to the array's number of microphones, which is growing intensively thanks to low-cost microphone sensor technology.

An omni-directional acoustic microphone detector is used in this research to determine the sound source direction of arrival (DOA) and angle of arrival (AOA) relative to the sensor. various types of DOAs are discussed in Chapter 2.

The microphone sensor measures in a certain direction where the particle velocity component of a sound wave is present [16]. There is a method known as beamforming for an array of particle velocity sensors, where three different kinds of algorithms are mostly used. Namely, MVDR, MUSIC, and ESPIRIT, two of which are used in our work.

2 OUTLINE

2.1 Beamforming using microphone arrays

Almost all the MEMS microphones are Omni-directional, that is they respond equally to the sounds coming from any direction in 2D and 3D space. Multiple microphones are combined to form a directional response or a beam pattern, depending on the sensitivity coming from different directions [6]. Here, broadside summing arrays and differential end-fire arrays are included. The modeling of design considerations of spatial and frequency responses requires different array configurations. While designing these configurations there will be some advantages and disadvantages with the array response. A beamforming technique is a spatial filtering technique which maximizes the output power in the direction of the sound source [5][6].

2.1.1 Microphone Arrays

Any number of microphones operating in a tandem are microphone arrays. They create a response to the direction of sound or beam pattern when configured. There are many applications such as speech recognition systems, hearing aids, determining the flight path of bullets in the military, etc. The directionality of the microphone from which the sound arrives is defined based on the sensitivity towards the angle of a sound source. [17].

2.1.2 Types of Microphone

Omni-directional: These microphones have equal sensitivity to all the angles i.e., these microphones have the capability to detect sound with equal gain from all directions. These kinds of microphones are very helpful but have a drawback of capturing undesired sound sources [13].

Cardioid: These microphones have high sensitivity at the front and small sensitivity in the back. Cardioid microphones are much more resistant to feedback than the omnidirectional microphones.

The arrays are classified into two types: namely, a broadside array and an end fire array depending on the direction of the wave fronts emitting from source.

Broadside array: The wave front coming from the source will reach all the sensors at the same time ($\varphi = 0°$).

End fire array: The wave front coming from the source will reach the sensors of a linear array with a maximum delay at the same time ($\varphi = 90°$ or $270°$).

Figure 4 explains sound source location for 2D coordinate system with broadside array angle representation.

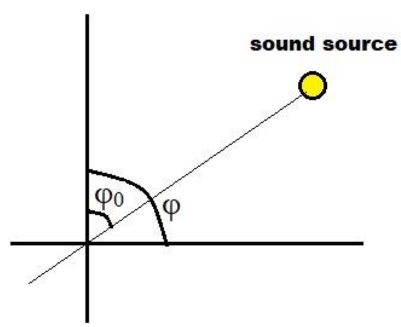

Figure 4: Coordinate system

The sensor that detects sound signals and converts them into an electrical signal is known to be a microphone that is acoustic to an electric transducer. Most of the microphones operate based on an electrical voltage signal from mechanical vibration of a thin piece metal called diaphragm, like how your eardrum vibrates when you hear a sound.

Microphones can also be worked as electromagnetic inductors, light modulators, piezoelectric generators, and capacitance changers [18].

Different types of microphones use various methods to convert energy. In this research, the microphone sensor that is shown in Figure 5 are being used.

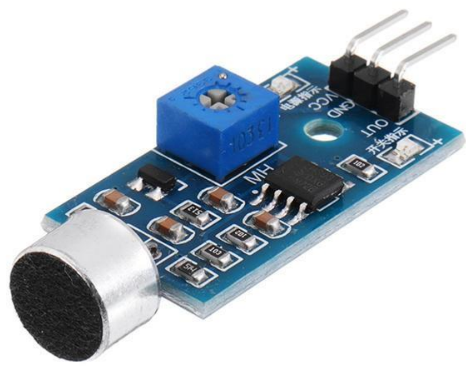

Figure 5: Microphone sensor LM393[19].

The sound sensor module shown in Figure 5 provides us an easy way to detect sound and is generally used for detecting the sound intensity. This module can be used for security, switch and monitoring applications. Its accuracy can be adjusted for the convenience of usage. It uses a microphone which supplies the input to an amplifier, peak detector, and buffer. When the sensor detects a sound, it processes an output signal voltage which is sent to a microcontroller, which then performs necessary processing.

The sensor module board has an output port used for both analog output (AO) and digital output (DO). When the sound intensity reaches a certain value, the output goes high with DO and a real-time output voltage signal of the microphone is given by AO. Mainly, the sensitivity can be adjusted via the potentiometer on the sensor module which is built in.

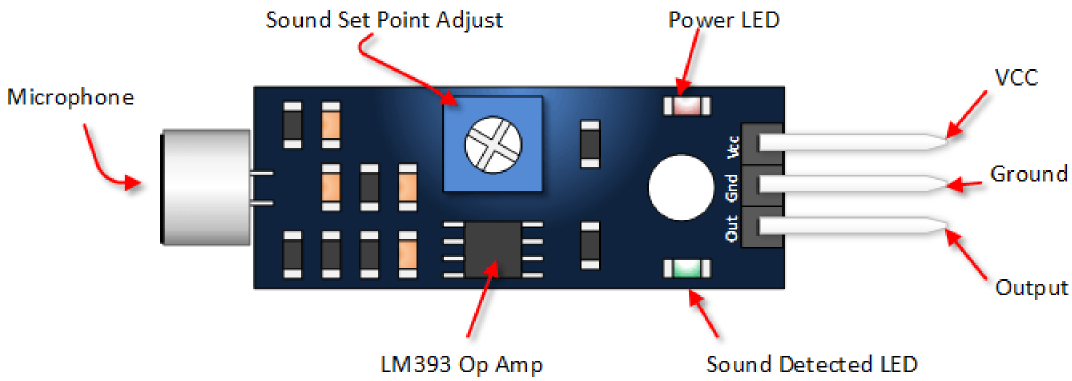

Figure 6: Microphone sensor module pinout diagram [20].

In this work, six of the above shown in Figure 6 microphone LM393 sound sensors are used. The circuit of the above sensor starts with a condenser microphone which picks up sound signals and transforms them into electrical signals. The electrical signals are then sent through the IC amplifier in the sensor which is then fed to the Arduino board for further processing.

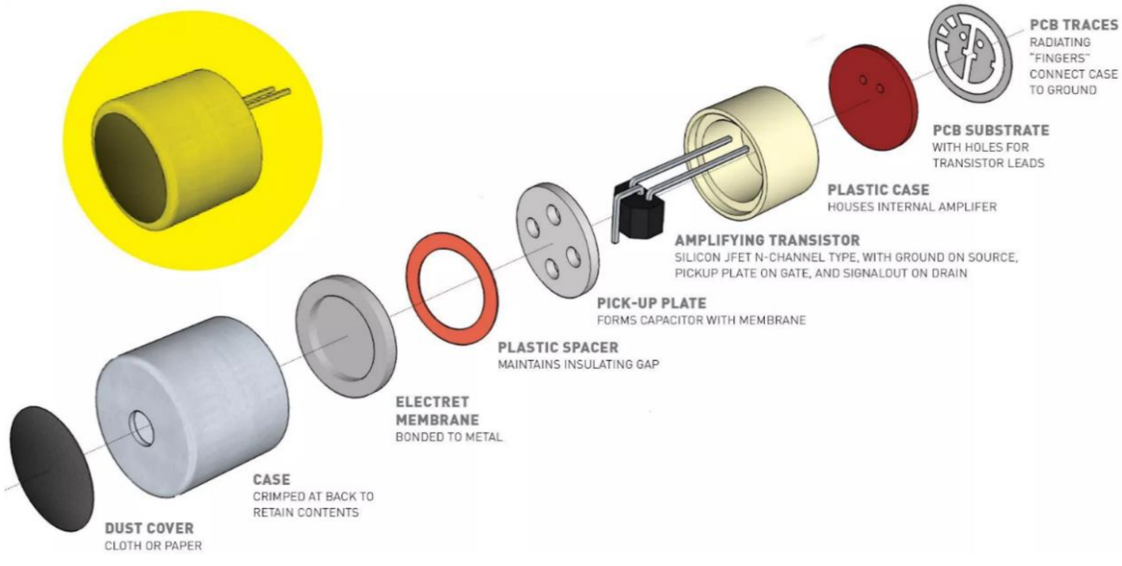

Figure 7: Inside a Microphone, how it works internally [21].

These kinds of microphones are called electret microphones (used in cellphones, hands-free kits, etc.). As shown in figure 7, they have an electret membrane with a pick-up plate and an amplifying transistor. The sound wave pressure pushes the membrane so the distance between the plates varies with the sound amplitude, acting as a variable capacitor. The capacitor is DC biased so that the vibration produces a weak AC voltage riding on top of a DC voltage[5]. A transistor is used to amplify the AC voltage. JFET types are an ideal choice here because they have a high gain. This is a capacitive microphone.

2.1.3 Explaining directionality and polar plots

Directionality describes about different kinds of patterns in which the microphone or array output changes with the change in sound source position in a plane/space. These microphone arrays are omnidirectional which means they have sensitivity towards the sound coming from all directions, regardless of the orientation of a microphone. The plot looks the same irrespective of the microphone's port orientation in the x-y, x-z, or y-z plane [6]. In this, any reference to the front of the array as on-axis direction is the direction of the desired audio and is labeled as 0° in polar plots. The rare side is at 180° and the sides are referred to space in between, centered on 90° and 270°.

2.2 Types of beamformers

2.2.1 Non-adaptive beamformer

The delay and sum beamforming are types of beamformers where the microphone signals are time aligned for a specific direction. Such a beamformer has an all-pass characteristic and their purpose is to delay the incoming signals by a certain number of samples.

The delays can be estimated by analyzing the array geometry of the plots. The advantage of this with filters would result in an improvement in the sound quality of speech recognition at low frequencies [18].

2.2.2 Adaptive beamformers

These beamformers can be implemented for two speakers when the location of the second speaker is unknown. This method is used to change the directionality of the array. When transmitting, a beamformer controls the phase and relative amplitude of the signal at each transmitter.

When receiving information from different sensors, the information is combined in a way where the expected pattern of radiation is preferentially observed. When applied to sensor networks, receiving beamforming is more challenging because it requires the collection of signals received at different antennas.

2.3 Arduino Uno

Arduino board is for developing real-time applications that makes a sense and control over more of the physical world needs than a desktop computer. It is an open-source hardware and open source software-based programming platform on a simple microcontroller board and a development environment for writing software for the board, which is extensible and easy for beginners, yet powerful enough for complex tasks.

The software is written in C, C++, java programming language. The development board-Arduino Uno is an implementation of fasting the computing world to real-time data capturing and analyzing with extensible hardware.

A basic model of Arduino is shown in Figure 8.

Figure 8: Arduino Uno Board [22].

Arduino allows the users to build new projects with an extendable hardware that is with additional sensors to the board, depending on the need of design. Depending on the real-world hardware need one can integrate things around us with external electronic devices. When the functionality of the task is complex, one can add a micro SD card to the board and make store more information.

ATMEGA328 microcontroller is present on the board. It also consists of timers, counters, PWM, interrupts, I/O pins, 16MHz clock helps produce more frequency as instructions per cycle increases, there is a reset pin to control it.

Table 1: Arduino parameters with specifications.

Parameters	**Specifications**
Input Voltage	7-12V (recommended) 6-20V (limitation)
USB (operating)	5V
Digital I/O pins	14
Analog I/O pins	6
DC current I/O	20 mA (per pin)
DC Current	50 mA (3.3V pin)
Flash memory	32 KB (ATmega328P)
Clock speed	16 MHz

Reset Pin is inbuilt on the board, resets the entire board to return the program that is running to the initial stage. When the board hangs up while running a complex program or giving too many instructions by connecting many devices, pushing this pin helps to restart everything from the beginning.

The 6 analog pins are marked from A0 to A5 and comes with a resolution of 10bits. These pins measure from 0 to 5V, however, they can be configured to high range using ‘analogReference()’ function and AREF pin. Also, there are 14 I/O digital pins, it allows external access to any circuit board. All these pins provide flexibility and ease of use when the external devices are connected and configured to run.

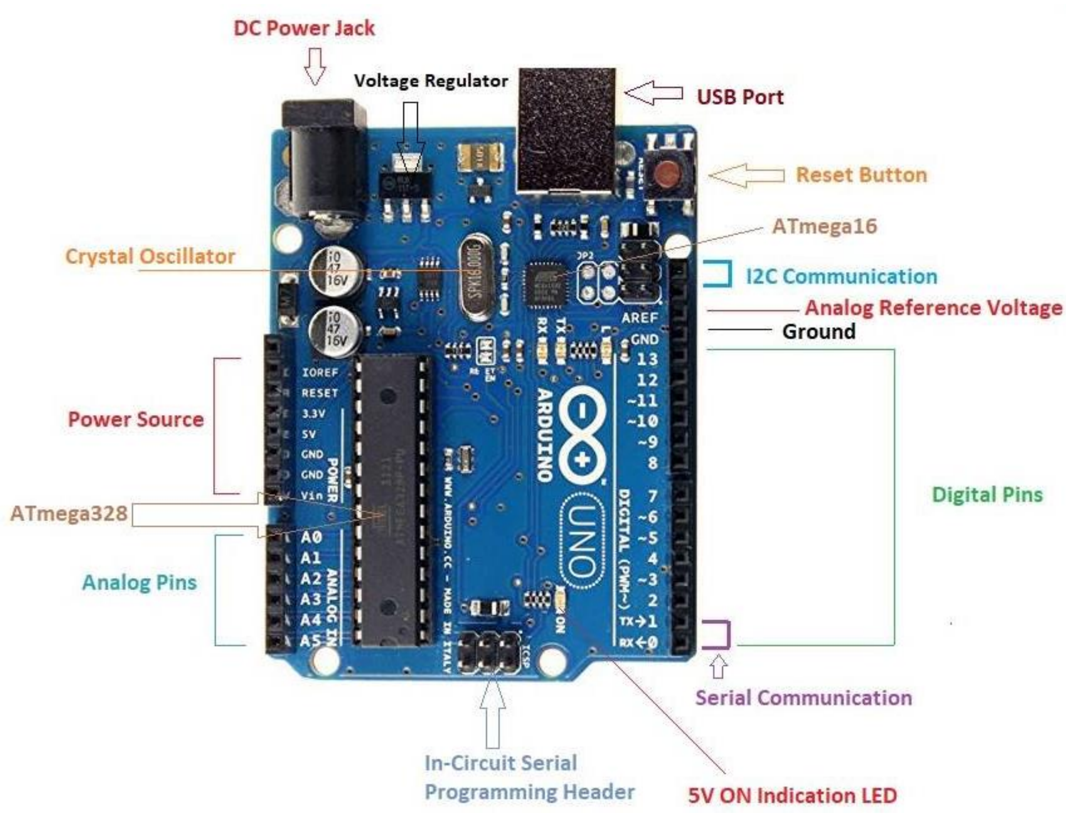

Figure 9: Arduino board with all components and pinouts [23].

This Arduino based AVR microcontroller comes with 2kB SRAM, 32kB flash memory, 1kB of EEPROM. On-chip ADC is used to sample these pins. A 16 MHz frequency crystal oscillator is present on the board. Figure 9 shows the pinout and on-board components of the Arduino Uno board.

2.3.1 Arduino IDE communication

The Arduino software-IDE (Integrated Development Environment) is a programming tool which allows the user to write programs and upload them into the Arduino board. Arduino IDE uses a programming language that is user friendly.
IDE compiles and translates the code into the assembler language after the user has written the code. The IDE code is uploaded onto the board by the user.

Once the code has been interpreted Arduino IDE notifies that the program has been uploaded and is ready to use. Arduino IDE has an integrated code parser to test the user’s written code before submitting it to the Arduino.

IDE software includes the set of different kinds of programs that are ready to be tested on the device. After testing the program, it can be uploaded to the Arduino by a USB cable to test different model sensors.

The ATMEGA328 provides the board to communicate serially using pins Tx and Rx for transmission and reception purpose and the ATMEGA16U2 incorporated onboard provides a pathway for serial communication using USB com drivers.
The serial monitor is provided on the IDE software which is used to send or receive data from the board. The LEDs flashing on the board at Tx and Rx pins indicates the transmission and reception of the data. Figure 10 is a java-based Arduino IDE programming window, where the integration of different sensors with Arduino is coded.

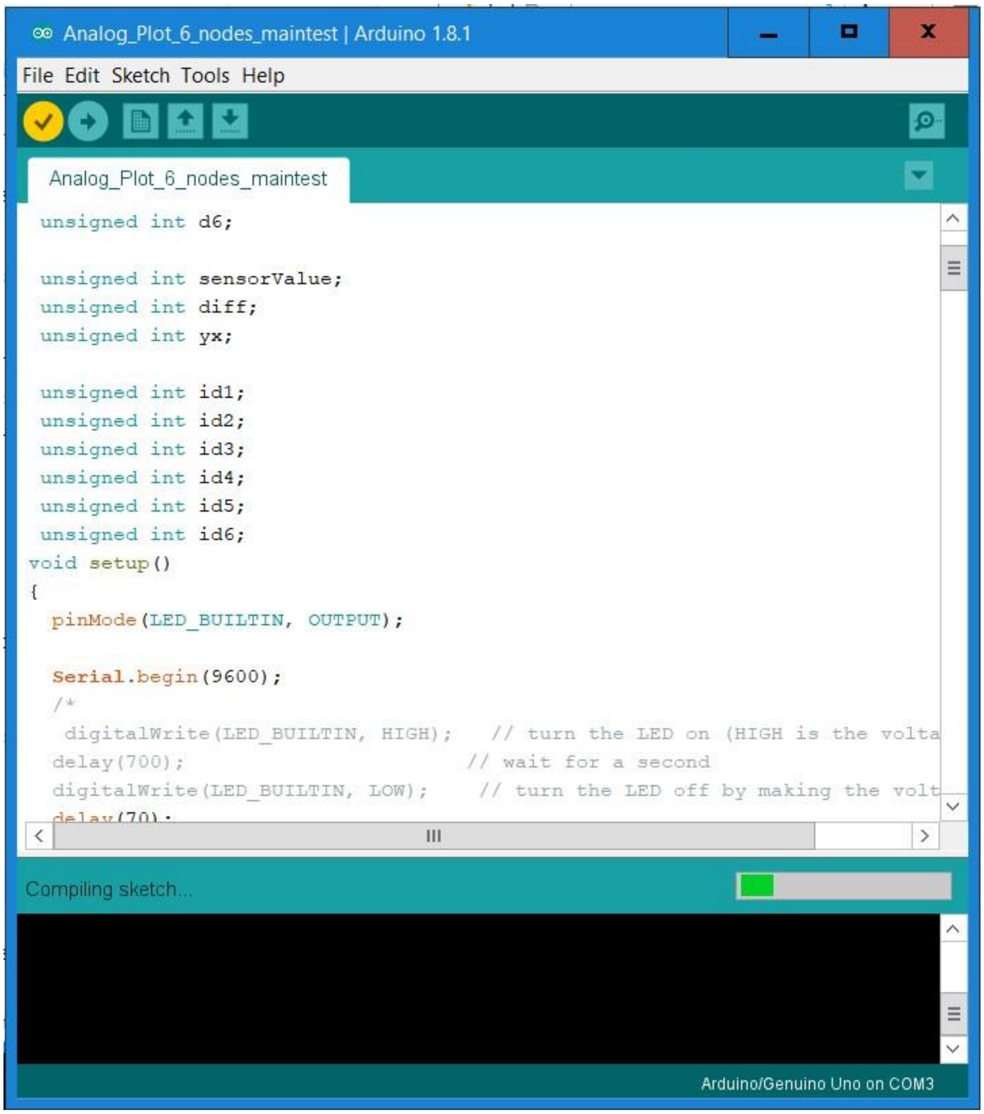

Figure 10:screen capture of java-based Arduino IDE

2.4 Spherical coordinates

Spherical coordinates define distance and two angles of a vector or a point in space. There are several performances of signals that is considered acceptable in defining the specifications of the two angles that are known as azimuth and elevation angles and the distance is usually an Euclidean norm. MATLAB software provides syntax functions for changing as well as altering between azimuth and elevation angle representations.

The angle between the x-axis and the orthogonal projection of the vector onto the XY-plane is known as the azimuth angle of a vector. Azimuth angles range from -180 and 180 degrees. The angle between the vector and its orthogonal projection onto the XY-plane is known as the elevation angle of a vector. The elevation angle is usually defined as the angle of a vector that makes with the positive z-axis [24].

Figure 11 illustrates azimuth angle and elevation angle of a vector, the coordinate system is relative to the center of a uniform linear array, whose array elements are appeared in purple disks. R represents the distance, az/el are the azimuth and elevation angles.

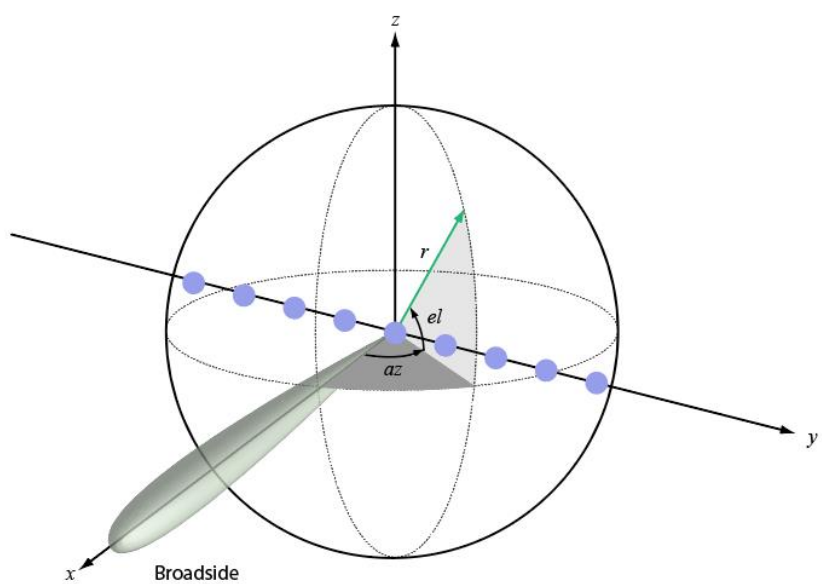

Figure 11: Coordinate system related to uniform linear array[25].

For our work, there is a need to convert the above spherical coordinates described to rectangular coordinates. Some of the equations that describe the relationship between spherical and rectangular coordinates are as follows,

$$R = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

$$az = \tan^{-1}(y/x) \tag{2}$$

$$el = \tan^{-1}(z/\sqrt{x^2 + y^2}) \tag{3}$$

To convert az, el, R to rectangular coordinates

$$x = R\cos(el)\cos(az) \tag{4}$$

$$y = R\cos(el)\sin(az) \tag{5}$$

$$z = R\sin(el) \tag{6}$$

When defining a target location for the phased array system, the length and the direction are also specified. Here it is seen that the distance corresponding to R from the array and the direction corresponding to the angles of azimuth and elevation are required for calculating the conversion of array response of the system.

After the conversion of array response, broadside angles are also required to estimate the DOA. A broadside angle is the angle between the plane and the signal direction. In the response of a uniform linear array these broadside angles are required to be included [25].

Usually the microphone array response of a system does not depend on the angle of azimuth and the angle of elevation, but it depends directly on a broadside angle. The broadside angle lies between -90 and 90 degrees. When it is zero, the signal path indicates that it is orthogonal to the array axis.

The broadside angle of the ULA is measured to be positive when considering towards the positive direction of the array axis. Other signal paths of a microphone array system that have same broadside angles will form as a cone around the ULA of the microphone sensors.

Broadside angle β is given as,

$$\beta = \sin^{-1}(\sin \text{az} \cos \text{el}) \tag{7}$$

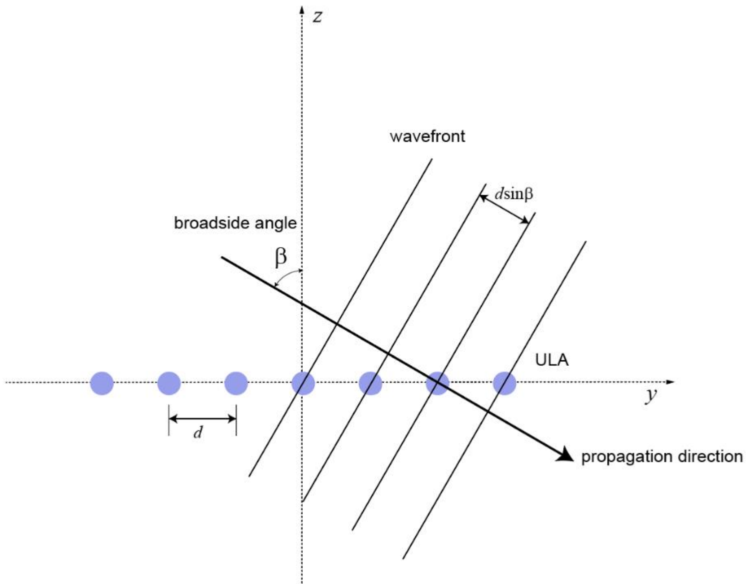

Figure 12: ULA elements along y-axis with respect to z-axis[25].

Figure 12 shows ULA elements along the y-axis spaced at d meters apart from each other. Since, the metrics in 2D space are used in this work, the elevation angle is considered to be 0°, the signal lies in xy-plane.

The broadside angle is reduced to the azimuth angle. Because in a plane wave the element in the microphone array system and angle of arrival are not simultaneously illustrated.

The distance that incident wave travels between array elements is $d \sin \beta$ where d is the distance between array elements and τ is a constant time delay between array elements. Which is given as,

$$\tau = \frac{d \sin \beta}{c} \tag{8}$$

Where c is speed of wave.

3 SOUND DETECTION AND LOCALIZATION TECHNIQUES

3.1 DOA techniques

The arrangement made to estimate the direction of arrival (DOA) is a given set of microphone sensors placed at known locations. Here in this work the aim is to estimate a two-dimensional coordinate of the sound sources that is created by a tap in our microphone array system.

The signals are captured by microphone sensor and then analyzed by the following methods: beamscan, minimum variance distortionless response (MVDR) which is identical to beamscan but depends on physical size of the microphone array system. MVDR uses an MVDR-beam for DOA estimation. Multiple signal classification (MUSIC) which is one of the high-resolution DOA estimation methods, does not depend on physical size of the microphone array system. From the mentioned beamforming methods, one of the methods is chosen that is suitable in this research.

The peaks of the output spatial spectrum of all these three methods will indicate the DOAs of the received signals at the microphone sound sensors. Here in this work, a comparison of the above mentioned three algorithms used to estimate the broadside angles will be made. Selection of an algorithm that is best suitable in our hardware implementation with uniform linear array (ULA) or uniform rectangular array (URA) will be chosen. Note that while considering the angle of elevation as well as the angle of azimuth the source localization point will be in 3D.

The DOA estimation algorithms are applied to beamforming techniques to obtain individual estimates of DOA for different frequency bands to obtain a suitable tap frequency to estimate the sound source. These estimates are then combined for statistical evaluations to find a correct estimation of the angle.

Usually, to estimate the DOA of a signal sound source when arranged uniformly in a linear array, all the microphone sensors in ULA will receive the same signal but with difference in time, because of the separation of these microphone sensors spatially as mentioned earlier.

So, a basic structure of microphone placement called ULA (Uniform linear array) is used in this work, as well as it is considered as two sets of parallel sets of ULAs in another implementation for better sound localization.

In this system, the microphone sound sensors are spaced at a distance, d between each microphone sensor.

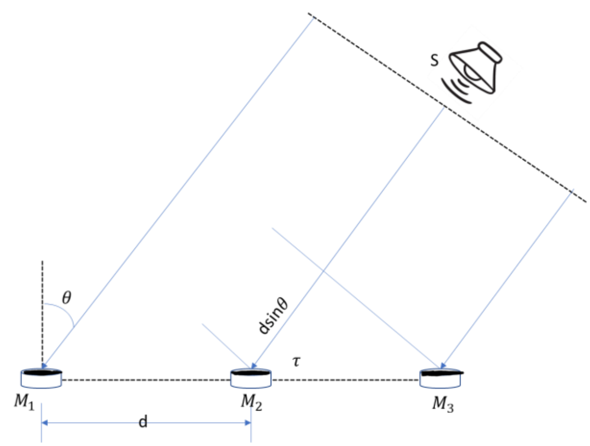

Figure 13: ULA with 3 microphone sensors.

In Figure 13 considering the distance between the sound source and the microphone array is supposed to be larger than the distance $'d'$ between the microphones, the sound signal DOA at microphones gives the angle, θ. $S(t)$ is the signal received by microphone 1, here the noises from the air is not considered. Then the signal received by microphone 2 would be the same signal $S(t)$ with a time delay or time advance of $\frac{d\ cos\,\theta}{c}$, where c is the velocity of sound. Similarly, consider it to the rest of the microphone array, when signals are received at a random microphone is given as[26].

$$x_i = S(t + \tau_i)\ , \qquad i = 2 \ldots N, \tag{9}$$

Where,

$$\tau_i = \frac{d_i\ cos\,\theta}{c}\,, \qquad i = 2 \ldots N, \tag{10}$$

And d_i is the distance between microphone i and reference microphone.

$$d_i = (i - 1)d, \qquad i = 2 \ldots N. \tag{11}$$

The signals that are equally separated by microphone array sensors are joined by beamformers, to output the array signals make more prominent from a specific viewing direction. These types of methods are called as steered beamforming-based methods.

If suppose a signal is present in a viewing direction, then the array output power is high and if a signal is not present then the array output power is low [9].

Therefore, the array is used to construct beamformers that inspect in all directions that are possible to estimate the DOAs. Depending on the array arrangement two major approaches are considered one is minimum variance distortionless response beamformer (MVDR) and the other is Multiple signal classification algorithm (MUSIC) beamformers.

3.1.1 MVDR

In minimum variance distortion less response beamformer, as to prevent the action of interferences to the maximum depending on the environment, the linear filter weights are calculated. It leaves the signal of interest undistorted.

In the process of optimal weight calculation, the calculation of this correlation matrix and its multiplication with other steering vectors are the most important parts.

The array of this correlation matrix is the measure of spatial correlation of the signal and the noise arriving at the array of microphones placed uniformly. This array correlation matrix measurement is later used to determine the spatial filter weights[9], [27]. When there are position errors in microphone sensors with MVDR, it shows a reduced performance when it is compared with the conventional beamformers. The MVDR beamformer's output is severely affected by the association between the signal path and the interferences.

3.1.2 MUSIC

Multiple signal classification algorithm is one of the high-resolution sub-space of DOA methods. MUSIC gives an estimation of number of signals arrived, that is DOA. MUSIC deals with the eigenvectors disintegration and eigenvalues of the matrix into two orthogonal matrices, namely signal subspace and noise subspace of the array sensors.

For the estimation of DOA for different sound sources it is required to consider their properties, that is signal frequency and signal noise. Other MUSIC algorithms have been proposed over the years to increase performance and resolution power with reduction in complexity[27]. The advantage of this is a directional calculation of the DOA by searching for zeros of a polynomial, which is necessary in case of MUSIC method. This method is limited to linear microphone sensors placed uniformly and are separated spatially.

The advantage of Normalized Beamscan is that it is strong against finite sample errors. But it has no reliable way to choose the diagonal loading factor, which affect its performance indirectly.

The MVDR beamformer gives a distortionless performance in the direction of interest. But it is unable to differentiate between two closely spaced signals.
Using MUSIC gives the system a high-level orthogonality between the signals, it has high power efficiency, hence high resolution and accuracy. But is limited to linear antennas.

Beamformers have the capability to improve the quality of signals from one direction. Also, decrease the quality of signals from other directions in DOA estimation. The direction at which the signal is present gives the largest power, as the estimated DOA of the sound source signal. The DOA displays a maximum peak value in each direction from which a sound source signal is detected. The array output power is plotted against that particular direction.

Basically, the two types of beamformers are broadband and narrowband. They are differentiated depending on the signal bandwidth on which beamformers are being used.

3.2 TDOA techniques

The other class method includes time delay estimation (TDE) methods. This technique includes the time delay estimations, which are calculated for each pair of microphones in the array system. Then, the time delay data that is acquired is collaborated with the information of known microphone sound sensor locations. The geometry of the microphone array range is considered and analyzed later to determine the best approximation of the sound sources' DOA.

One of the advantages of TDE based methods is that it has a lower computational load. when compared with the previously mentioned methods, because of the extensive search the DOA is avoided, which makes TDE method most productive. In addition, approaches based on TDE can also be extended to broadband signals.

3.2.1 Time delay estimation

There are a variety of techniques that exists to compute pairwise time delay. Here in our work, two such methods are considered. One, General-cross correlation (GCC) method and the other is General cross-correlation with phase transform (GCC-PHAT).

The time delay $\widehat{D}$ between two signals, x_i and a microphone pair in an array system is calculated with the help of following block diagram shown in Figure 14. Where x_i are incoming signals from sound source, which may be filtered through H to enhance the signal quality, denoise the signal and obtain signals y_i or it can be just passed as x_i [28].

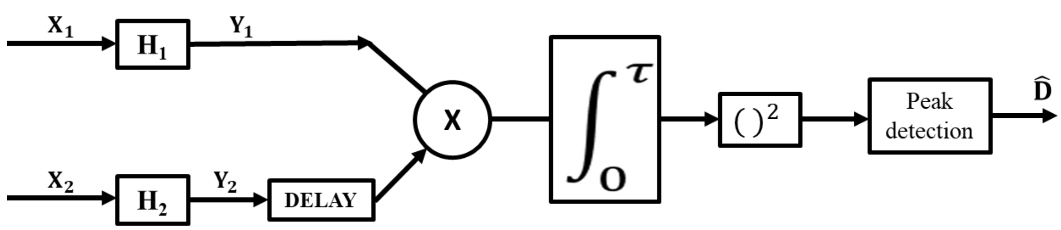

Figure 14: Block diagram for time delay of arrival estimation using GCC.

Signals $x_1$and x_2 arrive at two spatially separated microphones. The signals are mathematically classified as

$$x_1(t) = s(t) + n_1(t) \tag{12}$$

$$x_2(t) = \alpha s(t + \tau) + n_2(t) \tag{13}$$

To find the cross correlation between the signals $x_1(t)$ and $x_2(t)$.

Where, $s(t)$, $n_1(t)$, $n_2(t)$ are real stationary random signals, described by the sound signal $s(t)$ fade characteristic of the which is supposed to be uncorrelated with noises $n_1(t)$ and $n_2(t)$, α is a linear coefficient.

Time delay estimation, which is given as,

$$\widehat{R_{x_1x_2}}(\tau) = \frac{1}{T - \tau}\int_{\tau}^{T} x_1(t)x_2(t - \tau)dt \tag{14}$$

Where argument τ that maximizes equation $\widehat{R_{x_1x_2}}(\tau)$ provides time delay estimation and T is observation interval and cross correlation between $x_1(t)$ and $x_2(t)$, is a well-known Fourier transform relationship is related to the cross-power spectral density[29].

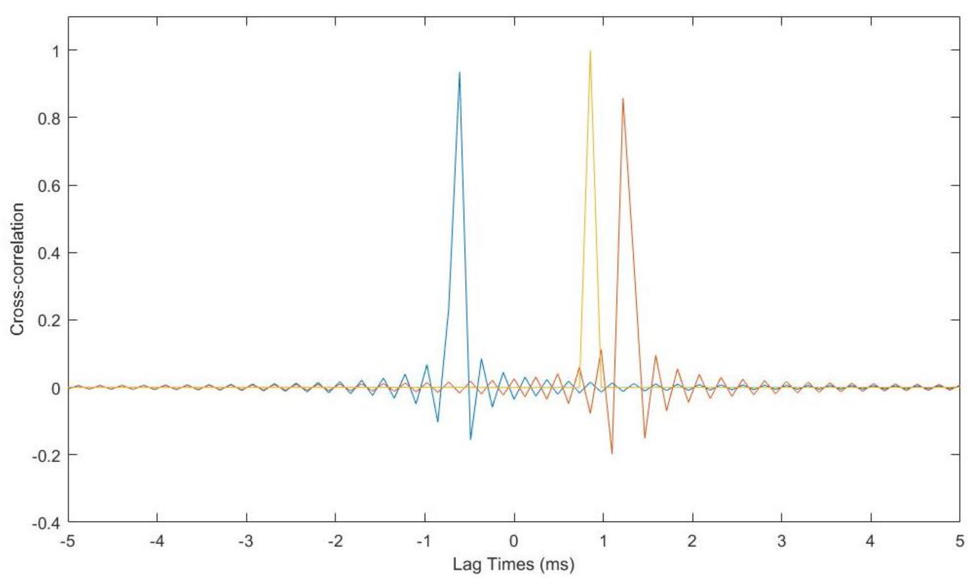

Figure 15: Cross correlation between two signals with respect to reference signal

To solve the problem of TDOA there are some conventional cross correlation techniques have been introduced. One such conventional method is Generalized cross correlation (GCC) function, which delivers the operational ease. Basic equation for GCC is given as,

$$R_{x_1 x_2}(\tau) = \int_{\tau}^{T} s(t)s(t-\tau)dt \tag{15}$$

3.3 Sound Source localization

In our work, after the peak detections information is collected, our designed system acquires data about the time delays of all the highest peak detected values of the microphones of an array which are close to the sound source. Then the system selects three such peak detections to locate the sound source using TDE based method.

After calculating the Time difference of arrival (TDOA) at each microphone, there needs to be a way to calculate the location of the sound source. The most popular methods for calculating the location of a sound based on the TDOA system are triangulation, trilateration and multilateration.

Triangulation is a method that works by using the TDOA between a pair of microphones to determine an angle of arrival (AOA). Which then uses another pair of microphones located at a different position to calculate another angle of arrival.
The intersection of the lines produced by these angles is then calculated and is defined as the location of a sound.

Trilateration is a method like multilateration which will be discussed in detail later. However, it relies on knowing the absolute time of flight of the sound wave. In order to use this method, one should know the time at which the sound was sent to the microphones while this works well in many other real-world applications such as communications and static ranging it would be unsuitable for our purpose as the time at which the sound initially occurs is not known.

Multilateration is a method of calculating position based on time difference or arrival that works based on the relative TDOA thus removing the necessity of knowing the time at which the sound occurs. Its resolution is determined by the sample rate of the microphones and a high enough sample rate can have a near-infinite resolution. Makes an attractive solution for our problem when designed with hardware. Additionally, it works well in the near and far-field making it a good solution to use for sound localization. Hence, our work adapts this with hardware, time of arrival (TOA) between the microphone sensors and then sound source position is calculated.

4 PROPOSED METHODOLOGY

Using multiple microphone arrays, sound source will be detected. Once the sound is detected the direction of the target sound source is found and then localize with a set of microphone arrays embedded with Arduino [9]. Hoping this is beneficial since it saves the runtime in sound source detection. Also, it’s flexible and scalable. Figure 16 shows how the 9 microphones M1-M9 are arranged in a 2D plane, where the sound emitting source is denoted as S. It is needed that the signals are compared with each microphone, using these results, the sound source is located.

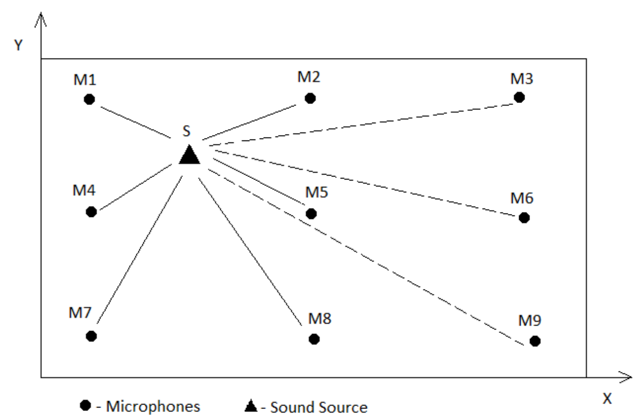

Figure 16: 2D Mapping of sound source

The below block diagram shown in Figure 17, explains the implementation of the proposed method. Different microphone arrays placed at a certain distance from each other to capture the audio as our input. Once the audio is captured, our audio detection system will consider only frequencies ranging between 50Hz to 10kHz and eliminate other unwanted frequencies. Following this the proposed algorithm will help in processing the required signals using different algorithms to localize the sound source.

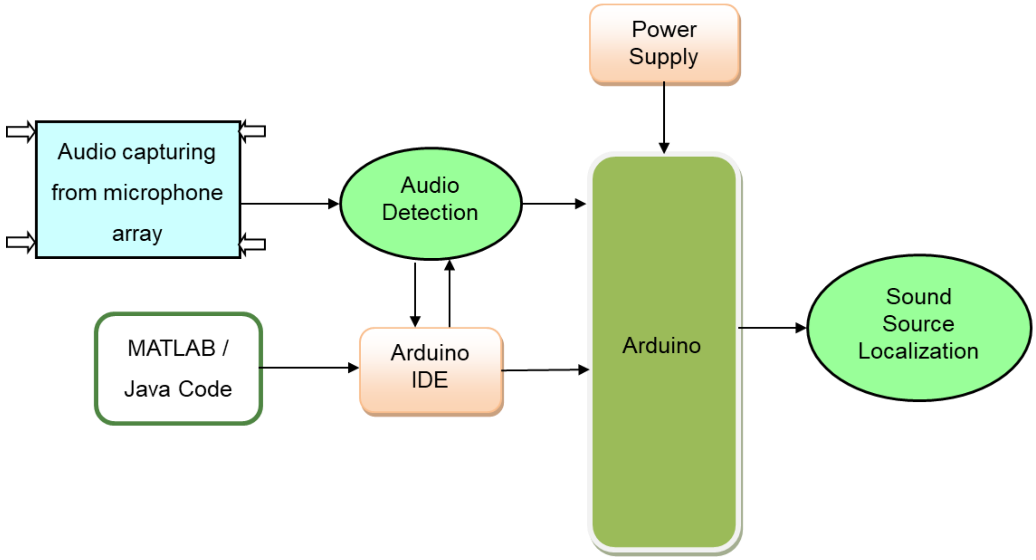

Figure 17: Block Diagram of Proposed Method

4.1 Beamforming method

Acoustic beamforming using a microphone array is done to extract desired signals in an interference-dominant, which is a noisy environment. Though noise is not considered in our work, this operation helps us to avoid unnecessary sounds and enhances signal quality for perception as well as further processing. As shown in Figure 16 with the use of 9 microphone arrays it is difficult to operate with signal quality.

It is decided to use a set of six microphone arrays, which are defined in two sets of ULA shown in Figure 18. Each array contains three omnidirectional microphones spacing at 50cms.

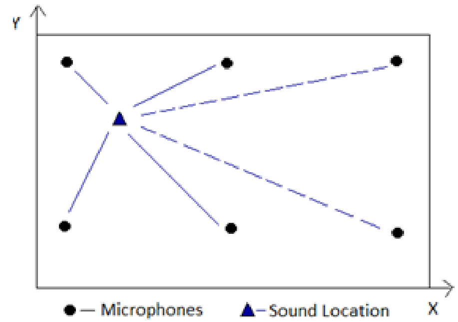

Figure 18: 2D mapping of source localization using six microphones.

Next, the received signals are simulated by the microphone array using the following algorithms and equations.

4.2 Classification of Algorithms and Equations

$$x_m(t) = S_i(t) \sum_{i=1}^{d} e^{j(m-1)\mu_i} + n_m(t), m = 1 \quad (16)$$

This equation can be written as,

$$x = [a(\mu_1), a(\mu_2), \dots a(\mu_d)] \begin{bmatrix} S_1(t) \\ S_2(t) \\ \vdots \\ S_d(t) \end{bmatrix} + n(t) \quad (17)$$

$$x = As(t) + n(t) \quad (18)$$

The total averaged power out of an array over K is expressed as,

$$P(w) = \frac{1}{k}\sum_{k=1}^{K} |Y(t_k)|^2 = \frac{1}{k}\sum_{k=1}^{K} w^H X(t_k) X^H(t_k) \quad (19)$$

Which implies,

$$P(w) = w^H R_{xx} w \tag{20}$$

Suppose ULA consists of M number of elements, where A(θ) is defined as the steering vector with a scanning angle θ.

Where, the idea is to scan across the angular region of interest, with weighting factor,

$$w = A(\theta) \tag{21}$$

Substituting w in above equation $P(w)$, the output power is calculated for the classic beamformer as a function of angle of arrival.

$$P_{classic}(\theta) = w^H R_{xx} w = A(\theta) R_{xx} A(\theta)^{\ H} \tag{22}$$

$$w = \frac{{R_{xx}}^{-1} A(\theta)}{A(\theta) {R_{xx}}^{-1} A(\theta)^{\ H}} \tag{23}$$

This gives the expression for the MVDR spatial spectrum,

$$P_{MVDR}(\theta) = w^H R_{xx} w = \frac{1}{A(\theta) {R_{xx}}^{-1} A(\theta)^{\ H}} \tag{24}$$

Whereas for the MUSIC spatial spectrum, an estimate R_{xx} of the covariance matrix is obtained and its eigenvectors are separated into signal and noise subspace and the direction of arrival estimation is estimated from one of these subspaces. Assuming that the noise in each channel is uncorrelated, which leads to a diagonal covariance matrix.

$$R_{xx} = A(\theta) S_s A(\theta)^{\ H} + \sigma^2 I \tag{25}$$

Where, $A(\theta) = [a(\theta_1), a(\theta_2), \dots\ a(\theta_d)]$ is a M × d array steering matrix. σ^2 is the noise variance and I is an identity matrix of size M×M.

Which implies,

$$R_{xx} = A(\theta)S_s A(\theta)^{\,H} + \sigma^2 I = Q\Lambda Q^H \tag{26}$$

With Q unitary and a diagonal matrix Λ = diag { $\lambda_1, \lambda_2, \ldots\ \lambda_L$}, the vector orthogonal to A is the eigenvector of R having the eigenvalue σ^2 [15].

Now, $R_{xx} = S_s\Lambda_s\, {S_s}^H +\ S_n\Lambda_n\, {S_n}^H$, where $\Lambda_n = \sigma^2 I$.

All the eigenvectors with noise are orthogonal to A, so S_s columns should span A's space range. While the S_s columns span their orthogonal addition. The projection operators of the signal and noise subspace are defined as,

$$\prod =\ S_s\, {S_s}^H = A\ (A^H A)^{-1}\ A^H \tag{27}$$

$$\prod\nolimits^{\perp}\ =\ S_n\, {S_n}^H = 1 - A\ (A^H A)^{-1}\ A^H \tag{28}$$

Hence, the MUSIC spatial spectrum is defined as [16].

$$P_{MUSIC}(\theta) = \frac{A(\theta)\ A(\theta)^{\,H}}{A(\theta) \prod^{\perp} A(\theta)^{\,H}} \tag{29}$$

4.3 DOA based localization

4.3.1 Single Source Localization based on DOA

When one location must be detected and localized, it is a single source localization case. The line from a point of sound source to any of a single microphone sensor is said to be a bearing line. The intersection of these lines from the source locations to the sensors are the estimated DOAs and the calculation method for these estimated DOAs for source localization is known as sound triangulation. In Figure 19 an example of sound triangulation is illustrated with one active sound source and two microphones Mic 1 and Mic 2 with two estimated DOAs of $\theta_1 and\ \theta_2$.

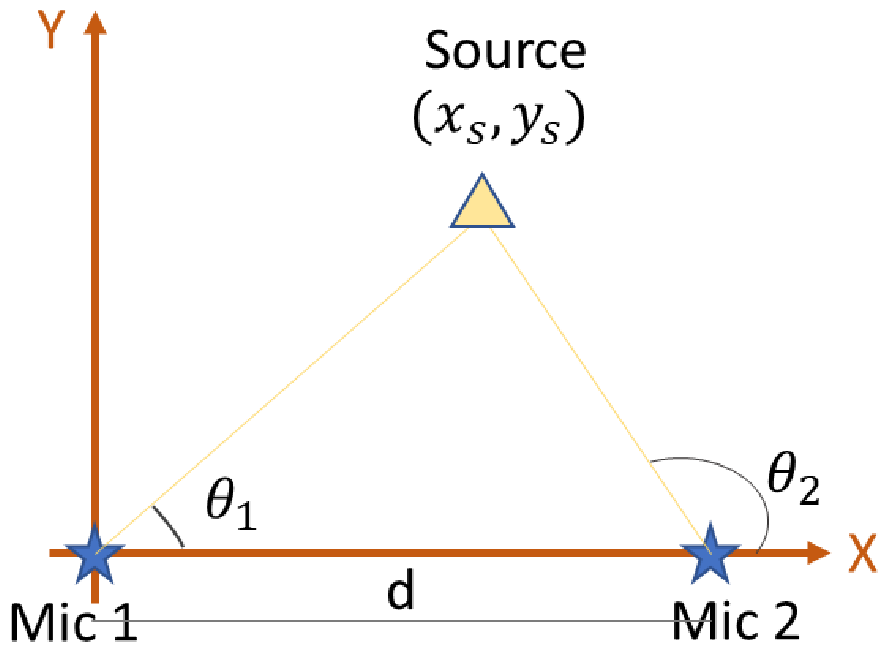

Figure 19: Triangulation with two nodes Mic 1 and Mic 2, and one active sound source.

4.3.2 Sound triangulation

Triangulation is a method to find an unknown point using known angles. That is in trigonometry and geometry, the process of determining the location of a point by forming triangles from known points is triangulation.

This process is mainly used in surveying along with other purposes such as navigation, astrometry, binocular vision, model rocketry and gun direction of weapons. One of the problems that closely relates to sound triangulation is a target motion analysis. The main aim in target motion analysis is to estimate source position but the difference would be to seek the target velocity from DOA measurements that a single moving or multiple moving observer has acquired.

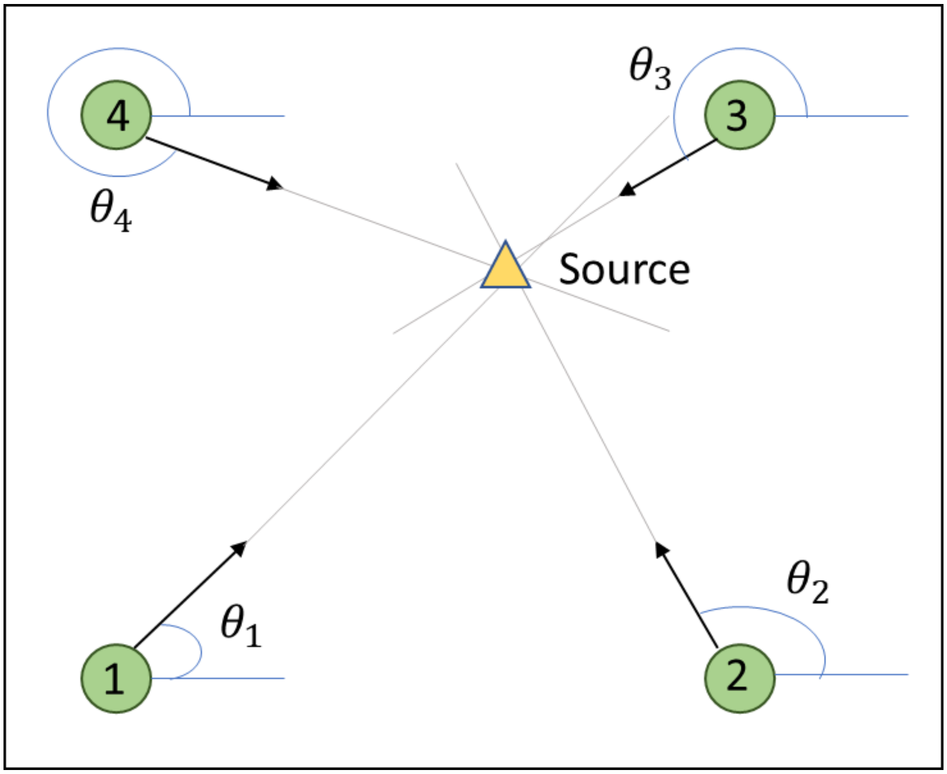

Figure 20: Triangulation using DOA estimates in a network of 4 nodes and one active sound source.

Figure 20 illustrates the target location with four microphones 1 to 4 and their estimated DOAs from the source are $\theta_1, \theta_2, \theta_3, and\ \ \theta_4$ respectively.

Considering the M nodes in a network of microphone sensors at $q_m = [q_{x,m}\ q_{y,m}]^T$, the function relating to x = $[x\ y]^T$ with its real azimuthal DOA estimate at node m is

$$\theta_m = arctan\left(\frac{y - q_{y,m}}{x - q_{x,m}}\right) \tag{30}$$

Where, arctan ( ) is the inverse tangent function.

This research mainly focuses on two- dimensional sound source location estimation problem. That is, here the main concern would only be with the azimuthal angle, which is needed to estimate the DOAs and TDOAs. When there is information about the elevation angle in some cases, the case would extend to a three-dimensional sound source localization. since this research is only on 2D tap localization the elevation angle is zero.

However, DOA estimates will be contaminated by noise and triangulation will not be able to produce a unique solution. To tackle the triangulation problem of statistical estimators while noise is not considered. Hence, maximum likelihood (ML) estimation is needed, it is possible to derive the ML location estimator by minimizing the cost function [30].

$$J_{ML}^{(DOA)}(x) = \sum_{m=1}^{M} \frac{1}{\sigma_m^2} (\widehat{\theta_m} - \theta_m(x))^2 \tag{31}$$

Where σ_m^2 is DOA noise variance at mth sensor.

Here, the computational efficiency is our primary concern by the intersection point method [11]. This method is based on intersection of the bearing lines from source to the microphones. That would be the centroid of intersection of all these bearing lines. Each point in the set is the mean of centroid that minimizes the sum of squared distances between itself at a set of intersection points.

To increase the vitality in lack of geometrical conditions, the method considers a systematic plan of identifying and denying the outliers that occur from the intersection of nearly spaced parallel pairs of bearing lines. Also, to attain the accuracy and computational complexity there is a method called grid-based method (GBM) [32]. Which is based on constructing a grid g of N_g grid points in the search space where the sound source is present in the localization area.

Here considering the measurements of angles is the criteria and the GBM suggests the use of angular distance, the GBM values are usually range between [0, π]. To the absolute distance, it is an appropriate measure of similarity. The source location estimation by GBM is done by identifying the grid point based method.

$$J_{GBM}^{(DOA)}(x) = \sum_{m=1}^{M} \; [A(\widehat{\theta_m}, \theta_m(x))]^2 \tag{32}$$

$$\hat{X}_s^{(GBM)} = argmin_{x \in \text{g}} \; J_{GBM}^{(DOA)}(x) \tag{33}$$

Where, arg denotes the two arguments.

A dense grid is required to minimize the total position error. That is what can remove a high N_{g} value.

In [33] it is mentioned that the GBM is much more efficient in computational terms than the nonlinear least square estimators with same accuracy.

4.3.3 Multiple Source Localization based on DOA

A fundamental problem in localization of sound origin from each of the nodes is that there must be a correct association of unknown DOAs. In order to perform a triangulation process, one must first estimate the right DOA combinations form the microphone sound sensor nodes that correspond to the same source when considering multiple sound sources or a single sound source in multiple locations at different time instants.

When using the DOAs that are belonging to different sources, the result will be estimated as the source to be ghost sources i.e., estimation of false sound locations is calculated. Source positions that do not match to original sound sources may affect the performance in estimating the true position of the sound source. Such affect is commonly addressed as the data-association problem in signal processing.

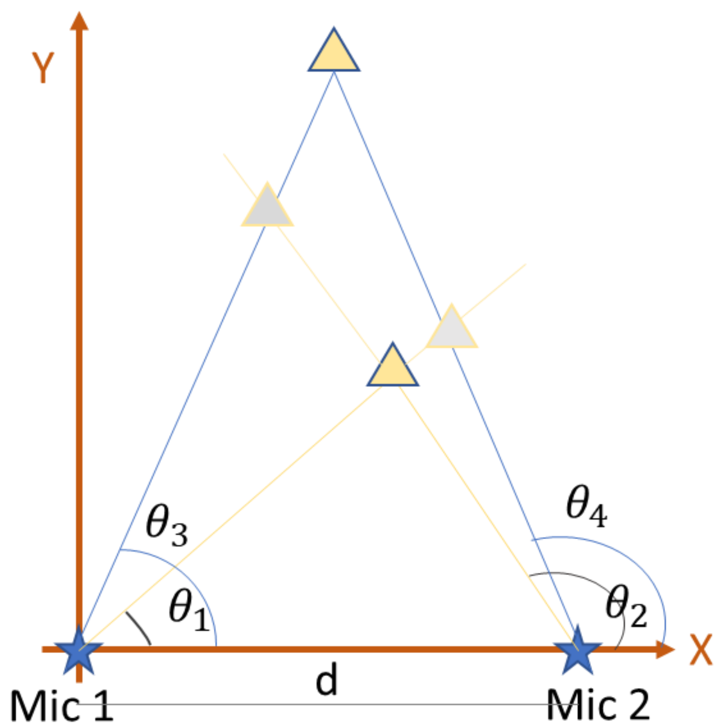

Figure 21: Illustration of the data association problem.

The data association problem is explained in Figure 19, the network with two nodes Mic 1 and Mic 2 with a separated distance of d and two active sound sources. The lines with angles $\theta_1 and\ \theta_2$ show the DOAs to the source that is near to two microphones and the lines with angles $\theta_3 and\ \theta_4$ are the DOAs which far from the microphones. The intersection of all these lines refer to the bearing lines from sensors which will result in sound source localization at 4 possible points that would result in false estimation of sound sources.

In Figure 19, the triangles in yellow color correspond to the true sources' locations. The true DOA combinations are estimated by using the DOAs that are from the node, which have a close similarity to the same source. The triangles in grey color are called as ghost sources which results in misleading DOA estimation, often leading to false estimation of sound source location.

When nodes are far away from sources some arrays might underestimate their number of multiple active sources or when nodes are closer to the sources in terms of their angular separation also underestimate in number of active sources[34]. Hence, there is a huge change of missed detections that can occur, means some arrays may be missing from DOA estimations of the sources.

In this study, the implementation of the single source localization is done and will be focusing mainly on this type of localization in the following chapter.

4.4 Time difference of arrival (TDOA) based localization

In a network, when there are multiple microphones, localization can be achieved with maximum productivity with less effort. It is Required that the signal processing to each node in all possible ways should lead to a better understanding of TDOA measurements and then later combining all these measurements will yield to a better sound source localization in a 2D-plane.

When all the microphone sensor nodes are connected via a low bitrate channel, the internal clock is not guaranteed to be synchronized. When considering the microphone arrays for sound detection and localization this network strategy is a must. Among most of the possible measurements, there is a possible solution in TDOA method[35].This method of TDOA to locate sound source is a smarter work and can be done in real-time.

Since it can work well with only single sound source localization, opting this technique in this work would be a better choice.

4.4.1 Generalized cross correlation in TDOA generation

An equivalent and worthy comparison parameter is need in this work. To estimate the TDOA that could be used for signal source transfer, from sound generating source to sound sensor and internal calculation. Which is common thing to be done in many sound source localization methods before estimating the TDOAs between the microphone pairs. The comparable parameter that is known to us in signal source localization tracking system is a time difference of sound source arrival among the microphone sound sensors which can be constituted by the geometric characteristics and implicit by its propagation characteristics [36] [35].

Using a Fourier transform of cross-correlation function the cross power could be generated. The distribution of signal power is described by power spectrum that will be considered among different frequencies ranges, random structure of microphone array system and relative power of the TDOA signals [37]. The TDOA is defined from the Equation (15) there will be a maximum representation of the power spectrum signal with certain weighting factor, as given in the equation below,

$$\dot{\tau} = argmax(\beta) \int W\,(\omega) M_1(\omega) M_2(\omega) e^{-j\omega\beta} d\omega \tag{34}$$

Where, $\dot{\tau}$ is an estimated value of τ, $M_1(\omega)$ and $M_2(\omega)$ are the Fourier transform of the first and second compared signals respectively. $W(\omega)$ is a cross-correlation weighting function. There are two different choices for $W(\omega)$, that could be shown in the below equation,

$$W_{PHAT}(\omega) = \frac{1}{|M_1(\omega)||M_2(\omega)|} \tag{35}$$

Or,

$$W_{GCC}(\omega) = 1 \tag{36}$$

In case of reverberant environments while using this cross correlation function with a weighting function know as Phase transform (PHAT) provides a better accuracy to the system and is known to be effective [29].

A normal cross-correlation function without adding any weights corresponds to an unfiltered cross correlation technique, when using UCC weights.

GCC for discrete-time signals is expressed as,

$$\dot{\tau} = argmax(\beta) \sum_{k=0}^{\frac{N}{2}} W(k)|M_1(k)||M_2(k)|\, cos(\theta(k)) \tag{37}$$

$$\theta(k) = \angle M_1(k) - \angle M_2(k) - \frac{2\pi F_s k\beta}{N} \tag{38}$$

F_s is the sampling frequencies, k is the index of discrete Fourier transform, N is the total number of the samples in each segment and θ(k) is defined as the phase error. preferably, phase error would be close to zero, in Equation (37).

4.4.2 Triangulation Algorithm

In this approach before the use of the generalized cross-correlation algorithm, there will be a comparable parameter generated for each sensor which is TDOA between the microphone sound sensors and is a relative value among different pair of sound sensors. In geometrical approach finding these TDOAs will not solve the problem in sound localization, that is the TDOA has no meaning. In order to reach the main goal of localization using these TDOAs, sound triangulation method is being used.

Geometrical solution is also needed, that can locate the sound source point to represent the known TDOA values and the know sensor positions, such as the coordinate system that could describe the actual position of the sound source. It may be spherical, tubular or cartesian [38].

Spherical microphone arrays are becoming increasingly important in acoustic signal processing systems for their applications in sound field analysis, beamforming, spatial audio, etc. In many of the previously mentioned applications, target positioning and interfering with sound source is a crucial step.

Localization of 3D sound sources is therefore a very relevant topic in the field of acoustic signal processing. However, spherical microphone arrays are usually composed of many microphones and it is an important issue to run signal processing methods for localization in real-time.

Some works have already shown the potential of graphics processing units (GPUs) for developing high-end real-time signal processing systems. New embedded systems with integrated GPU accelerators providing low power consumption are becoming increasingly relevant. In this new era of smartphones and tablets, these new systems play a very important role, opening additional possibilities for designing high-performance compact processing systems.

The real-time capabilities of these platforms are analyzed, providing also a performance analysis of the localization system under different acoustic conditions [24].

By this approach, implementing our hardware will be easy and to run it, in such a way that it localizes the sound in a coordinate system.

Considering the TDOA with at least 4 microphones, using a reference microphone at the center of the array which is the origin of cartesian coordinate and the array is arranged with an X-shape in the x-y plane, as shown in Figure 20.

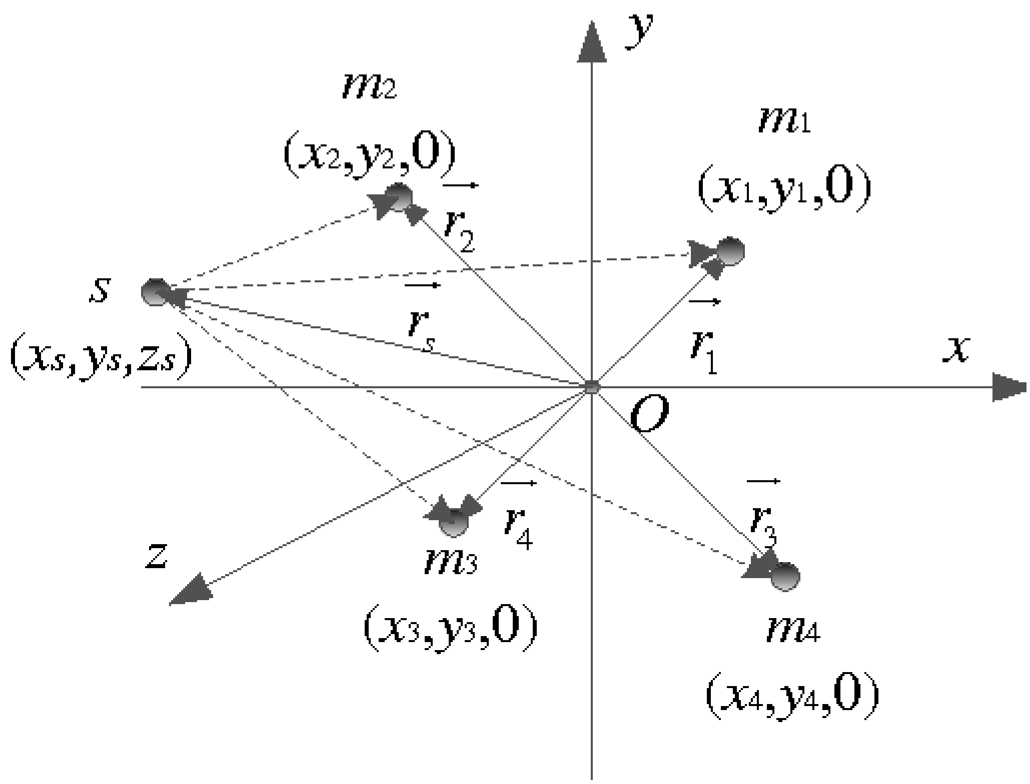

Figure 22: TDOA sound source localization model with multi-sensor array configuration [39].

The microphones are numbered as $m_1, m_2, m_3 and\ m_4$ with their respective coordinates. In geometry the microphones m_i $(i = 1\ to\ 4)$ are located at a fixed position, with coordinate $(x_i, y_i, 0)$. The sound source is located at an unknown location point in the space, with coordinate (x_s, y_s, z_s). One microphone act as a reference microphone, in this case it is microphone m_1.

The geometry relationship can be written as,

$$\left\|r_s - r_j\right\| = \left\|r_s - r_i\right\| + c\Delta t_{i,j} \tag{39}$$

Where $j = 2,3,4$ is the TDOA between m_j and m_1.

If j is positive, it is considered that m_j is at a greater distance to sound source than m_1. If j is negative, m_j is at a shorter distance to sound source and c is the speed of sound which is a constant of 340 m/s. The symbol $\|.\|$ represents the length of a vector.

Equation (34) can be rewritten as,

$$\sqrt{(x_s - x_k)^2 + (y_s - y_k)^2 + {z_s}^2} = \sqrt{(x_s - x_1)^2 + (y_s - y_1)^2 + {z_s}^2} + c\Delta t_{1,k} \tag{40}$$

Equation (36) can be obtained by deriving from Equation (35),

$$x_s = \frac{b_2 - b_1}{a_1 - a_2}, \mathrm{y_s} = \frac{\mathrm{a_1 b_2 - a_2 b_1}}{\mathrm{a_1 - a_2}},$$

$$\mathrm{z_s} = \pm\sqrt{\left(\frac{2(\mathrm{x_1 - x_2})\mathrm{x_s} + 2(\mathrm{y_1 - y_2})\mathrm{y_s} + \mathrm{c_2}}{2\mathrm{c}\Delta \mathrm{t_{1,2}}}\right)^2 - (\mathrm{x_s - x_1})^2 - (\mathrm{y_s - y_1})^2} \tag{41}$$

Where,

$$\mathrm{a_1} = \frac{(\mathrm{x_1 - x_3})\Delta\mathrm{t_{1,2}} - (\mathrm{x_1 - x_2})\Delta\mathrm{t_{1,3}}}{(\mathrm{y_1 - y_2})\Delta\mathrm{t_{1,3}} - (\mathrm{y_1 - y_3})\Delta\mathrm{t_{1,2}}},$$

$$\mathrm{a_2} = \frac{(\mathrm{x_1 - x_4})\Delta\mathrm{t_{1,2}} - (\mathrm{x_1 - x_2})\Delta\mathrm{t_{1,4}}}{(\mathrm{y_1 - y_2})\Delta\mathrm{t_{1,4}} - (\mathrm{y_1 - y_4})\Delta\mathrm{t_{1,2}}},$$

$$\mathrm{b_1} = \frac{\mathrm{c_3}\Delta\mathrm{t_{1,2}} - \mathrm{c_2}\Delta\mathrm{t_{1,3}}}{2\left[(\mathrm{y_1 - y_2})\Delta\mathrm{t_{1,3}} - (\mathrm{y_1 - y_3})\Delta\mathrm{t_{1,2}}\right]}, \tag{42}$$

$$\mathrm{b_2} = \frac{\mathrm{c_4}\Delta\mathrm{t_{1,2}} - \mathrm{c_2}\Delta\mathrm{t_{1,4}}}{2\left[(\mathrm{y_1 - y_2})\Delta\mathrm{t_{1,4}} - (\mathrm{y_1 - y_4})\Delta\mathrm{t_{1,2}}\right]},$$

$$c_2 = x_2^2 + y_2^2 - x_1^2 - y_1^2 - \left(c\Delta t_{1,2}\right)^2$$

The localization of the sound source is presented in Equation (36).

Here since the z-coordinate is the square root of a function, it will contain two solutions, one positive and the other negative.

In practical application, the sign of z-coordinate is established in advance, so that the z-coordinate can be determined [36] [40].

5 IMPLEMENTATION

At first, uniform linear array needs to be modeled that contains 6 microphones spaced at 0.5 meters apart. Assuming, that the two narrowband signals are striking on the linear array at two different angles of azimuth at the operating frequency of 300 MHz, with a thermal noise power of 0.01 watts at each microphone. Keeping the elevation angle to 0° and assuming the first and second angles of azimuth are at +40° and -40° respectively.

Two received signals are used to find the direction of arrivals. Because the signal received at the linear array is symmetric around its axis, it is sometimes difficult to obtain azimuth angle.

But broadside angle can be estimated with respect to the linear array. Broadside arrays with more than two or more microphones can be constructed by simply adding the additional microphones in line with the original three. The relationship between frequency and wavelength of sound is given as $c = f \times \lambda$, where c is speed of sound in air which is 343 m/sec. Distance of sound traveled in a specific time $d = c \times t$, where t is time in seconds.

At +40° and -40° respectively, implementation of broadside angles corresponding to two incident directions are described. Sometimes it is seen that if the elevation angle is zero and the azimuth angle is within [-90° 90°], at that instant the broadside angle will be same as the azimuth angle.

Other methods used in improving the resolution are Minimum variance distortionless response (MVDR) and Multiple signal classification (MUSIC) estimations [9]. A subspace method of high-resolution DOA estimation with accuracy is maintained by MUSIC algorithm. To indicate the DOAs of all the received signals at all the six microphone sensors. It is considered that the peaks of the output spatial spectrum, for all these three methods that are being used to select one method that is suitable for us in this work. Figure 22 illustrates the flow chart of complete implementation to be used in obtaining the SSL point of the microphone array system.

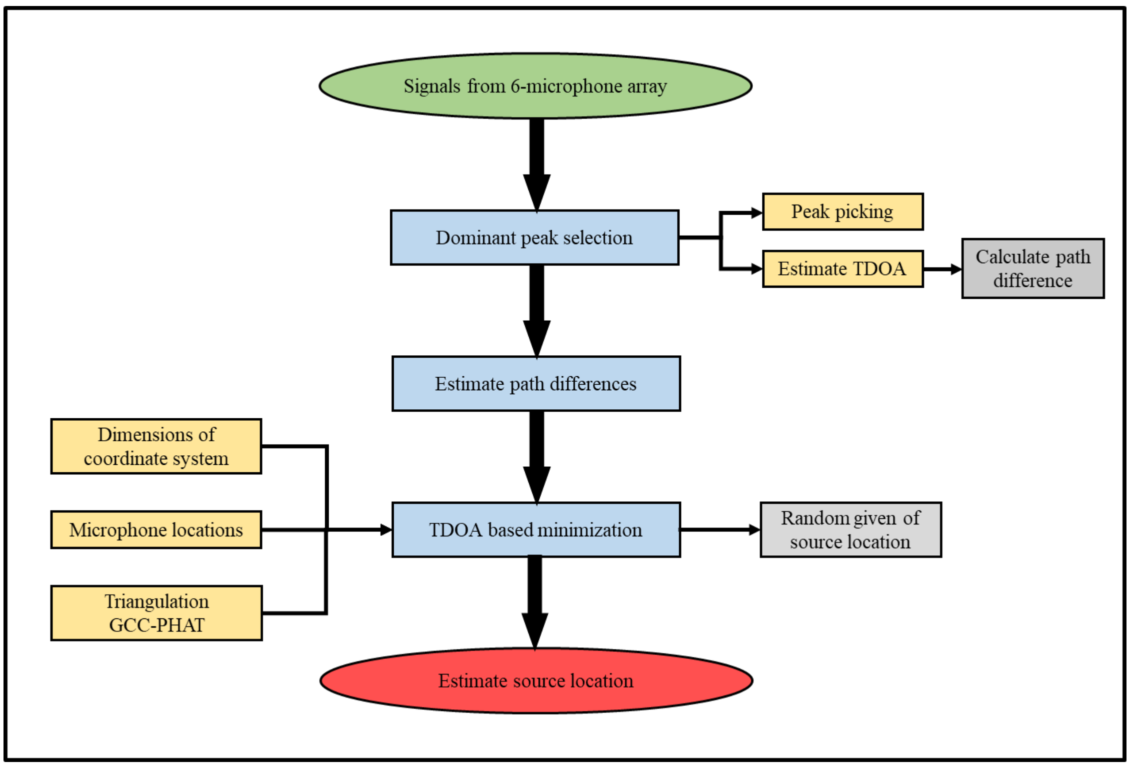

Figure 23: Flow chart of complete hardware and software implementation.

5.1 Considerations, limitations and challenges

There are several aspects to be considered when developing these prototypes

1. The device must know its sensor locations
2. Manual configuration (fewer costs)
3. GPS configuration (greater costs)

4. May be both
5. The device must have a good sampling rate of its sensors
6. The sensors location and configuration are of major importance, which should be considered carefully
7. The number of sensors affects the precision. The more the sensors, more the precision.

5.2 MATLAB simulation

Different effects of signals are found when changing their parameters on the performance of different DOA algorithms. In our case, beamscan, MVDR and MUSIC algorithms are used to see this change and the suitable method is taken into consideration. Considering the angles that are near to each other because while performing calculations using hardware, the best method is needed that can identify the peak values of closely spaced angles.

Across all the scan region to estimate the source point a spatial spectrum is formed when using a beamscan estimator. It is needed to set the DOA output port property, the DOA output property is needed to set to true to estimate the true DOAs and the number of signals property to of the output port is set to 2 to find the top two peaks values of the true source locations.

Plotting the spatial spectrum of the beamscan output. Estimating DOAs show the correct power output values which are +40° and -40°. It is also seen that the conventional beam of two different signals at different angles cannot resolve to identify the difference between two closely spaced signals, shown in Figure 24.

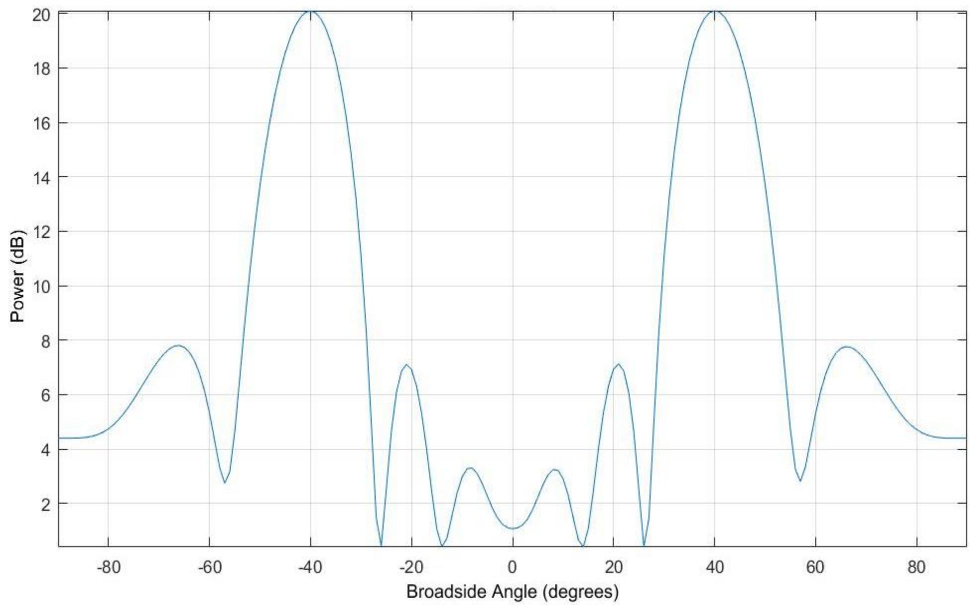

Figure 24: Beamscan output.

In Figure 23, when the signals are not closely spaced it is seen that beamscan output is able to differentiate the power outputs of two different angles.

But when the two angles are closely spaced, it is unable to differentiate between two closely spaced signals that is because when two signals are less than the beamwidth. Hence it is understood that beamscan will not pass to estimate the DOA of two closely spaced signals which is one of the drawbacks. For example, assuming that there are two signals with a 10° difference in azimuth angle. Let’s consider the two signals with azimuth angles 50° and 60°.

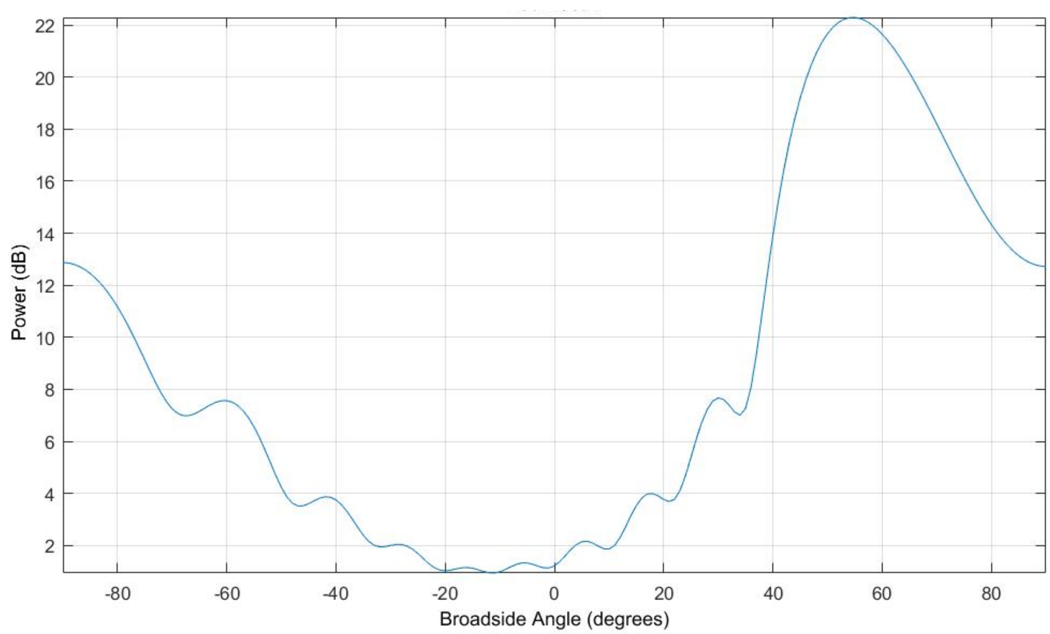

Figure 25: Beamscanned output of two closely spaced signals.

As you can see in Figure 23, the output has only one dominant peak. Therefore, confirming that it cannot resolve two closely spaced signals. To resolve this issue, the use of MVDR algorithm and the MUSIC algorithm helps.

First, an analysis on MVDR estimation is performed because an MVDR beam has a higher resolution when compared to Normal beamscan. Also considering the beamwidth, MVDR has a smaller beamwidth than beamscan[10].

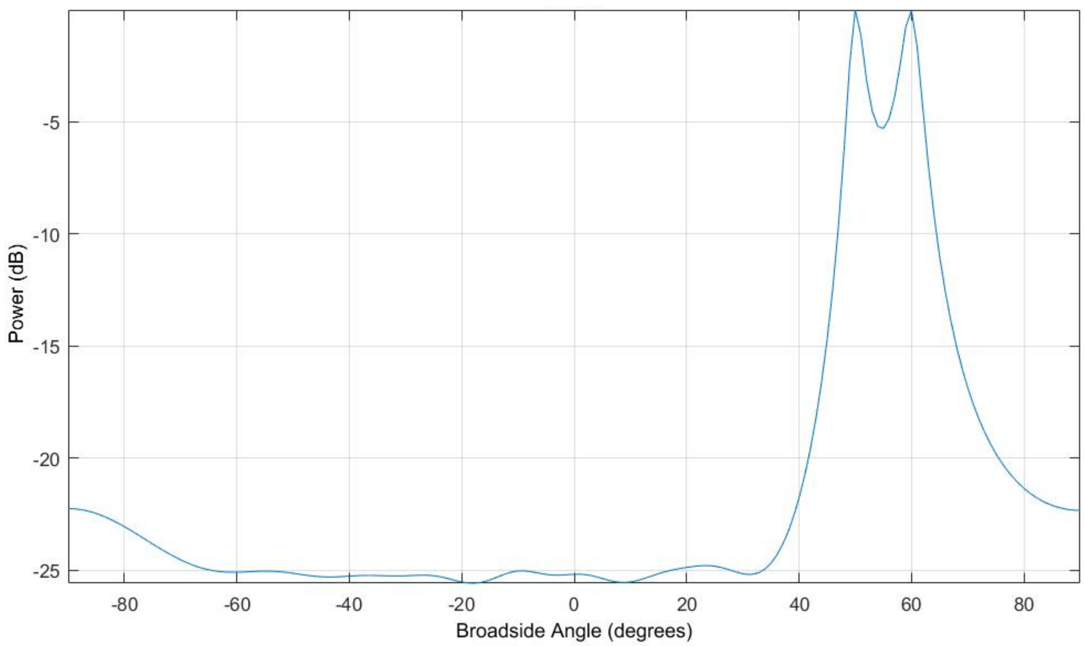

Figure 26: MVDR beamscan output.

From the above result, it is seen that the MVDR algorithm can differentiate between two closely spaced signals and correctly estimates the DOAs which was not resolved by Normalized beamscanning.

This proves that the MVDR is more accurate in differentiating two closely spaced signals. If the sensor positions are inaccurate then the MVDR may fail to estimate the correct DOAs.

It is true that MVDR is more sensitive towards the sensor positioning errors. Hence, if the two DOAs are much more closely spaced, there is a chance that the MVDR algorithm estimation of DOA may also fail.

Now the MUSIC algorithm is checked to see if it can resolve these closely spaced signals problem and estimate the correct DOAs with higher accuracy than MVDR.

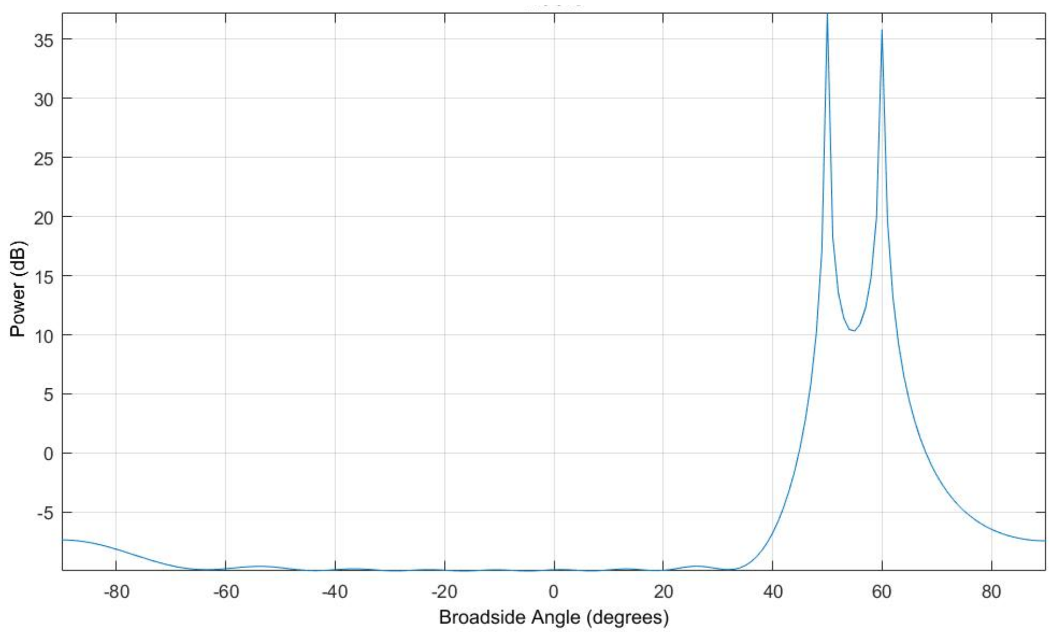

Figure 27: MUSIC beamscan output

The DOAs using the MUSIC algorithm for two closely spaced signals are correct and accurate.

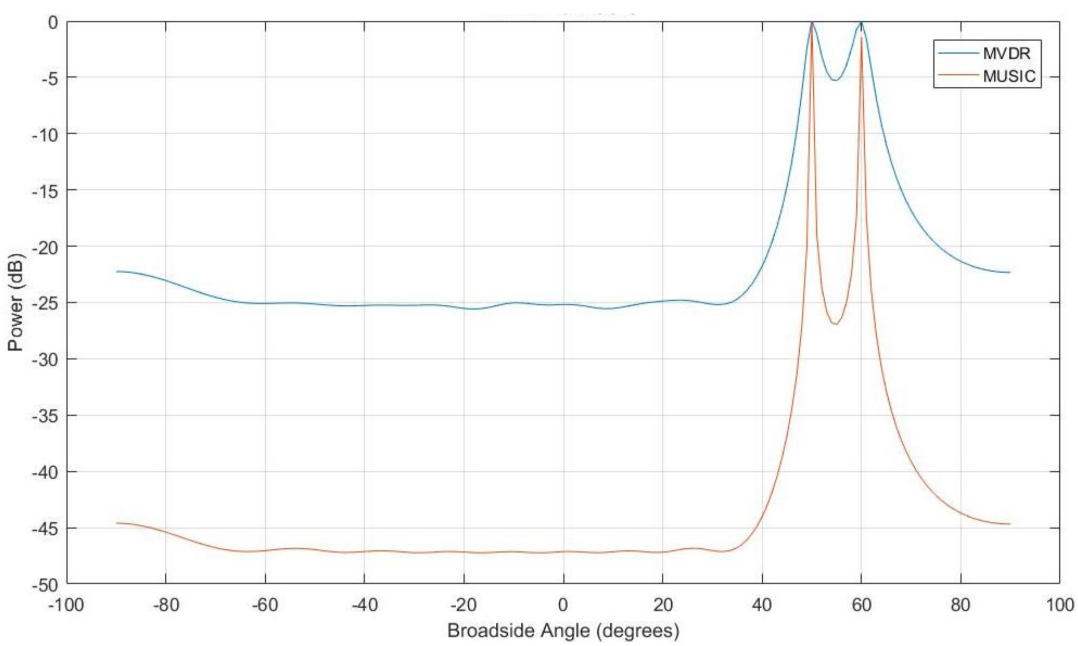

Figure 28: Comparison of MVDR and MUSIC

Estimating the DOAs of the two sources that are closely spaced and comparing the spatial spectrum power output peaks of MVDR to the spatial spectrum power output peak of MUSIC.

MUSIC provides a better gain, better accuracy and high resolution when compared to MVDR and Normal Beamscan. Hence, the MUSIC algorithm estimates the correct spatial spectrum for two closely spaced signals. If the number of sound sources are not specified to the MUSIC algorithm, even MUSIC may be inaccurate to itself in estimating the correct DOAs. If the number of sources is more then there is a chance that it might be even difficult for MUSIC to estimate the correct DOAs for all the closely spaced signals.

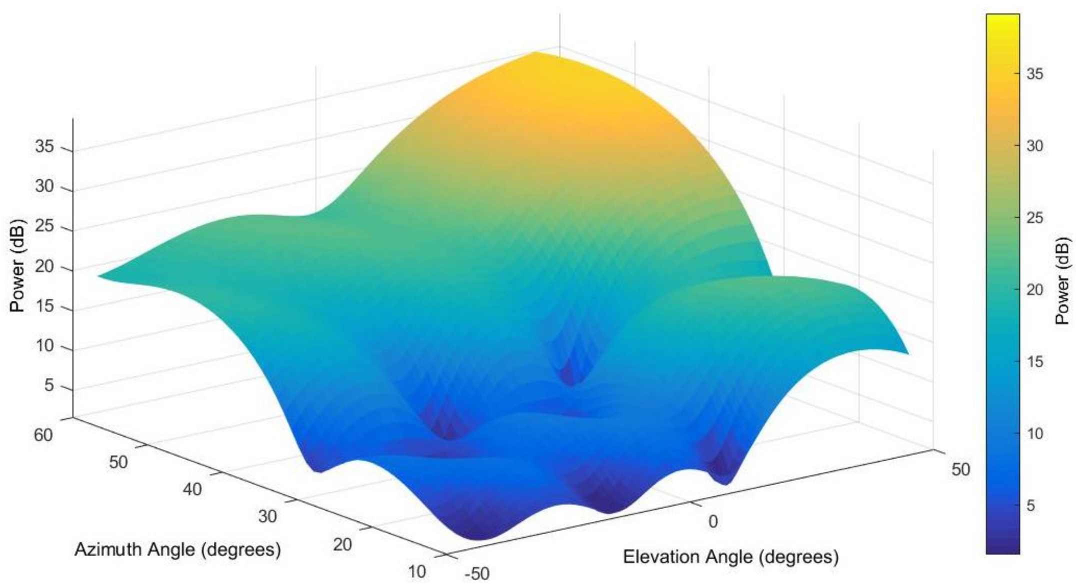

Figure 29: Power spectrum output of Beamscan

Figure 28 illustrates the power spectrum output of a beamscan signal of two closely spaced signals at 50° and 60° respectively. It is seen that the power output is having a high peak near those two angles, but it is difficult to differentiate because of the other signals. The power gain is high but unable to differentiate between closely spaced signals.

MVDR power spectrum output shows a peak value between two closely spaced signals eliminating the other signals.

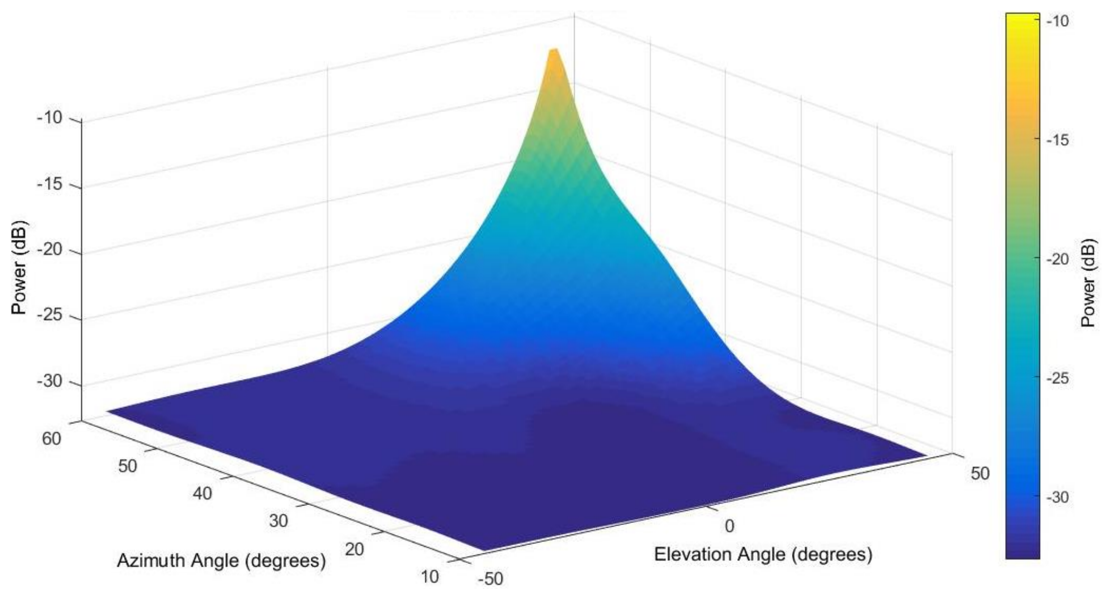

Figure 30: Power spectrum output of MVDR.

Even MVDR is unable to differentiate between closely spaced signals. The power gain output is comparatively low than the normal beamscan method shown in Figure 29.

The MUSIC power spectrum output is shown in Figure 30, the power gain output is better than MVDR power spectrum but is not great enough than the beamscan power spectrum output.

But the MUSIC is able to differentiate between two closely spaced signals, when compared with MVDR and Beamscan power spectrum outputs.

The power gain output is decent enough to perform the DOA estimation using MUSIC and can procced with other parameters like TDOA techniques. Comparing all the three power spectrums it is seen that with decent power can and able to differentiate between closely spaced signals MUSIC is the algorithm that is better suitable in this research for better results for sound source location calculations.

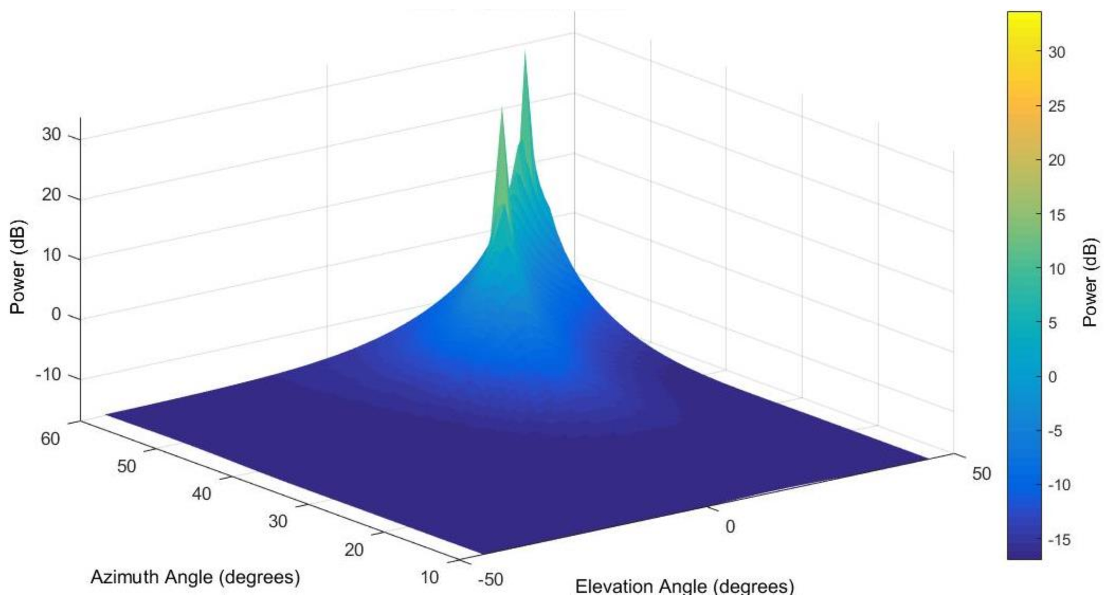

Figure 31: Power spectrum output of MUSIC

It is needed to have a generated comparable parameter, prior in approach of using the generalized cross correlation algorithm for each of the sound sensors respectively and in proportion to the value among them is the TDOA of a the signal[41]. Once the DOA estimations are done and the values of this are obtained, these values will have no meaning if it is not approached in a geometrical way. The geometric approach would be sound triangulation after calculating the values for TDOA. So, the main goal of localization is yet to be reached.

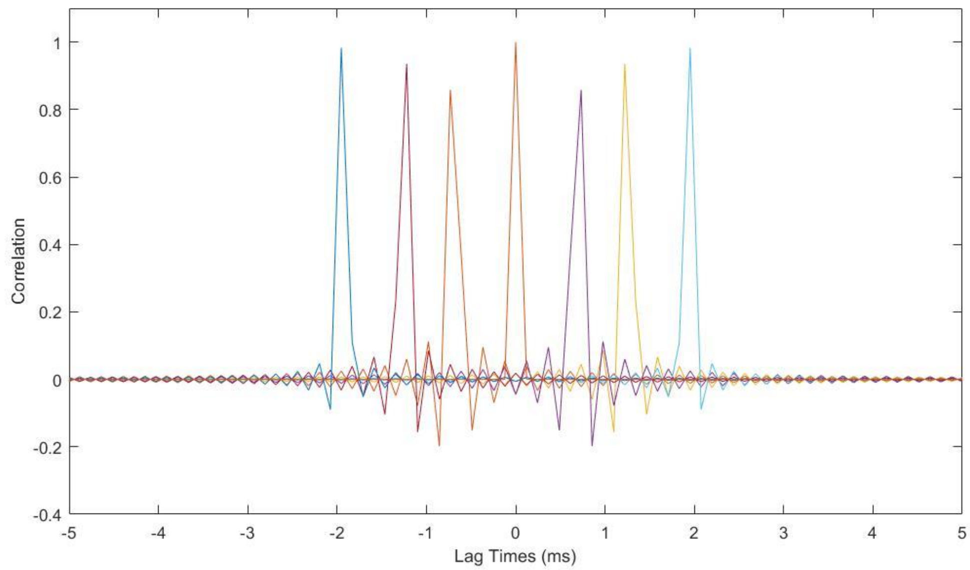

Figure 32: GCC result of all six microphones with one reference microphone.

Figure 29 gives us the GCC-PHAT-MUSIC of all the six microphones with one reference microphone when sound is generated exactly at the center. You can use correlation to estimate the TDOA at all the six sensors. From TDOA using sound triangulation and can estimate the sound source location.

Once the data is being collected from the microphone sensors, using the MUSIC algorithm, the sharp peak values from all the six microphone sensors are found. Take the best two or three peaks and estimate the source location using the triangulation method. The best peak detected values are needed because the highest peak or a greater number of peaks tells that the sound source is located near it.

5.3 Hardware implementation

5.3.1 Implementation 1-using rectangular array

Figure 32 illustrates the implementation of a rectangular array of about 1x1meter using microphone sound sensors network for source localization. It consists of six microphones, three of the microphone sound sensors, namely M1, M2 and M3 are placed along the x-axis at 50cms distance from each other and the other three microphones M4, M5 and M6 are placed parallel to the x-axis at about 100cms and 50cms from each other. They all are connected to Arduino Uno board, which is connected to the computer.

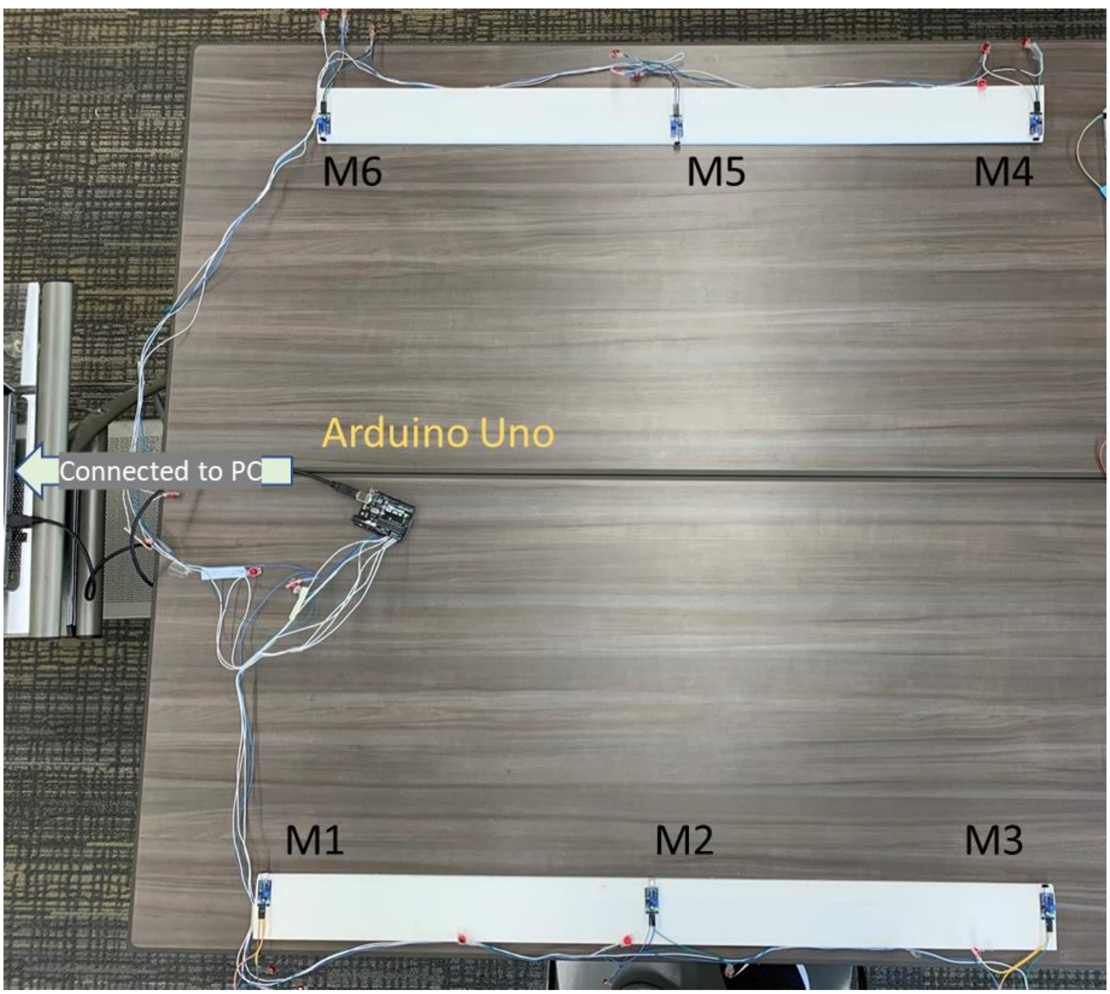

Figure 33: Hardware implementation for 2D-mapping using two parallel linear arrays.

The topology in between these microphone sensor nodes is derived from the centralized scheme. Even though it generates a traffic for our network topology, its reduced physical space with a point to point links between each sensor node and main system makes our sound localization hardware prototype attractive and reduces the programming complexity.

This network is easy to build due to its connection procedure; low-cost hardware communication system and it is easy for further modifications.

As far as the hardware and software components are concerned, the computer is core i5 with windows 10 as an operating system. There is a USB adapter to communicate with Arduino and the network nodes.

The functionality at the system was coded in a java programming language using Arduino IDE framework, to run the Arduino uno hardware-based sensor nodes as the low-cost embedded processor and standard acoustic transducer. Lastly, the localization module is coded in MATLAB to map the localized point and run on the same computer in which performs other tasks such as TDE calculation and sound triangulation methods.

The location of the sound source can be calculated by means of the arrival direction when each node is integrated with several microphones. Sometimes it took more computational complexity at the microphone sensor nodes to perform correct DOA estimated measurements that could achieve low-bandwidth use when transmitted to the network [42].

In approach to sound location estimation, the DOA measurements will explain the direction from where the sound is propagated to the microphone sensor at each instant of time whenever a sound tap is done.

5.3.2 Implementation 2-using single linear array

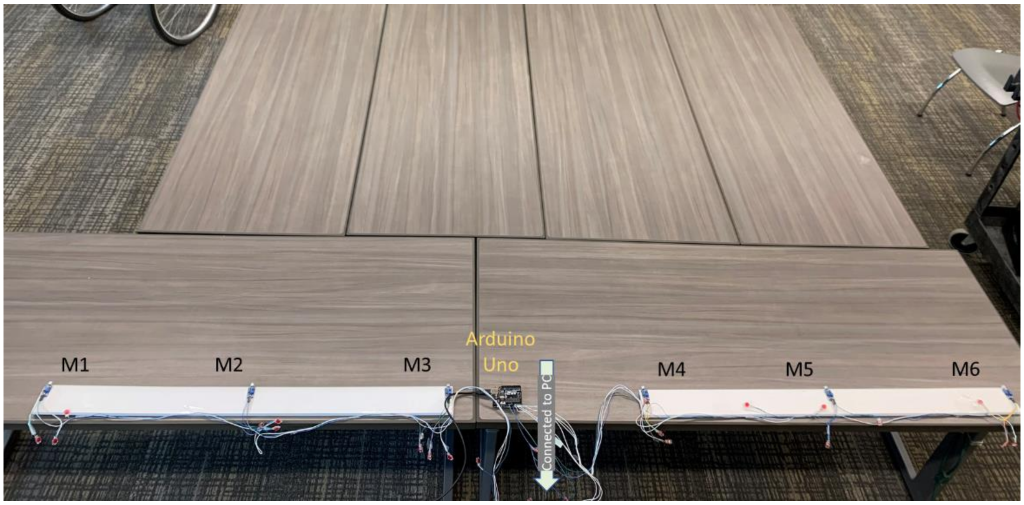

Figure 34: Hardware implementation for 2D-mapping using single linear array.

Figure 34 illustrates the implementation of single linear array microphone sound sensors network for source localization. This implementation also consists of six microphones, all of these six microphone sound sensors, namely M1, M2, M3, M4, M5 and M6 are placed along the x-axis at 50cms distance from each other. They all are connected to Arduino Uno board, which is connected to the computer.

The topology is derived from the basic network uniform linear array system, it takes a large physical space when compared with the other implementation. The network is easy to build due to its linear connectivity.

The sound tapping is done in front of the array system and the source position is determined by the DOA. The microphones that are near to the sound source position will detect the signal and once the microphone sensor nodes are known, the TDE values are calculated for 2D sound source positioning in MATLAB.

In this TDE-based localization approach, first it is analyzed to when a single source is active and then try presenting strategies for identifying multiple times by continuously tapping at different locations. Finally, methods of estimated source locations are discussed to that of original source locations as well as the number of sources.

5.4 MATLAB interface with Arduino Uno

To interface Arduino Uno with MATLAB it is required to have a support package. MATLAB provides an Arduino hardware support package where a serial COM port acts as a communication channel between the hardware that is being used in this research and MATLAB software, this helps in understanding the data calculations that is being received by microphone sensors.

Once the Arduino hardware support package has been installed the functionality of the system is checked thoroughly and if it fails, then the data logger is developed to assess the functioning of the system. Some care should be taken while performing trails when testing the tap detections using microphone sound sensors connected with Arduino because the sampling process introduces some noise, which can be adjusted using the on-board potentiometer on microphone sound sensor.

There are several functions associated with serial communication in MATLAB. Main step is to ensure that the Arduino board that is being used in the system should have access to serial COM port. In order to check it, in the Arduino IDE programming window, if you see COM31 as the port, it needs to be replaced with COM port to which your panel is connected and later the communication speed is set.

The following commands are used to implement the previously mentioned steps:

```
s = serial('COM31');
```

s.BaudRate=115200;

Next, the Serial port needs to be opened.

fopen(s);

Then, to read from it and store to var data, write:

dat = fscanf(s);

dado = str2double(dat);

It first reads the data and later converts it from char to double.

After finishing with the Serial COM port. Remember, it is very important to close it else other applications cannot access it.

fclose(s);

Lastly, the combination of these commands will check the ADC sampling. If you notice that everything is working well, then the connection of the A0 port from the Arduino is 5/3.3/0 volts which should provide a precise value on the logger.

If the scroll width parameter is changed from 10 to 0. Instead of just a section, a graph will show the entire log history. The data log is accessible after the plot has been closed by accessing the workspace information factor.

After executing the start function when a single microphone is connected to Arduino, the sound it receives will be converted into multiple parameters, such as b-bit number, x-sample data, and f-sampling rate.

In Figure 31, you can see the Arduino IDE output of all the six sensors when tapped continuously at different locations.

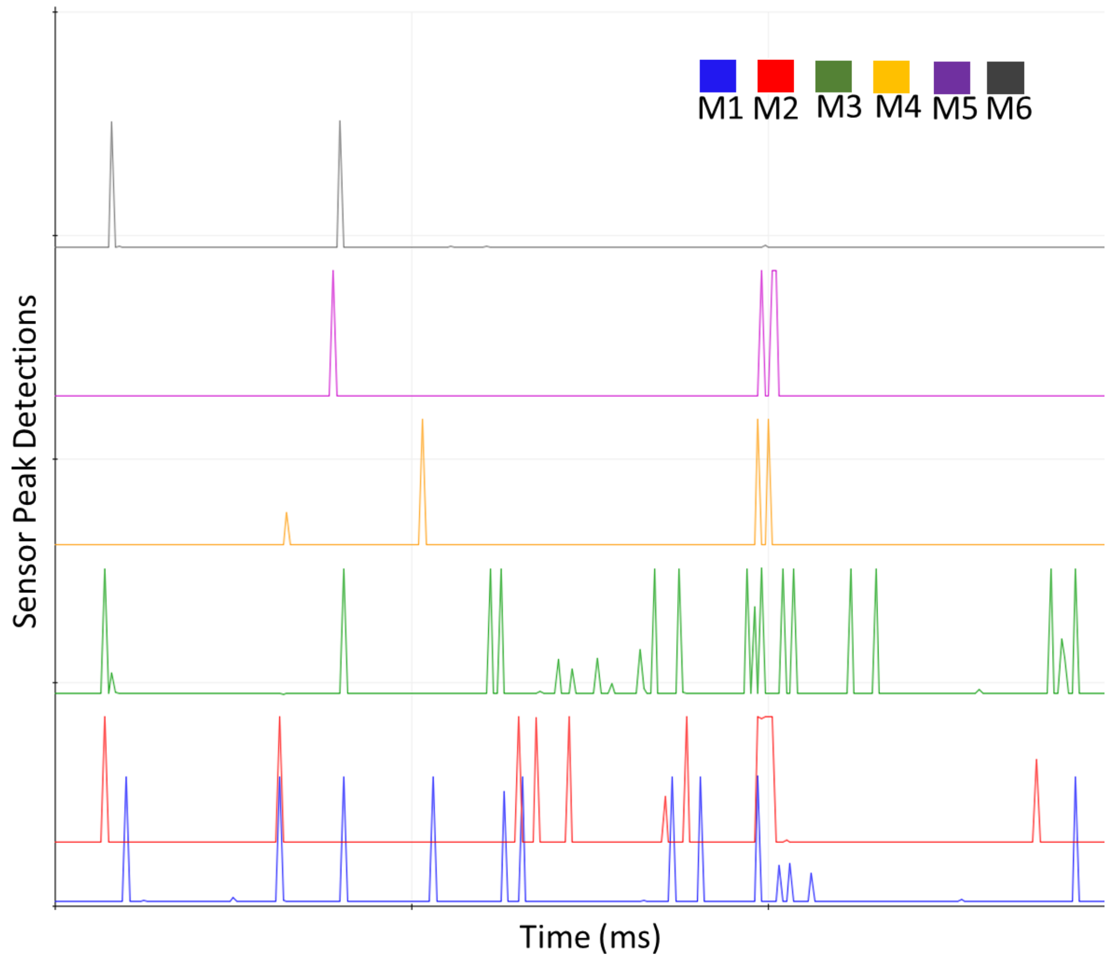

Figure 35: Microphone sensors data output on Arduino IDE screen.

Figure 35, you can see the Arduino IDE output of all the six sensors when tapped continuously at the midpoint of the table. Whenever the tap is detected near the microphone sensor the peak is generated. The time lag between each peak is our time difference of arrival (TDOA) which is in milli seconds.

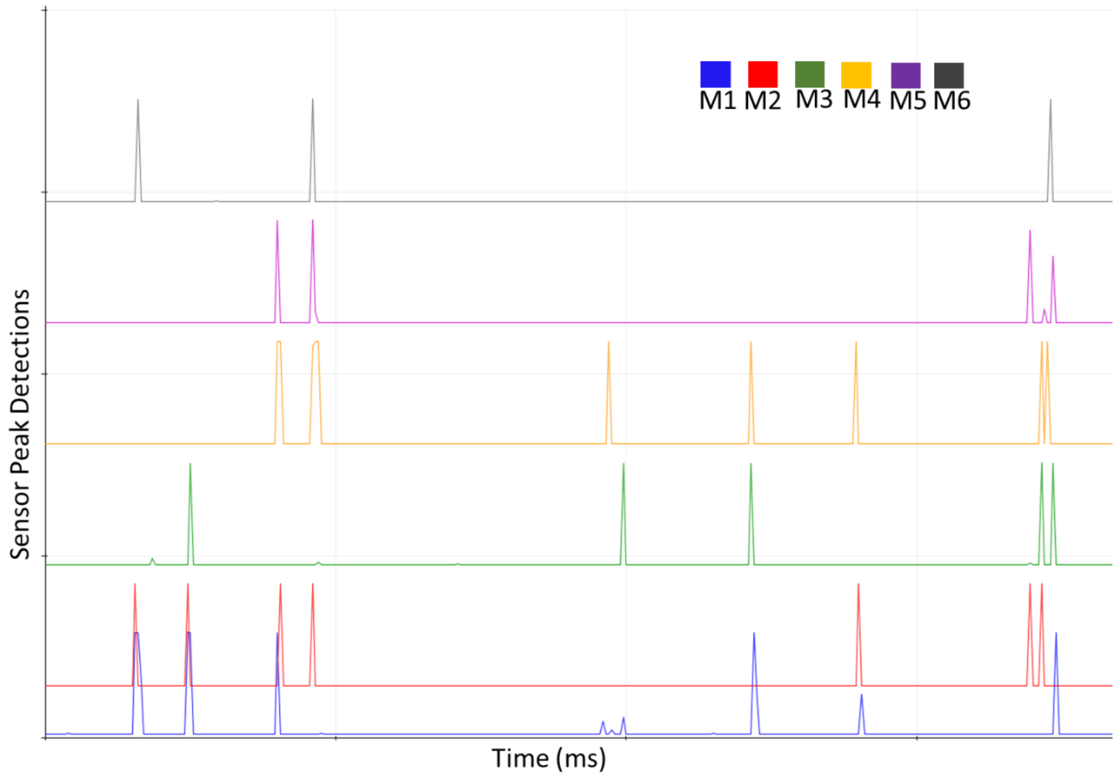

Figure 36: Peak detections of sensors data when tapped at center of table.

Since, the data from all six microphone sensors, using the MUSIC algorithm are known, the sharp peak values from all the six microphone sensors are found. Also, four best peak detected values are known by microphone sensors to perform the TDOA calculation. After the calculation is done with the data, the tap-sound source location is estimated using the triangulation method.

The best peak detected values are needed because the highest peak or a higher number of peaks means that the sound source is located near to it. Using MATLAB, the input of microphone sensor locations and true source locations are given. The system needs to calculate the estimated source locations using the TDOA based GCC-PHAT algorithm and Triangulation algorithm.

5.5 Results of software analysis and hardware analysis

5.5.1 Implementation 1 software results

Some of the ten random source locations are shown in Figure 35 and results are represented in Table 2.

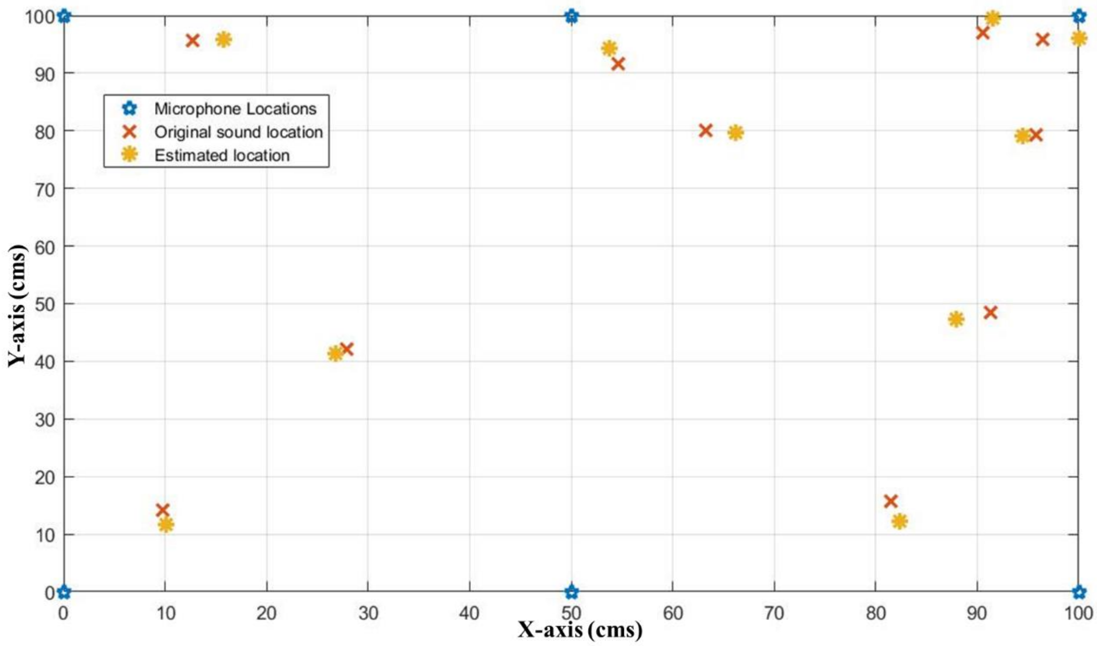

Figure 37: MATLAB results of sound source locations in a rectangular array.

The red cross mark refers to the original sound location, the yellow asterisk represents estimated source locations, and blue stars represents the know locations of microphone sound sensors. The x-axis and y-axis are about 100cms each. Table 2 represents the values of ten actual locations (x1, y1) and estimated locations (x2, y2). Some of the estimated source locations are accurately mapped with less error.

In table 2 it is clearly seen that most of the locations are very nearly mapped to original source locations. Few of the estimated locations are far from original locations.

Table 2: Actual and estimated location results for software implementation 1.

Location No.	Actual location		Estimated location	
	X1	Y1	X2	Y2
1	81.47	15.76	82.42	12.25
2	90.58	97.06	91.51	99.53
3	12.70	95.72	15.82	95.80
4	91.34	48.54	87.91	47.33
5	63.24	80.03	66.18	79.66
6	9.75	14.19	10.05	11.66
7	27.85	42.18	26.83	41.36
8	54.69	91.57	53.76	94.32
9	95.75	79.22	94.53	79.19
10	96.49	95.95	100.00	95.98

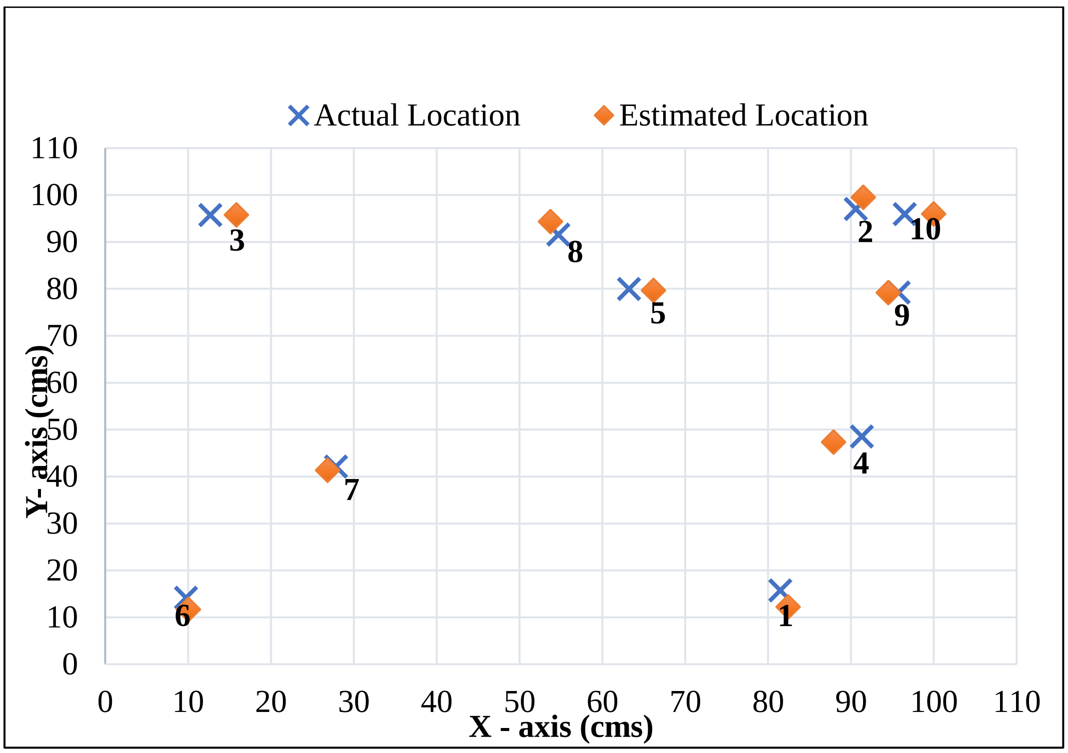

Figure 38: Representation of individual sound location for software Implementation 1

Finally, the localization error is calculated as the euclidean distance from the estimated sound source to the actual position for each localized coordinate. The resulting error is shown in Table 3 with an accuracy of 96-97% on an average.

Table 3: Error calculation for software implementation 1.

Distance Error (cm)	Localization Accuracy
3.63	96.4%
2.64	97.4%
3.12	96.9%
3.63	96.4%
2.97	97.0%
2.54	97.5%
1.31	98.7%
2.90	97.1%
1.23	98.8%
3.51	96.5%

5.5.2 Implementation 1 Hardware results

Figure 39 is the screenshot of the command window in MATLAB after sound triangulation and the sound localization is plotted on the graph in real-time.

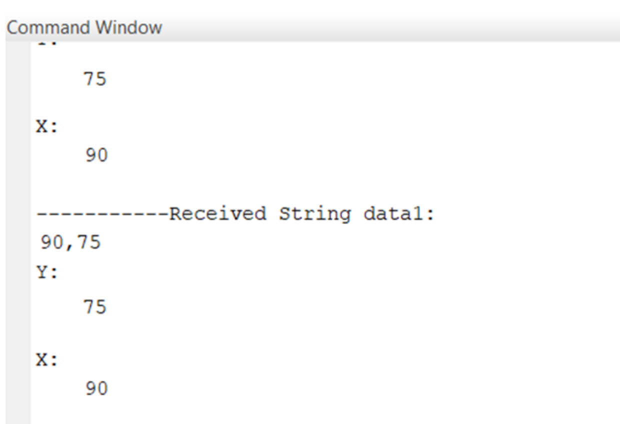

Figure 39: MATLAB command window output of implementation 1.

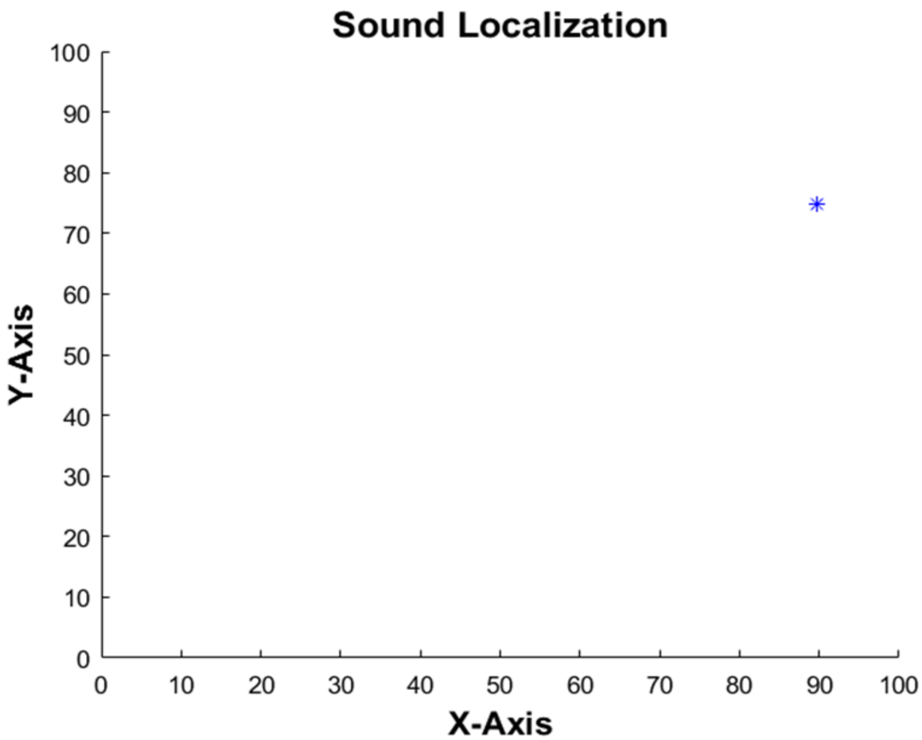

Figure 40: MATLAB output graph after sound source localization for implementation 1.

When a tap is done at a location (90,90), it was resulted at (90,75). As, mentioned in Chapter 3, the localization point is plotted in a 2D graph shown in Figure 40. Received calculated source location point from the command window using sound triangulation method. One of the source localization point is plotted on the graph as an example. The other localization points are shown in Figure 41 as to know the difference between the original and estimated source locations. Also, by seeing Figure 41 one can estimate how close the estimation calculations are done and how accurately a sound source localization system like this can be developed.

Table 4 shows the actual tapping locations performed and estimated locations that are estimated by the experimental results performed with the hardware implementation 1.

Table 4: Actual and estimated location results for software implementation 1.

Location No.	Actual location		Estimated location	
	X1	Y1	X2	Y2
1	81	16	95	10
2	91	97	99	84
3	13	96	27	86
4	91	49	97	36
5	63	80	75	69
6	10	14	27	8
7	28	42	40	34
8	55	92	63	88
9	96	79	100	63
10	96	96	100	77

Representation of individual source location points using the Table 4. To understand the difference that is present between the actual and estimated locations.

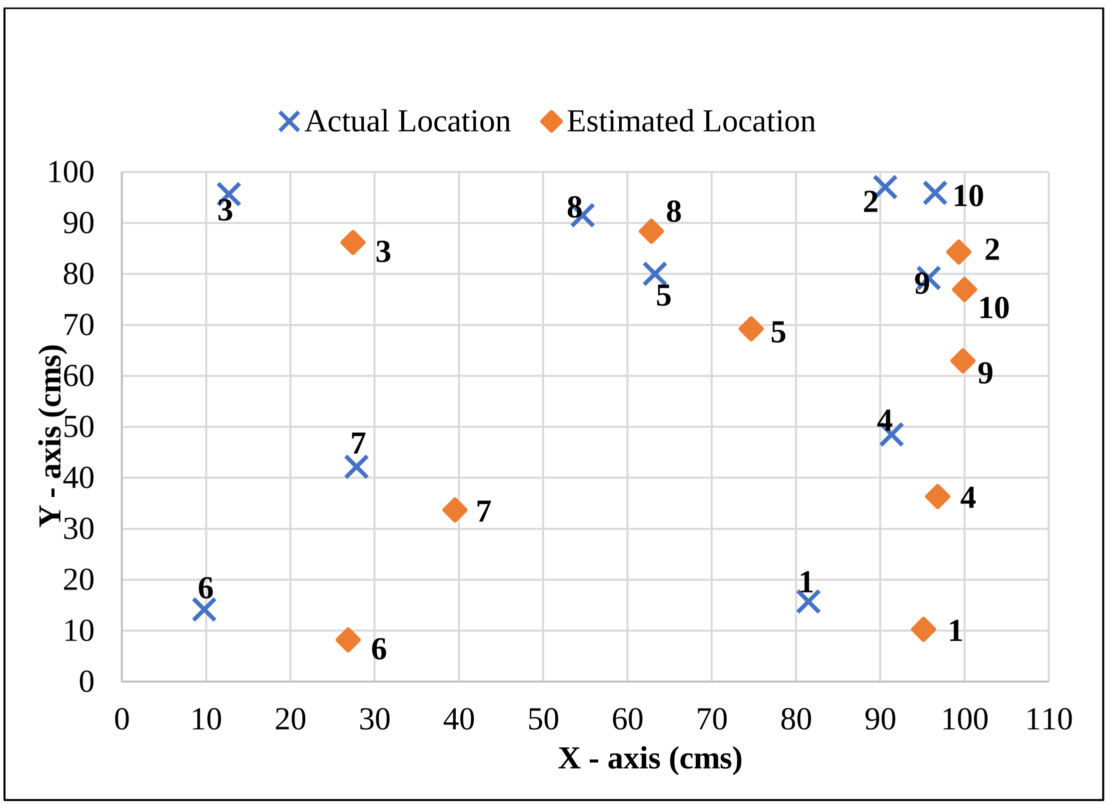

Figure 41: Actual and estimated locations with each localization point from Table 4.

The distance error using hardware is about 8% - 10% on an average, when compared to software implementation 1. Table 5 shows the distance error in centimeters.

Table 5: Distance error calculation for hardware implementation 1.

Distance Error (cm)	Accuracy
15	85.3%
15	84.6%
18	82.5%
13	86.7%
16	84.3%
18	81.9%
14	85.6%
9	91.3%
17	83.2%
19	80.7%

5.5.3 Implementation 2 software results

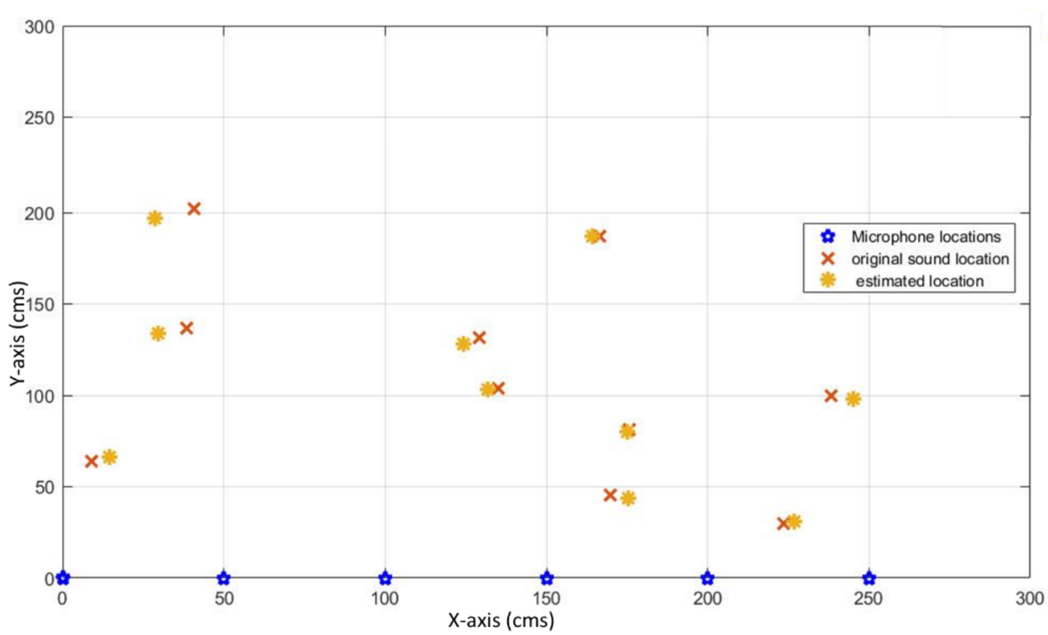

Figure 42: MATLAB results of sound source locations in a single linear array.

Figure 42 shows some of the ten random source locations and results are represented in Table 6 for a uniform linear array.

Table 6: Actual and estimated location results for software implementation 2.

Location No.	Actual location		Estimated location	
	X1	Y1	X2	Y2
1	40.89	202.30	28.54	196.90
2	166.50	187.15	164.20	187.02
3	223.60	30.05	226.92	31.01
4	129.14	131.26	124.44	127.76
5	175.68	81.46	174.90	79.89
6	38.40	136.61	29.72	133.96
7	238.36	99.72	245.13	98.25
8	135.22	103.77	131.89	103.37
9	169.93	45.18	175.33	43.79
10	9.14	63.85	14.66	66.06

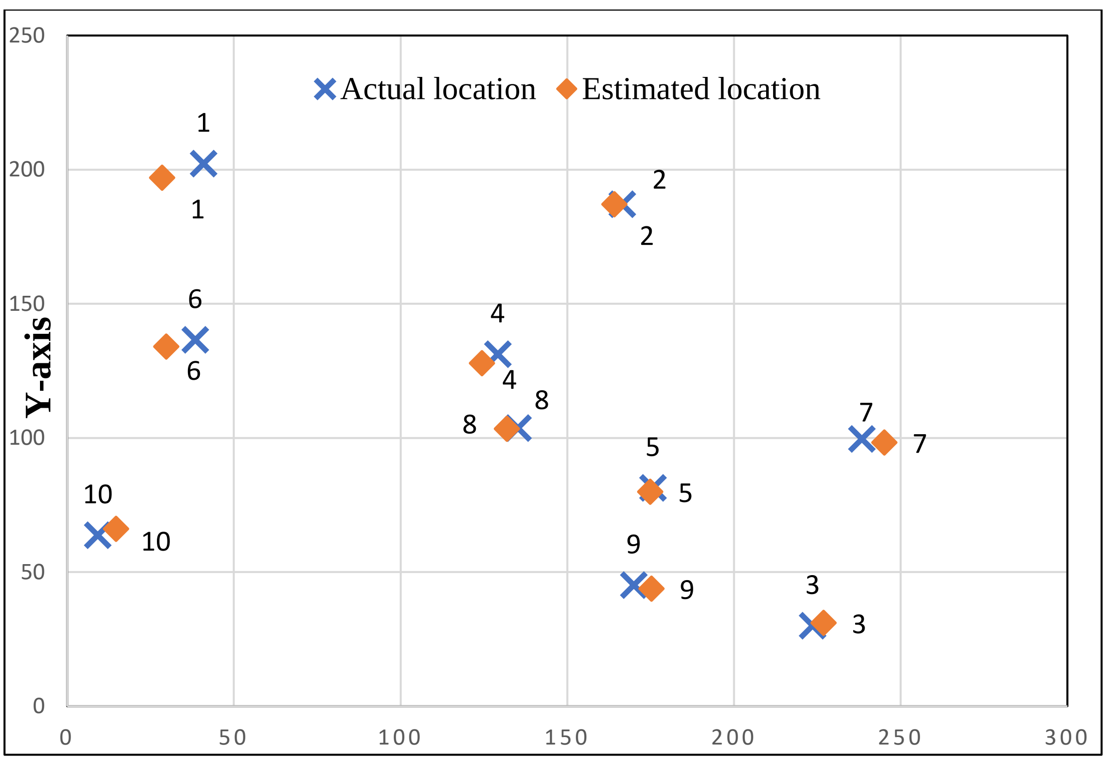

Figure 43: Representation of individual sound location for software Implementation 2.

Most of the locations are very nearly mapped to original source locations. It is seen clearly in Figure 43, that using linear array system design with same triangulation method, few of the estimated locations that are closer to the array have shown better accuracy in estimating the source locations.

Table 7: Distance error calculation for software implementation 2.

Distance Error (cm)	Accuracy
13.48	86.52%
2.30	97.70%
3.46	96.54%
5.86	94.14%
1.75	98.25%
9.07	90.93%
6.92	93.08%
3.35	96.65%
5.57	94.43%
5.95	94.05%

Table 7 explains the above-mentioned statement that when the source is nearer to the linear array the accuracy is more than 96%. That is the error distance less when compared to the rectangular array system.

5.5.4 Implementation 2 Hardware results

Figure 44 is the screenshot of the command window in MATLAB after sound triangulation and the sound localization is plotted on the graph in real-time.

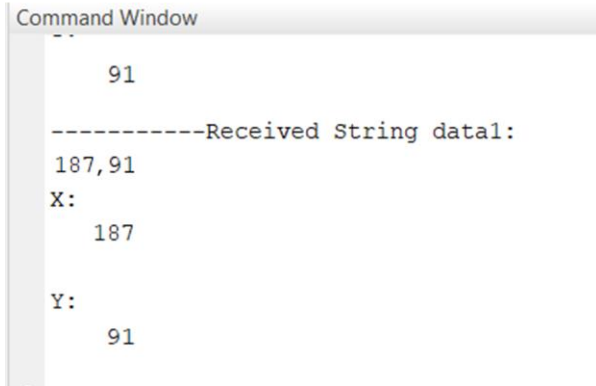

Figure 44: MATLAB command window output of implementation 2.

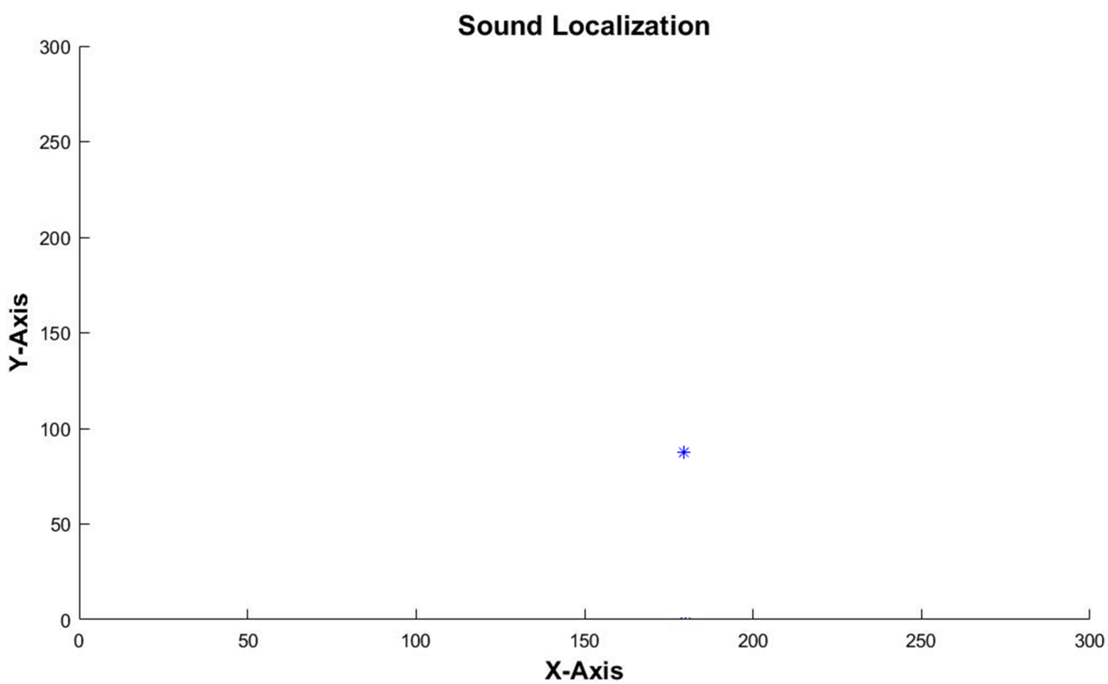

Figure 45: MATLAB output graph after sound source localization for implementation 2.

When a tap is done at a location (200,100), it was resulted at (187,91). Table 8 shows the actual tapping locations performed and estimated locations that are estimated by the experimental results performed with the hardware implementation 2.

The estimated locations for location 1 and 2 are (0,0) because the hardware was unable to detect the sound signal properly, because the tap location was far from the linear array system.

The microphone pairs were unable to detect the sound that is about more than 140 to 150cms. It is because the microphone sound sensors that are being used is sensitive towards other sounds, when tap is more than the mentioned distance in centimeters.

Table 8: Actual and estimated location results for hardware implementation 2.

Location No.	Actual location		Estimated location	
	X1	Y1	X2	Y2
1	41.00	202.00	0.00	0.00
2	167.00	187.00	0.00	0.00
3	224.00	30.00	235.00	30.00
4	129.00	131.00	127.00	103.00
5	176.00	82.00	180.00	66.00
6	38.00	137.00	22.00	112.00
7	238.00	99.00	220.00	98.00
8	135.00	104.00	123.00	84.00
9	170.00	45.00	184.00	44.00
10	9.00	64.00	14.00	50.00

Representation of individual source location points using the Table 8. To understand the difference between the actual and estimated locations. Locations points 1 and 2 were undetected while estimating with hardware.

The distance error using hardware is about 15% - 20% on an average, when compared to software implementation 2. Table 9 shows the distance error in centimeters.

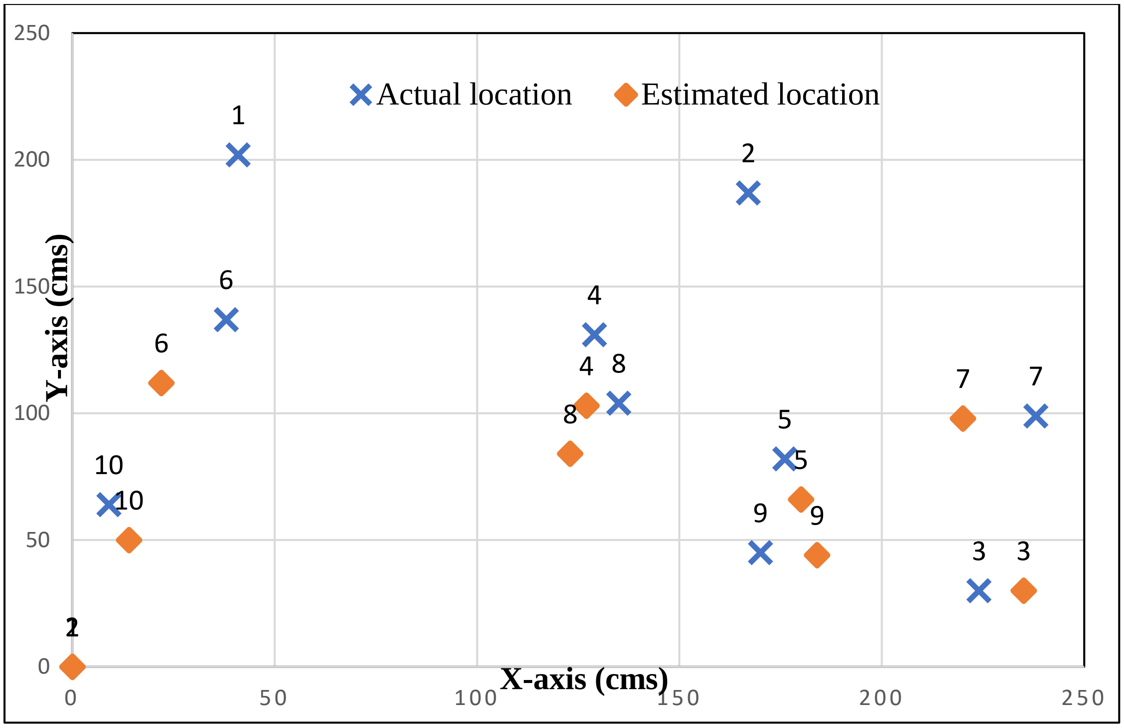

Figure 46: Representation of individual sound location for hardware Implementation-2.

Table 9: Distance error calculation for hardware implementation 2.

Distance Error (cm)	Accuracy
error	error
error	error
11.00	89.00%
28.07	71.93%
16.49	83.51%
29.68	70.32%
18.03	81.97%
23.32	76.68%
14.04	85.96%
14.87	85.13%

Using linear array system with hardware design, it is seen that few of the estimated locations that are closer to the array have shown better accuracy in estimating the source locations. Other sound sources that are more than 150cms in front of uniform linear array system have shown poor results.

The design constraints are very important while designing microphone array systems like these. Comparing all the results after analysis, it is clear and better to use a rectangular array system for real time applications. Single linear array system is better only for near localization estimations.

6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

Localization of sound sources offers us a great potential for location-aware applications development in modern world to help track events such as screams, gunshots, explosions, etc.

Different types of DOA estimations were studied and analyzed. A TDE based method was chosen because it is easy for implementation. GCC-PHAT with MUSIC was accurate in calculating the AOA.

Sound triangulation implementation for localization of sound source gave us better results in precision to DOA estimation. Also, there were several aspects that are needed to be considered while developing this sound source localization prototype. Considering it to be a manual configuration for a less cost. Also, learned that the system should have a greater number of sensors for more accurate SSL, which makes it more complex in calculating source location. But were able to make it possible with a smaller number of sensors, as this system had a decent sampling rate.

Also, learned how important it is to give sensor locations for calculations. Configuring all these parts was a major task, which needs to be done carefully. More the sensors, more the precision.

In our simulations, sound tapping based live audio is being used with two kinds of approaches. At first, the performance improvement with Arduino IDE is checked by incorporating Arduino with microphone sensors. Secondly, we have tested the whole system embedded with MATLAB to increase the performance and tested the tap-based sound source localization system. Significantly, this approach enables us to reduce the problem of dimensionality by producing low computational cost and produce acceptable results.

Although many methods have been proposed for locating sound sources during the last decade. A new source localization technique is obtained by the proposed system that works jointly with the Arduino IDE platform and MATLAB code. This technique combines the MUSIC beamformer and GCC-PHAT using a sound triangulation method with a 2D map-based coordinate system to locate the sound source.

Also, it is shown that Arduino and MATLAB can be used to link physical and computational worlds. A better system of sound analysis can be used because Arduino has limitations. There are various other boards that perform faster than the Arduino Uno. Unfortunately, they are not cost effective. The number of test samples found over the number of detections is our metric, which is defined to be our accuracy.

6.2 Future work

One of the goals would be to design a system to implement this on a larger scale. The idea is to achieve the detection of direction estimation and localization at a GPS level configuration for gunshot detection and localization as well as auditorium-based speaker localization.

This microphone array can be implemented at IOT level.

Using this kind of array system with pressure sensors, an impact localization system can be created.

Neural Networks-Based approach could be more effective for acoustic signal propagation when collecting large datasets and want to differentiate the TDOAs with increased computational power. The calculations of DOAs and TDOAs can be learned, when designing complex algorithms for SSL in the area of robotics.

Implementation of this can also be done is hospital rooms to monitor patience cough history, suffering with respiratory diseases.

Using a different type of microphone array that can detect heart rate irregularities.

REFERENCES

[1] P. Dostalek, V. Vasek, V. Kresalek, and M. Navratil, “Utilization of audio source localization in security systems,” in *43rd Annual 2009 International Carnahan Conference on Security Technology*, 2009, pp. 305–311.

[2] “PROFESSOR MAYER’S TOPOPHONE,” *michael leong’s poetry blog*, 14-Dec-2009.

[3] “World War I Acoustic Locator – History By Zim.” [Online]. Available: https://www.historybyzim.com/2014/07/world-war-i-acoustic-locator/. [Accessed: 25-Oct-2019].

[4] H.-G. Kang, M. Graczyk, and J. Skoglund, “On pre-filtering strategies for the GCC-PHAT algorithm,” p. 5.

[5] R. Chakraborty and C. Nadeu, “Sound-model-based acoustic source localization using distributed microphone arrays,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 619–623.

[6] S. A. Raj, G. Sreelatha, and M. H. Supriya, “Gesture recognition using field programmable gate arrays,” in *Devices, Circuits and Systems (ICDCS), 2012 International Conference on*, 2012, pp. 72–75.

[7] A. Lombard, W. Kellermann, and H. Buchner, “A real-time demonstrator for the 2D localization of two sound sources using blind adaptive MIMO system identification,” in *Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008*, 2008, pp. 41–44.

[8] H. Do, H. F. Silverman, and Y. Yu, "A Real-Time SRP-PHAT Source Location Implementation using Stochastic Region Contraction(SRC) on a Large-Aperture Microphone Array," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Honolulu, HI, USA, 2007, pp. I-121-I–124.

[9] V. Krishnaveni, T. Kesavamurthy, and B. Aparna, "Beamforming for direction-of-arrival (DOA) estimation-a survey," *Int. J. Comput. Appl.*, vol. 61, no. 11, 2013.

[10] X. Sun, W. Zhang, and D. Chen, "Movie Retrieval Based on Shazam Algorithm," in *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2018, pp. 1129–1133.

[11] C.-F. Chan and W. M. Eric, "An abnormal sound detection and classification system for surveillance applications," in *Signal Processing Conference, 2010 18th European*, 2010, pp. 1851–1855.

[12] S. Shon, D. K. Han, and H. Ko, "Abnormal acoustic event localization based on selective frequency bin in high noise environment for audio surveillance," in *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, 2013, pp. 87–92.

[13] T. Damarla, "Detection of gunshots using microphone array mounted on a moving platform," in *SENSORS, 2015 IEEE*, 2015, pp. 1–4.

[14] B. da Silva, L. Segers, A. Braeken, and A. Touhafi, "A runtime reconfigurable FPGA-based microphone array for sound source localization," in *Field Programmable Logic and Applications (FPL), 2016 26th International Conference on*, 2016, pp. 1–1.

[15] “Pin by Centro Auditivo Cuenca, audifonos Valencia Audifonos en Valencia on Centro Auditivo Cuenca | Distance.” [Online]. Available: https://www.pinterest.com.mx/pin/480055641521748915/?nic=1a. [Accessed: 25-Oct-2019].

[16] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, “Audio based event detection for multimedia surveillance,” in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 2006, vol. 5, pp. V–V.

[17] “(10) (PDF) Microphone Sensors for In-Vehicle Applications,” *ResearchGate*. [Online]. Available: https://www.researchgate.net/publication/317598111_Microphone_Sensors_for_In-Vehicle_Applications. [Accessed: 21-Oct-2019].

[18] E. Zwyssig, M. Lincoln, and S. Renals, “A digital microphone array for distant speech recognition,” in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 5106–5109.

[19] “microphone sound sensor module voice sensor high sensitivity sound detection module whistle module for arduino Sale - Banggood.com.” [Online]. Available: https://www.banggood.com/Microphone-Sound-Sensor-Module-Voice-Sensor-High-Sensitivity-Sound-Detection-Module-Whistle-Module-p-1235446.html?cur_warehouse=CN. [Accessed: 25-Oct-2019].

[20] “Arduino-Sound-Detection-Sensor-Pin-Outs.png (681×248).” [Online]. Available: https://robu.in/wp-content/uploads/2017/09/Arduino-Sound-Detection-Sensor-Pin-Outs.png. [Accessed: 20-Oct-2019].

[21] "main-qimg-c7ea8d6104a4685253426ff5b004eb9b-c (1989×978)." [Online]. Available: https://qph.fs.quoracdn.net/main-qimg-c7ea8d6104a4685253426ff5b004eb9b-c. [Accessed: 20-Oct-2019].

[22] "Arduino UNO Rev3 | 3DSOMA," *3DSOMA Selected Creative Electronic Technology and 3D Printing Store.* [Online]. Available: https://www.3dsoma.com/fr/plaques/194-arduino-uno-rev3-8058333490090.html. [Accessed: 20-Oct-2019].

[23] A. Aqeel, "Introduction to Arduino Uno," *The Engineering Projects*, 21-Jun-2018. .

[24] I. Lita and D. A. Visan, "Spherical coordinates control system for accurate orientation of directional antennas," in *2012 IEEE 18th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 2012, pp. 233–136.

[25] "timedelaybeamform2.png (1631×956)." [Online]. Available: https://www.mathworks.com/help/phased/ug/timedelaybeamform2.png. [Accessed: 20-Oct-2019].

[26] T. E. Tuncer and B. Friedlander, *Classical and Modern Direction-of-Arrival Estimation*. Academic Press, 2009.

[27] Z. jaafer, S. Goli, and A. S. Elameer, "Performance Analysis of Beam scan, MIN-NORM, Music and Mvdr DOA Estimation Algorithms," in *2018 International Conference on Engineering Technology and their Applications (IICETA)*, Al-Najaf, 2018, pp. 72–76.

[28] J. Choi and H.-T. Choi, "Preliminary Results on Three Dimensional Localization of Underwater Acoustic Sources," p. 2.

[29] B. Onl, S. Iml, and S. Kim, "Performance comparison of FFT-based and GCC-PHAT time delay estimation schemes for target azimuth angle estimation in a passive SONAR array," p. 4.

[30] L. M. Kaplan, Qiang Le, and N. Molnar, "Maximum likelihood methods for bearings-only target localization," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, Salt Lake City, UT, USA, 2001, vol. 5, pp. 3001–3004.

[31] "LOCALIZING MULTIPLE AUDIO SOURCES FROM DOA ESTIMATES.pdf." .

[32] A. Griffin, A. Alexandridis, D. Pavlidi, and A. Mouchtaris, "Real-time localization of multiple audio sources in a wireless acoustic sensor network," p. 5.

[33] A. Griffin, A. Alexandridis, D. Pavlidi, Y. Mastorakis, and A. Mouchtaris, "Localizing multiple audio sources in a wireless acoustic sensor network," *Signal Process.*, vol. 107, pp. 54–67, Feb. 2015.

[34] D. Pavlidi, A. Griffin, M. Puigt, and A. Mouchtaris, "Real-Time Multiple Sound Source Localization and Counting Using a Circular Microphone Array," *IEEE Trans. Audio Speech Lang. Process.*, vol. 21, no. 10, pp. 2193–2206, Oct. 2013.

[35] Han Yi and Wu Chu-na, "A new moving sound source localization method based on the time difference of arrival," in *2010 International Conference on Image Analysis and Signal Processing*, Zhejiang, China, 2010, pp. 118–122.

[36] B. Sahbani, P. Ramadhan, and M. K. Napitupulu, “VLSI design for 3D-Ultrasonic source localization using General Cross Correlation and Triangulation algorithm,” in *2016 International Symposium on Electronics and Smart Devices (ISESD)*, Bandung, Indonesia, 2016, pp. 193–198.

[37] S. Lee, Y. Park, and Y. Park, “Cleansed PHAT GCC based sound source localization,” p. 4.

[38] J. A. Belloch, “Real-time Sound Source Localization on an Embedded GPU Using a Spherical Microphone Array,” p. 10.

[39] F. Miao, D. Yang, R. Wang, J. Wen, Z. Wang, and X. Lian, “A moving sound source localization method based on TDOA,” 2014.

[40] P. Nimmy, K. R. Nair, R. Murali, K. R. Rajesh, M. Nimmy, and S. Vishnu, “Analysis of Acoustic Signatures of small firearms for Gun Shot Localization,” p. 4.

[41] J. Velasco *et al.*, “Novel GCC-PHAT model in diffuse sound field for microphone array pairwise distance based calibration,” p. 5.

[42] Y. Huang and J. Benesty, Eds., *Audio Signal Processing for Next-Generation Multimedia Communication Systems*. Boston: Kluwer Academic Publishers, 2004.