ACOUSTIC SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM)

by

Akul Madan

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Electrical and Computer Engineering

Department of Electrical and Computer Engineering

Indianapolis, Indiana

December 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Lingxi Li, Chair

Department of Electrical and Computer Engineering

Dr. Yaobin Chen

Department of Electrical and Computer Engineering

Dr. Brian S. King

Department of Electrical and Computer Engineering

Approved by:

Dr. Brian S. King

Head of the Graduate Program

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my committee chair Dr. Lingxi Li and co-chair Dr. Yaobin Chen for guiding me, providing their insightful advice and input during my research. I would also like to extend my gratitude to Dr. Brian S. King, a member of my thesis committee for his support and guidance. Lastly, I would like to thank Sherrie Tucker and the Department of Electrical and Computer Engineering at Indiana University-Purdue University Indianapolis (IUPUI) for their continued support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

dB	decibels
Hz	Hertz
kHz	Kilo-Hertz
cm	Centimeters
m	Meters

ABBREVIATIONS

SLAM	Simultaneous Localization And Mapping
MATLAB	Matrix Laboratory
LiDAR	Light Detection And Ranging
RADAR	Radio Detection And Ranging
ADAS	Advanced Driver Assistance System
ABS	Anti-Lock Braking System
MiDAR	Multispectral Imaging, Detection And Active Reflectance
DOA	Direction-Of-Arrival
SPL	Sound Pressure Level
DRR	Direct-To-Reverberant-Ratio
HMM	Hidden Markov Models
ANN	Artificial Neural Network
SUV	Sports Utility Vehicle
GPU	Graphics Processing Unit
ToT	Time Of Travel
CMOS	Complementary Metal Oxide Semiconductor
DNN	Deep Neural Network
ASR	Automatic Speech Recognition
ANN	Artificial Neural Networks
AED	Audio Event Detection
RGB	Red Green Blue
IR	Infra Red
AC	Alternating Current
DC	Direct Current
SDK	Software Development Kit
FFT	Fast-Fourier Transform
WAV	Waveform Audio File Format
A/D	Analog To Digital

DSP	Digital Signal Processing
SNR	Signal To Noise Ratio
RAM	Random Access Memory
CPU	Central Processing Unit
IDE	Integrated Development Environment

ABSTRACT

The current technologies employed for autonomous driving provide tremendous performance and results, but the technology itself is far from mature and relatively expensive. Some of the most commonly used components for autonomous driving include LiDAR, cameras, radar, and ultrasonic sensors. Sensors like such are usually high-priced and often require a tremendous amount of computational power in order to process the gathered data. Many car manufacturers consider cameras to be a low-cost alternative to some other costly sensors, but camera based sensors alone are prone to fatal perception errors. In many cases, adverse weather and night-time conditions hinder the performance of some vision based sensors.

In order for a sensor to be a reliable source of data, the difference between actual data values and measured or perceived values should be as low as possible. Lowering the number of sensors used provides more economic freedom to invest in the reliability of the components used.

This thesis provides an alternative approach to the current autonomous driving methodologies by utilizing acoustic signatures of moving objects. This approach makes use of a microphone array to collect and process acoustic signatures captured for simultaneous localization and mapping (SLAM). Rather than using numerous sensors to gather information about the surroundings that are beyond the reach of the user, this method investigates the benefits of considering the sound waves of different objects around the host vehicle for SLAM. The components used in this model are cost-efficient and generate data that is easy to process without requiring high processing power.

The results prove that there are benefits in pursuing this approach in terms of cost efficiency and low computational power. The functionality of the model is demonstrated using MATLAB for data collection and testing.

1. INTRODUCTION

1.1 Motivation

Night-time conditions, dust, fog, snow, rain, or any other adverse conditions alter the functionality of various sensors incorporated in autonomous driving. Vision based sensors like cameras often struggle with recognizing objects within the host vehicle's surroundings in darker conditions. Some car manufacturers use various auto exposure or contrast adjusting algorithms, but that functionality is incorporated at the price of higher processing power. Sudden changes in lighting conditions like driving through an underpass or bright reflections render camera based sensors temporarily blind. Similarly, sensors that depend on the concept of reflection of light or sound waves like LiDAR, radar, or ultrasonic are rendered useless in rainy, snowy, or foggy conditions due to the presence of numerous suspended particles around the sensors.

Current technologies and algorithms require computers with high processing power. Capturing, processing, encoding, filtering, re-encoding, and syncing data with other sensors. All these tasks rely on computational precision and any discrepancy in processing the crucial data could lead to fatal casualties [1]. Radar sensors are available as long-range and short-range, with short-range for tasks like detecting nearby vehicles and long-range for further objects. While RADAR, although cost-efficient, but is prone to giving false results from reflections off of smaller objects like an empty soda can [2]. A majority of the sensors used for autonomous driving are mounted on the exterior of the vehicle, making them prone to wear and tear over time. Sensors like LiDAR contain moving parts, which makes it even more likely to see regular wear and tear with prolonged use.

Many car manufacturers either use a single sensor or a combination, in order to offer some form of Advanced Driver Assistance System (ADAS) features like, lane departure warning using image recognition, proximity warning using radar, anti-lock braking system (ABS), etc [3]. In general, the combination of different sensors offers higher levels of automation features in a vehicle. One of the most important factors that influence the adoption rate of fully or partially automated vehicles is price. The price is influenced by the number and type of sensors installed in a vehicle [4]. Typically the cost of some of the industrial grade sensors

like LiDAR, radar or camera that are commonly used for autonomous vehicles, is extremely high. There are some other alternatives that offer a combination of sensors like radar and LiDAR as one senor, commonly known as MiDAR, but the cost is still extremely high [5]. As the likelihood of sensors getting damaged by normal wear remains extremely high, the need for cost-efficient replacement parts is highly essential. An alternative to this problem would be to use sensors that are easily accessible and are cost-efficient. In summary, employing sensors that are widely available, cost-efficient, and do not produce complex data, eliminate the need for high processing power. In addition, a sensor that works during night-time, dusty, foggy or snowy conditions would be an optimum solution to the problems faced by some of of the current sensors.

1.2 Methodology

In this thesis, an alternative approach that uses a microphone array for autonomous driving is proposed. The main idea is to provide a cost-efficient and computationally minimal method, with results comparable to other technologies being used. The data collected by the model and its analysis is conducted using MATLAB. A two algorithm approach is considered for sound source localization and distance approximation [6]. The localization of the sound source is implemented using the concept of direction-of-arrival (DOA) of sound. The implementation method used for this purpose captures audio from the microphones and estimates the direction of the sound source by calculating the time of delay between signals. The governing concept behind this technique is that sound sources at larger distances lose their intensity due to their distance, and produce sounds with lower sound pressure level (SPL) [7]. This is true for both light and sound signals, and can be easily modeled by Inverse Square Law [7]. If a known sound source is at an unknown distance, the amplitude of sound reaching a listener is lowered as the sound pressure level (SPL) spreads over the distance [6].

$$I \propto 1/d^2 \tag{1.1}$$

I: Intensity of the sound in decibels (dB)

d: Distance of the sound source from the listener

Another segment of the MATLAB algorithm works in parallel with the direction-of-arrival algorithm. This segment is responsible for capturing the audio samples using the connected hardware. The ideology behind this segment relies on the understanding that objects at different distances produce different sound signatures. The auditory characters can vary in terms of pitch, amplitude, reverberation, and timbre [8]. Although, there is a method that makes use of the concept of direct-to-reverberant-ratio (DRR), those dereverberation algorithms are computationally intense and take time to process [9]. If an algorithm cannot process and provide results in near real-time scenarios, then the application for such algorithms is ill-suited for autonomous driving, where split-second decision-making is key. Therefore, the algorithm captures audio samples and processes the files to estimate their intensities in decibels (dB). This helps the algorithm provide a good estimate of the distance at which the sound source is located. The main goal of the algorithm is to provide distance approximations of moving vehicles, therefore for this purpose, changes in the intensity levels are calculated. For example, as an ambulance approaches closer to a listener, its sound intensity increases in terms of amplitude, as a majority of the sound pressure levels are directed to the listener without attenuation [8].

The structure of the remainder of this thesis is as follows: The next section in this chapter goes over some of the major contributions that led to the completion of this thesis. Chapter 2 performs a literature review of related research, current problems faced by autonomous driving, improving safety using sensors, shortfalls of sensors, and proposing a newer model and its advantages. Chapter 3 presents a design for the microphone array and its potential benefits over other methods. It also goes over the design that was considered for the purpose of this thesis, in order to provide proof of concept and verify the functionality of the model. In Chapter 4, a detailed overview is provided for the experimental setup in regards to hardware, software and integration of both, along with its advantages. Chapter 5 provides information on data collection, especially the different test scenarios considered, capturing, processing, and filtering of data. Then, the results drawn from the experiment are presented in Chapter 6, along with an accuracy comparison with other technologies and some benefits of using

a similar approach. Further, Chapter 6 also explores how this new method can be used to solve problems that cannot be solved by the existing technologies.

1.3 Major Contributions

A novel idea of Acoustic SLAM is presented in this thesis. The system's design is relatively simple and easy to operate. Setup of the system also does not require any complex procedures. Using a microphone array, the system captures acoustic samples and processes them in order to extract necessary information about the surroundings. The information is relayed using graphical interfaces, in terms of the general direction of the sound source, and the approximate distance from it. Graphical elements are updated frequently, in order to display information in near real-time. This provides information to the user that is easy to comprehend.

The system is also flexible enough to run on various hardware platforms. This aspect also allows for the possibility of sensor fusion or using AI for scene recognition. As the data captured by the sensor array is stored in a multi-channel audio file, thus making it easy to process. Compared to sensor types like LiDAR, the system has relatively low operational costs, as the data isn't complex. As the system detects acoustic activity, therefore it can be a great tool for blind-spot detection. Vehicles with a limited field of view like trucks or SUVs can benefit from it, and use it to detect pedestrians or other small vehicles.

One of the major benefits of this system is cost efficiency. Compared to some other sensors, the cost of the entire system alone is a fraction of their cost. In terms of operational costs as well, the system does not require high-end hardware to complement the hardware components. The simplicity, flexibility, and cost-efficiency of the system make it a good alternative to some other systems.

2. LITERATURE REVIEW

There is a limited amount of research that has been done in the area of localizing sound sources using acoustics. A sound sample for any object captured within an environment contains the source's own sound along with some unwanted environmental noise, which varies depending on the surrounding. In order to extract the source information from the samples, some pre-processing is required. The sampled signal is de-noised in a pre-processing unit, then it is passed on for further processing [10]. The signal's original amplitude is left untouched after noise removal in order to retain the original sound intensity levels of the source.

It is necessary to capture the entire field in order to detect objects from different directions. For example, a LiDAR sensor spins rapidly while firing about 1,000,000 pulses per second, in order to get an accurate map of the host vehicle's surroundings [11]. Some researchers have pursued similar implementations that can be applied to a binaural microphone pair, that constantly moves, thus allowing it to capture a wider acoustic field [12]. Although this method would cut down on the need to consider a complex microphone array, which would definitely lower the overall cost associated with the implementation of this method. A similar but slightly different approach uses a single moving microphone, and utilizes a probabilistic method for event prediction, commonly known as Bayesian inference through Kalman filters [13]. Another group of researchers makes use of algorithms designed for speech recognition using Hidden Markov Models (HMM) and Artificial Neural Networks (ANN), for audio-visual signal processing and event detection [14].

The methods mentioned above are well suited for very specific scenarios and would present serious disadvantages for acoustic SLAM. If some of these methods were implemented for acoustic SLAM, then the complexity of the system would be high. With moving microphone/microphones, there is a need to incorporate noise isolation algorithms, which would help reduce the amount of interference due to the wind that would get captured because of the microphone's motion. In order to ensure that the algorithm operates in real-time, it would be advantageous to use data this isn't complex and large in size. This is the rea-

son why some of the complex probabilistic approaches mentioned in other research papers wouldn't be ideal.

2.1 Safety Concerns In Autonomous Vehicles

As new trends in autonomous driving continue to evolve each day, the variation in the number of sensor manufacturers grows too. Each car manufacturer implements different algorithms and hardware for solving the same problem. The biggest problem associated with autonomous vehicles is that they are sold as a product to a consumer. While understanding the basic fundamentals of operation of most consumer products is relatively easy, contrary to the technologies incorporated for autonomous driving [15]. Therefore, it is crucial to keep the user interaction as straightforward as possible. A majority of problems associated with autonomous vehicles can be broken down into three basic levels, which can be streamlined in order to optimize error resolution [1].

2.1.1 Sensory Level

Sensors play a major role in autonomous vehicles and are manufactured by numerous different companies with varying fabrication methodologies. Different manufacturers can use varying manufacturing methods, thus producing sensors with nonuniform accuracy under various conditions. The need for standardized quality control for all sensor manufacturing would eliminate the possibility of errors caused by differences in production. Perception errors are also a major concern with sensors, especially over prolonged use, as the chances of degradation are high [1]. Sensors like LiDAR can cause perception error or discrepancies in reading due to the presence of suspended particles in the air. Similarly, vision-based sensors perform poorly in low light or sudden changes in lighting conditions. While RADAR and ultrasonic sensors that rely on the concept of reflection can easily be altered by soft surfaces that dampen the waves emitted by them.

Solutions to these problems would be to have alternative sensors that would work in conditions in which the others cannot. The cost is also a major factor that contributes to the overall safety. A consumer may choose not to replace a faulty sensor due to the excessive

replacement cost and limited availability. Therefore, choosing components that are widely available, and come at affordable prices, would limit the number of errors at the sensory level.

2.1.2 Algorithmic Level

In similarity with sensors, various different algorithms offer different functionalities, data processing speeds and are available through numerous sources. Algorithms are generally to synchronize data from different sensors and process the safest course of action of an event in near real-time. One major downside to algorithms is that they assume all objects in an environment follow the rules, which makes it hard for them to compute unpredictable circumstances [16]. More advanced the algorithm, the more computational power it requires for its operations. Many modern algorithms use machine learning and artificial intelligence to learn from a user's driving habits and react if an action seems out of the usual behavior. Some basic functionality offered by algorithms includes object detection, traffic light detection and trajectory planning. Many algorithms still find it a challenge to navigate through construction areas or react in areas where an accident has occurred [17]. Algorithms play a key role in decoding the complex data collected by different sensors, and the level of automation depends largely on it [18].

Therefore, in order for an autonomous vehicle to be efficient in terms of data handling and decoding, there is a need for an algorithm that does not prove to be computationally expensive. It should also be developed in a way that makes it future-proof, and can easily be updated or modified if need be. Complex algorithms are harder to debug and difficult to understand. An additional factor that would prove to be beneficial, is for the code to be platform friendly. This makes the system easily configurable in the future if additional features were to be added, or the current system needed to be modified.

2.1.3 Hardware Variation

Hardware used in autonomous consists of everything from sensors, computers, power supply, and equipment used for mounting the sensors. But this section mainly focuses

on computational hardware. Companies like NVIDIA offer GPUs that are built with the capability of handling intense sensor data, while some car manufacturers choose to design and use their own custom-built chips. The accuracy of data gathered by the sensors comes down to the efficiency of the hardware used for processing it. A set of hardware also needs to be tested in different scenarios in order to ensure that the system would behave optimally, and make safe decisions in tough scenarios [19]. Vehicles with more advanced and costly sets of hardware may have an advantage over other vehicles, therefore a need for a performance standard would help level the field in terms of safety [1].

The safety of an autonomous vehicle depends on the performance of the hardware, which also directly relates to the detection and measurement accuracy of the system. The hardware also needs to be cost-efficient, in order for it to be easily replaceable by the user or upgraded to include any additional safety measures in the future.

2.2 How To Improve Safety with LiDAR, Radar and Camera

A wide array of sensors are incorporated in autonomous vehicles, with the sole purpose of making the vehicle safer and prepared for any unpredictable circumstances. Although, the current array of sensors do a great job, there is still a huge room for improvement in terms of safety [1]. Each different type of sensor has some vulnerabilities, it could be due to changing lighting conditions or severe weather. As most sensors rely on analog feedback on the sensor end, that makes them prone to interference from the surroundings. These could be in the form of severe weather conditions like heavy winds, rain or snow, that hinder the propagation of sound waves, laser beams, or even the view of a vision-based sensor.

2.2.1 LiDAR

The technology used for LiDAR faces a challenge between higher range or operational power [20]. Although, LiDAR can achieve higher accuracy with an increase in operating power, but that option costs more. Another drawback to a higher operating power is the safety levels of the laser beams fired by the sensor. A stronger beam has the potential to harm people's eyes, as well as sensors of cameras in the vicinity [21]. An alternative to this

problem is to use a long-range ultrasonic sensor in conjunction with LiDAR. Another major disadvantage to LiDAR is that the sensor itself is relatively bulky and is usually mounted on top of the roof of the vehicle, where the beams are free from any obstructions. Almost all cars are designed with an aerodynamic structure in mind, in order to keep the drag coefficient as low as possible. Mounting a LiDAR sensor on top of a vehicle increases the amount of drag force generated by a vehicle, thus lowering the fuel efficiency. A possible alternative would be to use an array of smaller LiDAR sensors along the sides of the roof of the car, which would also prove to be power efficient and would not increase the amount of drag force by a significant amount.

2.2.2 RADAR

Sensors like ultrasonic and RADAR are typically used for measuring a vehicle's proximity to nearby objects using high-frequency sound waves. A RADAR sensor by itself is vulnerable to unwanted radio interference at the receiver. A RADAR also cannot distinguish between different sizes of objects and can provide inaccurate estimates by picking up reflections from small objects on the road [2]. They are also susceptible to weather conditions like rain or snow. Raindrops or snow reflect waves before they reach their target and provide inaccurate data. An optimal idea is to rely on multiple data sources for higher measurement accuracy, rather than just one. Although the sensor itself is cost-efficient and easy to replace, but offers limited functionality in terms of tracking the movement of an object. For example, a fixed RADAR can detect the general direction of motion of an object in terms of left or right, but cannot provide a higher degree of information [2]. Sensors like RADAR that rely on waves as a means of measurement are prone to damping by different surfaces. Different surfaces or materials have varying compositions, some objects are composed of harder substances while others make use of softer materials. In general, softer materials tend to absorb a majority of the sound waves and create a damping effect. Detection of such objects can pose a challenge to RADAR since a majority of the waves are absorbed and are not reflected back. To counter this kind of problem, using vision-based sensors to detect the presence of nearby objects can improve tracking results [22]. Having multiple smaller short-range RADAR

sensors for near-field detection, and a combination of two or more long-range ultrasonic sensors for detecting objects at greater proximity would be a cost-efficient solution. In most scenarios, a combination of RADAR and vision-based sensors like cameras would provide sufficient data for autonomous driving, but wouldn't be as precise in their measurements as some other alternatives.

2.2.3 Camera

Cameras or similar vision-based sensors offer cost-efficient solutions for object detection. The sensor's performance is usually complimented by an object or scene recognition algorithm. The data from a camera is processed in terms of distance between pixels, in order to classify different types of objects like people walking to biking [11]. On an algorithmic level, errors in processing or calculation of parameters could be a problem, and testing different scenarios or conditions can help debug such issues. Other possible errors that arise are usually caused by perception errors [1]. The lens of a camera is susceptible to obstruction of view in the form of debris like dust particles, drops of water, or snow. This could hinder the detection performance and can lead to fatal errors. Installing transparent shields or coverings in front of the camera module would prevent small particles from making direct contact with the lens. Poor lighting conditions also lower the accuracy of a sensor like a camera, since the detection of objects in darker conditions makes it harder for an algorithm to differentiate between a moving object and its surroundings [11]. While some algorithms incorporate additional elements like automatic exposure setting based on the lighting, but a more robust solution would be to use infra-red cameras.

Most vision-based sensors share similarities with human eyes. In some scenarios, when there is a sudden shift from a low light setting to a really bright environment, the human eye is temporarily blinded while the iris tries to control the amount of light entering through the pupil. Similarly, vision-based sensors face similar problems when there is a sudden switch from dark to a bright condition [11]. The temporary period of the sensor being blinded is when it tries to adjust parameters like focus or exposure levels. But this temporary period could result in fatal casualties. Therefore adding certain redundancies to a vehicle, like

different types of sensors that can achieve the same task when other data sources fail, would prove to be a safer approach in the long run [23]. The quality and price of a vision-based sensors is also a determining factor when it comes to accuracy. Choosing sensors based on the purpose, functionality and quality should be the major priorities followed by cost-efficiency.

2.3 Drawbacks Of Using LiDAR, RADAR and Camera

LiDAR, RADAR, and cameras are some of the most commonly used sensor types for autonomy applications. Each of these sensors has different principles of operation, which make them suitable for certain situations and scenarios, that require such functionality. When used in conjunction with each other, the feedback from each sensor complements the other, thus increasing the detection and response efficiency of the system. As each sensor has its drawbacks in certain conditions when used by itself, but even when sensors are used in a combination, some flaws could still exist.

In LiDAR sensors, pulses of laser beams are fired at a really high rate, and once the reflections are captured, the data is processed to estimate the topology of the surroundings. The distance estimates are calculated using the concept of time-of-travel (ToT). The distance to an object is estimated once the reflected beams are captured by the receiver. For the measurement, the time taken by the laser beam to get reflected off of the object's surface is calculated [11]. Since the laser beams propagate through the air and reflect off of surfaces, which creates some operational challenges in certain conditions. One such scenario is when the sun is at high angles, especially the time around noon. During this time the intensity of sunlight is extremely high, and easily overpowers the laser beams emitted by LiDAR [4]. Another similar challenge is presented by highly reflective surfaces such as tinted windows, metallic surfaces, or shiny surfaces. These surfaces reflect a vast majority of light incident on their surfaces, including laser pulses from LiDAR, which makes it a difficult task to capture the reflected laser beam [24]. Apart from these scenarios, LiDAR is also not cost-efficient both in terms of component costs and operation [24]. A LiDAR scanner has an extremely high sample rate, which is essential for highly accurate measurements. But due to the high sample rate and the vast amount of data LiDAR produces every second, it creates the

demand for systems with high processing powers, thus proving to be an expensive option for daily operations [4]. Installation of a LiDAR sensor also requires a lot of space, especially spots without any obstructions, so the laser pulses can propagate freely.

RADAR sensors use the operating principle of reflection of echoes. A transmitter emits electromagnetic waves and these waves get reflected back from the surfaces of nearby objects [11]. Once the reflected waves are detected by the receiver, the distance to the object from the sensor is computed using the speed, distance, and time relation.

$$D = c * T/2 \tag{2.1}$$

D: Distance to an object

c: 3x10^8 m/s, speed of propagation of electromagnetic waves

T: Time delay, from emission to reflection to being received

The electromagnetic waves are prone to attenuation through other radio waves in the surroundings, which affects the measurement accuracy. Extreme weather conditions also slightly affect the performance of RADAR, especially if the receiver or the transmitter gets blocked by some foreign particles [2]. Certain environmental factors such as rain and snow generate echo signals that can mask the desired target echoes [2]. A RADAR also lacks the ability to distinguish small objects in the path of the radio waves from the desired targets. Nearby RADAR or other transmission sources can also cause interference to reflected echoes, thus hindering the accuracy of the measurements. But compared to LiDAR, a RADAR is a cost-efficient component that does not require high processing power.

vision-based sensors like cameras share similar anatomy to an eye. Therefore, its operating principle depends on the light entering through the lens, which helps it generate a digital image of the surroundings based on the varying intensities of lights being reflected from different objects [11]. Similar to an eye, certain scenarios create challenges for cameras. For example, in low lighting conditions, the sensor has a limited amount of light available for the detection of different objects. Another scenario is when there is a sudden shift from low light to a dark environment and vice versa. A sudden shift in lighting conditions does not provide enough time for the sensor in a camera to adjust the aperture size [11]. Therefore,

this sudden shift drives the camera into a temporary state of blindness. Highly reflective surfaces or direct exposure to the sun can also temporarily disable a camera's ability to detect the surroundings. Similar to LiDAR and RADAR, extreme weather conditions like snow or fog can cause detection errors, especially if some foreign particles obstruct the view of the lens. In extremely cold weather conditions, there is also a huge possibility of frost covering the lens due to condensation.

Most sensors used for autonomous driving are usually installed at specific locations on a vehicle. For example, a LiDAR sensor is usually mounted on the top of a vehicle, a RADAR is typically installed in the front and the back of a vehicle. Similarly, cameras are installed in places which allow them to capture a better view, which can be in the front, back or on the sides of the vehicle. The positions at which the sensors are mounted can often create blind spots, which are dead zones that are outside the field of view of certain sensors [1]. In some cases, where objects are below the level of the sensors can often be hard to detect. For example, a small child behind a car or a small animal can be hard to detect if they are below the range of sensors.

2.4 Similar Implementations For Different Scenarios

This section provides a literature review on some different implementations that other researchers have considered, that make use of sound signatures of the surroundings [12] [25]. As there isn't a direct solution to SLAM using audio, therefore different papers investigate different scenarios using acoustics. Echoes of sounds contain enough information that allows to reconstruct the entire surroundings and localize the microphone within it [13]. Echolocation is another approach that is similar to how some mammals like bats and dolphins navigate by using echoes generated by them for spatial navigation [26] [27]. If a single moving microphone is present, then the echoes of the sound generated by a source contain information about the source's speed and distance. A moving microphone can generate a virtual map in the form of a matrix, with information like the positions of different sound sources [13]. This method utilizes an ultra wide-band multi-path probabilistic framework, which is similar to Bayesian inference using the Kalman filter. Although, this approach is

slightly faster to compute than extended Kalman filter, but it still requires hardware and algorithm capable of processing complex data in near real-time. This approach although very different from others, but has some downsides. The methodology requires sound emissions after repeated intervals in order to localize the microphone. Dealing with measurement uncertainties is also a challenging task in a real scenario, as there are multiple sources that modify the way echoes are handled by the algorithm. Also, a moving microphone is not an ergonomic solution in the long run, as it is more prone to wear and tear over time. There is also a need to synchronize the movement of the microphone with the sample rate, as a difference in synchronization can result in a detection failure.

Another approach uses deep neural networks (DNN) for multimedia event detection, using videos pulled from the web [28]. While another application for acoustics is in a multi-modal assisted living environment, which uses a combination of audio and vision for person tracking and activity recognition [14]. Audio signals could be human speech, environmental sounds like a door knock, water from a pipe, by using automatic speech recognition using HMMs, artificial neural networks (ANN), and Bayesian filters. This allows for automatic detection of sounds like walking, cough, human fall, cry, screams or distress calls [14]. These approaches use neural networks for processing the data with precision. Computing real-time data using DNNs or ANNs is a power-demanding computation, which requires something similar to a Tesla K40 GPU. The sampling rate also determines the precision of the algorithm. A high sample rate means better precision, on the other hand, computation of data is much more economic with a lower sample rate, but at the cost of precision [28] [14].

Acoustics can also be used for short intervals, to accomplish small tasks like answering the door, responding to smoke alarm or unloading the dishwasher when the alarm goes off. This method also uses audio and vision-based sensors. It uses a point goal method, where the goal position is given to the agent by its relative displacement vector from the agent's position, and for object goal, the agent receives a semantic object label rather than the goal location [29]. The agent uses its acoustic-visual perception and memory along with the spatial and semantic cues from the acoustic event for both long term and short term events to find the target. The only downside to this method is distraction through other sounds. In a much similar manner, using sound sources as landmarks for being included in the vehicle's

map, in order to localize it along the time provides better understanding of sound generating obstacles like car engines or pedestrians [30]. For processing the captured data, considering the fact that when sound waves spread, the sound source is considered as a core of sound waves, if the distance from the array to the source is bigger than the array's size then sound waves can be seen as a parallel wave for calculation [10]. These methods do not require extensive computational power but do require some sort of noise removal techniques on the algorithmic level.

Distributed SLAM is another approach that is based on matching acoustic events taking place in or around the agent's surroundings [31]. Each microphone is connected to a separate recording device which adjusts for their different clock shifts, and the estimation is done using particle filtering. Using an array of distributed microphones to capture the agent's natural acoustic landscape, a processing algorithm for Audio Event Detection (AED), can compute varying reverberations within the captured audio sample [31]. Since this approach has microphones present at different locations, therefore it requires costly multi-channel A/D converters for sampling synchronization, and the positions of microphones need to be known with high precision. The research also relies on other sources for inputs like speed, which can work in conjunction with this method.

Some researchers also consider a more precise but complex method using binaural estimation. This method uses reverberations can be used as a metric to accurately estimate the distance between the user and the sound source [9]. Using direct to reverberant ratio (DRR), the absolute distance component of the direct energy can be extracted by normalizing distance independent reverberant energy [9]. Segregating direct and reverberant signals from an acoustic mixture based on estimated source direction. The only major drawback is the extensive processing power required for this method, as it processes the left and right ear signals first in order to quickly localize the sound source.

As there are many possibilities on which method to select based on factors like complexity of data produced, the processing power required, cost of hardware, or purpose. The goal is to create a method that contains a precise amount of information about the surroundings, without requiring high processing power. Developing a method that does not produce complex data is also essential, as it dictates whether the approach is capable of working with

other sensors. As data from different sources ideally needs to be synchronized in real-time, therefore quick processing of data and low operational costs are essential.

2.5 System Model Proposal and Advantages

This section proposes a system model for acoustic SLAM and also discusses different scenarios where an alternative method works in a much more efficient and economic manner. For better detection and accuracy it is necessary to maximize the field of view of the acoustic array. Therefore, a sensor that captures the entire 360° field proves to be the ideal solution. The system contains hardware elements with software elements for controlling and processing the data. An ideal sensor would contain an array of eight microphones, each at an equal distance from the other, thus all microphones are separated from each other at an angle of 45°. The microphones can be placed around the car in a specific manner to ensure they are equally separated from each other. Another possibility is installing pairs of microphones along the front, back and sides of the vehicle or a simplified version of a circular array atop the vehicle. The microphone array also requires 12V DC for operating which can be supplied by a rechargeable battery. If a car moves in a certain direction, then there would be wind blowing directly into the microphones pointing in the direction of motion of the car. To combat this problem, a porous plastic shield or a muffler will help block the wind but will allow sound waves to penetrate through. While on the algorithm side, there is a pre-processing algorithm and a post-processing algorithm. The system also has two separate algorithms working in parallel with each other. The first block contains the processing and analysis algorithms. While the second block is responsible for the localization of sound sources. This algorithm localizes by estimation the direction of arrival of captured sound. The processing algorithm is responsible for noise removal, signal filtering, and analysis. Both the hardware and software elements of the system work in conjunction with others to ensure tracking and detection accuracy. The simplicity of the system allows this method to be a cost-efficient alternative to sensors like LiDAR. The data captured by the system is not complex and can be easily processed by hardware with average processing power, without losing accuracy.

The system design is relatively straightforward in order to ensure simplicity in terms of hardware installation. The simplicity of the hardware makes the system serviceable, which proves to be cost-efficient in the long run. The entire system is minimal and doesn't add bulk to the vehicle or affect the drag coefficient in a major way. The placement of the microphones can be modified based on the requirement. Each microphone is placed at an equal distance from the other in order to ensure variability in captures. If microphones are too close, then it makes it harder for the processing algorithm to differentiate between signals captured by two adjacent microphones. In terms of the intensity of the sound, two microphones placed close to each other would capture signals with almost comparable intensity levels. Therefore, spacing out the placement of the microphones ensures an evenly spread out capture field. Which makes it relatively easier for the DOA algorithm to estimate the direction with accuracy.

Some sensor types like LiDAR are expensive and require high data processing power. It also has some drawbacks in certain conditions. Similarly, RADAR and cameras, while cost-efficient, still have some potential drawbacks in various scenarios. Some components add additional bulk to the vehicle due to their size, thus having a direct impact on fuel efficiency due to an increase in the drag coefficient. These sensor types work really well in conjunction with each other but processing all the data drives the operational costs really high. A viable solution to this problem is to make use of a sensor type that is cost-efficient, in terms of operating power and processing the data collected by the sensor.

Considering the first scenario, that is cost-efficiency. Sensors like LiDAR are too expensive to be replaced on a regular basis, in order to ensure optimum functional accuracy. In terms of operational costs, LiDAR typically requires computers capable of handling a large sum of data every second. This is due to the extremely high sampling frequency of LiDAR that requires complex analysis and interpretation. But as a downside, this adds on to the operational and processing costs. In a situation like this, a microphone array model, proposed earlier would be a viable solution. As with most microphones, the data sets produced can be easily processed even with some commercially available, low-cost prototyping boards. The size of data is also directly related to the sample rate selected for the field recording. A parameter like sample rate is adaptive and can easily be adjusted. If a situation presented a highly crowded environment where precision is essential, then in those scenarios the sample

rate can be increased for better accuracy. A microphone array is also easily available and cost-efficient, and can be replaced on a regular basis, in order to ensure tracking accuracy.

There are no international protocols that state the requirements or threshold parameter levels for a LiDAR. This means that manufacturers control the strength of the laser beams since there are no hard-set protocols for operation in public spaces [24]. A beam intensity that is too low does not provide accurate results or a large range of detection. While a higher intensity beam can provide a better analysis of the surroundings and a higher detection range, but can prove to be harmful to people's eyes in the vicinity. A high-intensity laser beam is also potentially damaging towards CMOS sensors used in many cameras. Whereas, a microphone array does not necessarily require any protocols for safety, as long as it is in a fully functional condition. A microphone based sensor also does not generate any form of physical beams or waves, therefore it is not a sensor type that can cause any harm. A microphone essentially works by capturing the signals that reach the sensor and are within the audible frequency range. The data sets captured contain data from all the objects in the surroundings that generate sound. Therefore, the extraction of data can be tailored according to situational requirements. Additional steps on the algorithmic level can easily be implemented for filtering and noise removal from field recordings without data loss.

Some of the major factors that affect the performance of many sensors are extreme weather conditions. LiDAR does not perform well under conditions like heavy rain and snow [24]. It is due to the fact that the presence of water droplets in any form prevents the laser pulses from reaching the desired targets. In a situation like this sensors like cameras would have difficulty detecting and tracking objects. Similarly, beams transmitted by a RADAR would also get slightly hampered by extreme weather. In extreme weather conditions, a microphone array works relatively well for the most part, with slight modifications on the processing algorithm side. The data produced by a sensor like a microphone contains acoustic signatures of the entire surroundings, using advanced processing and cognition techniques like active filtering can differentiate between different types of sounds [32]. Usually, snowfall does not generate any sound, because the surface area of a snowflake is relatively large compared to raindrops. Due to this, snow falls gently and other snowflakes generally dampen any

sound that may be produced by heavier particles. Therefore, in a scenario with snowfall, a microphone based sensor works without any issues, as there isn't a need for adaptive filtering.

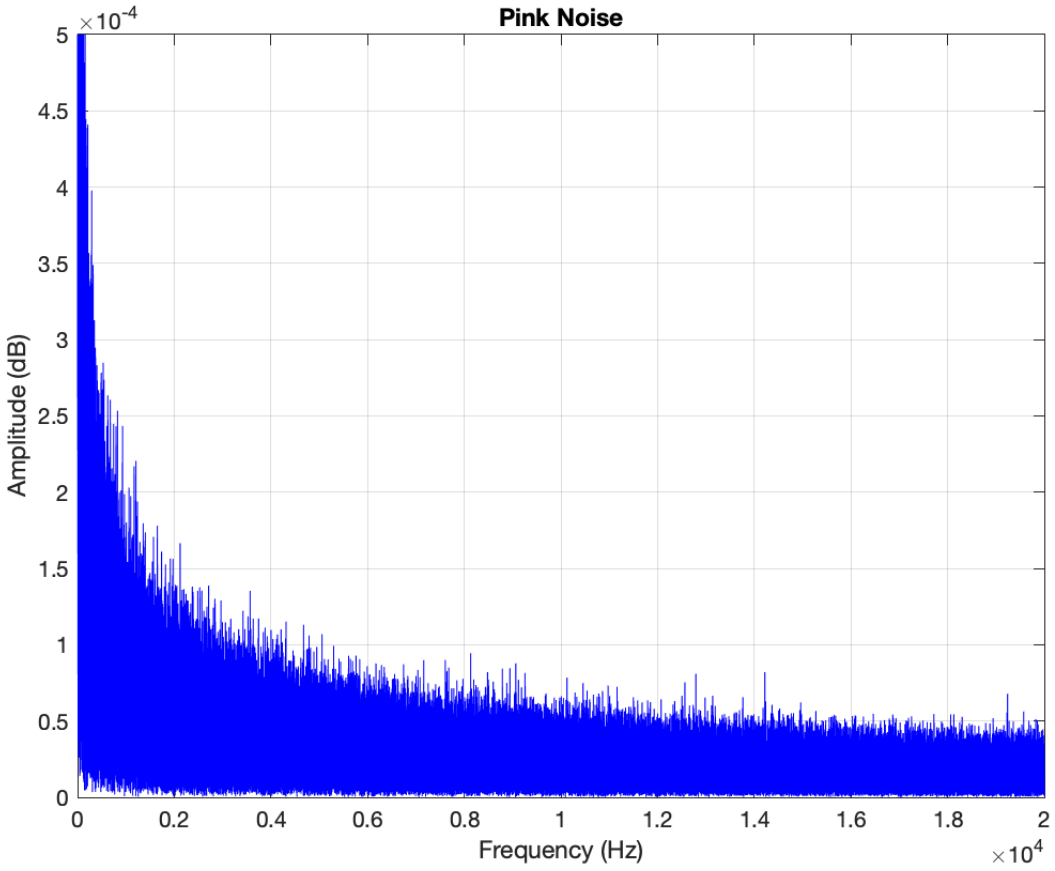

Figure 2.1. Pink Noise FFT

Another case is a scenario with heavy rainfall. As a microphone captures the entire acoustic field of the surroundings, therefore removing the sound of rain depends on the processing algorithm. By definition, pink noise contains equal energy per octave and contains less energy in the high-frequency spectrum (Figure 2.1) [33]. In theory, sounds like rain, leaves rustling, ocean waves and a vast majority of sounds in nature are generally considered pink noise (Figure 2.2) [33].

Compared to white noise, which contains an equal amount of energy throughout the entire audible frequency spectrum, pink noise mostly contains higher energy in the low-frequency region [33]. This makes the task of removing the sound of rain a whole lot easier because

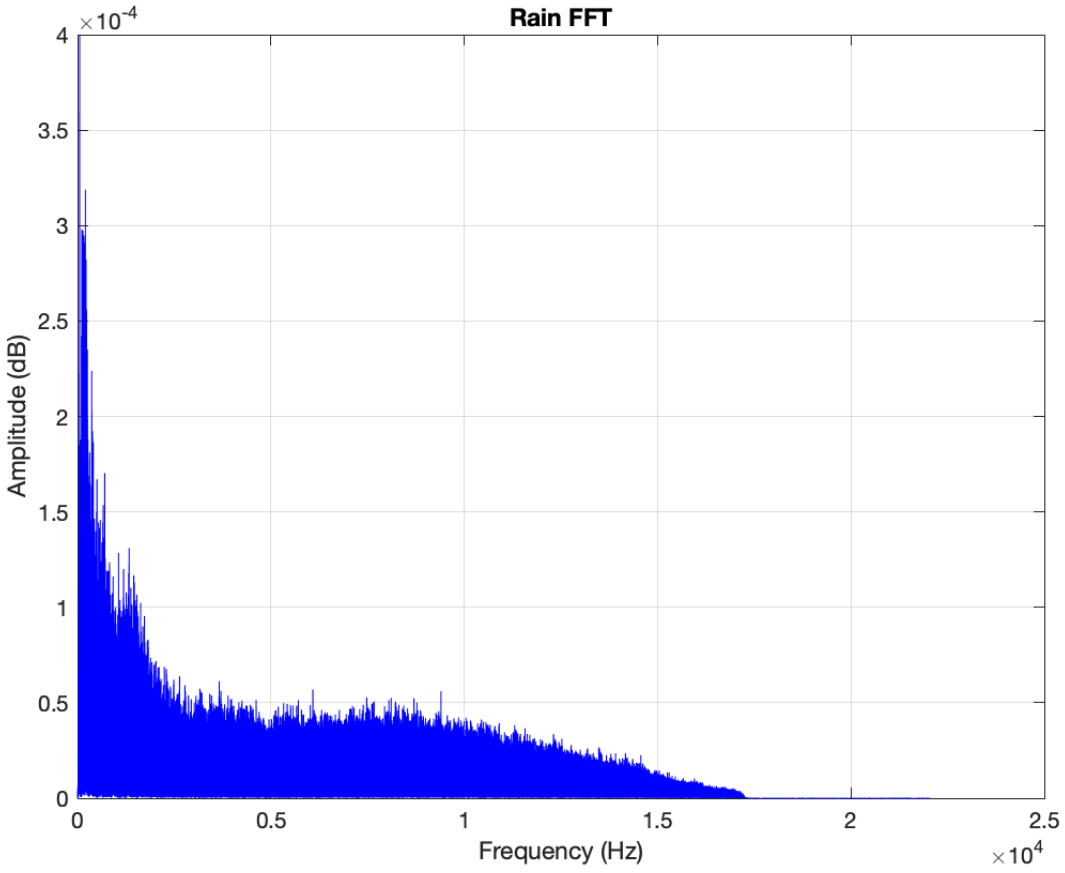

Figure 2.2. Pink Noise Like Characteristics of Rain

incorporating a simple high pass and a low pass filter generates desirable results. This can be improved further by incorporating an adaptive filter in the processing algorithm, that relies on inputs such as the amplitude of the low frequencies compared to other frequencies. Based on these inputs, the strength of the adaptive filter can be controlled. The strength of the filter refers to the sharpness of the cutoff for the adaptive filter or the order of the filter that determines the transition band. Scenarios with heavy rainfall or a higher low-frequency component would require a sharper cutoff, rather than a smooth cutoff.

A vast majority of the sensors used for autonomous driving are mounted at specific locations on a vehicle. This allows the vehicle to capture different types of data with minimum interference from other sensors or external factors. The placement of different sensors can often create certain dead zones or blind spots. These can be referred to as areas that are

out of reach or beyond the detection range of a certain component used for measurement. In situations like such, a microphone array is slightly better in terms of detection solely based on the sound of objects. This works more efficiently when used in conjunction with a sensor like a camera. This allows to detect stationary objects like small rocks, and detect small moving objects like an empty can. A microphone array can detect the sounds generated by objects, while a vision-based sensor helps to identify the class of the object [17]. Based on these inputs, an algorithm can predict what decision is based on the level of priority. A similar situation where a microphone array is a better alternative is for blind-spot detection in bigger vehicles like trucks or buses. Even in cars, detection of small animals moving or a small child can be easily picked up by a microphone, but other sensors types might be unable to track them, as they are below the detection level of the sensor. In larger commercial vehicles, a microphone array can also help to detect the presence of pedestrians, cyclists, pets, kids and even cars.

Almost all types of sensors still function in challenging scenarios, but their accuracy is not as reliable. Especially vision-based sensors need adequate lighting conditions to function properly. For example, in extremely dark conditions, cameras cannot detect objects since they cannot differentiate between the contract of an object and its surroundings. An infra-red based camera or headlights work really well in a situation like such, but they are relatively expensive. A microphone array should not have any problems detecting other vehicles in the dark, as it relies on sound waves reaching the sensor as the input source for detection. Another scenario that has a drastic effect on the functionality of a camera, is a sudden shift in lighting. This can be a situation in which a vehicle enters a dark tunnel from a brightly lit environment and vice versa. This sudden shift leaves the sensor temporarily handicapped, which is a cause of big concern. Failing to detect other objects or vehicles can lead to fatal casualties [1]. Therefore, in a scenario with varying lighting conditions, a microphone based sensor works relatively well.

Similar to varying lighting conditions, bright lighting can also hinder the performance of some sensors. Reflections from metallic surfaces or windows also impact the performance of sensors like cameras. A LiDAR's performance is also gets diminished by bright reflections. For example, when the sun is at relatively higher angles around noon, then the intensity of

sunlight is strong. The high intensity of the sunlight overpowers the laser pulses generated by a LiDAR. In a situation like this, the laser pulses get attenuated by the bright sunlight and thus making it harder to detect the target objects. Similarly, bright reflections also have the same effect on LiDAR's laser pulses as well as a vision-based sensor. On the other hand, an acoustic sensor has no effect on its tracking performance in extremely bright conditions.

In general, an acoustic method that uses a microphone array is more advantageous in scenarios like these, compared to some other sensor types. Since an acoustic sensor relies on the sound waves reaching the microphone's membrane for detection, therefore its performance is not hindered by external factors to a big extent. With conditions like heavy rain, slight modifications are needed on the algorithm side using a well-tuned combination of a high pass and a low pass filter to mask the pink noise like characteristics of the sound. Whereas, noise generated from wind can be eliminated by using a muffler on the microphone or porous plastic shield that allows the sound waves through. Extreme weather conditions like snow do not have any major impacts on the performance of a microphone when compared to LiDAR and cameras. Even in varying lighting conditions, whether the situation is too bright or too dark, a microphone's performance is not affected by it, as it solely relies on sound waves. This method also has potential applications for blind-spot detection, especially in large commercial vehicles, where the field of view around the vehicle itself is very limited.

3. MICROPHONE ARRAY DESIGN

In order to achieve higher accuracy, it is necessary to have multiple microphones. Generally, one single microphone would be enough for distance estimation based on the sound intensity levels. But in order to localize the direction of the sound, microphone pairs help in this regard [34]. The idea is similar to stereo sound. In stereo audio, the sound is split into two channels, one left and the other is considered right [9]. This creates a virtual perception of how our ears audition sound. This creates an effect that seems familiar to our ears, and we feel like different elements of the sound are being transmitted from two different directions. While in reality, it is just the sound intensity levels being higher in one channel compared to the other, is what creates a stereo effect. In a much similar manner, using two microphones can help in providing the general direction of the sound source in terms of left or right. Which is a relatively low level of data, as it doesn't provide enough information to the user.

To provide a higher degree of estimation in terms of DOA, multiple microphones can prove to be beneficial [35]. This idea can also be modeled by a similar technology used in movie theaters, known as surround sound. In this technique, the audio is split into multiple channels and is played through multiple speakers that are placed around the host [8]. Usually in this method, the host or the listener is placed in the center of the field, as that is the point where the sound waves converge and enhance the listening experience. The placement of the speakers creates a virtual perception of the spatialization characteristic of sound, by altering sound localization [8]. This creates an effect that gives a perception of distant sound sources by altering their sound intensities in a horizontal sound field. Similarly, in order to localize sound sources at different distances, an approach that utilizes multiple microphones should provide better results. Multiple microphones, each at equal intervals from the other provide a better DOA tracking accuracy [35]. In order to ensure that the entire acoustic field is covered by the array, an eight microphone approach would be ideal under many conditions. This helps to localize a sound source by creating eight acoustic zones, each 45° wide. The sound source is then localized within one of the octets based on the sound intensity levels [36]. This is necessary for higher accuracy, as the outputs from the DOA algorithm are used as inputs for the processing algorithm.

3.1 Array Design

To achieve higher tracking and detection accuracy, a good sensor design is necessary. Tracking or failure in detecting all directions can often lead to fatal casualties. Therefore, capturing the entire 360° field gives a better estimate of the fast-changing surroundings of the host vehicle. This in turn allows the algorithm to capture sounds from different directions and process their intensity levels in real-time. The general idea is to have as many microphones as possible, in order to capture a wide sound field with precision. Although, a moving microphone array would limit the number of microphones needed by capturing the entire field by changing its position [13]. This is a cost-efficient solution as it limits the number of sensors needed, but sensors with moving parts wear out at a faster rate when compared to mounted sensors. Having at least four sensors, one at each corner of the vehicle, cover four distinct directions that provide general information about sound sources present in those specific quadrants of the vehicle. A six microphone array can do an even better job at tracking, detecting, and estimating intensity levels of sound sources around the vehicle, when compared to a four microphone array.

In an ideal scenario, it is better to have certain redundancies, as it creates a more robust system for the long run. Considering an eight microphone array proves to be a good solution. With an eight microphone array, there are enough sensors pointing in different directions, which can provide a vivid depiction of the surrounding sound sources. Even if one or two microphones are not functioning adequately, there are still enough sensors present to safely compute the data. With an eight microphone approach, the sound field of the host vehicle is divided into eight distinct acoustic zones. This helps separate various sound sources into different areas based on their sound intensity levels captured by the array. A Higher number of microphones also provides a better estimate of the direction of the sound, as the DOA algorithm has multiple data sources to compare the values with. This approach helps to increase the overall accuracy of the system.

The orientation and positioning of the array are not limited to just one fixed design. In an ideal case, evenly positioned microphones will provide the best results due to a symmetric capture of the acoustic field (Figure 3.1). But for practical applications, slight variations to

the design would also work as there are no restrictions to the placement of the microphones, as long as the sound waves can reach the sensor without any hindrance.

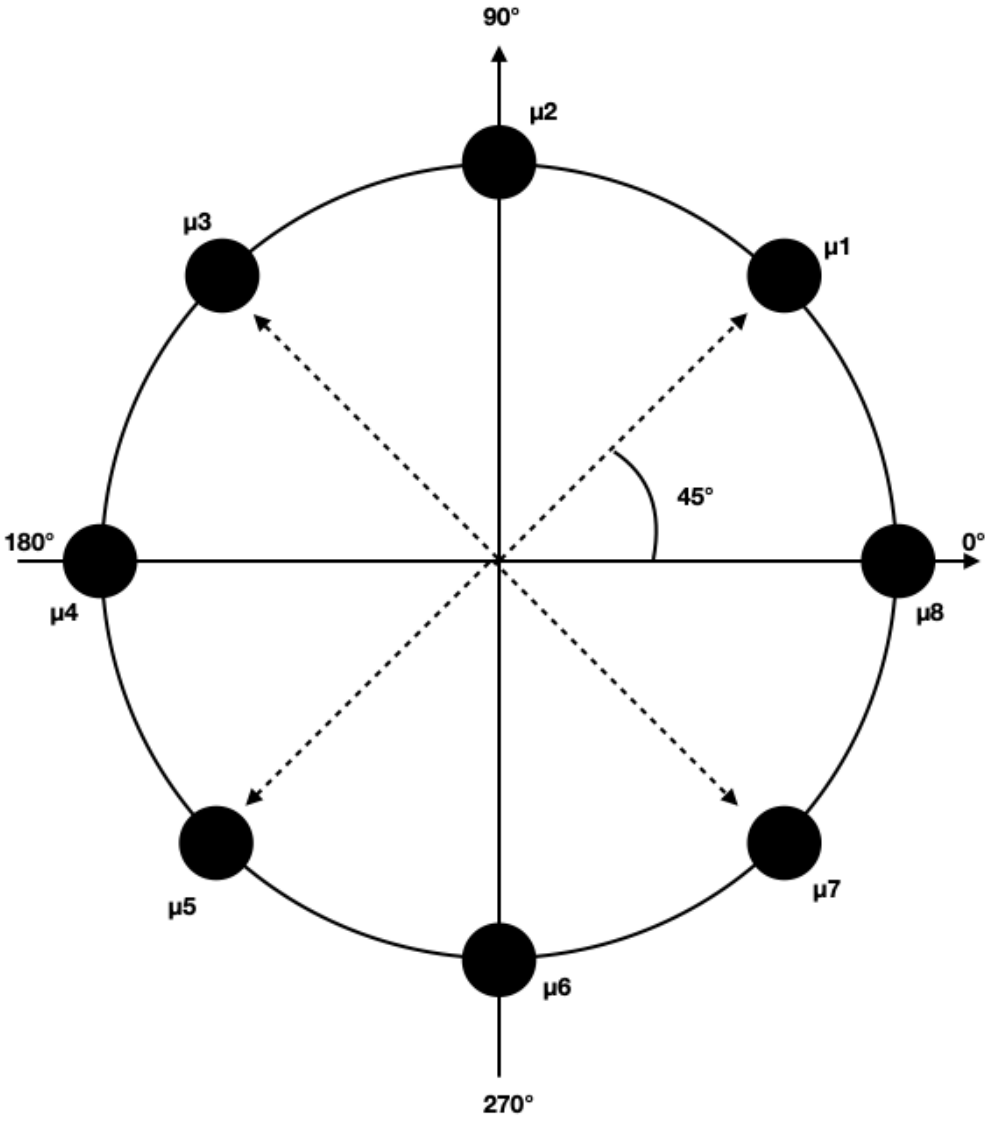

Figure 3.1. Microphone Array Design

Some other variations can include mounting four microphones, one on each corner of the host vehicle, and the remaining four microphones between the four sensors installed on the corners of the vehicle (Figure 3.2). This orientation ensures an even placement of all the sensors, based on a vehicle's shape and size.

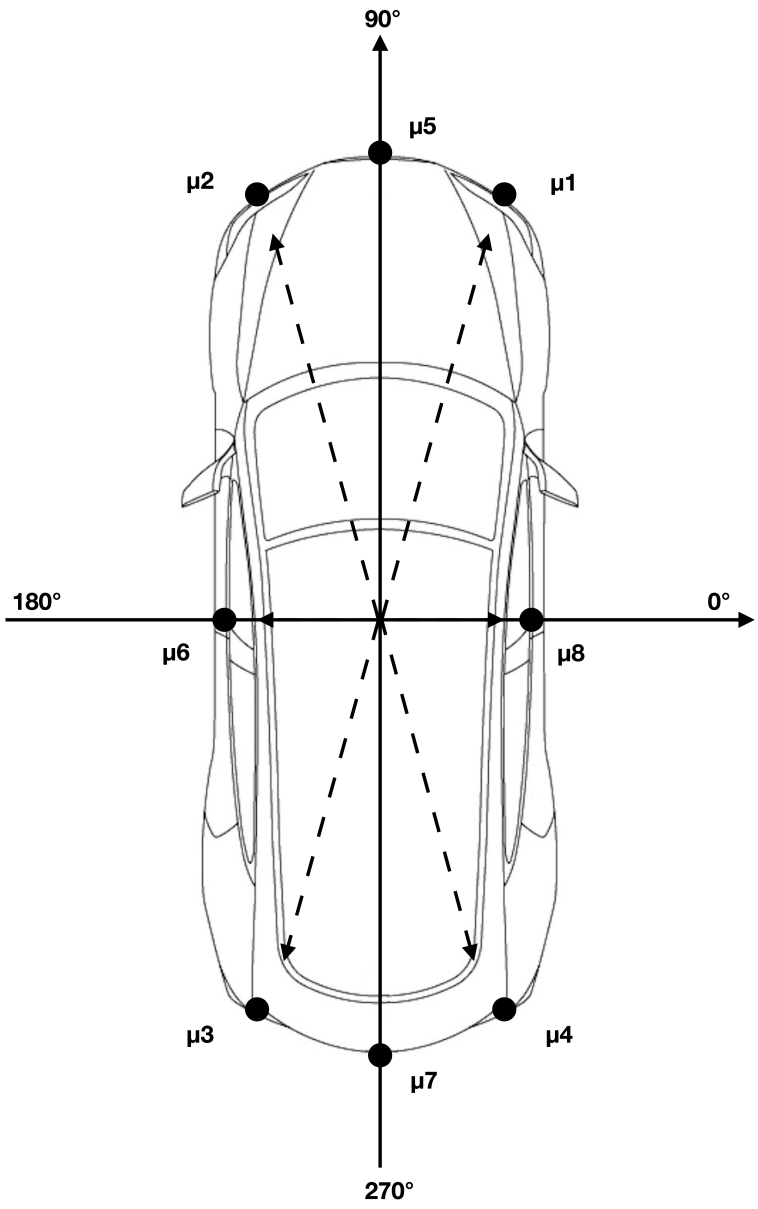

Figure 3.2. Microphone Array Placement On A Vehicle

3.2 Benefits of The Design

The symmetric array design and sensor placement have many advantages. A symmetric design allows the array to capture acoustic signatures in a symmetric manner. This means that if a sound source is right behind the vehicle and in the center, then the microphones on the left side have comparable data to the microphones on the right. The DOA algorithm benefits from a symmetrical array design, as it can provide finer tracking accuracy by comparing the values from the corresponding microphone pairs on each side. An array with eight microphones also provides some amount of redundancies in a scenario where a few sensors might fail over the course of time. Array with multiple sensors is more robust in terms of reliability and is also a cost-efficient solution. With multiple sensors, individual sensors are easily replaceable in a situation where a sensor might fail. The system setup isn't too complex and does not require intense processing power. In some severe weather conditions like heavy winds, where some of the sensors are directly facing the direction of the wind, in those cases, the remaining microphones can be used for detection. An additional porous shield can also be installed in front of the sensors in order to block heavy winds, rain, or snow. An even placement provides the DOA algorithm with enough comprehensible data to estimate the direction of the sound source. Some microphones can also be installed at a higher level than the other microphones or vice versa, in order to provide a better tracking range.

Dividing the 360° field into eight sub-sectors enables the system to capture acoustic signatures at locations that are beyond the range of some other sensor types. By placing multiple sensors, each facing in a different direction provides the ability for blind-spot detection. The microphone array can easily pick up acoustic signatures of small children or pets around the vehicle. This is extremely beneficial for large commercial vehicles with a limited field of view around the vehicle. A system like this can help detect the presence of pedestrians, bicyclists, or other moving objects around the host vehicle. A similar application can also be extended to recognize certain types of sounds and provide higher priority to them, solely through the sound signatures [25]. This can include sirens of emergency vehicles or people walking. In general, this system provides a cost-efficient alternative to some other sensor types and has

advantages in certain scenarios where other sensor types cannot function adequately. Especially, in low light or dark conditions, as the sensor's performance does not depend on it. The overall cost of the hardware and software components is also minimal in comparison to sensors like LiDAR.

4. EXPERIMENTAL SETUP

To test the functionality of the system, a combination of hardware and software components were selected. The experimental setup allows testing the algorithms, as well as the functionality of the hardware. The setup also allowed the system to have flexibility in terms of improvements. The experimental setup was also necessary for testing different scenarios and improving the systems tracking and detection accuracy. The components were selected based on the requirement of the system's design. In addition to different scenarios, different hardware components were tested as well, in order to find the ideal solution. The whole point of having an experimental setup was to provide a working proof of concept and compare its results with different measurement technologies used. This also provides a chance to test and debug scenarios that would occur beyond the ideal conditions, like the battery life of different components used for testing. The results from testing and debugging give the opportunity to make improvements to the system design, in order to achieve a higher tracking and detection accuracy.

4.1 Hardware setup

The system design requires a microphone array for higher tracking and detection accuracy. For this reason, a microphone array was selected for testing and fine-tuning the algorithm. The goal was to maintain simplicity in terms of hardware and software. For testing and debugging purposes, a simple hardware setup was considered. This was necessary to provide a proof of concept and demonstrate a working system. In order to provide an interface between the hardware and software, a portable computer was also used. The portable computer is used for communications between the hardware array and the algorithm. The computer is also responsible for capturing, storing, and processing the data gathered by the sensor array. For powering the computer and the array, a battery would be required in the vehicle. But for testing purposes, the built-in battery of the portable computer was used. In a situation where the system is installed in a vehicle, having direct power from the vehicle's oscillator and a backup battery for the system, would prove to be a better solution. In case of a technical snag, there is a second alternative that can still provide power to the system.

In order to test and verify the functionality of the tracking and detecting algorithm, a microphone array with a simple design was chosen. Therefore for the microphone array, a Microsoft Xbox One Kinect sensor was selected (Figure 4.1). This was chosen, due to its wide availability and being a cost-efficient component. The sensor has a four microphone linear array at the bottom, with each microphone evenly positioned from the other. This array is sufficient for testing the functionality of the system and providing proof of concept. In an ideal scenario, an eight microphone array that is capable of capturing the entire acoustic field would be a viable solution in the long run.

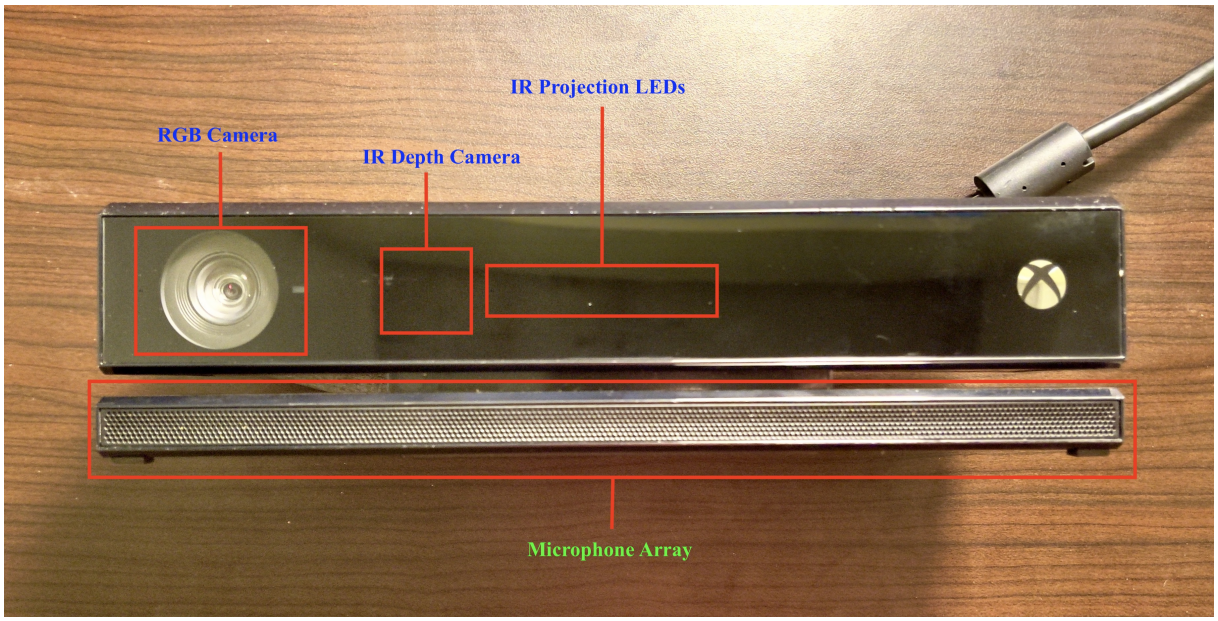

Figure 4.1. Microsoft Xbox One Kinect Sensor

The four microphone array present in the Kinect sensors also has an RGB camera, along with an IR depth finding camera. This provides an alternative to expanding the capabilities of the system through sensor fusion in the future. A combination of an acoustic sensor, a camera and a depth measuring camera would increase the system's measurement and detection accuracy. In order to mount the Kinect sensor, no additional clamps or mounts were needed as the sensor can be easily taped to the top of any surface. The microphones can also be protected with a plastic shield, that would act as a barrier against heavy winds. Another alternative is to cover the microphones with a porous foam windbreaker, which does

not require too much space for installation compared to a plastic shield. The sensor array also has a proprietary port for connection, therefore additional adapters were also required.

Figure 4.2. System Setup

The sensor array has specific power requirements for operation. The operation voltage for the Kinect sensor is rated at 12V DC. This can be provided using an external battery in a vehicle. Some cost-efficient alternatives for a 12V DC power supply include lead-acid batteries. A lead-acid battery has the capability of being recharged through a car's oscillator,

while the vehicle is moving. Lead-acid batteries are also generally safer compared to lithium-ion batteries. For demonstrating and testing, a regular universal serial bus adapter was selected. This allows the sensor array to be connected to a portable computer, and the functionality of the sensor can be tested. A speaker was also used to generate sounds of different vehicles and noises like rain or wind. The purpose of this hardware setup was to simulate different scenarios that would occur in a real-world situation.

In order to provide power to the sensor for testing and debugging, the array was directly connected to a wall outlet. A 12V AC to DC power adapter was used to supply power to the sensor (Figure 4.2). The overall cost of all the hardware components used for testing was around $80, excluding the portable computer used. This still proves to be a more economical choice compared to sensors like LiDAR.

4.2 Software setup

The software is an integral component of the entire system. The functionality and reliability of the system depend on the software setup. It is crucial to keep the algorithms separate for different types of measuring techniques, in order to keep the processing smooth. When it comes to the overall response time of the system, having a fast algorithm is necessary. The algorithm needs to be fast enough, to process data in near real-time. A delay in processing the real-time data can lead to fatal errors [1]. Testing and debugging different scenarios helped in increasing the reliability of the algorithm.

As the software elements were being tested to find different areas of improvement, therefore a flexible coding environment was needed. For this reason, MATLAB was selected as the ideal choice. MATLAB provides enough toolboxes for processing different types of data. It also supports a wide array of hardware, along with toolboxes for each type of sensor. Using MATLAB, data can be captured from different data sources and can be stored dynamically in different variables, which can be used later in the algorithm. The data stored in the variables can also be exported to different file types for further analysis using various software tools. The most important resource provided by MATLAB is to view the data or the results

graphically. This allows the system to show the computed results obtained from the data through a visual interface, which makes it easier for an average user to understand the data.

The software setup is split into two separate algorithms. Both the algorithms have separate sub-algorithms and functions responsible for computing different elements needed for approximation. The first algorithm is the DOA estimation algorithm which is responsible for localizing different sound sources within the host vehicle's proximity. The DOA algorithm uses acoustic signatures of surrounding vehicles and computes an approximation of the direction of the sound source. The direction of the sound source is then displayed in the form of a vector pointing in the approximated direction of the sound source.

The second main algorithm that is part of the software setup is the proximity approximation algorithm. This algorithm is vital for the system's reliability and accuracy. There are many sub-algorithms within the main algorithm. Communications with the hardware components like the sensor are carried out by both the DOA as well as the proximity approximation algorithms. The communication between the sensor and the algorithms is carried out using the bus connection. Additional drivers are also needed to be installed on the computer, in order for the sensor to work. The driver for the sensor (Kinect SDK 2.0) is available via Microsoft's website.

Outputs from the algorithms can be more accurate if measurements of quantities are relatively close to actual values. In order to ensure that, the exact positioning of the microphone array is needed. The Kinect sensor has a four microphone array housed in the bottom segment of the sensor module. The exact placement of the microphones was extracted in centimeters. This step is needed before setting up parameters in the DOA algorithm. The experimental setup for extraction of the microphone positions was done by generating a tone from a sound source in front of different microphones and comparing their values.

Positions of four microphones in the Kinect sensor were calculated by using a free and open-source software tool known as audacity. This was essential, as a four microphone array produces a four-channel audio file, which can be captured and processed using Audacity. In the previous versions of the Kinect sensors, the microphones were not evenly spaced, therefore adjusting for a similar placement is necessary, without disassembling the unit. The position of each microphone was determined in the form of x and y coordinates. Where x is

the distance of the microphone from the initial position, and y is the height of the microphone from the bottom of the array. The procedure to determine the position was conducted by playing a tone in front of each microphone through a speaker. For this purpose an open source library was used [37].

The tone being played was captured using Audacity (Figure 4.3), and the highest intensities were extracted by mixing all four channels. The microphone with the highest intensity of capture stands out, and this process was repeated multiple times, in order to get a better estimate of the positions of the four microphones. For finding the positions, the center of the array was used as the initial point or the origin of the x and y coordinate field. This means that the positions of the microphones to the left of the origin are marked by negative values for the positions. While microphones to the right of the origin are marked by positive values for distance (Figure 4.4). The total length of the microphone module was measured to be $24.6cm$, with the origin at $12.3cm$. A sine wave was played close to the microphone array and the highest intensities were captured (Figure 4.3). The point of the highest value was the position of the microphone. This process was performed by sweeping the sound source from left to right and marking the position of the highest intensity. Calibration of the microphone positions was performed five times and the values were recorded in Table 4.1. The average of all five trials was taken to get a good estimate of the positions.

Table 4.1. Calibration Trials Average

Trials	**Mic. 1**	**Mic. 2**	**Mic. 3**	**Mic. 4**
Trial 1	11cm	3.4cm	7.1cm	10.9cm
Trial 2	11.1cm	3.7cm	7.3cm	11.1cm
Trial 3	10.9cm	3.5cm	7.2cm	11cm
Trial 4	11cm	3.6cm	7.4cm	11cm
Trial 5	11cm	3.4cm	7.2cm	11cm
Average	**11cm**	**3.52cm**	**7.24cm**	**11cm**

Optimization of the algorithms is necessary for run-time efficiency of the system. Since the software elements operate in real-time to process the data, therefore code efficiency is an important factor. In an ideal condition, the data should be processed in real-time, but that is hard to accomplish as the computer requires some amount of time to compute the data.

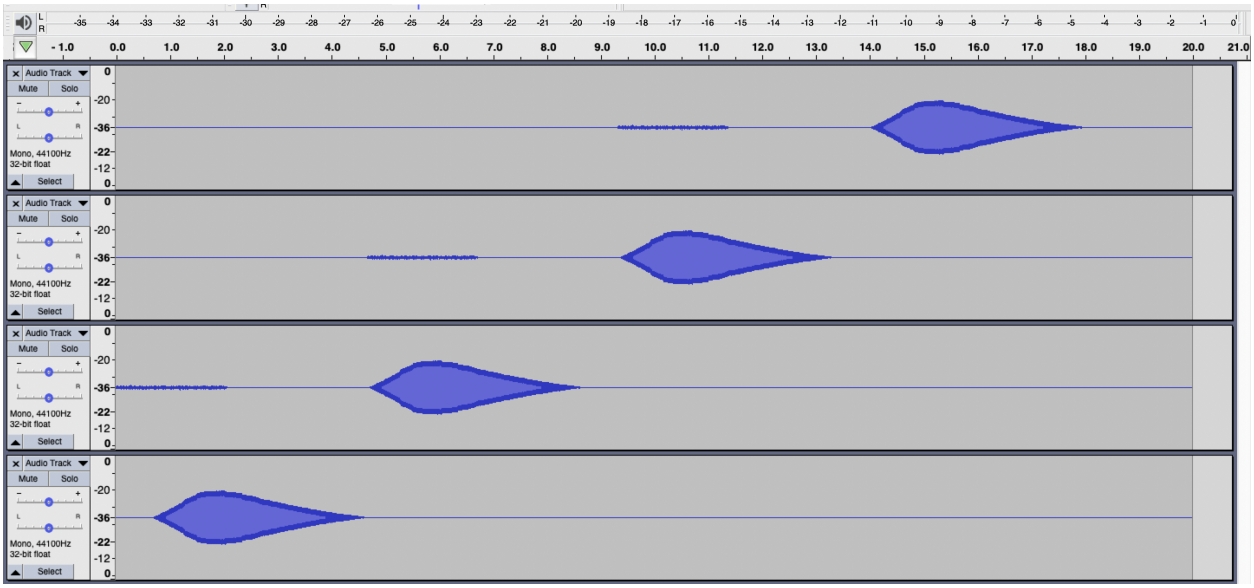

Figure 4.3. Microphone Position Calibration Using Audacity

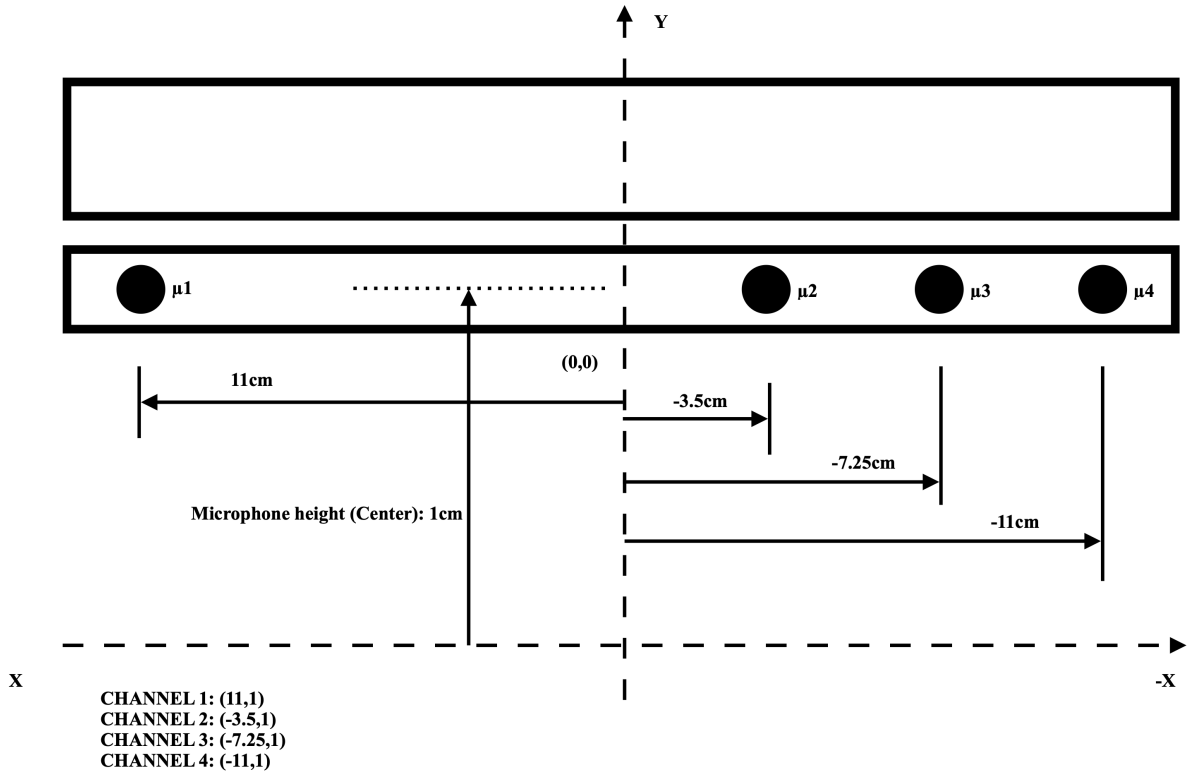

Figure 4.4. Microphone Position Layout

But the time delay can be minimized to near real-time values by using different loops for different tasks. Organizing the code based on the order of occurrence also helps to prioritize tasks that are higher level, and provide inputs to other sub-algorithms. The optimization of the code was done by making use of variables in a dynamic manner. This method avoids the initialization of different variables, which adds additional processing load onto the computer. Rather, using variables under conditional loops helps to assign values to them when certain requirements are satisfied. Using separate files for different functions also helps to reduce the execution time. For example, the DOA algorithm displays the results in the form of a vector that is updated every half a second. If this process is done as a part of the main DOA algorithm, then there is a noticeable amount of lag to the rate at which the graphics for the display vector are updated. Whereas, updating the vector graphics as a separate function executes at a much faster rate.

In a much similar manner to updating the vector graphics, having a separate function for capturing audio for both the algorithms also increases the execution speed. Another method incorporated for optimizing the code for higher efficiency was to set a buffer size. The buffer is used for capturing small bits of audio and then pushing it to other functions for further processing. A buffer size that is to big can capture large amounts of data, but requires additional processing power. On the other hand, a buffer size that is too small can miss out on some important elements of the data while sampling and capturing the data. Therefore, the buffer size for both algorithms was kept different, as both achieve different tasks. The DOA algorithm has a buffer size of 64, while the proximity approximation algorithm uses a buffer of length 32. As more precision and speed are required from the proximity approximation, which is why a buffer length of 32 works best. The DOA algorithm does not necessarily need high processing speed, therefore a 64 buffer length is sufficient. The buffers for both the algorithms are cleared when the data is pushed to other functions for processing, which prepares them for storing new data from the sensor source.

The structure of the DOA algorithm is as follows. The algorithms are divided into two main parts. The first one is the main algorithm with other sub-algorithms for achieving various computations. While the second algorithm is more of a function that is responsible for updating the vector graphics. Execution of both the algorithms parallel to each other

increases the overall response time of the DOA algorithm. Starting from the main algorithm, the first section is responsible for capturing the acoustic signatures of the surroundings. In order to ensure that the data quality is of a high standard, a decent sample rate is necessary. A sample rate that is too low can miss out on some important elements while sampling the audio, whereas a high sample rate is generally good, but requires high processing power. According to Nyquist Sampling Theorem, the sample rate should be at least two times higher than the highest frequency value in the expected data [38]. Since the human hearing range is from 20 Hz to 20 kHz, therefore a sample rate of 44.1 kHz works well. Another element to the audio capturing sub-algorithm is frame length. This decides the length of the signal to be captured and stored in a variable. This is necessary in order to process the data in near real-time, as a specific frame length allows the system to capture new data while processing the most recently captured data in the background.

The next segment of the DOA algorithm utilizes the positions of the four microphones found using Audacity. This is needed to compare the sound intensity levels over the distance of separation of the microphones. If a given sound source is closer to one microphone than the others, then the corresponding microphone returns a captured signal with higher sound intensity level. Then, on comparing theses intensity values with the rest of the microphones, the algorithm can approximately localize the sound source. This is done by computing the position of the sound source in radians, based on the microphone that returns the highest intensity levels. The algorithm compares the intensity levels for each microphone, with a pair of adjacent microphones, in order to get a precise approximation of the direction of the sound source [39]. The entire algorithm loops, over and over again, in order to capture, process, and update the vector graphics. Since for testing and demonstration, the system uses a four microphone array, but this same setup can easily be replaced with an eight microphone array. The only variable that would need to be modified is the array for the positions of the eight microphones, with respect to an initial position or the origin.

The proximity approximation algorithm relies on inputs from all the microphones in the array. This helps the algorithm distinguish between sound sources that are closer to the host vehicle. The proximity approximation algorithm prioritizes sound sources with a higher intensity over sources with lower intensity levels. High-intensity level directly relates to the

sound source being closer, which is why those sound sources are given a higher priority. Then, the following sources are computed based on the decreasing order of intensity levels.

The proximity approximation algorithm is structured as follows. As mentioned earlier, the sample rate is selected to be 44.1 kHz, based on the Nyquist Sampling Theorem. The audio capture segment works almost identically to the DOA algorithm, but the buffer size is used as 32. The audio captured is stored in a lossless uncompressed audio file with a *.wav* extension. Another factor that affects the quality of the recording, is the bit depth of the recorded file. The bit depth of the audio file can either be 8, 16, or 24. Generally, a higher bit depth of 24 is considered to be the best, but it also produces data with large file size. Therefore, a bit depth of 16 was selected, and it works well for capturing detailed acoustic information.

The next factor responsible for the fast computation of data is the sample frame length. An ideal frame length should be long enough to capture enough data, but not too long as it would take time for the data to process. A long sample frame length increases the time needed for computation, and the results would be displayed after the event has occurred. Therefore, a sample frame length of 1024 samples per frame was selected. This ensures that enough data is captured every second, without skipping any important information. The captured frames are then passed on to the pre-processing block for further analysis.

After audio capturing has occurred, the data is pushed to the pre-processing block for further analysis. The main role of the pre-processing block is to remove any unwanted noise from the audio captures. The amount of noise removal required for each recording is not fixed, as it depends on the level of noise present in each capture [40]. The transients in the signal are analyzed in real-time, and if any unwanted or non-periodic transients are present then they are considered as external noise. The noise removal process lowers the amplitude of any non-periodic transients in the captured audio. These can be due to uneven surfaces of the road, car horns, through natural elements like birds or people walking.

After pre-processing of data is complete, the data is pushed for further computations. In this step, only the highest intensity level and its corresponding microphone are considered. The intensity levels of the captured data are then used as inputs to the proximity approximation block. The inputs from other microphones are also compared and then are pushed to

the post-processing block for displaying the results computed from the captured data. The computation of an approximate distance uses a function that contains the mathematical equations and relations that translate the intensity levels to a distance value. The computation of the distance using a separate function lowers the feedback time of the computation block of the algorithm.

The next step of the proximity approximation algorithm is post-processing the data. In this section, since the data contains fours audio channels, the loudest channel is considered for displaying the results. A comparison between all four audio channels decides the most dominating audio channel, which directly relates to the loudest intensity microphone. The next step is to clear all the previously used variables for newer sets of data. After this step, the array length is adjusted in order to display the results graphically. Once the vector lengths are set, the data is pushed then pushed to the graphics block.

The final segment of the algorithm is to display the results graphically. The results are kept simple and easy to understand for the user. In order to display the results, a waveform display, a Fast-Fourier Transform, and the distance in a colored scale are shown. These are updated in near real-time, when newer results are computed and pushed to the graphics block. The waveform display shows data in an Amplitude vs Time form. Whereas the Fast-Fourier Transform display shows the arrangement of different frequency elements in the data in a Amplitude vs Frequency assortment [36]. The distance approximation is shown as a color meter, where green represents sound sources that are further away from the host vehicle. Brighter colors like red represent close proximity of the sound source to the host vehicle. An additional spectral display can also be accessed if needed, but it adds some amount of delay to the results. The data presented by a spectral display is also complex and a bit difficult to comprehend for the user, but can be accessed if it is required for a detailed analysis of the captured data. For simplicity, the results are displayed in a manner that is easy to be comprehended by the user and are mapped according to the spatial arrangement of the sensor array.

4.3 Hardware and Software Integration

Optimal hardware and software integration is essential for the system's feedback and response to be as close to real-time as possible. The sensor array captures raw analog data from the host vehicle's surroundings, which is encoded into a digital signal using a universal serial bus adapter. The conversion is needed, as it allows the data to be imported into a variable form and analysis can be performed on the data. The system runs two separate algorithms in parallel with each other, in order to compute and display different types of information. The sensor array also has extra redundancies like an IR depth camera and a regular RGB camera. These can act as a good alternate for providing multiple types of data using only one sensor module.

On the software side of the setup, various sub-functions are utilized, in order to execute the computations at a faster rate. The code was also tested and optimized to ensure optimal functionality in different types of scenarios. On the software side of the setup, additional libraries were also installed for the sensor. These libraries were provided by Microsoft, which allows the sensor module to be set up and used with a computer. For the algorithms to provide accurate results, the exact positions are also needed, which were calculated using the calibrations performed using Audacity. Additionally, Simulink was also used to test the performance of the microphones in terms of their sensitivity, by computing a frequency response.

As the entire system uses and processes captured audio for computation, therefore MATLAB's audio toolbox was selected to streamline the process. This provides the functionality for the algorithm to adjust values of specific parameters, that would increase the quality of the captured data. The toolbox also provides functions for displaying the results in a graphical manner with the axis supporting decibel, amplitude and frequency data types. Additionally, digital signal processing system toolbox and signal processing toolbox were also included as a part of the audio toolbox. These provide mathematical functions that can be accessed to convert the amplitudes of the data into other forms like logarithmic or power values.

MATLAB environment was selected due to its support for a wide array of hardware components. Different manufacturers can provide their own library or can be downloaded directly from MATLAB. Multiple hardware components can also be used within one algorithm, with each sensor component providing different data types. In the case of a Kinect microphone array, it provides a four-channel audio file, which is a bit more complex to process compared to other audio samples with two audio channels. For this reason, only the loudest channel is considered and the rest are eliminated, as it enables the algorithm to process the data faster. The loudest channel also means that the sound source is closer to that particular microphone compared to the rest. The intensity of sound and distance of the sound source from the listener can be modeled by an inverse relation.

Integrating hardware and software components together exposes the test setup to different test conditions. This helps to adapt the system to different conditions and fix problems that arise during testing. This helps to create a robust hardware and software system setup. A setup like such enables the system to be mobile, and can easily be mounted on a vehicle. This also enables the system setup to be modifiable in the future, if extra features were to be added. As the algorithms were designed specifically for sensors like microphone arrays, therefore adding extra microphones is a seamless process. As all parameters of the sensor module can be controlled by the software setup, which eliminates the need for additional tools that are needed for analysis.

Within MATLAB, the algorithms can also be exported to C-programming language. The benefit of this is to run the software setup on a small portable computer, which would prove to be more power efficient and compact. A similar logic of the algorithms can also be translated to other programming languages, in order to make the code more adaptable for different systems. The integration of the hardware and software components allows the system to run independently, without being affected by the performance of other sensor types. The system needs to be ready when new data is available for processing without any additional delays. Therefore, making use of variables for storing and processing new data allows a smooth flow of data elements within the algorithm. Mapping the computed results in a graphical manner makes the task of comprehending the surroundings easier for the user.

4.4 Benefits of Test Setup

Having a test setup helps to simulate different scenarios and find potential points of failure. Many bugs within the code were debugged and the overall structure was improved after testing the system setup. The overall structure of the system setup was relatively simple and cost-efficient. The data gathered by the sensor array is not too complex and can be easily changed based on the capture setting. Processing the captured data requires a moderate amount of processing power, making it a good alternative for some other sensor types that require extensive computing power. The flexibility of the system made it easier to test different hypothetical scenarios and test the system's response in those cases. Based on the results obtained from those simulations, the parameters were tweaked further, in order to eliminate flaws from the system. An example of such a flaw was a relatively high processing delay due to large buffer size. This was easily fixed by lowering the buffer size, so the data can flow at a much smoother rate.

One major advantage that the system carries over other alternatives is the ability to function independently. This eliminates the need for additional complex sensor components, that increase the overall costs associated with the system, and in turn require higher operational power. The ability to run independently provides the freedom to install other sensors for providing additional features to the user without interfering with the system setup. The test setup is flexible enough, that additional microphone arrays can be combined with the initial setup to provide better accuracy, without the need for modifying the algorithms. Since the test system is a working prototype of the actual system, therefore it can be mounted on a host vehicle to test real world scenarios. Apart from these, the system is also cost-efficient compared to some other sensor types like LiDAR. It also has major advantages for blind-spot detection, especially on large commercial vehicles with a limited field of view of the surroundings.

The software setup of the system is simple and comprehensive in terms of logic. Information is relayed to the user through graphical interfaces, making it easier to understand the complex data. The structure of the algorithms is also flexible, which allows room for modifications in the future. The functions used by the algorithm can also be written in

different programming languages, making it platform-friendly. The simplicity and flexibility of the system give full freedom for sensor fusion. Sensor fusion would allow the system to track and detect multiple types of objects in real-time. This would increase the overall accuracy of the entire system. An example of sensor fusion is to use vision-based sensors for detecting still objects or object moving around the host vehicle [41]. Using a microphone array for detecting acoustic signatures of objects, while using vision sensors for confirming the presence of those objects would provide better tracking results. Similarly, combination of sensors like RADAR with a microphone array can help localize still and moving objects [41] [42].

Another major benefit is the flexibility of the algorithms. This is in regards to the ability to run on different platforms for better performance or additional features. The portability factor also means that the code can be flashed onto a small prototyping board, and using the ports on the board, various hardware peripherals can be connected. This allows creating variations of the system based on the vehicle and the available budget. By using MATLAB as the software environment, the code can easily be translated to different programming languages. The most convenient option is to export the algorithms to C-programming language, directly from MATLAB. This would then allow the code to be embedded on portable microprocessors. The basic logic of the algorithms can also be translated to other programming languages using some third-party tools or directly by importing the logic of the functions. This provides the freedom to select different platforms for the system based on the budget and the requirement for the scenario.

In general, the test setup provides a real emulation of how the actual system setup would perform. This also provides enough flexibility testing and make changes to the system design by modifying the algorithms or hardware setup. The test setup also allows to test different real world conditions and scenarios, in order to test the performance of various components. Due to the simplicity of the hardware setup, real situations were also tested by mounting the sensor array at the rear end of a vehicle, without any additional mounts or clamps. A simple graphical interface relays information to the user about the vehicle's surrounding in a simple, efficient, and timely manner. When the test system is installed on a vehicle, it does

not add any additional bulk to the vehicle or require high operational power. The system is a cost-efficient solution to other sensor types like LiDAR.

5. DATA COLLECTION

Data collection is necessary in order to validate the system's performance, response time, and overall efficiency. This is a procedural approach, that allows the system to be tested in various conditions. This section provides an in-depth review of different types of data collected. This section also provides an overview of different data processing techniques used for each step within the algorithms. These steps are essential for verifying the integrity of the system's test setup. Additional tools are also used for verifying the results provided by the system. These include distance measurement instruments and data blocks produced by other sensor types for comparison of accuracy.

The purpose of data collection is to test hypothetical situations that the system can encounter in real-world applications. This allows the system to be modified further in order to increase the overall tracking and detection accuracy of the system. During data collection, additions changes to the algorithm were made for noise removal and the overall sensitivity of the system. Some extra variables were also created to store new data which allows the processing of data to occur at a faster rate. The data collected can be transformed into different forms, and the system's performance under different conditions can be identified [43]. Based on these results, extra physical or software changes can be implemented. Analyzing the data after collection helps to understand certain points of failure of the system. This allows creating a more robust system for the long run.

The majority of data collection depends on different test scenarios. These help to understand how the hardware and software components perform under such conditions, and their performance can be improved further. In our case, the system was tested for noisy conditions, windy conditions, and heavy rain. This helps to establish the ideal conditions for the setup and implement changes to increase the accuracy in the non-ideal scenarios. The whole point of collecting multiple sets of data is to reduce the likelihood of errors in real-world scenarios. Sections in this chapter further explore some different experimental procedures followed to collect data, and the types of tools used. They also provide an in-depth overview on three possible cases in which the host and the sound source can exist in real-world scenarios. The final sections of this chapter provide an explanation on the different models and techniques

used to process the data collected. The last section provides a comprehensive evaluation of the different types of data collected and the conclusions that can be derived from it.

5.1 Data Collection Procedure

The proper procedure needs to be followed in order to obtain good quality data. For initial testing and data collection, the experimental setup can be used for testing the system's response in various conditions. In order to simulate different types of sound sources, a wireless speaker was used. A single sine wave tone can be played to demonstrate a sound emitted by a source. Different variations of the monotonic tones can be used by varying the frequency of the sine wave. Varying the positions of the speaker can also simulate a moving sound source. In order to test the performance with noise, a different wireless speaker was used to play white noise, pink noise, or sounds of nature.

To test the system and verify the accuracy of the data, measurement tools, and a test setup were used (Figure 5.1). As sound sources were placed at different positions are the distance was computed by the algorithm, therefore the results were verified using distance measurement tools. In order to simulate windy conditions, a fan was placed close to the microphone array that creates the same effect as wind blowing into the microphones. For blocking the wind, a plastic shield was placed in front of the sensor array that still allows the sound was through it. The connector on the sensor array uses a proprietary port, therefore a universal serial bus adapter was used to connect to the computer. The sensor also carries a power requirement of 12V DC for operation, which is indirectly supplied through the adapter. This power was supplied using the wall outlet, but when the setup is installed on a host vehicle, a portable battery or the auxiliary port of the vehicle can be used.

Simulating different scenarios and situations are the most important factors of testing. This allows the system's response to be evaluated under these conditions. Further, the verification of the system can be performed to check if additional changes to the hardware or software components are needed. In order to simulate the scenarios, the three possible cases for the states of the host and the sound source were evaluated. The three cases for the states of the host vehicle and the sound source are as follows:

1. Static source

2. Static host and moving source

3. Moving host and source

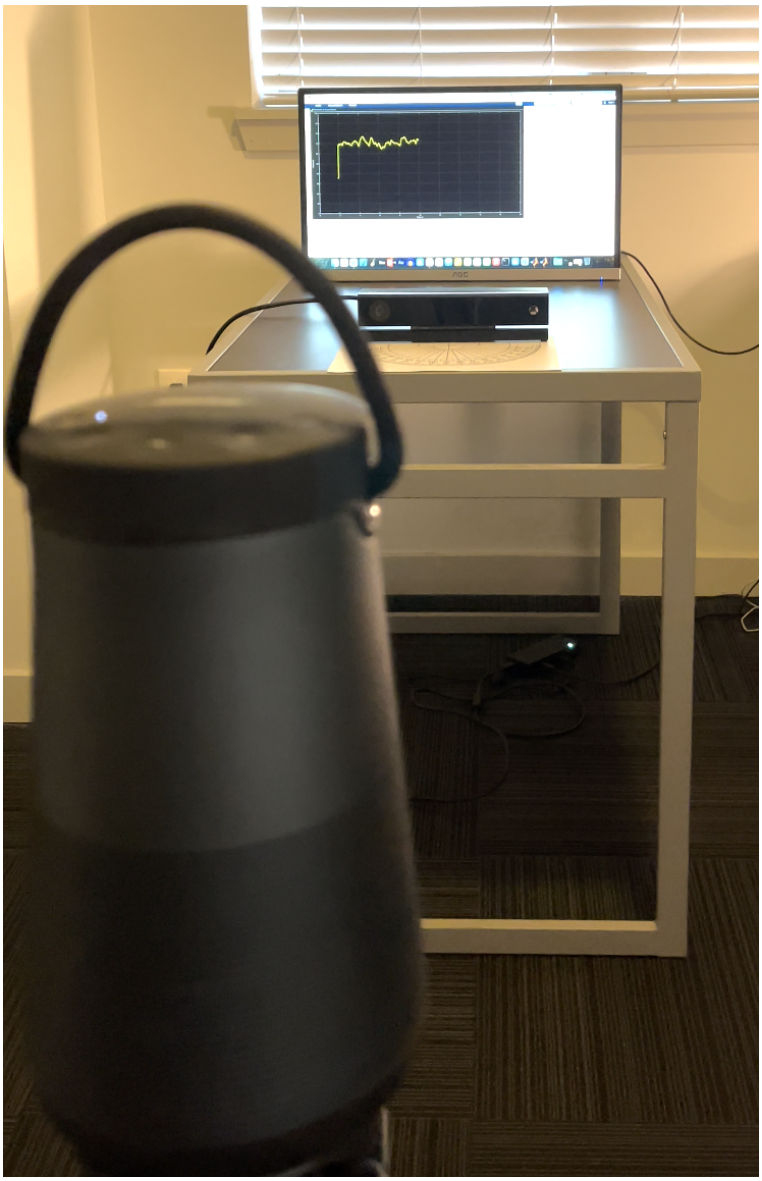

Figure 5.1. Data Collection Test Setup

5.1.1 Static Source

The first scenario is a case in which the sound source is stationary. In this case the host can be moving or at a static position as well. A situation like such can occur at a traffic light or in a parking lot. This can also be a case where an emergency vehicle is parked on the side of the road. Similar applications of such scenarios can be used for blind-spot detection. With a static source, the task of detection is relatively simple. The only variable that needs to be removed is additional environmental noise.

The data collection procedure for the case with a static source is as follows. Sounds generated by different sound sources are generally produced by the vibrations caused inside the vehicle's engine [38]. In order to simulate the presence of a static sound source, a monotonic sine wave was played using a wireless speaker. The speaker was placed at a fixed distance of two meters from the microphone array. This procedure was repeated with the monotonic tone being replaced by an engine sound, which simulates a static vehicle with the engine running. To test the system's performance with noise, an additional wireless speaker was introduced to the test condition. This speaker played the ambiance of a busy city with sounds of people walking, car horns, and various other acoustic elements.

The collected data shows that the entire system setup depends on the sound waves being captured by the microphone array. Therefore, if there are objects like parked cars or other still objects, then detecting them would be a challenge. As still objects do not produce any sounds, which makes it impossible for the system to detect their presence. As long as a static source generates some amount of acoustic data, then the system can detect its presence. The major advantage that the system's design carries is setup flexibility. This means that the task of detecting still objects without any sound can be achieved through sensor fusion. A vision-based sensor like a camera can detect still objects in the frame, while the acoustic array can detect moving objects that generate sounds [44].

The data collected for the case with a static sound source suggests that environmental noise can slightly affect the performance of the system, as it is unstructured data from random sources [45]. This is bound to happen, as when two sound waves interact with each other, then the varying energy levels collide with each other. This is known as the

attenuation of signals. The attenuated signal can be recovered using specific algorithms, which would increase the processing time of the current algorithms. Although, for testing purposes, relatively loud volumes of noise were used. The purpose was to test the most extreme conditions which prepare the system for adverse conditions. The types of noise vary in real-world conditions, the variations can be in terms of the amplitude of sounds of different objects [45].

The first type of sound source was using the monotonic sine wave (Figure 5.2). This type of sound generally mimics the sinusoidal vibrations produced by the movement of pistons in an engine [38]. A sound like this is easily detected by the system and the proximity estimation was found to be really accurate. The second type of source was the sound of a static car's engine. The distance was kept fixed for all test conditions at two meters. With the idle engine sound, the detection and proximity modeling was still really close to the actual state of the source (Figure 5.3). Testing with the third type of source was engine sound with a city landscape noise. In this scenario, the tracking results after noise removal are close to the results with the second type of sound source, with only a few variations in the blue line (Figure 5.4).

In general, the results for all three types of sound sources were found to be good approximations of the actual distance. This means that the results computed by the system are accurate in a scenario with a static sound source. Although the approximations of distance were modeled accurately, these computations were not consistent for all three sound sources. Since the microphones in the array are arranged close to each other, therefore computing the variation of intensity levels across each microphone is a bit tricky. This problem would be solved by placing microphones around the vehicle, with each evenly spaced from the other. One of the main observations that can be made from the data is that when the sound source gets more complex, then the modeled values for the approximated distance see small variations and can be observed in Figure 5.4. This is because of the fact that a noise removal block is incorporated within the algorithm. With signal processing, the original data is bound to lose some amount of data as a result of processing.

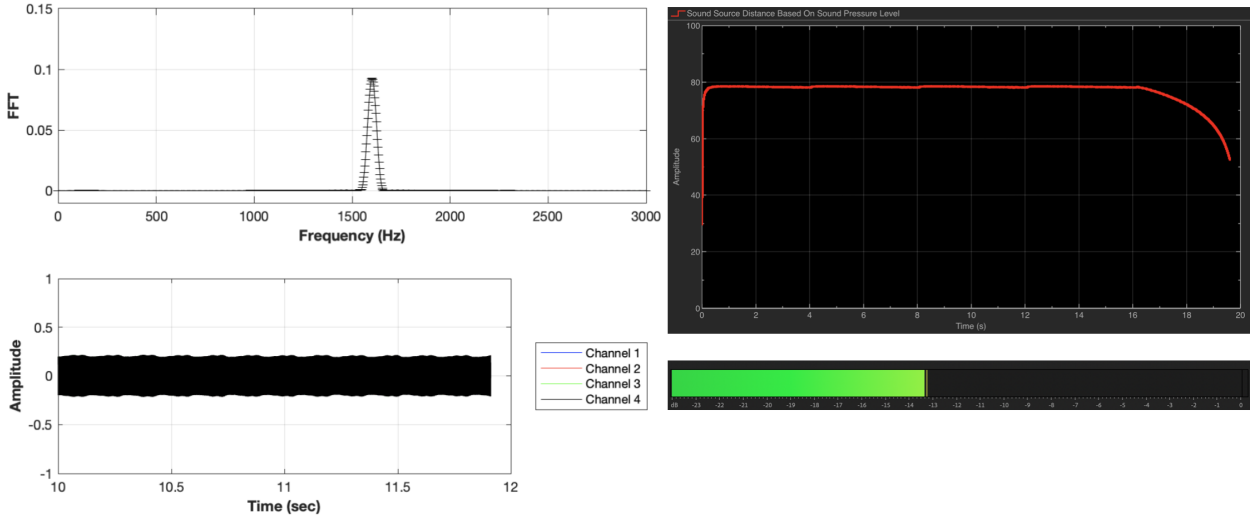

Figure 5.2. Static Monotonic Sound Source

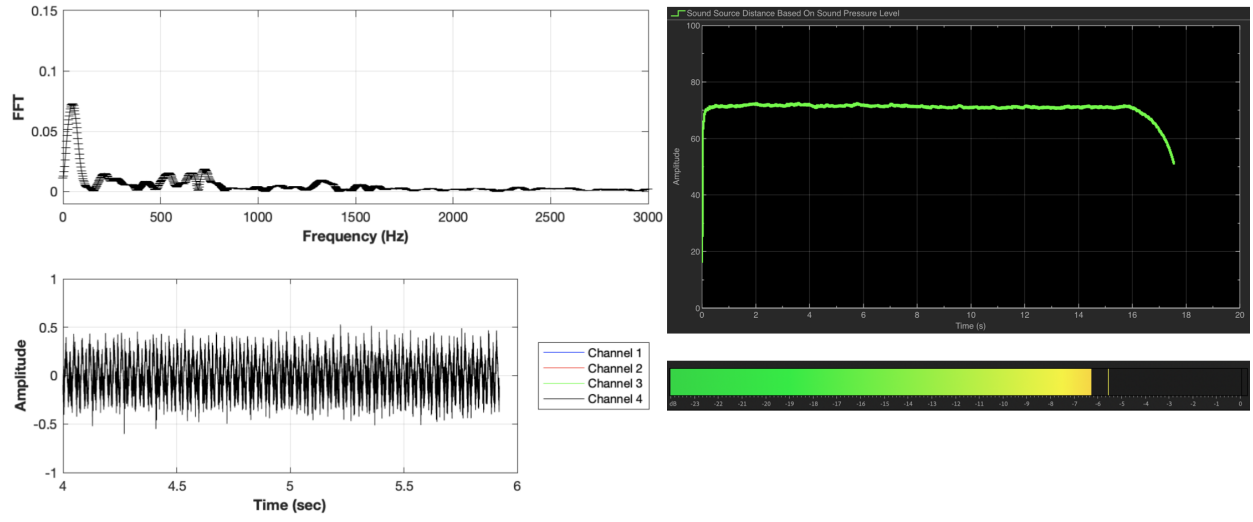

Figure 5.3. Static Car Engine Sound As A Source

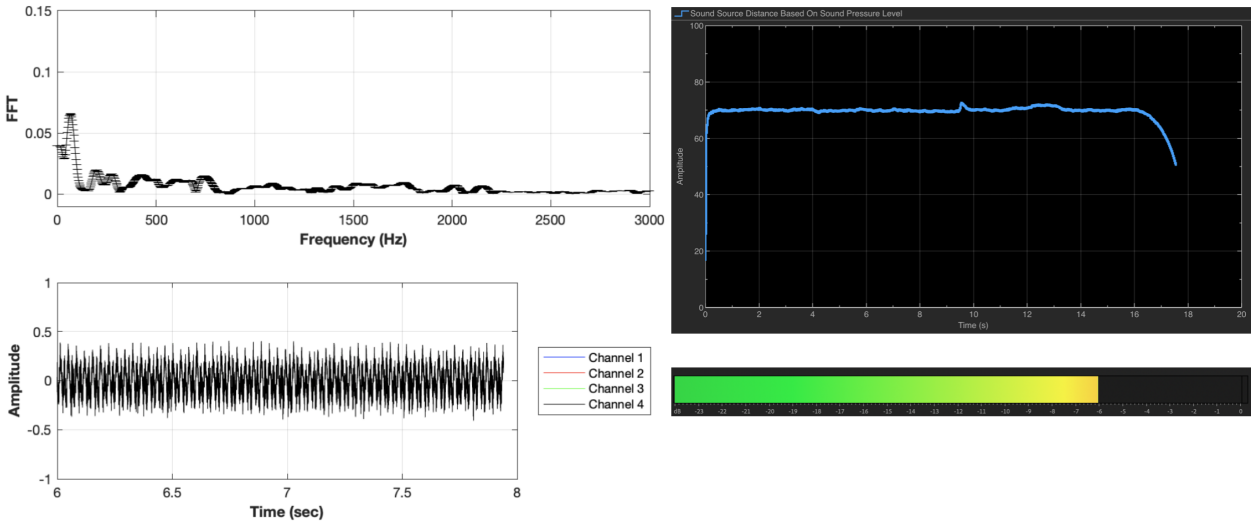

Figure 5.4. Static Car Engine Sound With City Noise As A Sound Source

5.1.2 Static Host and Moving Source

The second scenario is the case with a static host vehicle, with moving sound sources in its surroundings. An instance like such can occur when a host vehicle is at a red light and the vehicle in the opposite lane is passing by or the host vehicle is pulled over at the side of the road. This can even occur in a parking lot when the host is trying to reverse the vehicle, but there are moving vehicles present. In such a case, the system would also work well for blind-spot detection. A condition like this also presents a situation in which multiple sound sources can be present around the host vehicle. In such cases, the system uses a combination of three to four adjacent microphones, in order to localize each sound source [34]. But for testing only a limited number of wireless speakers were available and only a four microphone array was used. Therefore, computing multiple sound sources with just four microphones would not provide accurate direction estimations. Although, an eight microphone array can do that relatively well.

The procedure for this scenario is very similar to the first scenario, but with slight modifications. First, a monotonic sound source was used to play a fixed pitched sound. In order to simulate a moving sound source, the wireless speaker was being moved around the Kinect sensor. The motion of the movement was similar to a vehicle moving on a road,

which was in a fixed direction with slight movements to the left or right. The initial starting distance was again used as two meters. This process was again repeated with the monotonic tone being replaced by a real-world sample of a car passing by, this provides a more accurate representation of a moving vehicle. Similar to the first condition, the performance was tested again with environmental noise. Another wireless speaker was used to play the sounds of a city landscape with sounds like traffic, car horns, and people walking. The speaker playing environmental noise was placed at a fixed position and was not moving like the other speaker.

The sound waves reaching the microphone array are all collectively considered as data, which is further processed for noise removal and proximity approximation. In this situation, the moving objects are easy to detect, as they generate sound while moving. If there are still objects present in the surroundings, then they can be detected through sensor fusion. Sensors like RADAR or vision-based sensors like cameras can detect stationary objects and work in parallel with the microphone array. Although a RADAR sensor is considered an acoustic sensor due to the emission of high-frequency beams. But this sensor can still work without causing any interference with the microphone array. This is because the frequency of the ultrasonic waves is way beyond the audible range or the detection sensitivity range of a microphone.

The first type of moving sound source tested was the monotonic sound. This sound is generally easy to detect as it doesn't contain any complex elements in the sound waves. The results for the changing distance of the sound source are also modeled well by the system (Figure 5.5). With the second sound source, that is with the sample of a car passing by, the approximations are still within a good range. But since the sound contains many harmonics when the source moves, the system takes a bit longer to process the data (Figure 5.6). In the third testing scenario with environmental noise added to the car engine sound, the distance approximations of the moving were similar to the first sound source (Figure 5.7). Noise removal adds additional time delay to the processing of the results when complex acoustic elements are present in the data.

The results for all three sound sources were found to be within relatively close range to the actual values. The system does experience a slight processing delay when the sound sources are moving when compared to the scenario in which the source is static. As the case with a

static sound source, the model of distance approximations was similar to the actual values, but the results were not consistent throughout all three types of acoustic conditions. The condition with the monotonic sound had approximations that was an exact representation of the actual changing distance values. But for the engine sound and the condition with noise, the results were not as accurate as of the first case and contain slight deviations (Figure 5.7). This is due to the fact that the data is more complex compared to the first case. The system also compares the intensity levels between adjacent microphones, which requires additional processing time for complex data. In general, the system can detect, track and provide proximity approximation that can be used to detect nearby objects with good precision. The reliability and detection accuracy of the system can be increased through sensor fusion. Another method that would lower the processing time would be to run the noise removal block on a separate portable computer, and push the processed data to the secondary portable computer for further analysis.

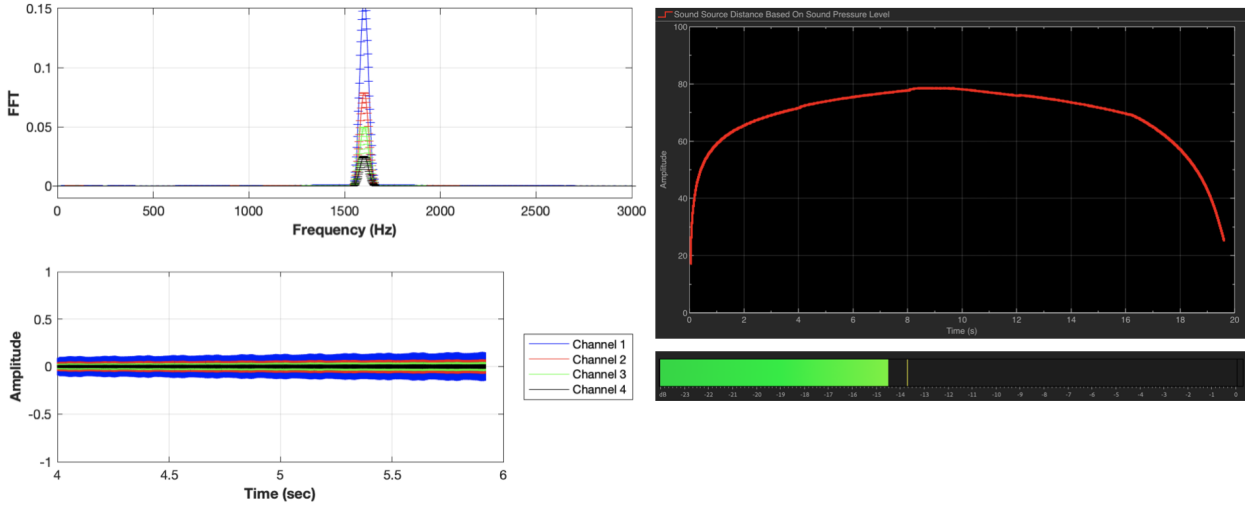

Figure 5.5. Moving Monotonic Sound Source & Static Host

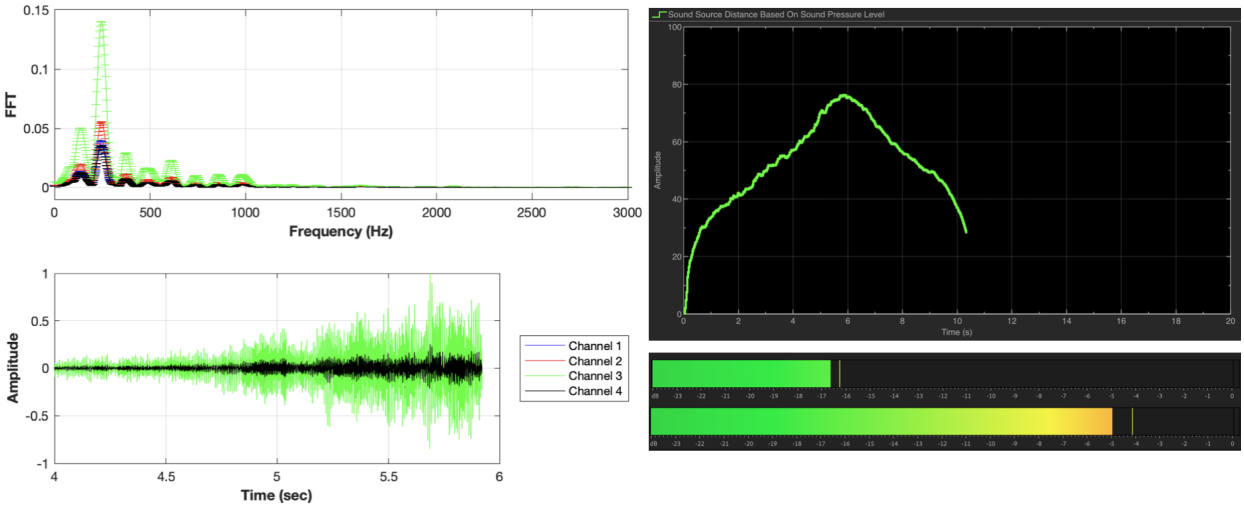

Figure 5.6. Moving Car Sound As A Source & Static Host

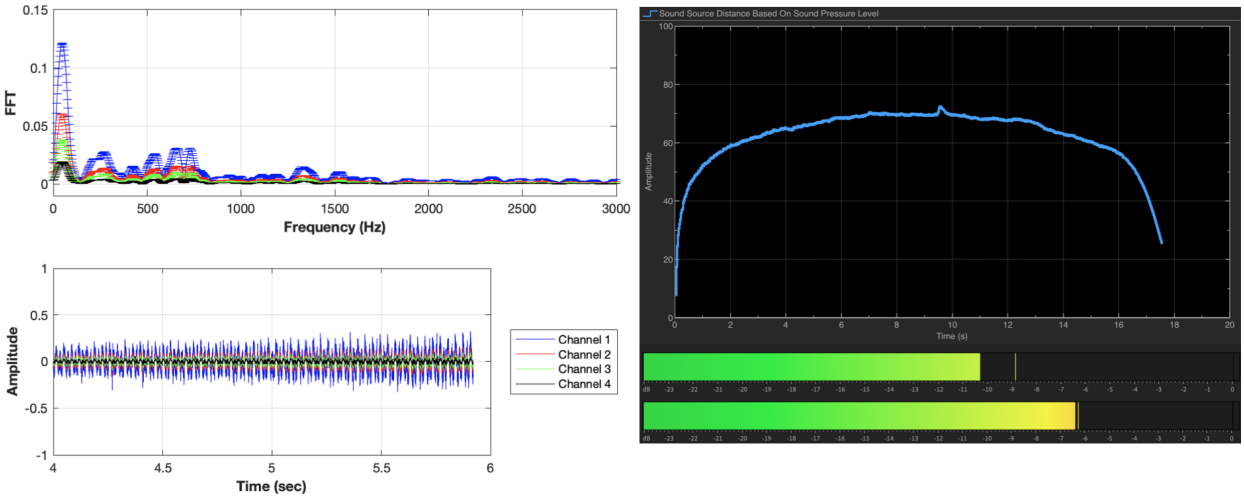

Figure 5.7. Moving Car Sound With City Noise As A Source & Static Host

5.1.3 Moving Host and Source

In the third test scenario, both the host and the source are moving. This situation is most commonly encountered while driving on the road. A situation like such occurs most of the time where the host is surrounded by other moving sound sources. The processing power need is highest for this scenario, as the system tries to capture and analyze the data in real-time. In this condition, noise removal is necessary for eliminating low amplitude noise due to the wind. Localization of different sound sources is performed by comparing the intensity levels of adjacent microphone pairs. This helps to localize the DOA of sounds from different sources. Testing multiple sound sources is a challenging task, as it requires multiple wireless speakers in order to simulate multiple sound sources. Another limitation was that only a four microphone array was used to provide a proof of concept, therefore an eight microphone array would provide better results. An array with evenly spaced microphones covers the entire acoustic field and can provide better DOA accuracy, compared to a four microphone array.

The testing procedure for the scenario with moving host and the sound source is as follows. Similar to the first two testing conditions, first the monotonic sound source is tested. For this test scenario, both the microphone array and the wireless speaker are moving. Different variations were tried to simulate different possibilities of motion of vehicles. For this scenario, the initial starting distance was not needed as the positions of both the microphone array and the sound source keep changing over time. Any arbitrary starting point for both the array and the source can be selected. This setting was repeated again but this time with the car engine sounds, which provides a better representation of the sounds on a road. Then finally, the engine sound was tested with environmental sounds like traffic noise, which was played through a different wireless speaker. The speaker playing the noise was kept at a fixed position but the volume of the noise was altered over time. This helps create a more natural representation of noise sources moving within the host vehicle's surroundings.

All the acoustic data captured by the microphone array goes through the pre-processing and noise removal blocks. Similar to the second condition, tracking moving objects is not a challenging task, as they generate sound while moving due to the vibrations that arise from

the engine [38]. Sensor fusion can be used to detect still objects like emergency vehicles that are parked along the side of the road. A two-stage approach to processing the data would work in a more time-efficient manner. The current test setup relies on only one computer for processing and analysis, which takes some amount to process the previous sets of data. Running the noise removal block on a separate portable computer will help in decreasing the amount of processing time needed, as the task of analyzing the data can be performed in parallel to each other.

First, the monotonic sound source was tested by itself. The sound being relatively simple is easy to detect and track. Results for this sound source are within a close range of changing distance values and depict the actual motion of the sound source (Figure 5.8). For the second sound source, which is the car engine sound, the results were also found to be within a good range as they depict the actual values of varying distance due to the motion (Figure 5.9). The third sound source was the engine sound with the environmental noise playing through a wireless speaker. In this case, the proximity estimation was good but wasn't as accurate as with the first two cases (Figure 5.10). The blue line in Figure 5.10 is not as smooth in the third segment due to the loss of data during processing. This is because the number of microphones available in the array is limited to four. Which cannot provide an ideal representation of the entire acoustic field around the host. In an ideal situation, an eight microphone array would solve this problem. An additional case was also considered in which both the host and the source are moving at the same speed. For this scenario, all three cases provided accurate estimations of distance. When two objects are moving at the same velocity, then the distance between them is constant. This is generally easy to process as the sets of data, in this case, are not complex.

The results computed by the algorithms model the distance values for all the test cases and were able to provide an accurate representation of the sound source's varying motion in real-time. The only observation made was the processing delay for the DOA algorithm with moving sound sources. This was caused by a limited number of microphones available in the array module. The delay is expected, due to the fact that the data sets get more complex as the source and the host start moving. Separating the hardware needed for processing will also help in reducing the processing time needed. Running the noise removal block on a

separate computer, while the proximity approximation block can run in parallel separately. The localization and mapping in terms of distance can be more accurate if sensor fusion is used. Vision-based sensors or RADAR can help to detect still objects with precision, while the microphone array can localize and map the moving objects around the host vehicle.

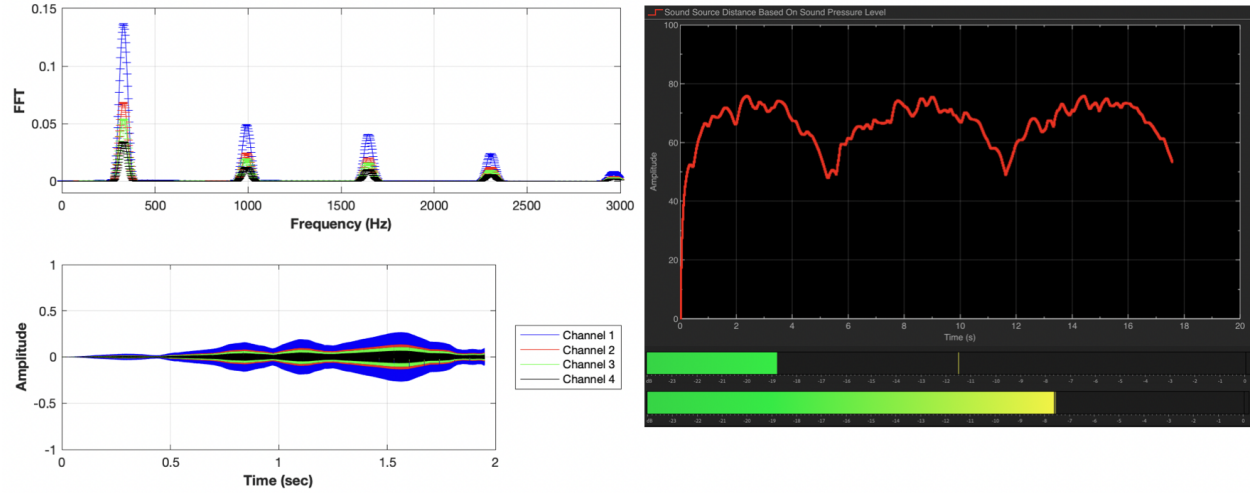

Figure 5.8. Moving Monotonic Sound Source & Host

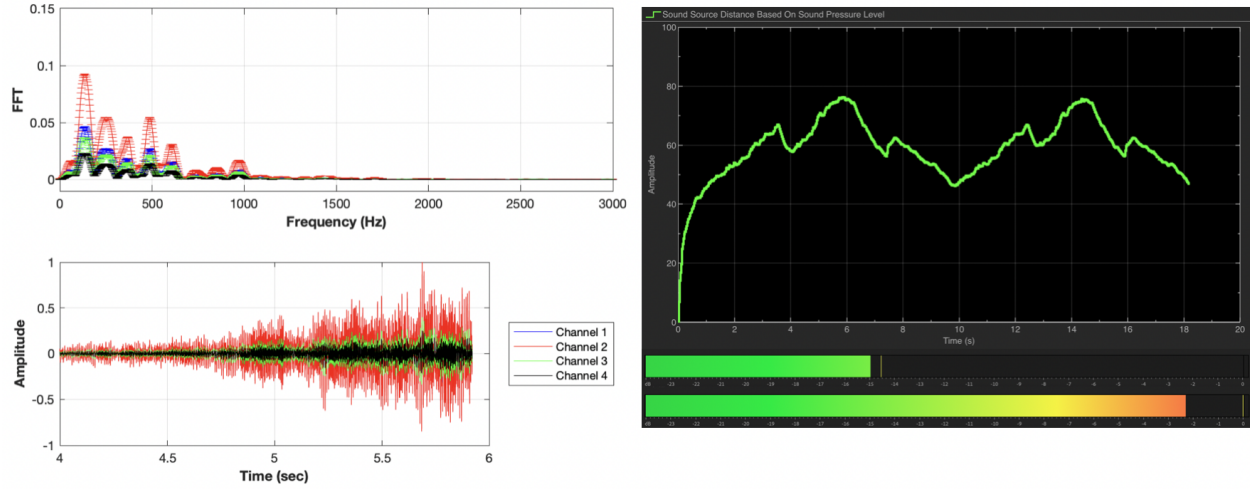

Figure 5.9. Moving Car Sound As A Source & Moving Host

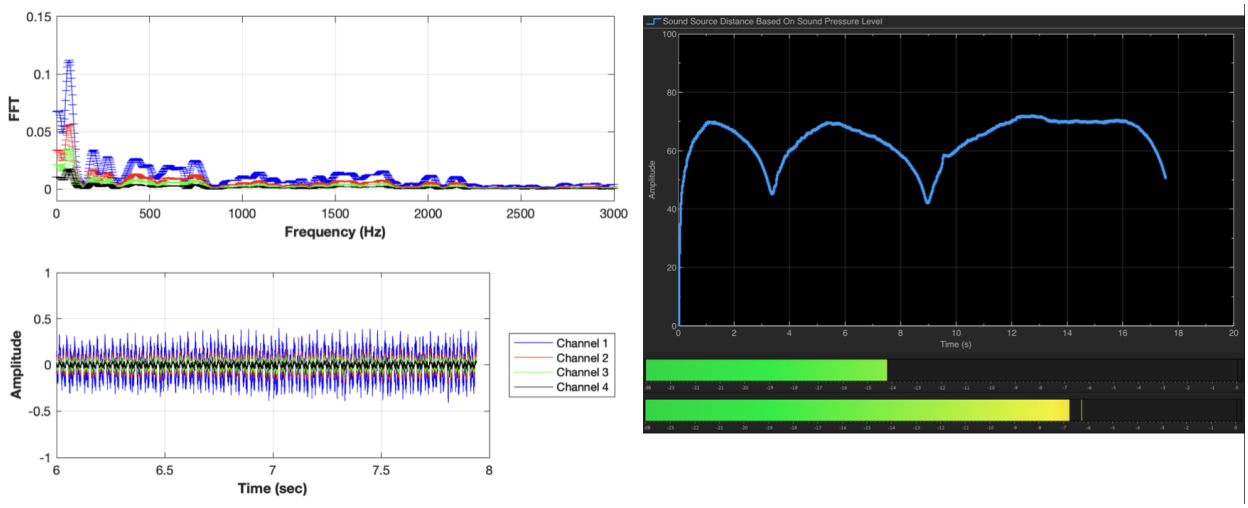

Figure 5.10. Moving Car Sound With City Noise As A Source & Moving Host

5.2 Pre-processed Data

The pre-processing of data is needed before it can be analyzed further and important elements can be extracted from it. The captured data is considered raw in terms of the amount of information it contains. Raw data contains a lot of unwanted elements that increase the file size, thus creating storage issues. Larger sets of data also take much longer to process, and require higher memory for buffer storage. The process of pre-processing data is also important for the removal of redundant elements. Some sets of data can also contain certain artifacts that occur during the capturing or storing process. This can create irregularities in the analysis and estimation results concluded from the data. Therefore, in order to avoid this, a filtering block is needed. This filtering block is executed after the data has been pre-processed by the algorithm. The framework for the filter blocks is created using MATLAB's *filterDesigner* (Figure 5.11 & Figure 5.12). The parameters for the cutoff frequencies can be controlled within the code of the main algorithm.

The raw data captured has redundant information that would slow down the computation process if it is not eliminated. This redundant information can be environmental noise, vibrations from other sources, or even artifacts that arise due to digital sampling of analog information [38]. One example of such a case is using the DOA and proximity approximation system in rainy conditions. The sounds of other vehicles in rainy conditions would get

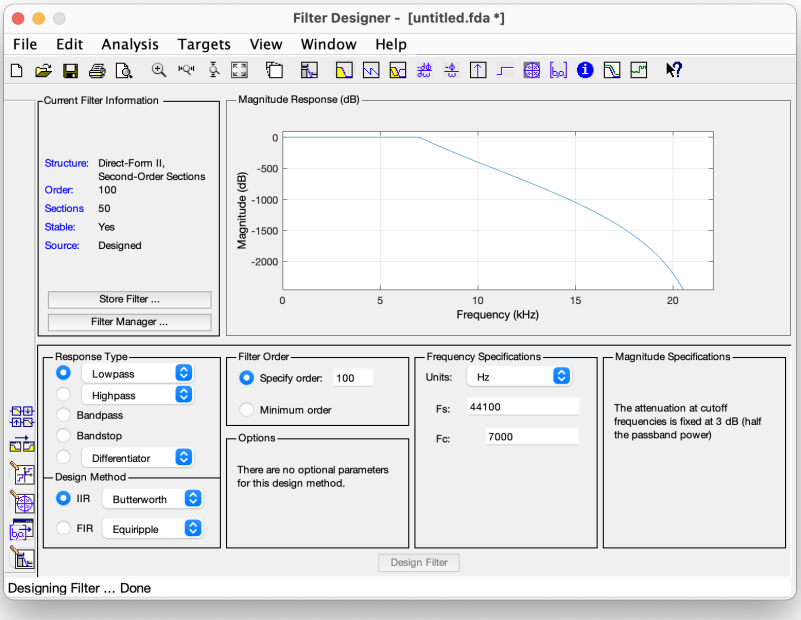

Figure 5.11. Low Pass Filter Framework Design

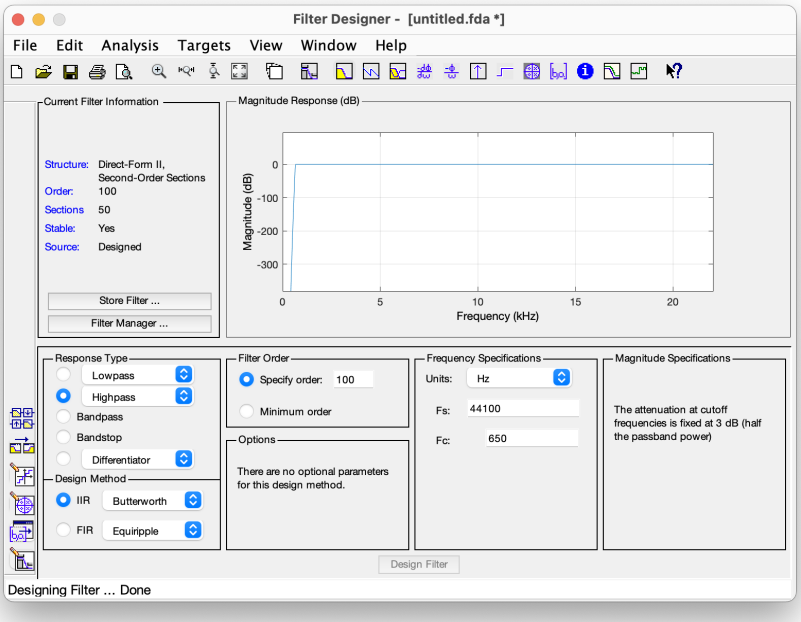

Figure 5.12. High Pass Filter Framework Design

masked by the sound of the water droplets falling on different surfaces. As mentioned in the earlier sections, the characteristics of rain are similar to pink noise. Therefore, a majority of noise originating from rain can be eliminated from the captured samples by implementing a combination of a high pass filter and a low pass filter [46]. This removes the unwanted noise of the rain from the samples and focuses on the sounds of the vehicles. This can be observed from Figure 5.13 & Figure 5.14, where the sound of rain has been filtered out from the noisy signal.

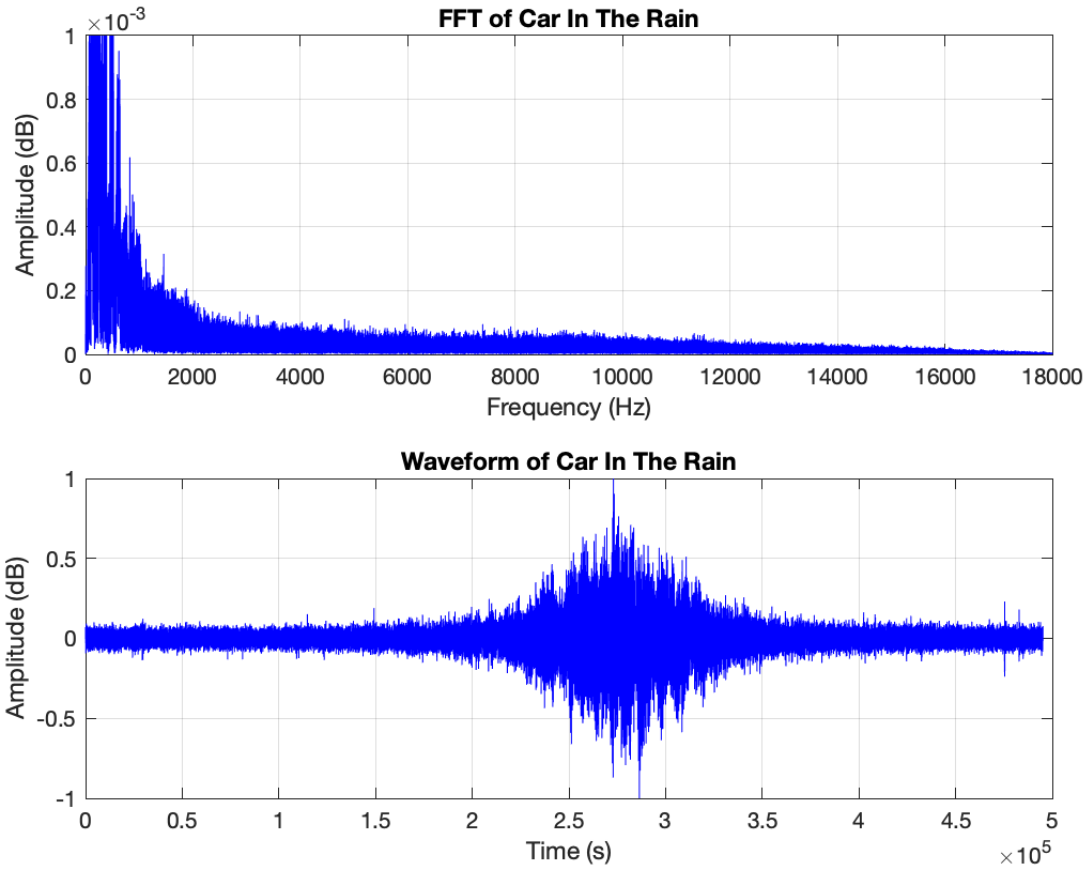

Figure 5.13. Sound Of A Moving Vehicle In The Rain

Noise removal or pre-processing of captured data ensures that the size of the data is small and only contains relevant information for the processing algorithms. This step helps to ensure that the processing of data occurs in near real-time. Complex amounts of data take longer to process, and the results wouldn't be relevant if the events have already occurred.

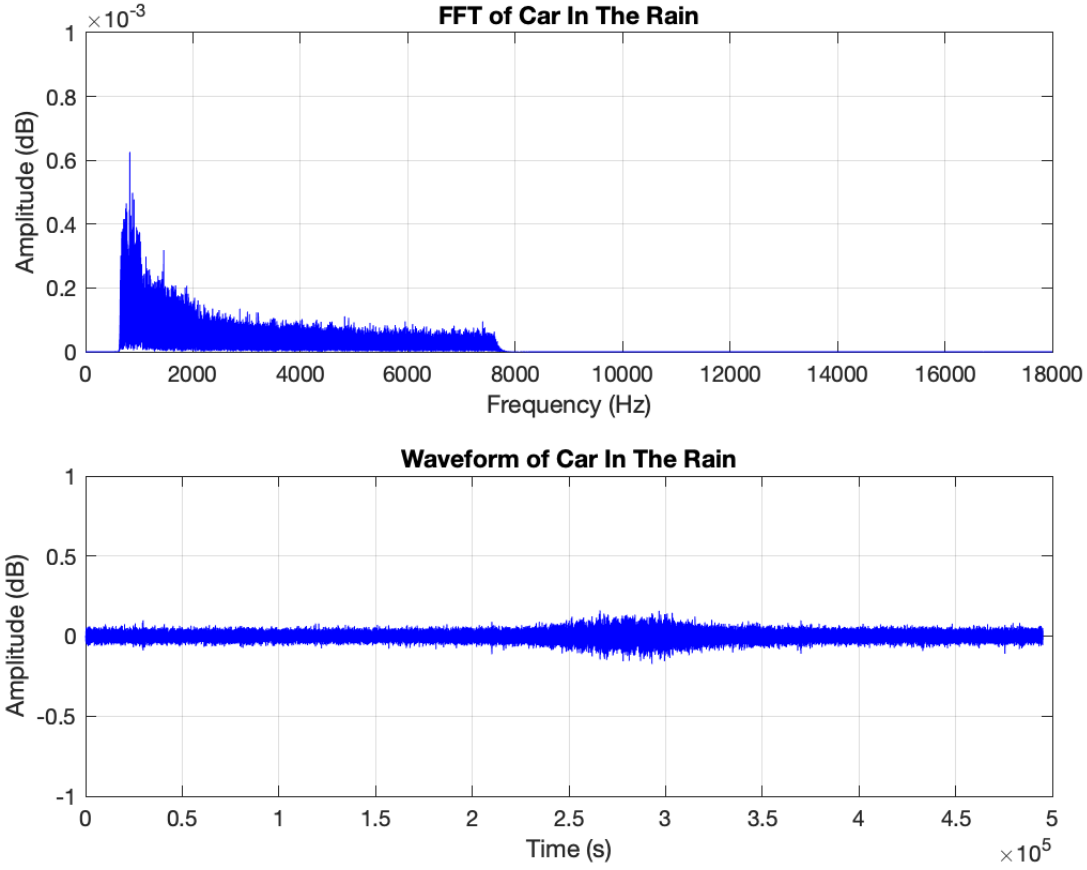

Figure 5.14. Filtered Sound Of A Moving Vehicle In The Rain

Pre-processing stage lowers the size of the data by eliminating redundant information from the audio samples. The information eliminated usually contains unwanted environmental noise, vibrations from the engine, or noise from wind or rain.

Noise is the environment exist in various forms. It can be sounds of nature like wind, rain, or leaves rustling, these are clubbed as pink noise [33]. Some other sources of noise can be due to the vibrations from the engine or the wheels of the host vehicle, which cannot be classified into one specific form, but share similar characteristics to Gaussian noise [38]. The noise is separated from the sounds of vehicles, by identifying the periodic transients of sound. Both the external noise and the vehicle sounds have different transients. The noise removal step separates these transients from each other, and the noise is eliminated.

Noise removal block is part of the pre-processing process. This ensures that any unwanted data is removed from the data file before it is pushed for further analysis. Smaller data size ensures that the data flow is smooth and efficient. The size of captured data varies based on the scenario. A situation with static sources produces smaller sets of data, while scenarios with a lot of movement have larger sets of data. This is because if there is an activity in the host's surroundings, then the acoustic data is more complex. Generally, complex sets of data take longer to process, therefore pre-processing the data reduces the data size significantly.

After the noise removal process is completed, the data is pre-processed further. Following the noise removal process, the size of the data is smaller than the original raw data. This makes it easier to process it in near real-time. The next step is to transform the data into a type that can be analyzed by the algorithms. The most commonly used method used for the storage and playback of audio data is amplitude vs time representation. Similarly, the audio data is stored in a variable, where the data is represented as an array. This allows the data set to be accessed by simply calling the variable name within the code. It also makes it easier to perform transformations on the data in a variable format, without altering the original data. Variables also make the process of moving the data sets across different sub-algorithms relatively faster.

After the noise removal step, the data is transformed into a variable format, which is then pushed for further processing. The next step to push the data further for filtering. This step is a post noise removal procedure and ensures that the data is an accurate representation of

the original acoustic signature of the sound source. The pre-processing of data requires the most amount of time compared to other sub-algorithms. Therefore, this process cab made faster by separating the hardware components. This means that the pre-processing block can execute and process the data set on a separate computer, and the rest of the algorithms can execute on a different computer. By using such an approach, the algorithms following the pre-processing block wouldn't have to rely on the timely export of data. This would help lower the overall processing time taken by the system.

5.3 Filtered Data

Filtering is the next step following the pre-processing of raw data. This step is essential, as it removes any high-energy elements from the pre-processed data. These are high amplitude elements that are present in the lower or higher frequency range, and they overpower mid-range frequency elements. Therefore, if their amplitudes are too high, then they are removed using low and high pass filters. This filtering process only executes if these elements are present in the pre-processed data. Filtering makes sure that the data is smooth and does not contain any arbitrary values [40]. If any unwanted information is present in the data, then the results computed by the proximity approximation algorithm would not be accurate. For example, a noisy signal was created, and the spectral image was generated [47]. The horizontal lines in the spectral image are the high-energy, unwanted noise elements (Figure 5.15). The FFT and waveform of the noisy signal is also shown in Figure 5.16.

The first step in the filtering process is to create a new variable with a copy of the pre-processed acoustic data. This helps to speed up the processing part, as different variables with the same data can be pushed for further analysis. It also ensures that the original data is left untouched if the results computed by some sub-algorithms are not accurate. After a copy of the pre-processed data is created, the noisy data is converted into the frequency vs amplitude format (Figure 5.16). This helps to recognize the presence of high amplitude elements along the frequency spectrum. Filtering of these unwanted elements is easier to implement if the data is in the frequency domain. Therefore, an FFT is performed on the pre-processed data. If any data points with amplitudes above the zero decibel threshold are

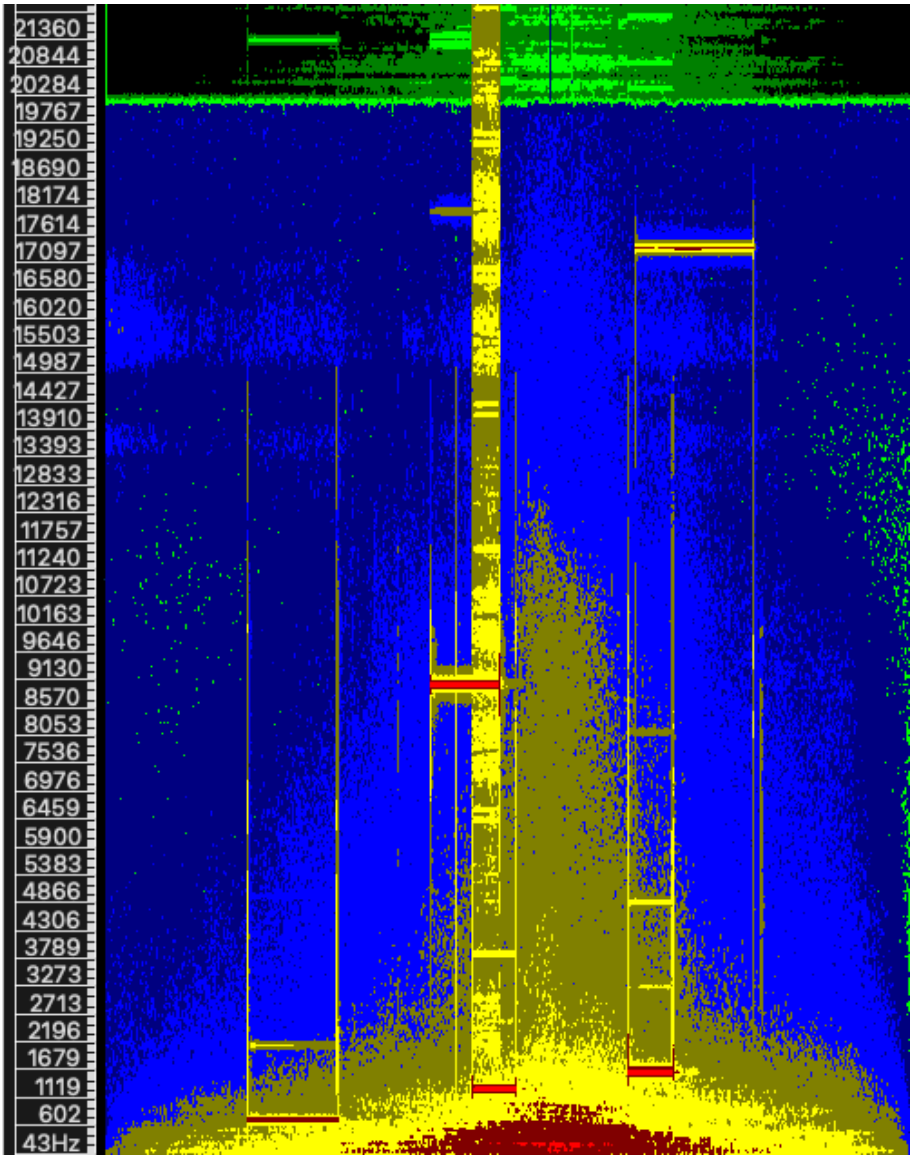

Figure 5.15. Spectral Image Of A Noisy Signal

present, then an appropriate filter is applied in order to remove those elements (Figure 5.17) [32].

The size of the data is still large after the filtering process, as the data contains information from four audio channels. The next step is common for both the DOA and the proximity approximation algorithms. This is to find the loudest audio channel, which directly corresponds to the position of the sound source is closer to the microphone returning the highest intensity value. The DOA algorithm compares different microphone pairs with each other,

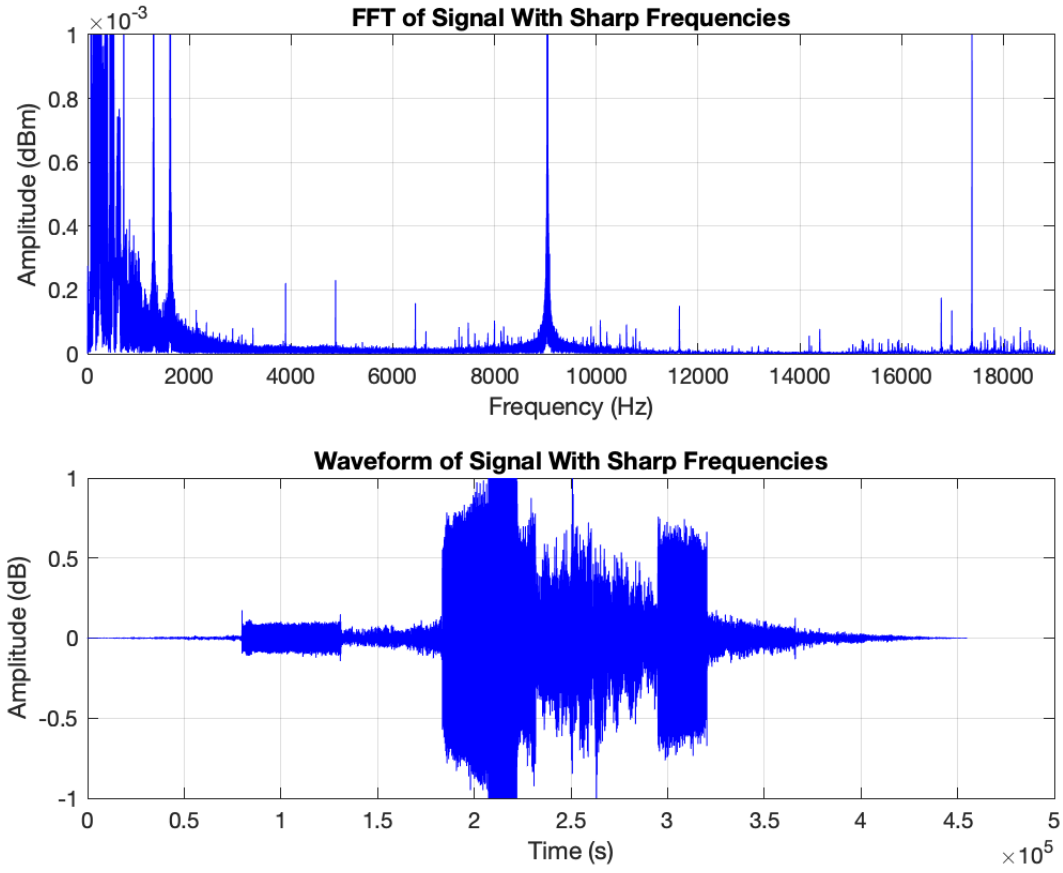

Figure 5.16. An Unfiltered Signal With Noise

in order to present an accurate DOA of the sound. While for proximity approximation, only the channel with the highest intensity level is needed. After this step, the data is small in size as it only contains information about one audio channel rather than four. When the system is tested with an eight microphone array, the processing time will get cut down to about one-eighth of the initial processing time.

After the appropriate filters have been applied to the pre-processed data, the next step is eliminating audio channels. This step of the filtering algorithms is similar for both DOA and proximity approximation algorithms. The DOA algorithm requires all four audio channels for the comparison of intensity levels. The main purpose of the DOA algorithm is to precisely estimate the direction of the sound source. The degree of presence is computed in radians by using the intensity values from the two loudest channels. While the distance

approximation only considers the loudest audio channel out of the four available. The sound source is localized by the DOA algorithm in terms of its location around the host vehicle. Whereas, the proximity approximation algorithm is responsible for mapping the sound source by computing its distance from the host vehicle.

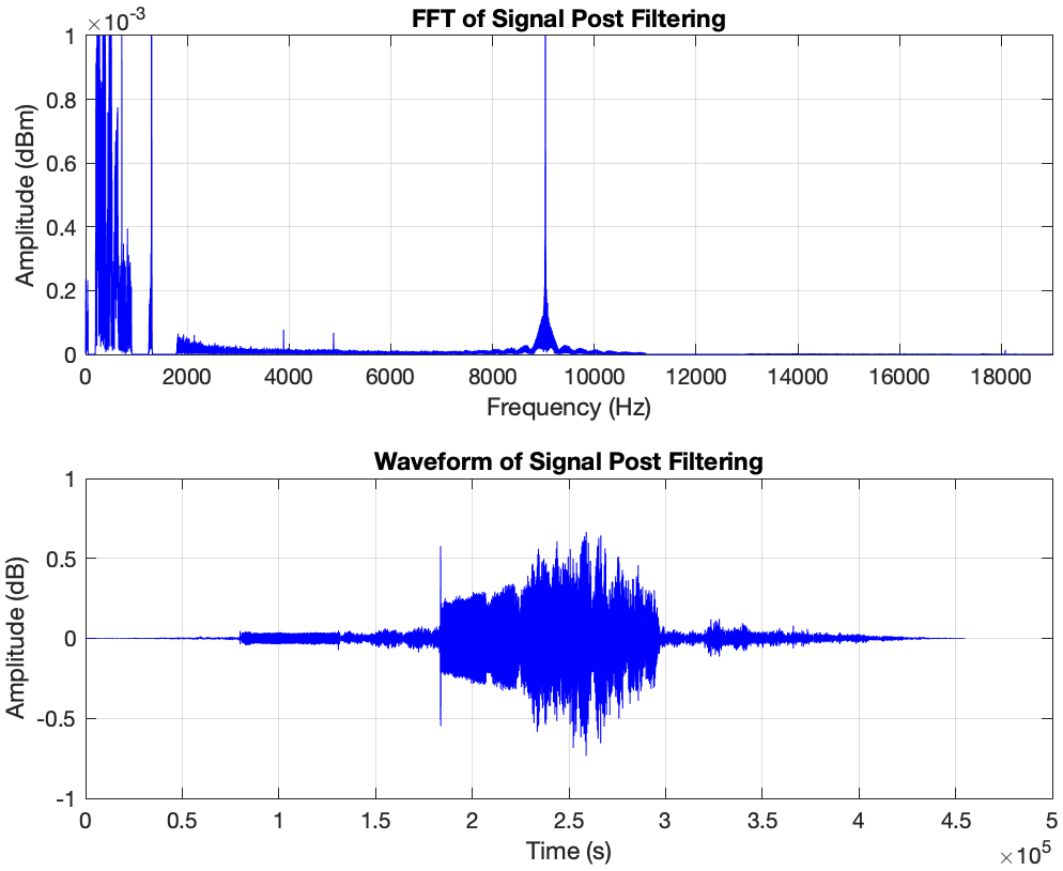

Figure 5.17. Noisy Signal Post Filtering

In general, the filtering of data can be done before the pre-processing process. This would help remove the unwanted noise more efficiently, but as the raw data is relatively large in the beginning, therefore the whole filtering process takes a long time to compute. Another disadvantage to filtering the data, in the beginning is with regards to the accuracy of the data. As noise removal occurs before filtering, then the data passes through the filters and high-energy elements are removed from it. At this point, the data is mostly free from any unwanted noise, and the filtering algorithm can perform efficiently. But if the data filtering

process executes before the pre-processing stage, then a lot of information is lost, as the algorithm aggressively removes elements that seem to have high-energy. The main goal of the filtering block is to apply smoothness to the data, so any discrepancies are eliminated from it [40].

5.4 Post-processed Data

Once the data has been filtered, it is then pushed further for post-processing. Post-processing of audio related data in the system's design refers to editing the data in order to fit the array dimensions. At this stage, a majority of the data is in its final form. In this step, the data is evaluated to compute the distance estimate based on the sound intensity levels. The amplitudes of the data are left untouched throughout the processing procedure. As the distance estimates inversely relate to the intensity levels, therefore altering it would cause discrepancies in the results provided by the algorithm. Post-processed data is simply adjusted for being displayed through the graphical interface.

In the preliminary steps, the data goes through a noise reduction process, followed by filtering. After these steps are executed, the data contains some empty spots. The noise reduction process simply lowers the amplitudes of elements it considers as noise. Some elements that already have been reduced are usually removed by the filtering process. These empty data points need to be replaced by arbitrary data values, so they can be scaled to a set of axes. These empty data points are padded by zeros, this is done so the original data values are left untouched. After this step, the data is normalized and can be displayed graphically in a two-dimensional manner.

The DOA algorithm doesn't require intensive noise removal or filtering. The general process of normalization works relatively well for estimating the direction of the sound source. After the DOA algorithm compares the audio channel pairs, the buffers are cleared so new data sets can be accommodated. The angle of presence is computed in radians, using the two concurrent audio channels as vectors. The angle of presence computed is the arbitrary separation between the two-channel vectors. This is also dependant on the positions of the microphones within the array. The wider the separation between consecutive microphone

pairs, the better is the accuracy of estimation. Microphones that are too close to each other provide similar levels. Comparing the data collected by such microphones, does not show much difference between intensity levels.

On the other hand, the proximity approximation algorithm relies purely on accurate data. Therefore, noise reduction is an integral part of the distance approximation algorithm. Filtering is also another essential component, as it helps to generate smoother data sets. This directly helps to get accurate estimation results. The distance of the sound source is computed using the Inverse Square Law. For the calculation of distance, the intensity levels of the acoustic data are used as inputs for estimation. Similar to the DOA algorithm, the distance of separation between two consecutive microphone pairs is an important factor in determining the distance accurately. The greater the distance of separation between the microphones, the better the intensity comparison results.

After the results are computed by both the algorithms, the finalized data is pushed further to the graphics block. The results for the distance are displayed using two graphical elements. The first one is a color gradient scale, which displays higher intensities as bright red color, and the lower intensities are shown using green. Colors between these gradients represent changing positions of the sound source, as it moves closer or farther away from the host vehicles. Red color can be seen as a warning when the sound source gets too close to the host vehicle, while green can be regarded as a relatively safe distance. The second set of results are displayed using a small section next to the color gradient meter, in which the relative distance is modeled by a moving line. DOA algorithm uses a separate graphics display for displaying the direction of the sound source.

Small buffers of fixed lengths are used for displaying the results graphically. Once the buffers for the graphics receive data, the old data sets are cleared. This allows for a seamless transition of new data to be processed by the algorithms. The buffers for the graphics are also cleared once the data is displayed. An additional step was implemented, this allows the system to run without any obstructions. If the response from different blocks of the algorithm is slow, then those sets of data are skipped. This allows the system to process the next set of data, instead of delaying the results for the old set. Execution of the code is timed, and if some blocks cause a delay, then the algorithm resets itself. An implementation

like this allows the system to self-configure problems that may arise during the data capture or processing.

5.5 Explanation of Data Types

The system produces different forms of data after each processing step. Data can vary in terms of size, the information contained, or the number of audio channels. Different sequential processes make data more portable and precise for analysis. After each processing step is executed, the form of data changes in some way that is different from the original data set captured by the microphone array. Data for the amplitudes of the captured samples is left untouched throughout the data handling process, in order to preserve characteristics of the sound source. Only the form of the data changes after the processing of data is completed. For example, after noise reduction, the data is smaller in size, and after filtering it is even smaller. But after the post-processing stage, it is the smallest in size since unnecessary channels are eliminated.

The microphone array captures the initial sets of data required for analysis. The sample rate used for this case was selected as 44.1 kHz. This allows the system to get all the information about the host's acoustic field without any data loss. The captured data contains four audio channels, as each channel corresponds to the data captured by each individual microphone in the array (Figure 5.18). In an ideal eight microphone array, the data would contain eight audio channels. The captured data is stored in a lossless file with a wave file extension. Lossless data format prevents any data loss due to compression of data.

Pre-processing of data performs noise reduction if the data requires it. This process is executed on four audio channels. After the execution of this stage, the size of the data is small only if noise reduction is performed on the original data set. Whether or not the noise reduction process is performed on the original data set, it is then transformed into a variable format for easier portability. The variable is an array of the same length as the sample rate selected. For this case the sample rate was selected as 44.1 kHz, therefore the length of the variable array is 44.1 kHz. This is enough to preserve all the acoustic information without any data loss or artifacts in the sampled data.

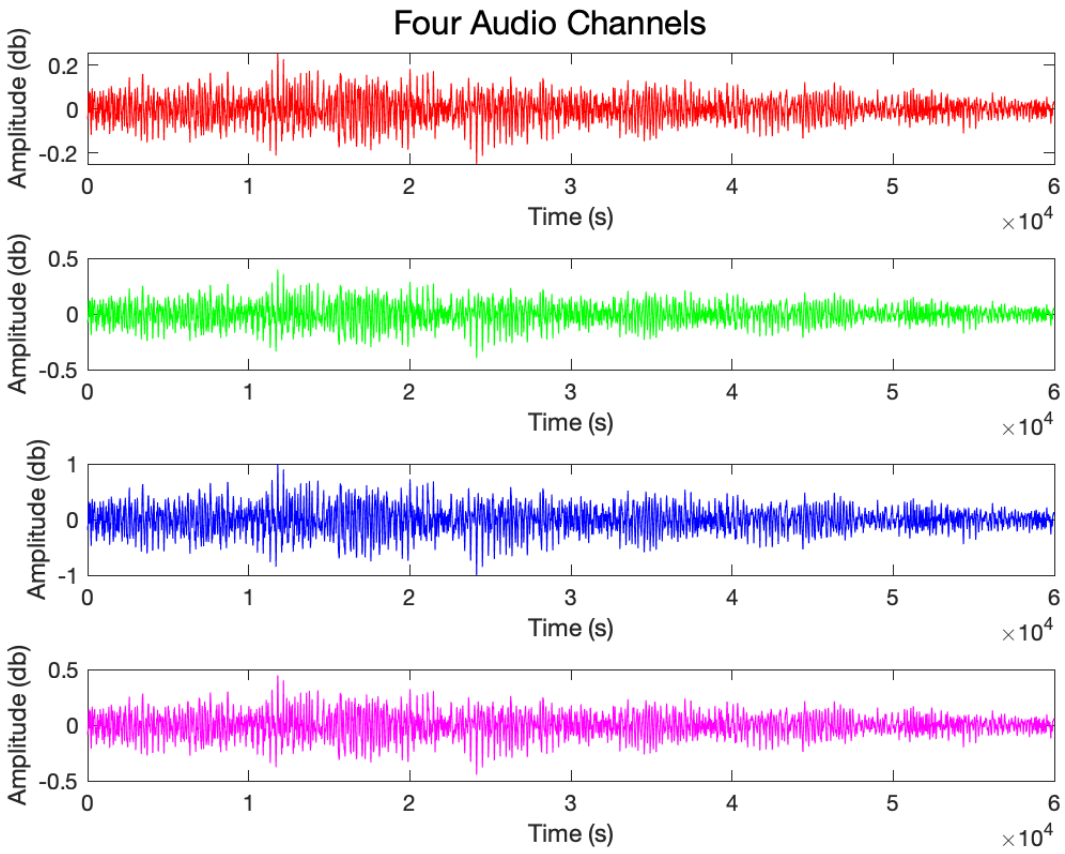

Figure 5.18. Raw Data With Four Audio Channels

The next step is filtering of data where smoother sets of data are generated as the output. During this process, any acoustic elements present with high energies or amplitudes above the zero decibel threshold are removed. The removal occurs by applying the adequate filter types as required. For this step within the filtering block, the data is transformed into the frequency domain, as it separates the existence of all frequency elements along with their amplitudes. The transform used for this purpose was Fast-Fourier Transform (FFT). A copy of the pre-processed data is created before being transformed into the frequency domain. This preserves the data from being corrupted by any possible errors that may arise from processing and transformations.

The filtered data then bifurcates into two forms for the DOA and proximity approximation algorithms. At this point, the data only contains information about the sound source

and is free from any noise. The DOA algorithm eliminates the audio channel data after computing the direction of the sound source. The DOA algorithm only pushes the computed results to the graphics block. While the distance approximation algorithm compares the audio channels, and only stores the loudest channel (Figure 5.19). This channel is then pushed for post-processing. At this stage, the data is the smallest in size and only contains the acoustic characteristics of the sound source, within the audible range of frequencies.

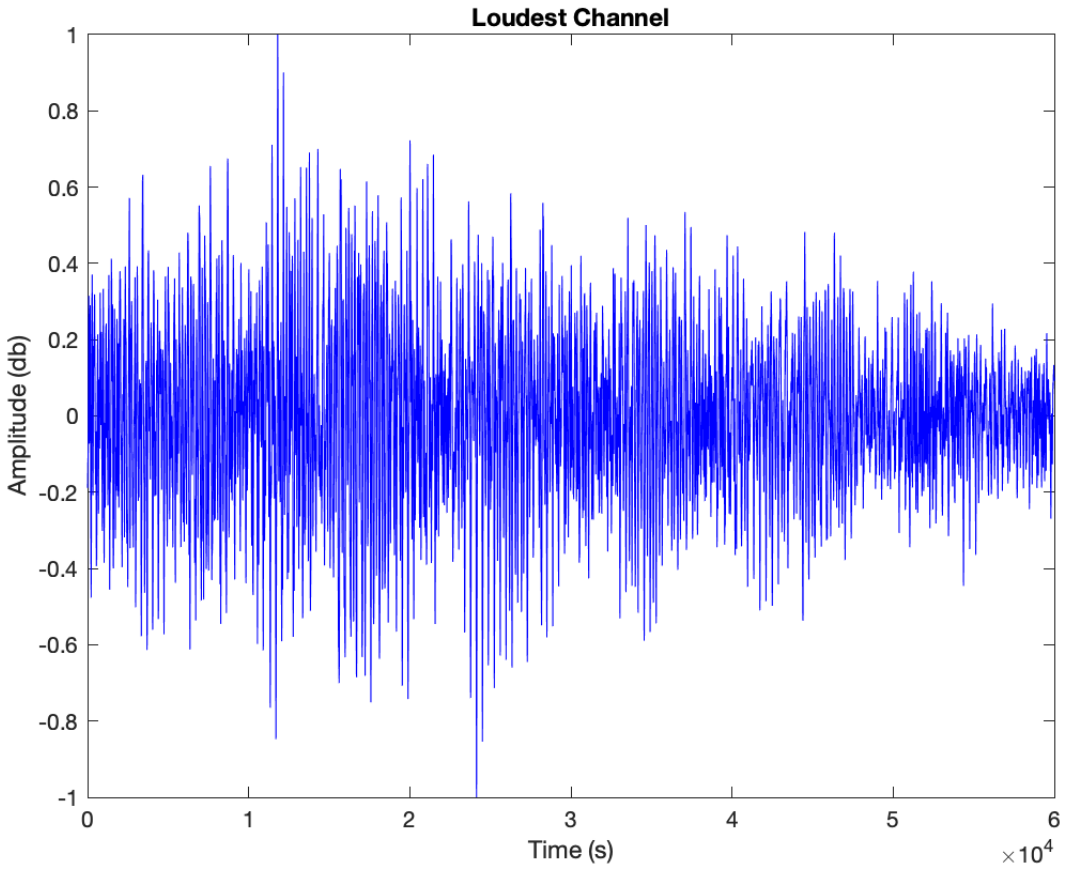

Figure 5.19. Channel 3 Selected, As It Has The Highest Intensity

The post-processing block within both algorithms receives the results for the direction of sound and the loudest audio channel. These values are then normalized if needed, in order to match the vector lengths. This allows the information to be displayed in a two-dimensional form. After the data is displayed in various forms, the buffers are cleared. This whole process from capturing, processing, displaying the data, to clearing the buffers loops over and over

again in real-time. To ensure a constant flow of data, the size of the buffers was selected adequately. In order to avoid processing delays, processing blocks were created to streamline the entire process.

The size of data is the largest when it is captured and stored in a lossless wave file. The file contains information from four audio channels, along with external noise. As the data passes through different processing stages, the size of data gets smaller and is thus easier to process. Each processing stage makes sure that the data is accurate and is free from any unwanted information. This ensures that the approximations made by the algorithms are accurate. Additional normalization is applied to the data set, in order to match the lengths of the arrays for displaying the data. The final form of the data for the distance approximation is then modeled by a moving line that represents the motion of the sound source and as a color gradient. While the final form of data for the direction of sound is displayed as a vector pointing in the direction of arrival of the sound.

6. RESULTS

The section provides a deep insight into the results and conclusions made from the system's performance during data collection. The comparisons are made based on accuracy and the system's response time, in different test scenarios and conditions. This segment also goes over some advantages the system has over other sensor systems. The results section is to verify and explain the functionality of the sensor array model proposed.

The main idea for the system is to model the distance based on the time delay of the captured sounds. In general, if an object is farther from a listener, then there are more reverberations present in the sound once it reaches the listener [9] [48]. Whereas, sounds closer to the listener have a smaller number of reverberations [9]. This is due to the fact that as distance increases, the sound gets reflected off of multiple surfaces [49]. While the sound wave continues to travel in the same trajectory, with the final acoustic signal being more complex due to the reflections being superimposed on the original signal. The transients of a sound source that has reverberations present are generally longer in length [48]. Estimating the distance solely based on the number of reverberations is also a good solution, but it requires extensive filters. For example, particle filters can compute the number of reverberations present but require high processing power [9]. The DRR or the intersection of reverberation level and the direct sound level provides the critical distance from the sound source (Figure 6.1).

This section also goes over some benefits that the setup offers. The benefits mainly pertain to the situations in which the system can perform relatively well, compared to other sensor types. It also proves an in-depth overview of the economic benefits. Economic benefits relate to the system hardware and software setup. The direction of arrival runs separately from the distance approximation algorithm due to the high number of computations performed by the algorithm. Execution of both algorithms in parallel allows the system to compute two different tasks at the same time. It also provides enough processing power needed for each algorithm. In an ideal system design or setup, these tasks can be computed on two separate computers for a faster response time.

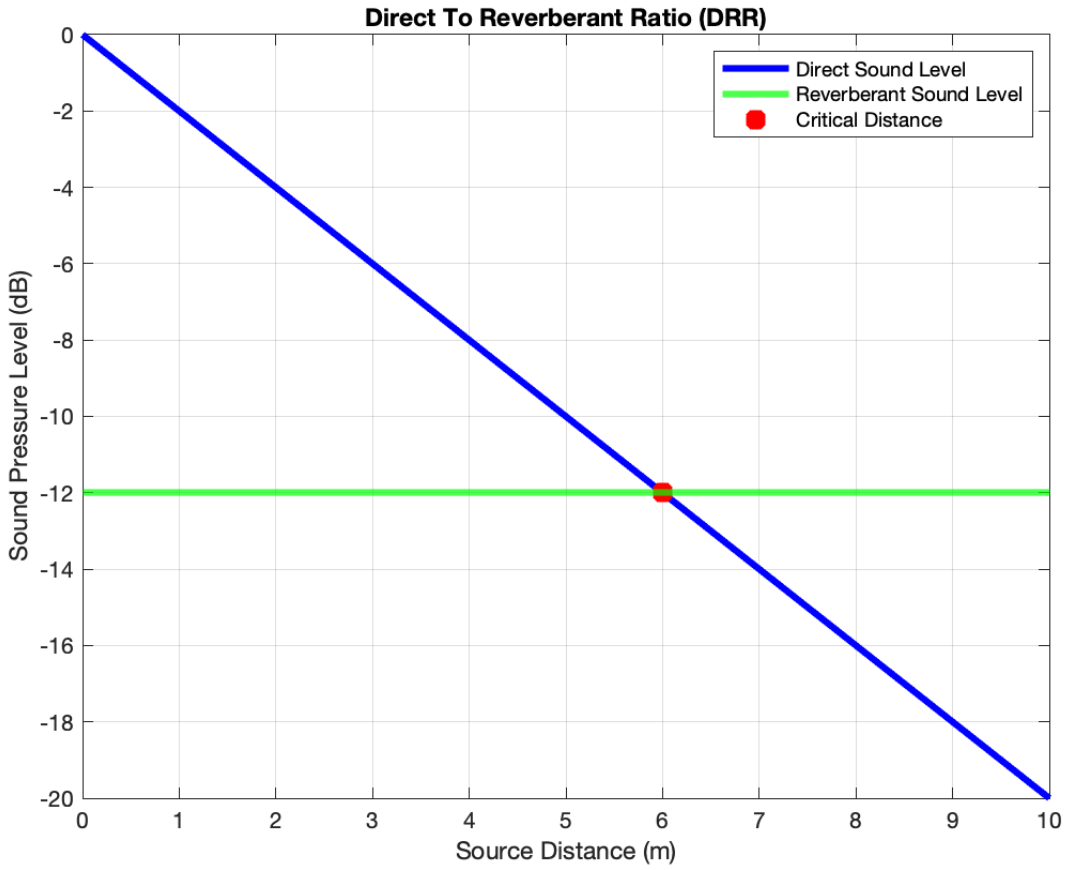

Figure 6.1. Correlation Between Reverberations & Critical Distance

The next section provides more information on how the direction of arrival algorithm computed the direction of the sound source. This includes the basic mathematical logic used for the system setup, as well as the equation for the setup. The model was designed for a two-microphone system for estimating the direction of the sound, by calculating the results as two possible outcomes, either left or right. This design concept was then extended for a four microphone array setup and can be used for an eight microphone array as well.

6.1 Direction Of Arrival Computation

The computation of the direction of the sound source is performed a bit differently than proximity approximation. It uses basic trigonometry and geometry to simplify the acoustic

arrangement. The direction of the sound originating from the source is displayed using the polar form of the acoustic field. Most signals captured by different sensor types are converted from analog to digital. In order to achieve that, a good sampling rate is selected, along with a buffer for temporarily storing the data. The sensitivity of the microphones detects the amplitudes or the intensities of the sound. Most microphones have an oscillating membrane that captures the varying pressure levels of the sound. Therefore, a microphone with a good sensitivity range for the frequency directly translates to higher accuracy.

The estimation for the direction of the sound source is as follows. The geometric representation of the system is shown in Figure 6.2. The model represents the comparison approach with microphones 1 and 2. This concept is further extended for all the microphones that are part of the array. Figure 6.2 has a sound source S, which is present between microphones 1 and 2. The direction of the sound source is given by the angle θ. The computation of the angle θ is performed by comparing the time delay between microphone pairs. For example, in Figure 6.2 the sound source is present between microphone 1 and 2, therefore the evaluation process considers the following microphone pairs: (1,2), (1,3), and (1,4).

The lengths of segments BS and DS can be calculated using the distance formula. Microphones 2, 3, and 4 have an equal distance of separation from each other. Similarly, microphones 1 and 4 also have an equal distance of separation between them. The time delay between the two consecutive microphone pairs is compared and based on that the angle θ is calculated. The algorithm also uses the speed of sound in the air to calculate the length of segment AD.

This is easy to calculate using the distance formula, where c is the speed of sound in air ($340m/s$), and t is the time delay in seconds (s):

$$AD = t * c \tag{6.1}$$

This gives the distance covered by the acoustic signal while the time delay of length t occurred. The positions on the x-axis remain fixed, as it represents the positions of the microphones in the array. The time delay can be detected by looking at the starting positions of the transients of the acoustic data. The smaller the time delay between two transients, the

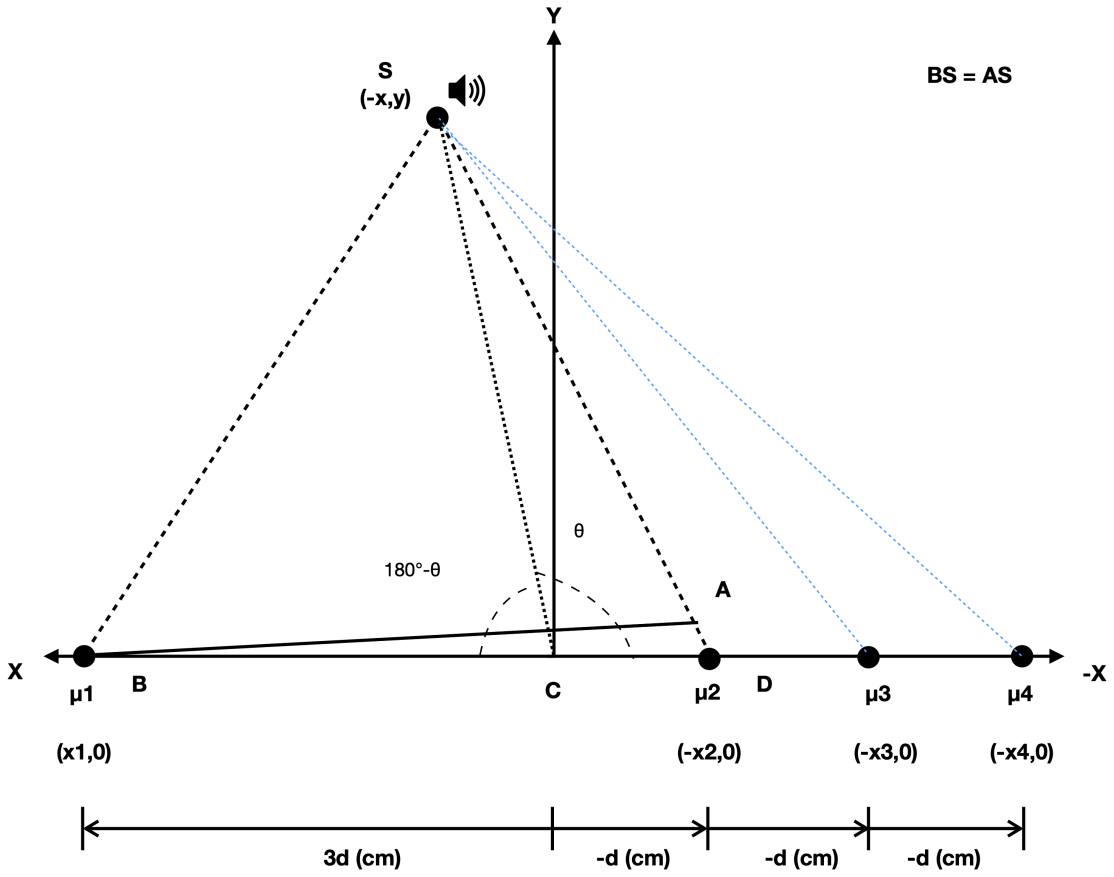

Figure 6.2. Direction Of Arrival Estimation System Model

more centered the sound source is between two consecutive microphone pairs. All possible microphone pairs are tested in order to get an accurate representation of the direction of the source. The positions of the microphones are essential for this measurement. With the Kinect sensor, the arrangement of the microphones is symmetric, with respect to the center point on the array module.

The implementation of the direction of arrival system can be extended to an eight microphone array. For this setup, microphones on the same surface of the vehicle can be grouped together for comparison of time delay. The difference in transients or the time delay in the signals captured by different microphones is used as inputs to compute the angle in radians. The results for the angle are computed by taking the components of the x & y axes. These

components are in terms of *sin* and *cos* computed in radians. The computed results for the angle in radians are then pushed to the graphics block. This is where the direction of the sound source is displayed by a vector in the polar form.

The direction of arrival (DOA) of the sound produced by the source is localized by the system using the microphone array. The samples collected are common for both the DOA and the proximity approximation algorithms. The data is processed in different ways to extract different kinds of information. Processing the data for the DOA algorithm is a bit more complex than proximity approximation, as there are many variables involved with the computation. As all possible combinations of microphone pairs are evaluated, the response time is a bit slow, but the results are highly accurate.

6.2 Accuracy Comparison

This subsection provides accuracy comparisons for different test scenarios, and how the system compares with actual measurements. The accuracy comparisons are made for the distance estimations and direction of the sound source. Starting from the different test conditions from the data collection procedure for proximity approximation, different measurements are compared for verification of data. Later on in this subsection, the results for the direction of arrival are compared for accuracy and precision. This provides a general idea for the system's performance in real test conditions. Further changes to the system's design or the software elements can be made based on the amount of deviation from actual measurements.

6.2.1 Distance Approximation Accuracy Comparison

Staring with the test conditions for the sound source distance modeling results from the data collection stage, this section provides a review of the results and their level of accuracy. In this subsection, all three possible test scenarios are considered for comparison. After the results for each scenario are presented, this section will also provide an overview of some

conditions in which the system did not perform relatively well. The intensity of the sound is given by the equation,

$$I = 2\pi^2\nu^2\lambda^2\rho c \tag{6.2}$$

I: Intensity of the sound in decibels (dB)

ν: Frequency of the sound wave (Hz)

λ: Amplitude of the sound wave (m)

ρ: Density of air (kg/m^3)

c: Speed of sound (m/s)

The speed of sound is given by the relation,

$$c = \nu\lambda \tag{6.3}$$

The equation for the intensity can be simplified to,

$$I = 2\pi^2\rho c^3 \tag{6.4}$$

Under ideal conditions, the speed of sound is approximately *340 m/s*, but this value varies based on the temperature, humidity, and location. Using the relation between speed, distance, and time.

$$c = d/\tau \tag{6.5}$$

Here, τ is the time delay between a sound wave captured by two consecutive microphone pairs, and the estimation of the distance is made using the speed of sound derived from the intensity levels.

The first test scenario was with the static sound source. In this case, sound sources like a monotonic sine wave, or less complex sounds are easier to detect and process. When additional environmental noise is introduced to the condition, the data gets more complex, and more processing is needed for extracting specific information. The results for the simple

sound sources like the monotonic sine wave and the engine sound have the exact same values for distance as the actual measurement. Slight deviations in accuracy are observed when environmental noise is added to the scenario. This occurs due to the fact that the noise goes through a reduction process, which can alter the original source sound in some way. But the data is not altered in a manner that would cause discrepancies in the results.

The second scenario was with the static host and a moving source. This scenario was simulated by moving the sound source around, in order to create an illusion of a moving vehicle. Similar to the first scenario, the monotonic sine wave, and the car sound produce accurate results. Sometimes if the object is moved too fast between the microphones in the array, then the tracking and the results are not quite accurate. A scenario like this wouldn't really occur in a real-world setting but is something that can be fixed by increasing the sample rate. But this also comes at a cost, in terms of an uptick in the time taken by the system to process the data. With the environmental noise induced into the scenario, the results are still observed to be close to the actual measurements.

The third and final scenario tested was with the moving host and source. In order to simulate this scenario, both the microphone array and the sound source were moving to replicate a moving vehicle. A test situation like such exposes the microphones to wind while in motion. To eliminate noise through wind, a transparent or porous shield can be mounted in front of the microphones that can block the wind but would let the sound wave pass through. The monotonic sine wave results were the most accurate in this scenario. Then the results for the engine sound were close to the actual motion of the source. After inducing environmental noise in the scenario, the results were close to the actual measurements but weren't as accurate as they were with other test conditions.

In conclusion, the data modeled by the system was observed to be in close agreement with the actual measurements for the distance of the sound source. Some additional improvements can be made on the software end by using neural networks for better separation of noise from the sound source [28]. The computers used for processing the data and analysis can be separated, which would lower the processing time.

6.2.2 Direction Of Arrival Accuracy Comparison

The verification process for the direction of the sound source is as follows. A printed protractor was used to measure the angle at which the wireless speaker was placed. In order to accurately get the angle of the speaker, a string was used to get the direction from the protractor to the source. This setup helps to verify the results provided by the DOA algorithm. The speed of the results depends on the complexity of the data provided as an input to the DOA algorithm. The verification of the results was done in a manner similar to the distance approximation algorithm. This provides a general overview of the performance of the system in terms of DOA estimation. The algorithm executes the A/D process, which helps to perform analysis on the data sets. Multiple DSP techniques such as sampling, noise reduction or frequency domain conversion is performed on the captured data, in order to extract information. The core concept of the noise reduction process is to minimize the noise in the raw data. For this purpose, the algorithm tries to maintain an SNR as high as possible, without losing any important information.

The description of the test scenario simulated is as follows. A wireless speaker was used to play the role of a sound source near the microphone array. The distance was kept at approximately 4 meters, as this was just to verify the functionality of the DOA tracking accuracy. In order to simulate a moving sound source, the speaker was approximately moved between -40° and 30° in random order. The DOA algorithm was able to accurately track the movement of the sound generated through the speaker, and direction is represented by the green arrow as shown in Figure 6.3 and Figure 6.4. As only a four microphone array was used to provide a proof of concept, therefore only the 180° acoustic field is considered. This approach can easily be extended to an eight microphone array in order to represent the entire 360° acoustic field around the host vehicle.

The results of the DOA algorithm were consistent with the actual values and can be also be observed from Figure 6.3, Figure 6.4 & Table 6.1. For complex acoustic data with loud environmental noise, the DOA results don't get affected by a significant amount. This is because the algorithm can easily pick out the loudest acoustic channel, as noise does not directly have any effect on the intensity levels of sound. In general, the overall performance

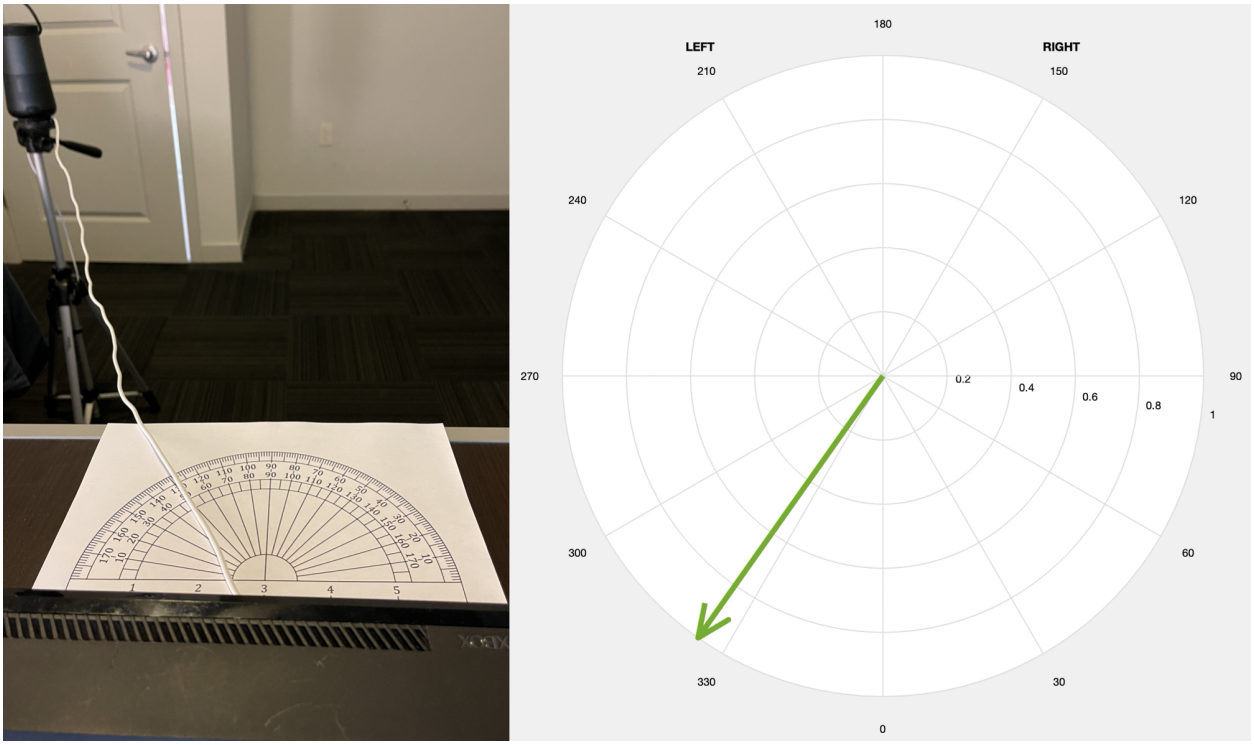

Figure 6.3. DOA -40° to the left

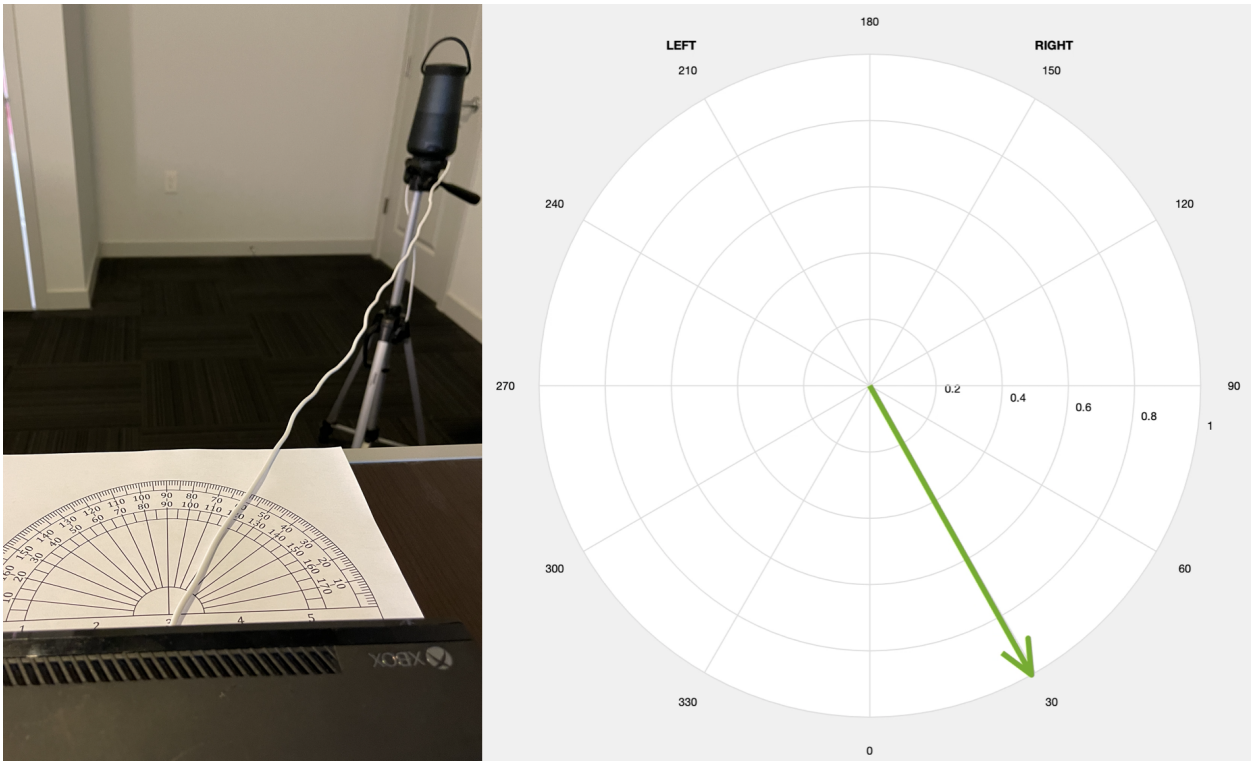

Figure 6.4. DOA -40° to the right

Table 6.1. DOA Algorithm Accuracy Comparison

TEST ANGLE	TRIAL 1	TRIAL 2	TRIAL 3	TRIAL 4	TRIAL 5
90°	95°	100°	90°	90°	85°
80°	80°	75°	80°	70°	80°
70°	60°	70°	65°	70°	65°
60°	60°	70°	60°	60°	65°
50°	40°	55°	50°	50°	50°
40°	40°	35°	45°	40°	50°
30°	30°	30°	35°	30°	30°
20°	20°	10°	20°	20°	20°
10°	5°	10°	5°	10°	10°
0°	-5°	0°	-5°	0°	0°
-10°	-5°	-15°	-10°	-10°	-10°
-20°	-20°	-20°	-15°	-10°	-15°
-30°	-25°	-30°	-25°	-30°	-35°
-40°	-40°	-45°	-45°	-40°	-40°
-50°	-50°	-50°	-50°	-60°	-55°
-60°	-55°	-55°	-60°	-60°	-60°
-70°	-70°	-65°	-70°	-70°	-70°
-80°	-70°	-85°	-80°	-75°	-80°
-90°	-90°	-85°	-90°	-95°	-80°

can be improved by integrating machine learning techniques or using artificial intelligence (AI), to recognize and differentiate characteristics of different sounds. Using smaller computers for easier tasks and more powerful ones for heavy processing can definitely provide a boost in the efficiency of the system.

6.3 Blind-Spot Detection

Compared to other sensors utilized for autonomous driving that produce beams for scanning and detecting the surroundings, a microphone array works a bit differently. Many sensors such as LiDAR produces large and complex data sets, that have a lot of redundant information, and deciphering the data requires high operational costs. In this regard, a microphone array is easier to setup and operate. An acoustic sensor like a microphone also has an advantage over other sensors as it can pick up on the acoustic activity around the vehicle. This opens up the possibilities for blind-spot detection.

Blind-spot regions or dead zones around a vehicle could be areas that are beyond the range of a sensor, or are below the level of detection [1]. Spots like these could be behind the vehicle, or by the sides where the field of view is limited. Especially in large commercial vehicles, blind-spot detection can prevent some major fatalities. A microphone array can pick up acoustic signatures of activities such as pedestrians, small vehicles, small children, or animals. This can be used for regular vehicles where the field of view is limited. This can also be used as a combination with some other sensors in order to detect still objects, as they do not produce any sound. Other applications of blind-spot detection can also be applied to large warehouses, where workers are moving around large commercial machines. Overall, an acoustic sensor has some benefits in terms of object detection over other sensor types, through detection of their acoustic signatures.

For example, the waveform of a pedestrian walking close to the vehicle is shown in Figure 6.5 and will be used for feature extraction. The changes in transients of the waveform can be used to recognize the type of activity taking place. This can be observed in Figure 6.6 when transient changes are detected on a small section of the sample. In Figure 6.6, a dense region of the vertical region represents a significant amount of change in the transient.

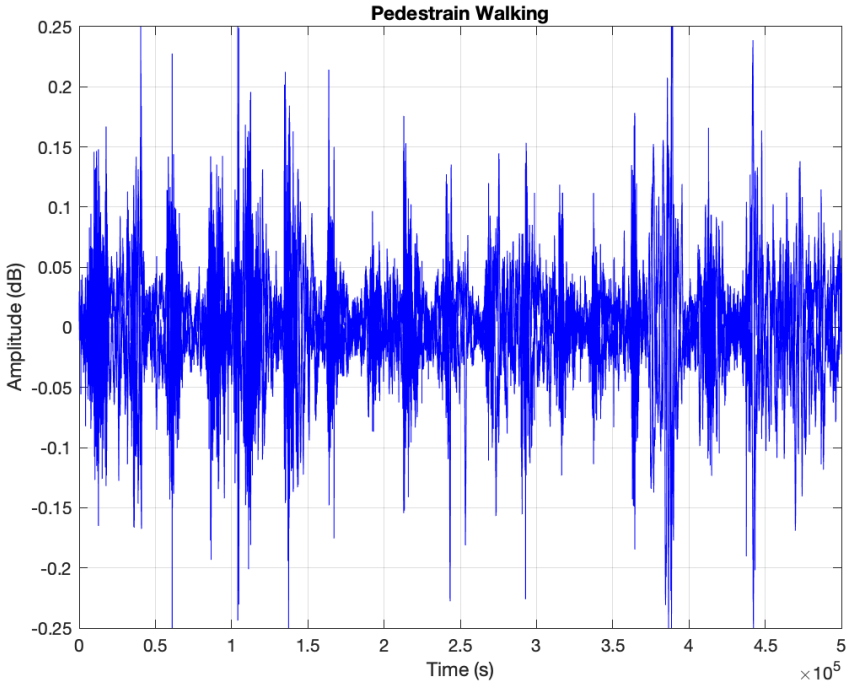

Figure 6.5. Transients Of A Pedestrian Walking

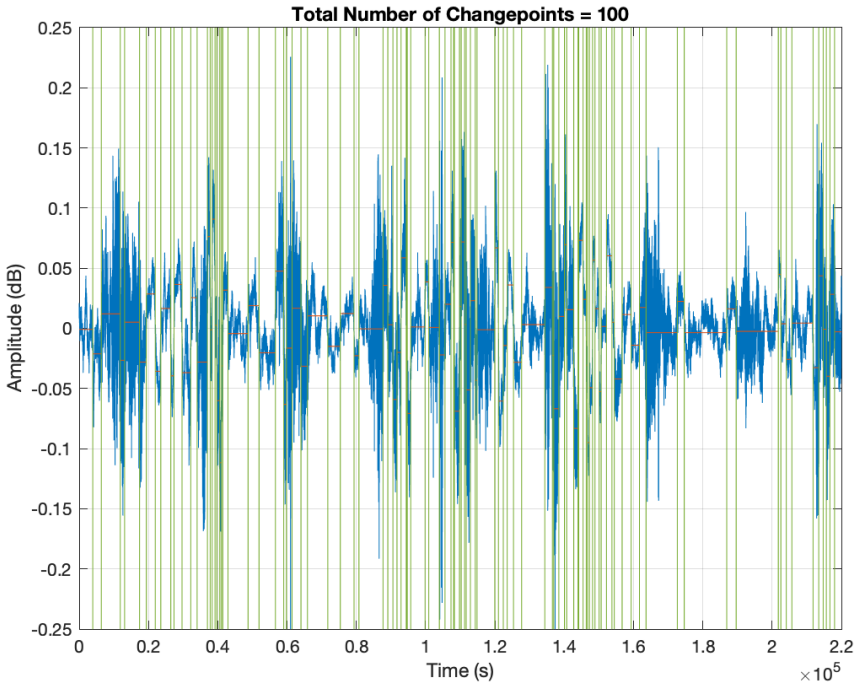

Figure 6.6. Transient Change Detection

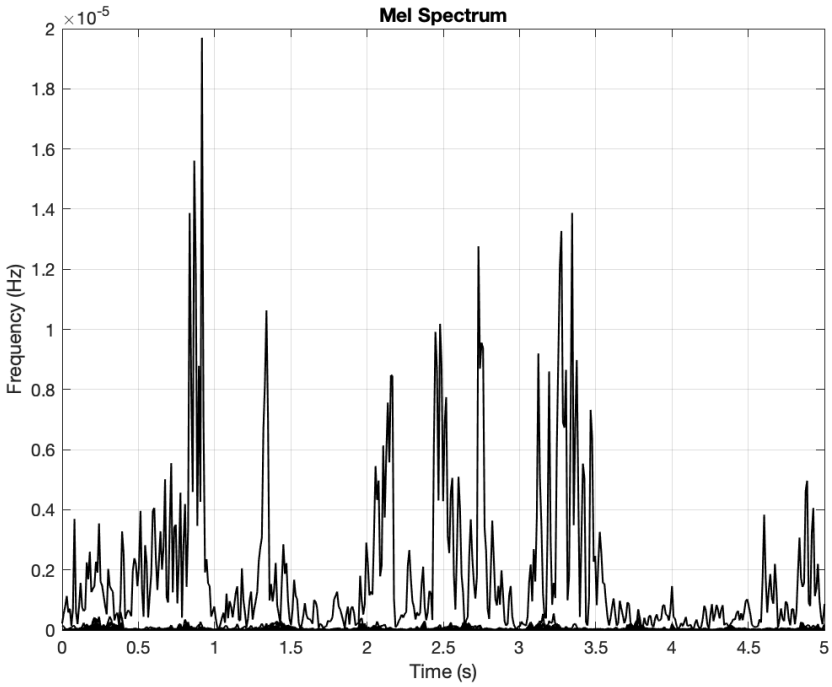

Figure 6.7. Mel Frequency Spectrum

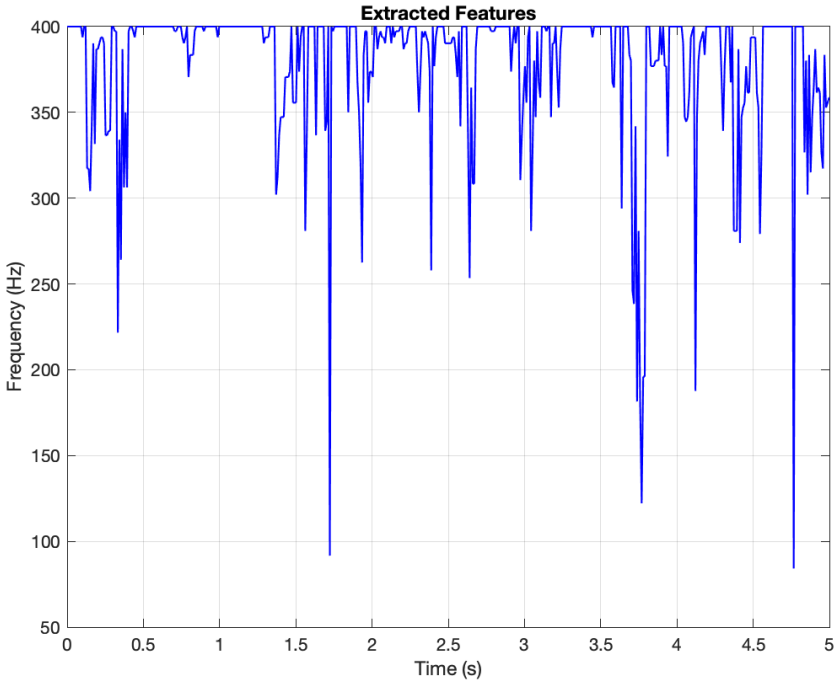

Figure 6.8. Extracted Acoustic Features

In Figure 6.7, a Mel-Frequency spectrum is shown. This provides an accurate representation of the way different sounds at specific distances are perceived by human ears [36] [50]. After performing a Mel Frequency analysis on the audio sample, the elements with the highest pitch are extracted and can be observed from Figure 6.8. These are the points at which the pedestrian's foot makes contact with the ground, and generates a periodic waveform with high pitch elements. The model for blind-spot detection can be enhanced even further using AI or machine learning. Acoustic event recognition or sound-activated decision-making can help avoid certain obstacles like firetrucks [45] [51]. Generating a risk evaluation matrix for different sounds can help prioritize tasks like emergency vehicles.

6.4 Economic Benefits

As microphone array only considers the data that reaches the sensor in terms of changes in acoustic intensities, thus producing data that is easy to process. A lot of redundant information present within the data about the surroundings is eliminated in the processing stages. This creates a system that is economic to operate. The data sets produced are not too complex and hence do not require extensive processing power. Overall, the cost of operation is relatively low compared to some other sensor types like LiDAR.

Cost efficiency is one of the major benefits of a system like such. Starting from the cost of hardware components, to the software elements, the maximum cost is under $200. The system also proves to be economic in the long run, with regard to the cost of replacing hardware components. In terms of software tools used, the code can be ported to Python which is free to use and provides efficient libraries for audio applications [52]. For example *pyAudioAnalysis*, an open-source Python library can efficiently extract audio information and perform analysis [53]. The data sets generated by the system are also economic in terms of processing, thus eliminating the need for extensive computational power. A low-cost system allows the user to replace faulty parts, which directly translates to better safety compared to costlier sensor types like LiDAR. In conclusion, the system's low operation costs, simple setup, flexibility for sensor fusion, tracking, and detection accuracy, make it a strong alternative to some other high-cost sensor systems.

7. FUTURE WORK

The system was simulated on a personal computer, therefore the number of resources such as RAM, CPU, and memory allocated were limited. If the sensor array and the software elements are connected to a computer that is specifically designed for the purpose of Acoustic SLAM, then the performance would see a significant boost. Similarly, running the DOA and distance approximation algorithms on separate computers can definitely help increase the processing speed. This parallel computational approach for processing and estimation will significantly reduce the system's response time. The overall accuracy and detection of the system setup can be increased through sensor fusion. Sensor fusion with a vision-based sensor can help detect still objects or vehicles, which cannot be detected by the microphone array. Even an IR camera can help detect still obstacles while the microphone array can pick up acoustic signatures of moving vehicles.

MATLAB was selected as the IDE because it provides flexibility in terms of testing and debugging. But it also has some limitations like requiring extensive CPU power and execution of statements in a sequential manner. The finalized version of the software elements can be ported to the Python programming language. This would enable the algorithms to run on various platforms. Python is free and also offers open-source libraries like *librosa* for processing audio data, which is faster than the audio toolbox offered by MATLAB. It is also the ideal language for machine learning and artificial intelligence applications, which can be applied to the system's design in the future.

8. SUMMARY

In this thesis, a novel idea that utilizes an array of acoustic sensors for SLAM was presented. Different test scenarios were simulated using MATLAB, and the system's performance was evaluated. The overall performance and accuracy of the system were comparable to the actual measurements. One major advantage that the system has over other sensor setups is cost-efficiency. The data sets captured by the sensor array are easy to process, as they are not too complex and contain acoustic data stored in four channels. The simplicity of the data means that the system does not have a high operational cost, in terms of computational power needed. This helps the system capture and process data, in order to display the results in near real-time.

Another advantage the system carries over sensors like cameras, LiDAR, and RADAR is that it can function in adverse conditions. Conditions like heavy snow, do not impact the system's accuracy. Similarly, dark or low light conditions also do not alter the system's functionality. Since conditions like rain and wind have white noise-like characteristics, therefore this can be removed during the processing stage by applying a high pass filter to the data. This makes it a viable low-cost alternative.

The system's design is relatively simple and only depends on the positioning of the microphones for better accuracy. This provides enough flexibility for modifications or for sensor fusion. Sensor fusion with a vision-based sensor can help the system reach even higher detection accuracy. The flexibility of the software elements allows the code to be ported to other programming languages for higher portability and compatibility. Regular vehicles, large commercial transportation, or warehouse machines can even benefit from a system like such through blind spot detection. In conclusion, this system carries major advantages over other sensor types in many situations, while being a cost-efficient alternative.

REFERENCES

[1] J. Wang, L. Zhang, Y. Huang, and J. Zhao, "Safety of autonomous vehicles," *Journal of advanced transportation*, vol. 2020, 2020.

[2] M. Skolnik. (Nov. 2020). "Factors affecting radar performance." Last date accessed: 05-13-2021, [Online]. Available: https://cleantechnica.com/2021/03/12/lidar-may-be-harmful-to-people-cameras/.

[3] B. A. Jumaa, A. M. Abdulhassan, and A. M. Abdulhassan, "Advanced driver assistance system (adas): A review of systems and technologies," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 8, no. 6, 2019.

[4] L. Tang, Y. Shi, Q. He, A. W. Sadek, and C. Qiao, "Performance test of autonomous vehicle lidar sensors under different weather conditions," *Transportation research record*, vol. 2674, no. 1, pp. 319–329, 2020.

[5] NASA. (Aug. 2018). "Midar - active multispectral imaging." Last date accessed: 08-29-2021, [Online]. Available: https://www.nasa.gov/ames/las/midar.

[6] M. Yiwere and E. J. Rhee, "Distance estimation and localization of sound sources in reverberant conditions using deep neural networks," *Int. J. Appl. Eng. Res*, vol. 12, no. 22, pp. 12 384–12 389, 2017.

[7] N. Voudoukis and S. Oikonomidis, "Inverse square law for light and radiation: A unifying educational approach," *European Journal of Engineering and Technology Research*, vol. 2, no. 11, pp. 23–27, 2017.

[8] J. O'Reilly, S. Cirstea, M. Cirstea, and J. Zhang, "A novel development of acoustic slam," in *2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, IEEE, 2019, pp. 525–531.

[9] Y.-C. Lu and M. Cooke, "Binaural estimation of sound source distance via the direct-to-reverberant energy ratio for static and moving sources," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1793–1805, 2010.

[10] P. Yang, H. Sun, and L. Zu, "An acoustic localization system using microphone array for mobile robot," *Int. J. Intell. Eng. Syst*, vol. 2, pp. 18–26, 2007.

[11] M. Zaffar, S. Ehsan, R. Stolkin, and K. M. Maier, "Sensors, slam and long-term autonomy: A review," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, IEEE, 2018, pp. 285–290.

[12] C. Evers, A. H. Moore, and P. A. Naylor, "Acoustic simultaneous localization and mapping (a-slam) of a moving microphone array and its surrounding speakers," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 6–10.

[13] I. Dokmanić, L. Daudet, and M. Vetterli, "From acoustic room reconstruction to slam," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Ieee, 2016, pp. 6345–6349.

[14] A. Karpov, L. Akarun, H. Yalçın, A. Ronzhin, B. E. Demiröz, A. Çoban, and M. Železnỳ, "Audio-visual signal processing in a multimodal assisted living environment," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[15] A. Nunes, B. Reimer, and J. Coughlin. (Apr. 2018). "People must retain control of autonomous vehicles." Last date accessed: 07-09-2021, [Online]. Available: https://www.nature.com/articles/d41586-018-04158-5.

[16] K. Gammon. (Sep. 2020). "Can a new algorithm make self driving cars safer." Last date accessed: 03-26-2021, [Online]. Available: https://www.insidescience.org/news/can-new-algorithm-make-self-driving-cars-safer.

[17] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE intelligent vehicles symposium (IV)*, IEEE, 2011, pp. 163–168.

[18] NHTSA. (Sep. 2017). "Automated vehicles for safety." Last date accessed: 03-18-2021, [Online]. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety.

[19] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang, and W. Shi, "Computing systems for autonomous driving: State of the art and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6469–6486, 2020.

[20] A. Armenta. (Mar. 2021). "Safety considerations for lidar sensors." Last date accessed: 08-04-2021, [Online]. Available: https://control.com/technical-articles/safety-considerations-for-lidar-sensors/.

[21] J. Sensiba. (Mar. 2021). "Lidar may be harmful to people and cameras." Last date accessed: 07-15-2021, [Online]. Available: https://cleantechnica.com/2021/03/12/lidar-may-be-harmful-to-people-cameras/.

[22] C. Chen, S. Majumder, Z. Al-Halah, R. Gao, S. K. Ramakrishnan, and K. Grauman, "Learning to set waypoints for audio-visual navigation," *arXiv preprint arXiv:2008.09622*, 2020.

[23] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[24] A. Ivankov. (Apr. 2020). "Advantages and disadvantages of lidar." Last date accessed: 08-8-2021, [Online]. Available: https://www.profolus.com/topics/advantages-and-disadvantages-of-lidar/.

[25] C. Couvreur, "Environmental sound recognition: A statistical approach," *Doctorat en sciences appliqu ees, Facult e Polytechnique de Mons, Mons, Belgium*, 1997.

[26] J. Steckel and H. Peremans, "Batslam: Simultaneous localization and mapping using biomimetic sonar," *PloS one*, vol. 8, no. 1, e54076, 2013.

[27] U. Saqib and J. R. Jensen, "A model-based approach to acoustic reflector localization with a robotic platform," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 4499–4504.

[28] K. Ashraf, B. Elizalde, F. Iandola, M. Moskewicz, J. Bernd, G. Friedland, and K. Keutzer, "Audio-based multimedia event detection with dnns and sparse sampling," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 611–614.

[29] C. Chen, Z. Al-Halah, and K. Grauman, "Semantic audio-visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 516–15 525.

[30] R. Munguía and A. Grau, "Single sound source slam," in *Iberoamerican Congress on Pattern Recognition*, Springer, 2008, pp. 70–77.

[31] Ł. Grzymkowski, K. Główczewski, and S. A. Raczyński, "Distributed acoustic slam," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, IEEE, 2015, pp. 1566–1570.

[32] S. Pfeiffer, S. Fischer, and W. Effelsberg, "Automatic audio content analysis," in *Proceedings of the fourth ACM international conference on Multimedia*, 1997, pp. 21–30.

[33] L. Niklasson, "Low intensity natural sounds and pink noise's effect on attention," *UMEA University, Department of Psychology*, vol. 2020, 2019.

[34] C. Rascon and I. Meza, "Localization of sound sources in robotics: A review," *Robotics and Autonomous Systems*, vol. 96, pp. 184–210, 2017.

[35] I.-J. Jung and J.-G. Ih, "Distance estimation of a sound source using the multiple intensity vectors," *The Journal of the Acoustical Society of America*, vol. 148, no. 1, EL105–EL111, 2020.

[36] G. Tzanetakis, G. Essl, and P. Cook, "Audio analysis using the discrete wavelet transform," in *Proc. conf. in acoustics and music theory applications*, Citeseer, vol. 66, 2001.

[37] L. Xiang, C. Kerl, F. Echtler, and L. Glud. (Mar. 2020). "Open source drivers for the kinect for windows v2 device." Last date accessed: 02-10-2021, [Online]. Available: https://github.com/OpenKinect/libfreenect2.

[38] H. Wu, M. Siegel, and P. Khosla, "Vehicle sound signature recognition by frequency vector principal component analysis," in *IMTC/98 Conference Proceedings. IEEE Instrumentation and Measurement Technology Conference. Where Instrumentation is Going (Cat. No. 98CH36222)*, IEEE, vol. 1, 1998, pp. 429–434.

[39] X. Chen, H. Sun, and H. Zhang, "A new method of simultaneous localization and mapping for mobile robots using acoustic landmarks," *Applied Sciences*, vol. 9, no. 7, p. 1352, 2019.

[40] L. Lu, H. Jiang, and H. Zhang, "A robust audio classification and segmentation method," in *Proceedings of the ninth ACM international conference on Multimedia*, 2001, pp. 203–211.

[41] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: A survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, pp. 1–11, 2017.

[42] C. Evers and P. A. Naylor, "Optimized self-localization for slam in dynamic scenes using probability hypothesis density filters," *IEEE Transactions on Signal Processing*, vol. 66, no. 4, pp. 863–878, 2017.

[43] J. Foote, "An overview of audio information retrieval," *Multimedia systems*, vol. 7, no. 1, pp. 2–10, 1999.

[44] Z. Zeng, J. Tu, M. Liu, T. S. Huang, B. Pianfetti, D. Roth, and S. Levinson, "Audio-visual affect recognition," *IEEE Transactions on multimedia*, vol. 9, no. 2, pp. 424–428, 2007.

[45] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time–frequency audio features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[46] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, pp. 1–14, 2014.

[47] C. Cannam, C. Landone, and M. Sandler, "Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files," in *Proceedings of the ACM Multimedia 2010 International Conference*, Firenze, Italy, Oct. 2010, pp. 1467–1468.

[48] G. Chen and Y. Xu, "A sound source localization device based on rectangular pyramid structure for mobile robot," *Journal of Sensors*, vol. 2019, 2019.

[49] E. Georganti, T. May, S. Van De Par, and J. Mourjopoulos, "Sound source distance estimation in rooms based on statistical properties of binaural signals," *IEEE transactions on audio, speech, and language processing*, vol. 21, no. 8, pp. 1727–1741, 2013.

[50] M. S. Puckette, M. S. P. Ucsd, T. Apel, *et al.*, "Real-time audio analysis tools for pd and msp," *University of California San Diego*, 1998.

[51] P. Dhakal, P. Damacharla, A. Y. Javaid, and V. Devabhaktuni, "A near real-time automatic speaker recognition architecture for voice-based user interface," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 504–520, 2019.

[52] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, Citeseer, vol. 8, 2015, pp. 18–25.

[53] T. Giannakopoulos, "Pyaudioanalysis: An open-source python library for audio signal analysis," *PloS one*, vol. 10, no. 12, e0144610, 2015.