Detection of Emergency Signal in Hearing Aids using Neural Networks

A major project report submitted in partial fulfillment of the requirements for the award of the degree of

Master of Science in Electrical Engineering with Emphasis on Signal Processing

By

VAMSHI KRISHNA LAKUM ARSHINI GUBBALA

**Contact Information:**

Author(s):
Vamshi Krishna Lakum
Address: Karlskrona, Sweden
E-mail: vakr11@student.bth.se

Arshini Gubbala
Address: Karlskrona, Sweden
E-mail: argu11@student.bth.se
ACKNOWLEDGEMENT

First and foremost we offer my sincere gratitude to my supervisor, Prof. Nedelko Grbic and Prof. Sven Johansson, who has supported us throughout my thesis with their patience and knowledge. I attribute the level of my Master’s degree to their encouragement and effort and without them this thesis would not have been completed or written. Finally, we would also like to thank our beloved parents for supporting us both morally and financially and we would also like to thank Sridhar Bitra for his extended support. Without their encouragement and motivation we could not able to complete this project. We would also like to thank our friends who helped us with their valuable suggestions and support.

Vamshi Krishna Lakum,
Arshini Gubbala.
ABSTRACT

The detection of an emergency signal can be estimated by the cancellation of surrounding noise and achieving the desired signal in order to alert the automobilist. The aim of the thesis is to detect the emergency signal arriving nearer to the automobilist carrying hearing aids. Recent studies show that this can be achieved by designing various kinds of fixed and adaptive beam formers. A beam former does spatial filtering in the sense that it separates two signals with overlapping frequency content originating from distinctive directions. In this contribution, robust beam former namely Wiener beam former is designed and analyzed collaboratively in a group under the consideration of hearing aid constraints such as the microphone distance. A fractionally delay (FD) are designed to get a maximally flat group delay. The studies had been carried out by comparing noise cancellation algorithms like LMS, NLMS, LLMS and RLS algorithms. By comparing Omni-directional and multi-directional microphones the SNR can be studied.

In this thesis work, first proposing appropriate microphone array setup with improved beam forming techniques by using required adaptive algorithm (NLMS) in order to get better quality using the Microphone arrays. Microphone arrays have been widely used to improve the performance of speech recognition systems as well as to benefit for people who need hearing aids. With the help of microphone arrays, it can choose to focus on signals from a specific direction. To getting better signal quality in microphone array using adaptive algorithms, these are help in the noise suppression in accordance with the different beam forming techniques.

The proposed system is implemented successfully and validated using MATLAB simulation tool. The emergency signal is different in different countries, so we identify any type of emergency signal by training through neural networks.
# TABLE OF CONTENTS

**ACKNOWLEDGEMENT**

**ABSTRACT**

**LIST OF FIGURES**

1. **Introduction**
   1.1 Introduction 10
   1.2 Objective of thesis 11
   1.3 Expected Outcomes 11
   1.4 Outline of Thesis 11

2. **Literature Review and Background Study** 12
   2.1 Microphone Array in Hearing Aids 12
   2.2 Beamforming in Hearing Aids 13
   2.3 Noise Reduction in Hearing Aids 13

3. **Microphone Array and Fractional Delay** 14
   3.1 Overview of the chapter 14
   3.2 Introduction of Microphone 14
   3.3 Microphone Polar Patterns 14
      3.3.1 Omni-directional 15
      3.3.2 Bi-directional 15
      3.3.3 Cardiod 15
   3.4 Microphone Array 16
   3.5 Working Principle 16
   3.6 Array Geometry of Microphone 17
   3.7 Fractional Delay 18
   3.8 Calculation of Fractional Delay 19

4. **Beam Formation** 20
   4.1 Beam Formation Basics 20
   4.2 Delay and Sum Beam Former (DSB) 21
   4.3 Adaptive Beam Former 22
4.4 Wiener Beam Former  
4.5 Generalized Side Lobe Canceller

5 Adaptive Algorithms

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Least Mean Square (LMS) Algorithm</td>
<td>26</td>
</tr>
<tr>
<td>5.2 Normalized Least Mean Square (NLMS) Algorithm</td>
<td>27</td>
</tr>
<tr>
<td>5.3 Leaky Least Mean Square (LLMS) Algorithm</td>
<td>28</td>
</tr>
<tr>
<td>5.4 Recursive Least Mean Square (RLS) Algorithm</td>
<td>29</td>
</tr>
</tbody>
</table>

6. Direction of Arrival

6.1 MUSIC Algorithm

7 Neural Networks

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Back Propagation Algorithm</td>
<td>34</td>
</tr>
<tr>
<td>7.2 Training Procedure</td>
<td>35</td>
</tr>
<tr>
<td>7.3 Working of Algorithm</td>
<td>36</td>
</tr>
</tbody>
</table>

8 Simulation and Results

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1 Experimental Setup</td>
<td>38</td>
</tr>
<tr>
<td>8.2 Experimental Block Diagram</td>
<td>41</td>
</tr>
<tr>
<td>8.3 Microphone Array Setup</td>
<td>42</td>
</tr>
<tr>
<td>8.4 Wiener Beam Former Implementation</td>
<td>49</td>
</tr>
<tr>
<td>8.5 Implementing GSC for the system</td>
<td>50</td>
</tr>
<tr>
<td>8.6 Signal to Noise Ratio</td>
<td>52</td>
</tr>
<tr>
<td>8.7 Direction of Arrival</td>
<td>53</td>
</tr>
<tr>
<td>8.8 Neural Networks using Back Propagation</td>
<td>54</td>
</tr>
</tbody>
</table>

9 Conclusion and Future Work

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1 Conclusion</td>
<td>57</td>
</tr>
<tr>
<td>9.2 Future work</td>
<td>57</td>
</tr>
</tbody>
</table>

10 Bibliography

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>References</td>
<td>58</td>
</tr>
<tr>
<td>References</td>
<td>58</td>
</tr>
</tbody>
</table>
**LIST OF FIGURES:**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig.1</td>
<td>Brief idea of thesis</td>
<td>10</td>
</tr>
<tr>
<td>Fig.2</td>
<td>Patterns of different microphones</td>
<td>15</td>
</tr>
<tr>
<td>Fig.3</td>
<td>Microphone Array layout</td>
<td>17</td>
</tr>
<tr>
<td>Fig.4</td>
<td>Microphone Array geometry</td>
<td>18</td>
</tr>
<tr>
<td>Fig.5</td>
<td>Representing fractional delay</td>
<td>19</td>
</tr>
<tr>
<td>Fig.6</td>
<td>Representation of beam forming</td>
<td>20</td>
</tr>
<tr>
<td>Fig.7</td>
<td>Beam formation layout</td>
<td>21</td>
</tr>
<tr>
<td>Fig.8</td>
<td>Delay and Sum Beam former</td>
<td>22</td>
</tr>
<tr>
<td>Fig.9</td>
<td>Adaptive Beam former</td>
<td>23</td>
</tr>
<tr>
<td>Fig.10</td>
<td>Generalized Side-lobe Canceller</td>
<td>25</td>
</tr>
<tr>
<td>Fig.11</td>
<td>Representation of different Adaptive Signals</td>
<td>26</td>
</tr>
<tr>
<td>Fig.12</td>
<td>DOA estimation problem</td>
<td>30</td>
</tr>
<tr>
<td>Fig.13</td>
<td>Back Propagation Training set</td>
<td>34</td>
</tr>
<tr>
<td>Fig.14</td>
<td>Applying a training pair to a network</td>
<td>35</td>
</tr>
<tr>
<td>Fig.15</td>
<td>Single connection learning in a Back Propagation network</td>
<td>36</td>
</tr>
<tr>
<td>Fig.16</td>
<td>Experimental Setup</td>
<td>38</td>
</tr>
<tr>
<td>Fig.17</td>
<td>Input Signals (a) Emergency signal (b) Traffic noise</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>(c) Emergency signal with Traffic noise</td>
<td></td>
</tr>
<tr>
<td>Fig.18</td>
<td>Power Spectral Density of the Input Signals</td>
<td>39</td>
</tr>
<tr>
<td>Fig.19</td>
<td>Algorithm Output (a) LMS algorithm (b) NLMS algorithm</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>(c) LLMS algorithm (d) RLS algorithm</td>
<td></td>
</tr>
<tr>
<td>Fig.20</td>
<td>Power Spectral Density of Algorithm Output</td>
<td>41</td>
</tr>
</tbody>
</table>
Fig.21: General Experimental Block diagram 42
Fig.22: Case 1 (a) Setup (b) SNR values (c) Graphical representation 43-44
Fig.23: Case 2 (a) Setup (b) SNR values (c) Graphical representation 45-46
Fig.24: Case 3 (a) Setup (b) SNR values (c) Graphical representation 47-48
Fig.25: Wiener Beam Former Implementation 49
Fig.26: Graphical representations at (a) Mic 1 (b) Mic 2 (c) WBF1 (d) WBF2 50
Fig.27: Emergency Signal at the system output 51
Fig.28: Structure of GSC implementation 58
Fig.29: Plot of Direction of Arrival 54
Fig.30: Flow Chart for finding error 55
Fig.31: Error Graph 56
Introduction
1. Introduction

1.1 Introduction:
Technology has been increased now-a-days and it is being shaped as per the requirements of the generation. These days automobilists have been carrying hearing aids like ear phones during the journey for wide variety of reasons. In traffic, it is difficult to detect an emergency signal like ambulances, siren, police, and fire-fighting vehicle while using any of these hearing aids. This leads to several disasters in many ways, therefore a system has been investigated for the detection of incoming direction of an emergency vehicle. Acoustic detection methods based on a cross microphone array have been implemented[1]. There is a huge necessity of detection of the signals of these emergency vehicles. A siren detector is a system able to perform detection of siren sounds, the instantaneous frequency components of change at know rate within a selected frequency band and with a known period. Detection of siren sounds emitted by an emergency vehicle may be used for the development of an efficient control of traffic. This issue can be solved by using the microphone in the hearing aids in order to detect the emergency signal[2]. Microphone has the capability of reducing the additive noise in received signals and there is a vast development that has to be adopted[3]. In general microphones are used to detect and transducer sounds and the effectiveness of the microphones are limited. Every microphone is characterized by a ‘directivity pattern’ which specify the gain and the phase shift that microphones gives a signal coming from specific direction. Beam forming is a spatial filtering technique that isolates sound sources based upon their positions in space [4].

Fig.1: Brief Idea of the thesis
1.2 Objective of the thesis:
Beam Formation has played an extensive role in the estimation of Direction of Arrival (DOA) [5]. By using a microphone array model with the Fractional delay (FD) [6] DOA can be optimally measured. The aim of this research is to detect any type of emergency signal like ambulances, siren, police, fire-fighting vehicle etc., by which effectively filtering the emergency signal from external noise and also the emergency signal should be trained through neural networks in hearing aids like head phones.

1.3 Research questions:
1. Which direction is the emergency signal coming from?
2. How to effectively filter the emergency signal from the external noise?
3. How to train neural networks to identify different emergency signals?

1.4 Expected outcomes:
The expected results are as follows:
1. The emergency signal has been detected by two microphones and the direction of arrival is estimated.
2. The emergency signal is enhanced from the detected sound signal (emergency signal with noise) using beam formation technique.
3. Different types of emergency signals like ambulances, siren, police, and fire-fighting vehicles are trained to the neural network successfully.

1.5 Outline of Thesis:
The report is organized as follows:
- Chapter 2 describes Literature Review and Background Study.
- Chapter 3 describes Microphone array and Fractional Delay.
- Chapter 4 describes Beam formation Techniques.
- Chapter 5 describes Adaptive Algorithms.
- Chapter 6 describes Direction of Arrival.
- Chapter 7 describes Simulation Results and Experimental Setup
- Chapter 8 describes Conclusion and Future Work.
- Chapter 9 describes References.
2. Literature Review and Background study

It is difficult for the normal hearing person to understand the speech signal containing the background noise. Many algorithms like minima controlled recursive averaging (MCRA) algorithm[7] and Psychoacoustically Motivated Spectral Weighting Rules (PMSWR)[8] have been developed for the enhancement of the speech signal from the background noise[9]. Several signal processing techniques has been evolved to increase the quality and intelligibility of the speech signal in the hearing aids[10].

In the previous decades, many sophisticated signal processing algorithms such as beam forming, noise reduction techniques and feedback cancellation has been used to increase the development of hearing aids.[11] There are some other signal processing technologies like adaptive filtering, echo cancellation and array processing which have been widely used in hearing aids[11].

2.1 Microphone Array in Hearing Aids

Microphone array consists of a multiple microphones arranged in a spatial domain. Microphone array hearing aids is one of the best solution in order to help even for the hearing impaired person to listen the speech signal in the presence of background noise[12]. The main aim of the microphone array hearing aids is to increase the speech to interference ratio when the interference is arrived from different directions rather than the desired speech signal. There are basically three major components in the microphone array hearing aids namely the microphone array, processing unit and receivers in which all the three components are interconnected. Microphone array acts as a preprocessor to the system followed by the speech enhancement system [13]. In order to maintain high signal to noise ratio in a noisy environment microphone array is very much capable. The advantage of microphone array is their ability to exploit, reduction of noise based on the knowledge of the position of speech signal[14].
2.2 Beamforming in Hearing Aids

Beamforming is a widely used signal processing technique which is used for signal transmission or reception. In order to create a constructive interference in a particular direction and destructive interference in other direction beamforming technique is used. Beamforming is used to create a null in the direction of the noise source and this technique allows only signal in which it is coming from a specific direction. Beam forming is performed in hearing aids to enhance the SNR and to increase the speech intelligibility in hearing aids [15].

There are several beamforming techniques that has been developed now a days for hearing aids in order to enhance the desired signal from various types of noises. Fixed beam forming is one of the technique which is used to obtain the beam in a particular direction and it doesn't change the direction as that the incident source direction changes. In order to steer the directional pattern to the location of the desired source and to maximize the attenuation of noise source an adaptive beam former is used in hearing aids.

2.3 Noise Reduction in Hearing Aids

In many applications noise which is unwanted signal plays a major role. In various devices such as telecommunications, radar, sonar, medical application etc., noise exists in various forms and creates problems. To enhance the speech signal in hearing aids several noise reduction techniques have been developed. A reference signal is available and is used to reduce the noise.

Amplitude modulation is the key technology used to separate speech from noise signal. Amplitude modulation works on the principle that desired speech signal has a harmonic structure and the amplitude of this harmonic component will change over the time and produces amplitude modulation. The amplitude signal of the speech and noise may vary. Speech signal has higher amplitude signal compared to that of stationary and pseudo stationary noises. Pseudo noise has very low amplitude modulation. The amplitude modulation of the environmental noises such as babble, traffic noise has higher amplitude than the stationary noise and lower amplitude compare to the speech signal[16].
3. Microphone Array and Fractional Delay

3.1 Overview of the chapter:
This chapter deals with the introduction of microphone, microphone polar patterns, microphone array, working principle, array geometry of microphone and fractional delay filters and calculation for fractional delay.

3.2 Introduction of Microphone:
Microphones are transducers which detect sound signals and produce an electrical image of the sound which means they produce a voltage or a current which is proportional to the sound signal. However, the effectiveness of a microphone can be limited. Most microphones which we are using today use electromagnetic induction, capacitance change, piezoelectric generation or light modulation to produce an electrical voltage signal from mechanical vibration. Acoustic noise and multipath distortion make hands-free sound capture particularly difficult to achieve. There is a wide use of microphones in our day to day life such as telephones, tape recorders, mobile phones etc. Microphones can be designed with different patterns and different impedances. Microphone array is an attractive approach in order to represent the array of microphone a system is constructed of a number of microphones distributed in space[17]. The output of each individual microphone is processed and all outputs are summed. In a reverberant environment, the microphone array improves the quality of sound capture [18]. Now-a-days, advanced technology in electrets provides low-cost, high-performance transducers. In addition, the increased computational power and decreased cost of digital signal processors (DSP’s) make computationally demanding systems implementable. By utilizing digital signal processing techniques, it is practical to design large scale microphone array systems for high quality sound capture for group conferences in large enclosure [19].

3.3 Microphone Polar Patterns:
The sensitivity of microphone to sound from various directions defines the property known as directivity [20]. The directivity depends on the inner elements of a microphone and describes the microphone's sensitivity to sound from various directions. Some microphones pick up sound equally from all directions, others pick up sound only from one direction or a particular
combination of directions. The types of directionality are divided into three main categories omni-directional, bi-directional and cardiod. The microphones different characteristics are shown in fig.3.

3.3.1 Omni-directional:
Microphones response in omni-directional microphone is generally considered to be a perfect sphere in three dimensions. Omni-directional microphone captures the sound coming from many directions. The mic position must retain the same and fixed while the sound source is moving. Although omni-directional microphones are very useful in the right situation, picking up sound from every direction is not usually what you need. Omni sound is very general and unfocused; if you are trying to capture sound from a particular subject or area it is likely to be overwhelmed by other noise.

3.3.2 Bi-directional:
Bi-directional microphone picks up sound equally from two opposite directions and uses a figure-of-eight pattern. We do not have a lot of situations which require the polar pattern of bi-directional microphone. In an interview the bi-directional microphone would be used with two people facing each other (with the mic between them) [20].

3.3.3 Cardiod:
Cardiod microphone picks up mostly from the front, but to a lesser extent the sides as well uses heart-shaped pattern. Cardiod microphone is used in emphasizing sound from the direction the mic is pointed while leaving some latitude for mic movement and ambient noise. The cardiod is a very versatile microphone, ideal for general use. Usually cardiod microphones are handheld [20].

![Fig.2: Patterns of different microphones](image)
3.4 Microphone Array:

The use of microphone array signal processing techniques in hearing aids is becoming increasingly widespread [21]. In a simple way array of microphones is just like a normal microphone but instead of having one microphone will have them in multiple to record the input signals. Microphones in the array work combine in a balanced way to record the sound simultaneously. The main advantage of using more than one microphone is that it helps in determining the position of the sound source by allowing the software that is processing the microphone signals. By analyzing the arrival times of the sound to each of the microphones in the array this can be achieved. For example if the sound arrives into the microphone on the right before it enters the microphone on left, then it comes to know that sound source is to right of the system. During the capture of sound, the microphone array software searches for the sound source and aims at making a beam in that direction. If the concerned sound source moves the capture beam will follow it eventually. It is like having two high directional microphones one being scanning the workspace measuring the sound level and other being pointing out to the direction with highest sound level i.e., is to the source of the sound. In addition to this the huge directivity of the microphone array reduces the surrounding noises and reverberation which results in the much clearer representation of the speaker’s voice [22].

3.5 Working Principle:

In a noisy place, hearing aids will amplify the noise as well as the desired speech signal. In any direct or indirect form microphone array processing consists of two main procedures one is sound source localization and beam former. First one helps in finding where the sound source is and should work certainly under reverberation and noisy conditions and tells the beam former where to focus the microphone array beam. There are many ways in finding the direction of the sound source’s and among them one is Time Delay Estimates (TDE) based methods which uses the facts that the sound source’s reaches the microphones at different times. Delays are easily calculated using the cross correlation function between the signals from the different microphones. Another method is to steer the beam and to measure the direction based on the maximum output signal. This method gives the similar results as that of the time delay estimate [23].
3.6 Array geometry of Microphone:

In order to calculate distance between source and microphones by using mathematical geometry formulae we consider a uniform linear array (ULA) consisting of two microphones which are placed in a three dimensional co-ordinate system with their respective coordinates as $m1 = (x_1, y_1, z_1)$; $m2 = (x_2, y_2, z_2)$ and also assumed the positions of the sources in the same 3-dimensional space. The distance between these microphones is 1.4 cm and it is arranged in such a way that to avoid aliasing in spatial frequency domain. The distance between the source and the microphones can be calculated from the formulae shown below [4]. Fig 5 shows the microphone array geometry.

$$Bm_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2}$$  \hspace{1cm} (1)$$

$$Bm_2 = \sqrt{(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2}$$  \hspace{1cm} (2)$$

$$Am_1 = \sqrt{(x_1^* - x_1)^2 + (y_1^* - y_1)^2 + (z_1^* - z_1)^2}$$  \hspace{1cm} (3)$$

$$Am_2 = \sqrt{(x_2^* - x_2)^2 + (y_2^* - y_2)^2 + (z_2^* - z_2)^2}$$  \hspace{1cm} (4)$$

where in fig (4) $A$ and $B$ denotes the sound sources and $x_1^*, y_1^*, z_1^*$ denotes the distances.
3.7 Fractional Delay:

Fractional delay filters (FD) [24] approach many deeps into the Digital signal processing applications i.e., in the field of speech coding and synthesis, communication, music technology. The standard applications of FD filter are time delay estimation, audio and music technology, speech coding and synthesis, time delay estimation etc [25]. It is not only the sampling frequency but also the sampling instants that play a huge role in these applications. Fractional delay filters provides their contribution building blocks that can be used for tuning the sampling instants i.e., executes the required band limited interpolation [6], which means a signal sample at any approximate point in time even though the point is situated in between two points. The FD filter is commonly applied in matching of data bits or symbols when dispatched through systems like digital modems. The main function at the receiving end is to find the appropriate dispatched data symbols as accurate as possible. Matching of the sampling frequency and sampling instants are mandatory for attenuating defective decision in digital communication, as it plays a vital role in concluding the decisions of the receiving bits or symbols by considering the samples from the incoming received continuous-time pulse sequence. Designing of the FD filters is to delay the input signals samples by a fractional amount of sampling period. As the delay is in fractional
amount, the inter-sample, performance of the genuine signal becomes too important. The expectation in designing the FD filter is that incoming consecutive time signals are fully band limited up to the nyquist frequency and it is constructed in discrete time domain.

3.8 Calculation for Fractional Delay:

Let the signal \( x(n) \) is delayed, and the delayed version of the signal

\[
x_d(n) = x(n - \beta)
\]  

(5)

The delay in sample can be calculated as

\[
\beta_{\text{samples}} = \frac{f_s d}{c}
\]  

(6)

where \( n \) is an integer or sample index, \( f_s \) is the sampling frequency, \( d \) is fractional delay and \( c \) is the velocity of sound i.e., is equal to 343.3 m/s.

The best way of implementing fractional delay is by utilizing sinc filter that shifted by the fractional amount [26]. This can be carried out using a standard FIR structure. In order to calculate the signal instants values at any point in time can be detected by using sinc interpolator according to Shanno’s sampling theorem [27]. So by convolving the delayed signal \( x_d(n) \) with \( \sin(k - \beta) \) to given signal. Sample \( \beta \) at any arbitrary time and \( k \) is a sample index.

\[
\gamma(\beta) = \sum_{n=-\infty}^{\infty} x_d(n) \sin(n - \beta)
\]  

(7)

The delayed sinc function is assigned to as ideal fractional delay interpolator [8]

\[
h_\beta(k) = \frac{\sin(\pi(n-\beta))}{\pi(n-\beta)}
\]  

(8)

![Fig.5: Representing Fractional Delay](image)
4. Beam Formation

4.1 Beam Formation Basics:

The basic idea of beam formation is to receive a signal radiating from a specific location and attenuate signals from other locations by adding the spatial dimension to filtering [28]. Beam formation is a spatial filtering technique that divides sound sources based on the position in space we have main lobe and side lobes where the main lobe is corresponding to the direction of interest, and the side lobe corresponding to the undesirable directions. Fig 7 shows the representation of beam formation.

![Fig. 6: Representation of beam forming](image)

In general beam former is used to estimate the signal arriving from a desired direction in the presence of noise and interfering signals, and also the beam former is used to optimize the reception of a target signal minimizing background noise. The principle of the beam formation technique mainly depends on the array of sounds arriving from only one particular direction to an array of microphones. Beam former plays a major role in our work as we used this to listen to the sounds particularly from one direction and making rest of them to ignore to some extent. Beam forming technique was originated in radio astronomy 1950’s as path of combining antenna information from collection of antenna dishes, it started to explore as a generalized signal processing in numerous applications involving spatially distributed sensors by 1970’s. Example of this expansion include sonar, to allow submarines greater ability to detect enemy ships using hydrophones or in geology, enhancing the ability of ground sensors to detect and locate tectonic plate shifts [17]. It was around during that time microphone arrays have become an active area of
research which made to keep the virtual microphone at some position instead of physical sensor movement. Applications of beam forming include laptops, hands free telephony, conference mikes etc. This is an important concept because it is not just used for array signal processing it is also used in many sonar systems as well. Fig 8 shows the layout of beam formation.

![Fig.7: Beam formation layout](image)

Now the signal is kept under windowed discrete fourier transform and then for the extraction of emergency signal we are using speech enhancement using beam formation technique. Beam forming techniques are broadly classified into two types: data-independent and data-dependent[29]. Data-independent or fixed beam formers are those algorithms where their parameters are fixed during operation. The delay-and-sum beam former is an example of fixed beam former. Data-dependent or adaptive beam formers continuously update their parameters based on the received input signals. The Griffiths-Jim beam former is an example of adaptive beam former [30].

### 4.2 Delay and Sum Beam former (DSB):

DSB is the simplest of all beam forming techniques. It is developed based on the idea that if a linear equi-spaced microphone array is being used then the output of each microphone will be the same except that each one will be delayed by a different amount. So, if the output of each mic is delayed appropriately, then we add all the outputs together. The desired signal that was propagating through the array will reinforce, while the noise or interference will tend to suppress. A block diagram of delay and sum beam former can be seen below in Fig.9.
Fig. 8: Delay and Sum Beam former

Based on the intra-element distance of the array the delays are calculated. In DSB the geometric arrangement of the microphones and weights associated with each microphone. The signal-to-noise ratio (SNR) of the output signal is greater than that of any individual microphones signal [30]. The major drawback of the DSB is the requirement of large number of microphones to improve the SNR. If the interference or noise signals are completely uncorrelated with the desired signal then, for every doubling of number of microphones will result in an additional 3dB of SNR increment. Another disadvantage is that, no nulls are being placed directly in the interference signal’s direction [30].

4.3 Adaptive Beam Former:

While rejecting interferences or noises coming from other directions a data-dependent beam forming technique adaptively filter the incoming signals in order to pass the signal from desired direction. An adaptive beam former is a data-dependent beam former that is able to separate signals collocated in the frequency band but separated in the spatial domain [31]. An adaptive beam former is able to automatically optimize the microphone array pattern by adjusting the elemental control weights until a prescribed objective function is satisfied. The choice of selecting the adaptive algorithm for deriving the adaptive weights plays key role in determining
the convergence speed and system complexity [30]. General adaptive algorithms that are being used for beam forming include LMS algorithm, NLMS algorithm and RLS algorithm. Each algorithm has its own advantages and disadvantages. A typical adaptive beam former structure can be seen in Fig.10.

The weight vector $\mathbf{w}$ can be chosen according to the input signal vector $\mathbf{x}(n)$. The main aim of the adaptive beam former is to optimize the system response with respect to the prescribed criteria, so that the output signal $y(n)$ will be free from noise or interference. The optimum adaptive criteria for beam former are Minimum Mean Square Error, Maximum Signal-to-Noise/Interference ratio and Minimum variance [31].

4.4 Wiener Beam Former:

The Wiener beam former is also referred to as minimum mean square error beam former. It is defined as the weights of the beam former which minimizes the mean square difference between the beam former output (when all sources are present) and a single microphone output (when only the desired signal is present) [32].

$$W_{opt} = \arg \min_{r \in \{1,2,...,N\}} E\{y[n] - s_r[n]^2\}$$  \hspace{1cm} (9)

The optimal weights that yield the optimum output signal is given by equation 31, the output $y[n]$, from the beam former is given by,

$$y[n] = \sum_{i=1}^{N} \sum_{j=0}^{L-1} w_i[j]x_i[n - j]$$  \hspace{1cm} (10)
where, \( L-1 \) is the order of the filters and \( w_i[j], j=0,1,\ldots,L-1 \) are the filter taps for \( i^{th} \) microphone. \( N \) is the number of microphones and \( x_i(n) \) is the \( i^{th} \) microphone observation.

\( s_r[n] \) in equation 31 is the single reference microphone observation when only desired signal is chosen as input. The optimal weights which minimize the mean square error between the output and reference signal is given by, [33]

\[
W_{opt} = [R_{ss} + R_{nn}]^{-1}r_s
\]  

(11)

where \( R_{ss} \) and \( R_{nn} \) are the auto correlation matrices of the signal of the signal of interest and noise respectively and are given by,[32]

\[
R_{ss} = \begin{bmatrix}
    r_{s1s1} & r_{s1s2} & \cdots & r_{s1sN} \\
r_{s2s1} & r_{s2s2} & \cdots & r_{s2sN} \\
    \vdots & \vdots & \ddots & \vdots \\
r_{Ns1} & r_{Ns2} & \cdots & r_{NsNs} \\
\end{bmatrix}
\]

(12)

\[
R_{nn} = \begin{bmatrix}
    r_{n1n1} & r_{n1n2} & \cdots & r_{n1nN} \\
r_{n2n1} & r_{n2n2} & \cdots & r_{n2nN} \\
    \vdots & \vdots & \ddots & \vdots \\
r_{Nn1} & r_{Nnn2} & \cdots & r_{NnN} \\
\end{bmatrix}
\]

(13)

\( r_s \) is the cross correlation vector and is given by,

\[
r_s = [r_1 \ r_2 \ \cdots \ r_N]^T
\]

(14)

where, each element is given by,

\[
r_{ir}[k] = E\{s_i[n]s_r[n+k]\}
\]

(15)

\( i = 1,2,\ldots,N, \quad r \in [1,2,\ldots,N], \quad k = 0,1,\ldots,L-1. \)

The performance of this wiener beam forming algorithm and the results will be presented in the following sections.
4.5 Generalized Side-Lobe Canceller (GSC):

The GSC is a simplification of the Frost Algorithm presented by Griffiths and Jim and some ten years after Frost’s original paper was published [34]. This section discusses the layout of the GSC and its implementations. GSC is the most frequent and achievable approach used in microphone applications. It is used to decrease the noises or interferences from non target location in array beam forming and can be used as an adaptive noise canceller in array processing [35]. The structure consists of two non-adaptive filters (wiener filters) and one adaptive filter (NLMS). The two non-adaptive filters which are at the down side of the below fig.11 are connected to the adaptive filters, which means the adaptive part is mixture of both adaptive and non adaptive filters. As the GSC is adaptive technique the weights keep on changing based on the input signal given. Adaptive techniques present a higher capacity at reducing noise interference but are much more sensitive to steering errors due to the approximation of the channel delays. The general layout of GSC is as shown below:

![Fig.10: Generalized Side-lobe Canceller](image-url)
5. Adaptive Algorithms

The block diagram of adaptive filtering is shown in fig 11:

5.1 Least Mean Square (LMS) Algorithm:

Basically the LMS algorithm [36] is being developed from steepest descent adaptive filter, which has a weight-vector update equation given as:

$$ w(n+1) = w(n) + \mu E\{e(n)x(n)\} $$

(17)

The signals are shown in fig 11

The LMS algorithm is most accepted adaptive algorithm due to its robustness, simplicity and its need of least computation power so it is widely used in many adaptive applications. This LMS is gradient search adaptive algorithm which tries to compute a set of filter weights that seeks to minimize mean square error. In this algorithm there is a tradeoff between misadjustment and speed of adaptation due to choice of step size[36]. So, this adjustment of step size it effects both convergence speed and residual error level. This convergence characteristic is analyzed with the objective to demonstrate the convergence factor that will ensure filter stability. The things get attracted to usage of LMS algorithm is its proof of convergence in stationary environment, its low computational complexity and its stability nature behavior when implemented with finite
precision arithmetic[36]. The entire algorithm is expressed in vector notation as shown below.

The LMS algorithm for a pth order FIR adaptive filter:

Parameters: \( P = \) filter order
\( \mu = \) step size

Initialization: \( \mathbf{w}(0) = \mathbf{0} \)

Computations: \( n = 0,1,2,3 \ldots \)

\[ \begin{align*}
    a) & \quad y(n) = \mathbf{w}^T(n)\mathbf{x}(n) \\
    b) & \quad e(n) = d(n) - y(n) \\
    c) & \quad \mathbf{w}(n + 1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n)
\end{align*} \]  

where \( y(n) \) is estimated output of the filter, \( e(n) \) is error, \( x(n) \) is the input signal.

The term \( \mu \) is known as step size which is computed as given below:

\[ 0 < \mu < \frac{2}{\lambda_{\text{max}}} \]  

where \( \lambda_{\text{max}} \) is maximum eigen value of correlation matrix \( \mathbf{R} \) of input signal.

\[ \mathbf{R} = \begin{bmatrix}
    r(0) & r(1) & r(2) & \ldots & r(10) \\
    r(1) & r(0) & r(1) & \ldots & r(9) \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    r(10) & r(9) & r(8) & \ldots & r(0)
\end{bmatrix} \]  

5.2 Normalized Least Mean Square (NLMS) Algorithm:

In order to converge rapidly i.e., convergence speed of LMS algorithm without any knowledge of estimation of input process correlation matrix, a new variable convergence factor introduced instead of constant factor according to eq.(5). So, this normalized LMS algorithm converges faster than traditional LMS algorithm as it utilizing variable convergence factor for aiming at minimization of instantaneous output error. In this algorithm we employ a variable convergence factor \( \mu(n) \) in the updating equation of traditional LMS Algorithm then we improve the convergence rate the updating equation is given by eq.(1). So replacing \( \mu \) in LMS by \( \mu(n) \) which results into NLMS algorithm [6].

\[ \mathbf{w}(n + 1) = \mathbf{w}(n) + \mu(n) e(n) \mathbf{x}(n) \]  

where \( \mu(n) \) must be chosen with an aim of achieving a faster convergence.

\[ \mu(n) = \frac{\beta}{\mathbf{x}^H(n)\mathbf{x}(n)} = \frac{\beta}{\|\mathbf{x}(n)\|^2} \]
where $\beta$ is normalized step size of $0 < \beta < 2$.

The noise amplification problem is diminished by using the NLMS algorithm due to the normalized step size. NLMS algorithm is analyzed as given below:

Parameters: $P =$ filter order

$\mu(n) =$ step size

Initialization: $w(0) = 0$

Computations: $n = 0, 1, 2, 3 \ldots$

\begin{align*}
a) \ y(n) &= w^T(n)x(n) \\
b) \ e(n) &= d(n) - y(n) \\
c) \ w(n + 1) &= w(n) + \beta \frac{x(n)}{\|x(n)\|^2} e(n)
\end{align*}

\section*{5.3 Leaky LMS Algorithm:}

By introducing leaky factor we can successfully stabilize the system and in alleviating ‘stalling’ of adaptive coefficients due to very low input signal and quantization effects[37]. If the autocorrelation matrix of the input signal is ill conditioned i.e., having one of its eigen values close to zero lead to the undammed modes in the LMS algorithms so we introduce leaky factor ($\gamma$) into autocorrelation matrix of input signal $x(n)$ so we stabilize adaptive filter coefficients. This algorithm is known as LLMS Algorithm [38]. The update equation for the LLMS algorithm is given by:

\begin{equation}
\begin{aligned}
    w(n + 1) &= (1 - \mu \gamma)w(n) + e(n)x(n)
\end{aligned}
\end{equation}

Here $\gamma$ is the leakage coefficient which has limits $0 < \gamma \ll 1$. Excess parameter drift can be avoided by selecting proper value of positive leakage parameter $\gamma$ (Leaky factor). LLMS algorithm is analyzed as shown below:

Parameters: $P =$ filter order

$\mu =$ step size

$\gamma =$ leakage factor

Initialization: $w(0) = 0$

Computations: $n = 0, 1, 2, 3 \ldots$

\begin{align*}
a) \ y(n) &= w^T(n)x(n) \\
b) \ e(n) &= d(n) - y(n) \\
c) \ w(n + 1) &= (1 - \mu \gamma)w(n) + e(n)x(n)
\end{align*}
5.4 Recursive Least Squares (RLS) Algorithm:

In gradient descent algorithm such as LMS and its based algorithms require the knowledge of autocorrelation of input signal $x(n)$ and cross correlation of input process and desired output signal $d(n)$. But this statistical knowledge may be unknown sometimes so we have an alternative to find minimization of mean square error by recursive least square algorithm [37] so this algorithm utilizes least square approach so by this approach it minimizes least square error. So this least squares approach method whose objective is minimizing the sum of squares of difference between model filter output and desired signal. This is analyzed by updating new set of samples of incoming signals which are received at every iteration then the solution for least squares problem can be calculated in recursive form resulting in Recursive Least Squares (RLS) algorithm i.e., obtained by providing negative feedback of error signal $e(n)$ and desired signal $d(n)$. The weight update equation is given by:

$$w(n) = w(n-1) + \alpha(n)g(n)$$

$$\alpha(n) = d(n) - w^T(n-1)x(n)$$

which is the difference between desired signal $d(n)$ and the estimate of the $d(n)$ that is formed by applying the $w(n-1)$ set of filter coefficients to the new data vector $x(n)$. Recursive least curve algorithm minimizes weighted least square error at time ‘$n$’ given by the equation and ‘$\alpha$’ is the general constant.

$$E(n) = \sum_{i=0}^{n} \lambda^{n-i} |e(i)|^2$$

Parameters: $P =$ filter order
$\lambda =$ exponential weighing factor
$\delta =$ Value used to initialize $P(0)$

Initialization: $w(0) = 0$

$P(0) = \delta^{-1}I$

Computations: For $n = 1,2,3 \ldots$ compute

a) $z(n) = P(n-1)x(n)$

b) $g(n) = \frac{1}{\lambda + x^T(n)x(n)} z(n)$

c) $w(n) = w(n-1) + \alpha(n)g(n)$

d) $P(n) = \frac{1}{\delta} [P(n-1) - g(n)z^H(n)]$
6. Direction of Arrival

As we know that there is a one-to-one relationship between the direction of a signal and the associated received steering vector [5]. It should be therefore possible to invert the relationship and estimate the direction of a signal from the received signals. For estimation of direction of arrival an antenna array should be provided and we also know that there is a Fourier relationship between the beam pattern and the excitation of the array. This allows the direction of arrival (DOA) estimation problem to be treated as equivalent to spectral estimation. The problem setup shown in Fig. 12 has several (M) signals impinge on a linear, equi-spaced, array with N elements, each with direction \( \phi_i \). The goal of DOA is to use the data received at the array to estimate the value of \( \phi_i, i = 1, 2, ..., M. \)

\[ x = \sum_{m=1}^{M} \alpha_m S(\phi_m) + n \]  

(39)

![Fig.12: DOA estimation problem](image)

There are several methods to estimate the number of signals and their directions. But as per theoretical purpose we describe five techniques generally: correlation, maximum likelihood, MUSIC, ESPRIT, and Matrix Pencil. Here we are using music algorithm to detect the signal.

6.1 MUSIC Algorithm:

MUSIC stands for Multiple Signal Classifier[39]. MUSIC algorithm is probably the most popular technique of all the techniques used for knowing the direction of arrival. With the help of MUSIC algorithm we are able to find direction of multiple sources in one shot. MUSIC algorithm is independent on the correlation matrix of the data. We consider a model of M signals incident on the array, computed by noise, i.e.,
where $s(\phi)$ is the steering vector of the signal whose direction ($\phi$) has to be estimated and $n$ is zero-mean Gaussian with covariance $\alpha^2 I$. Here the goal is to estimate $\phi_m$, $m = 1, \ldots M$. The easiest way to estimate the angles is through correlation. Using this data in the model we will compute the following:

$$x = S\alpha + n$$

(40)

$$S = [s(\phi_1) \ s(\phi_2) \ldots s(\phi_M)]$$

(41)

$$\alpha = [\alpha_1 \ \alpha_2 \ \ldots \ \alpha_M]^T$$

(42)

The matrix $S$ is a $N \times M$ matrix of the $M$ steering vectors. Assuming that the different signals to be uncorrelated, the correlation matrix of $x$ can be written as:

$$R = E[xx^H],$$

(43)

$$= E[S\alpha\alpha^H S^H] + E[nn^H],$$

(44)

$$= SAS^H + \alpha^2 I,$$

(45)

$$= R_s + \alpha^2 I,$$

(46)

where,

$$R_s = SAS^H$$

(47)

$$A = \begin{bmatrix}
E[|\alpha_1|^2] & 0 & \ldots & 0 \\
0 & E[|\alpha_2|^2] & \ldots & 0 \\
0 & 0 & \ldots & E[|\alpha_M|^2]
\end{bmatrix}$$

(48)

The signal covariance matrix $R_s$ is clearly a $N \times N$ matrix with rank $M$. It therefore has $N - M$ eigenvectors corresponding to the zero eigenvalue. Let $q_m$ be such an eigenvector. Therefore,

$$R_s q_m = SAS^H q_m = 0$$

(49)

$$q_m^H SAS^H q_m = 0,$$

(50)

$$S^H q_m = 0$$

(51)

where this final equation is valid since the matrix $A$ is clearly positive definite. Eq. (51) implies that all $N - M$ eigenvectors ($q_m$) of $R_s$ corresponding to the zero eigenvalue are orthogonal to all $M$ signal steering vectors. This is the basis for MUSIC. Call $Q_n$ the $N \times (N - M)$ matrix of these eigenvectors. MUSIC plots the pseudo-spectrum:

$$P_{\text{MUSIC}}(\phi) = \frac{1}{\sum_{m=1}^{N-M} |s^H(\phi)q_m|^2} = \frac{1}{s^H(\phi)Q_nQ_n^H s(\phi)} = \frac{1}{\|Q_n^H s(\phi)\|^2}$$

(52)

Note that since the eigenvectors making up $Q_n$ are orthogonal to the signal steering vectors, the denominator becomes zero when $\phi$ is a signal direction. Therefore, the estimated
signal directions are the $M$ largest peaks in the pseudo-spectrum. However, in any practical situation, the signal covariance matrix $R_s$ would not be available. The most we can expect is to be able to estimate $R$ the signal covariance matrix. The key is that the eigenvectors in $Q_n$ can be estimated from the eigenvectors of $R$.

For any eigenvector $q_m \in Q$,\[ R_s q_m = \lambda q_m \]
\[ \Rightarrow R q_m = R_s q_m + \sigma^2 I q_m, \]
\[ = (\lambda + \sigma^2) q_m, \quad (53) \]
i.e., any eigenvector of $R_s$ is also an eigenvector of $R$ with corresponding eigenvalue $\lambda + \sigma^2$. Let $R_s = Q \wedge Q^H$. Therefore,\[ R = Q[\Lambda + \sigma^2 I]Q^H \]
\begin{equation}
\begin{bmatrix}
\lambda_1 + \sigma^2 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & \lambda_2 + \sigma^2 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_M^2 + \sigma^2 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & \sigma^2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & \cdots & \sigma^2 \\
\end{bmatrix} \end{equation}
(54)

Based on this eigendecomposition, we can partition the eigenvector matrix $Q$ into a signal matrix $Q_s$ with $M$ columns, corresponding to the $M$ signal eigenvalues, and a matrix $Q_n$, with $(N - M)$ columns, corresponding the noise eigenvalues ($\sigma^2$). Note that $Q_n$, the $N \times (N - M)$ matrix of eigenvectors corresponding to the noise eigenvalue ($\sigma^2$), is exactly the same as the matrix of eigenvectors of $R_s$ corresponding to the zero-eigenvalue. This is the matrix used in Eq. (52). $Q_s$ defines the signal subspace, while $Q_n$ defines the noise subspace.

There are few important observations to be made:

- The smallest eigenvalues of $R$ are the noise eigenvalues and are all equal to $\sigma^2$, i.e., one way of distinguishing between the signal and noise eigenvalues (equivalently the signal and noise subspaces) is to determine the number of small eigenvalues that are equal.
- By orthogonality of $Q$, $Q_s \perp Q_n$.
Using the final two observations, we see that all noise eigenvectors are orthogonal to the signal steering vectors. This is the basis for MUSIC. Consider the following function of $\phi$:

$$P_{\text{MUSIC}}(\phi) = \frac{1}{\sum_{m=-M+1}^{N} |q_m^H s(\phi)|^2} = \frac{1}{s^H(\phi)Q_n Q_n^H s(\phi)'},$$

(55)

where $q_m$ is one of the $(N - M)$ noise eigenvectors. If $\phi$ is equal to DOA one of the signals, $s(\phi) \perp q_m$ and the denominator is identically zero. MUSIC, therefore, identifies as the directions of arrival, the peaks of the function $P_{\text{MUSIC}}(\phi)$. 

7. Neural Networks

7.1 Back Propagation Algorithm:

In neural networks the determination of weights ‘w’ is done by learning algorithm. For neural network training back propagation is one of many algorithms. The term back propagation algorithm is defined as “A common method of training a neural network in which the initial system output is compared to the desired output, and the system is adjusted until the difference between the two is minimized”. Back Propagation network is considered as the perfect example of the neural network[40]. As just mentioned, to train the network you need to give it examples of what you want— the output you want (called the Target) for a particular input as shown in Fig.13.

![Fig.13: Back Propagation Training set](image)

So, if we put in the first pattern to the network, we would like the output to be 0 1 as shown in figure 14 (a black pixel is represented by 1 and a white by 0 as in the previous examples). The input and its corresponding target are called a Training Pair[41].

Fig.14 explains applying a training pair to a network.
7.2 Training Procedure:

In the network first we initialize by setting up all its weights to be small random numbers (say between 0 and +1). Next, the input pattern is applied and then the output is calculated. The calculation will give an output that is completely different to what you want (the Target), since all the weights are random. We then calculate the error of each neuron, which is essentially: \( \text{Target} - \text{Actual Output} \). This error is then used mathematically to change the weights in such a way that the error will get smaller. In other words, the Output of each neuron will get closer to its Target (this part is called the reverse pass). The process is repeated again and again until the error is minimal [41].

We’ll just look at one connection initially, between a neuron in the output layer and one in the hidden layer, fig.15.
7.3 Working of Algorithm:

Fig.15 shows a single connection learning in a Back Propagation network.

![Diagram of a Back Propagation network with connections between neurons A, B, and C.](image)

The connection we’re interested in is between neuron A (a hidden layer neuron) and neuron B (an output neuron) and has the weight $W_{AB}$. The diagram also shows another connection, between neuron A and C (next output neuron).

1. First apply the inputs to the network and work out the output – remember this initial output could be anything, as the initial weights were random numbers.

2. Next work out the error for neuron B. The error is *What you want – What you actually get*, in other words:

   $$\text{Error}_B = \text{Output}_B (1-\text{Output}_B)(\text{Target}_B - \text{Output}_B)$$

   The “Output(1-Output)” term is necessary in the equation because of the Sigmoid Function – if we were only using a threshold neuron it would just be $(\text{Target} - \text{Output})$. Sigmoid Function: Sigmoid function provides more flexibility while processing in hidden layer. Sigmoid function used for computational procedure.

3. Change the weight. Let $W_{AB}^+$ be the new (trained) weight and $W_{AB}$ be the initial weight.

   $$W_{AB}^+ = W_{AB} + (\text{Error} \times \text{Output}_A)$$

   Notice that it is the output of the connecting neuron (neuron A) we use (not B). We update all the weights in the output layer in this way.

4. Calculate the Errors for the hidden layer neurons. Unlike the output layer we can’t calculate these directly (because we don’t have a Target), so we *Back Propagate* them from the output layer (hence the name of the algorithm). This is done by taking the Errors from the output neurons and running them back through the weights to get the hidden layer errors. For
example if neuron A is connected as shown to B and C then we take the errors from B and C to generate an error for A.

\[ \text{Error}_A = \text{Output}_A (1 - \text{Output}_A)(\text{Error}_B W_{AB} + \text{Error}_C W_{AC}) \]

Again, the factor “\( \text{Output} (1 - \text{Output}) \)” is present because of the sigmoid squashing function.

5. Having obtained the Error for the hidden layer neurons now proceed as in stage 3 to change the hidden layer weights. By repeating this method we can train a network of any number of layers.
8. Simulation and Results

8.1 Experimental Setup:

The experimental set up describes the input signal which is an emergency signal and there exists a traffic noise which can be observed while travelling in heavy traffic therefore it is considered as a traffic noise.

Fig.16: Experimental Setup

As the emergency signal and the traffic noise will be mixed and the desired emergency signal can be extracted by performing different adaptive algorithms LMS, NLMS, LLMS, RLS algorithms. Fig 17 (a) shows the Emergency Signal, Fig 17 (b) shows the Traffic Noise, Fig 17 (c) shows the signals combined with Emergency signal and Traffic noise.

Fig.17: Input Signals (a) Emergency signal (b) Traffic noise (c) Emergency signal with Traffic noise
Fig. 18 shows the Power Spectral Density of Emergency Signal, Power Spectral Density of traffic noise and Power Spectral Density of the combination of emergency signal and traffic noise with respect to the frequency (Hz) and the power (dB).

The extraction of the desired emergency signal can be processed by the adaptive filters like LMS, NLMS, LLMS and RLS algorithms as proposed in previous chapters. Fig 19 (a) shows the LMS Algorithm, Fig 19 (b) shows the NLMS Algorithm, Fig 19 (c) shows LLMS algorithm and Fig 19 (d) shows the RLS algorithm.
Fig. 19: Algorithms Output (a) LMS algorithm (b) NLMS algorithm (c) LLMS algorithm (d) RLS algorithm

Fig. 20 shows the Power Spectral Density of LMS, NLMS, LLMS and RLS algorithms with respect to the frequency (Hz) and power (dB).

In order to compare different algorithms with the input signals from fig 18 the power spectral densities of LMS, NLMS, LLMS and RLS algorithms has been shown in fig 19. We can clearly see from the fig 20 that the adaptive algorithms LMS and LLMS cannot filter both the emergency signal and the traffic noise more robustly but by using NLMS and RLS algorithms can be filtered more robustly. But due to more complexity in RLS algorithm the NLMS algorithm has been preferred.
The complexity and time taken for the execution of the system is more for RLS when compared to NLMS. So we prefer NLMS algorithm instead of RLS algorithm.

8.2 Experimental Block Diagram:

The emergency signal (source signal) and the environmental noise (noise interference) is fed to MIC 1 and MIC 2 (i.e., MIC 1 and MIC 2 consists of both emergency signal mixed with environmental noise). These two microphone’s (MIC 1 and MIC 2) were steered by using fractional delay between the microphones. A powerful feature of the beam forming array is the ability to electronically steer the beam pattern without physically moving the array. This is simply achieved by adding a delay stage to each of the array elements. The steered output is fed to the beam former 1 and beam former 2. The steering process is explained in Fig.9. The basic idea of beam forming is to concentrate on the array of sounds arriving from one particular
direction to an array of microphones. The beam former thus obtained is fed to GSC-NLMS where the noise is filtered and therefore the desired emergency signal is obtained.

![General Experimental block diagram](image)

**Fig.21: General Experimental block diagram**

### 8.3 Microphone Array Setup:

The evaluation of results has been done by choosing an emergency signal at MIC 1 and MIC 2 and traffic noise at three different cases entirely in Matlab environment. These two signals are of 309888 samples of data. Among these two signals one is considered as a main speech and the other as noise. The main aim of our work is to suppress the noises. The tests are performed at various SNR inputs by changing the noise power based on the equations 58 to 59. For the evaluation, there are three different consideration points in which the distance in meters from where the source and noises originates:

**Case 1:** \( S = [1.2,1.3,0.6], \) MIC 1 = [1.1,1,0.8], MIC 2 = [1.2,1,0.8], N1 = [0.7,0.9,1.4];

**Case 2:** \( S = [2.2,2.3,1.6], \) MIC 1 = [2.1,2,1.8], MIC 2 = [2.2,2.1,1.8], N1 = [1.7,1.9,2.4];

**Case 3:** \( S = [3.2,4.3,2.6], \) MIC 1 = [3.6,1.4,2.1], MIC 2 = [2.3,4,1.2,8], N1 = [2.7,2.9,3.4];

For these three different consideration points the analysis were achieved at their respective systems i.e., at WBF 1 and WBF 2 is combined with NLMS and their respective results by varying input SNR values from 1,5,9,13,17,21,25 are shown in following sections and their values are tabulated respectively. SNR is calculated based on the equations from 58 to 59.
Case 1:
For Case 1, the main source is nearer to the microphone position and the noise is far from the source position, at this point the performance of the system is effective in cancelling the noise with an SNR of 4.2601 dB at WBF 1 and the output SNR is obtained at 8.9868 dB respectively. As the SNR at the wiener beam former decreases the output SNR increases from 8.9868 to 9.1183.

Fig. 22 (a) shows Case 1 set up Fig. 22 (b) shows SNR values Fig. 22 (c) shows Graphical representation.
Fig. 22: Case 1 (a) Setup (b) SNR values (c) Graphical representation

### Table: SNR Values

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Input SNR (dB)</th>
<th>SNR at WBF 1 (dB)</th>
<th>Out Put SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4.2061</td>
<td>8.9868</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4.0654</td>
<td>9.0329</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>3.9167</td>
<td>9.0702</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>3.5693</td>
<td>9.0980</td>
</tr>
<tr>
<td>17</td>
<td>17</td>
<td>3.447</td>
<td>9.1124</td>
</tr>
<tr>
<td>21</td>
<td>21</td>
<td>3.5608</td>
<td>9.1175</td>
</tr>
<tr>
<td>25</td>
<td>25</td>
<td>3.7317</td>
<td>9.1163</td>
</tr>
</tbody>
</table>
Case 2:
For Case 2, the main source is nearer to the microphone position and the noise is far from the source position, at this point the performance of the system is effective in cancelling the noise with an SNR of 4.2601 dB at WBF 1 and the output SNR is obtained at 8.9868 dB respectively. As the SNR at the wiener beam former decreases the output SNR increases from 8.9868 to 9.1183.

Fig.23 (a) shows Case 2 Set up Fig.23 (b) shows SNR values Fig.23 (c) shows Graphical representation.
Fig. 23: Case 2 (a) Setup (b) SNR values (c) Graphical representation

<table>
<thead>
<tr>
<th>Input SNR (dB)</th>
<th>SNR at WBF 1 (dB) (y1)</th>
<th>Output SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.2061</td>
<td>8.9868</td>
</tr>
<tr>
<td>5</td>
<td>4.0654</td>
<td>9.0329</td>
</tr>
<tr>
<td>9</td>
<td>3.9167</td>
<td>9.0702</td>
</tr>
<tr>
<td>13</td>
<td>3.5693</td>
<td>9.0980</td>
</tr>
<tr>
<td>17</td>
<td>3.447</td>
<td>9.1124</td>
</tr>
<tr>
<td>21</td>
<td>3.5608</td>
<td>9.1175</td>
</tr>
<tr>
<td>25</td>
<td>3.7317</td>
<td>9.1183</td>
</tr>
</tbody>
</table>
Case 3:
For Case 3, the main source is nearer to the microphone position and the noise is far from the source position, at this point the performance of the system is effective in cancelling the noise with an SNR of 4.3159 dB at WBF 1 and the output SNR is obtained at 8.2372 dB respectively. As the SNR at the wiener beam former decreases the output SNR increases from 8.2372 to 8.3209.
Fig.24 (a) shows Case 1 Set up Fig.24 (b) shows SNR values Fig.24 (c) shows Graphical representation.
Fig. 24 Case 3 (a) Setup (b) SNR values (c) Graphical representation
8.4 Wiener Beam Former Implementation:

We consider two wiener beam former systems WBF1 and WBF2. All the signals from different positions will reach the microphones, so the output of microphone consists of mixture of all the signals. These outputs from the microphone array will encounter with wiener filters \( W_{11} \) and \( W_{12} \). The outputs from the two different wiener beam former’s is \( Y_1(n) \) and \( Y_2(n) \) respectively. Fig.25 shows the Wiener Beam former implementation.

At Wiener beam former 1 (WBF1), one of the signal from the input \( W_{11} \) are taken as main signal and the other signal \( W_{12} \) is considered as the traffic noise which means that the output \( Y_1(n) \) resembles the desired signal, while the rest of the signal \( W_{12} \) is considered to be noise which gets attenuated or cancelled. The same procedure is done in the next Wiener beam former 2 (WBF2) respectively, but by considering the rest of signal \( W_{11} \) as main signal component in (WBF2) case and their respective output is \( Y_2(n) \) respectively. The outputs from \( Y_2(n) \) are given to the NLMS and it will be discussed in the following section.

Fig.26 (a) shows the output at Mic1, Fig.26 (b) shows the output at Mic2, Fig.26 (c) shows the output at WBF1 (\( Y_1(n) \)) and Fig.26 (d) shows the output at WBF2 (\( Y_2(n) \)).
8.5 Implementing GSC for the system:

This structure shown in the fig.11 express WBF based GSC. The preferred system consists of two non adaptive filters (Wiener filters) and one adaptive filter (NLMS) along with a microphone array. The numbers of microphones used are two. The input to the micro phones is a combination of emergency signal, and traffic noise. They reach the microphones with delay as they originate from different positions. The mixture of signals from the microphones is given to the Wiener Beam formers (non-adaptive filters). The top branch of the structure produces beam formed signal which is a fixed one and its output is taken as Y(n) and the other branch also
produces the beam formed signal respectively. The outputs from the second Wiener Beam former are given to the adaptive part which consists of NLMS algorithm. The description and concerned equations of the NLMS are discussed in the previous chapters. As the GSC is adaptive technique the weights keep on changing based on the input signal given. Adaptive technique (GSC) acts as a higher capacity at reducing noise interference but the adaptive techniques are much more sensitive to steering errors due to the approximation of the channel delays. Fig.27 shows the output of the system comparison with the input signal from fig.17.

The output from the adaptive filter is $y_b$

$$y_b(n) = \sum_{k=1}^{2} w_k^T(n)\nu_k(n)$$  \hspace{1cm} (56)

The output of the GSC is given by

$$Y(n) = y_1(n) - y(n)$$  \hspace{1cm} (57)

Fig.27: Emergency Signal at the system output
Fig. 28 shows the structure of GSC implementation.

8.6 Signal to Noise Ratio:

Objective test is used to measure the performance of the above mentioned system at WBF 1 and at system output. Signal to Noise Ratio (SNR) is calculated for them. Signal to Noise Ratio is defined as the variance of the output power to the variance of the noise power.

\[
SNR = \frac{signal\ output\ power}{noise\ output\ power}
\]  

\[
SNR = 10 \log_{10} \left( \frac{var(speech)}{var(noise)} \right)
\]  

\[
(58)
\]

\[
(59)
\]
8.7 Direction of Arrival:

Part 1: This part is designed to validate the working of the MUSIC algorithm. The program designed is simulating the input wave that is arriving at the array in the form of planar wave fronts (the source is assumed sufficiently far away). The array itself is assumed to be an $m$ element linear array with no directional properties in the given plane. The number of elements can be varied. Further the frequency of the input can be changed. dfactor is the ‘division factor’ of the wavelength that represents the direction between the elements (for eg: if d factor = 2, distance between the elements = $\lambda/2$). Thus we have to handle both the frequency and the distance between the elements in terms of the given frequency. The samples variable represents the number of samples that are considered for the given input and $f_s$ represent the sampling frequency. These two variables can be adjusted to simulate the processor available, the amount of accuracy and the system refresh rate. For example, if we choose a high value of samples, amount of memory utilization increases and the refresh rate (rate between consequent measurements) decreases however the accuracy of calculations increases (assuming input signal lasts for that long a time).

Part 2: The array which we used is to define the gain and phase change provided by the array elements to a given signal direction. The array is a function of $\theta$ and is saved.

Part 3: This part explains implementation of MUSIC algorithm. The function returns an estimate of the direction of arrival. From the input matrix $x$, the number of elements and the samples is obtained obviously. The first step is the calculation of the co-variance matrix. The next step involves the calculation of the eigen values and eigenvectors of $S$. On knowing the minimum eigen value, the multiplicity of eigenvalue is obtained. The multiplicity is being obtained using a naive comparison. However, the final aim is to use the MDL criterion to find the number of incident signals. The next step is to obtain the noise eigenvector matrix of these minimum eigen values and form the matrix. Using the calculated values from the above matrix we can plot to get direction of arrival in terms of azimuthal angle and elevation angle.
Part 4: Enter the number of signals as 1 and the angle in degrees in this case as 54. Obtain the values of the highest d peaks to get the direction of arrival. The below graph explains the degrees from 0 to 360 degrees on x-axis and the value of PMU(\theta) on y-axis. The peaks are obtained at 47.36 at y-axis. So in this case the direction is the highest peak value is 47.36°.

\[\text{Fig.29: Plot of Direction of Arrival}\]

8.8 Neural Networks using Back Propagation:

The neural networks has been trained with four single hidden layers. The single hidden layers each with 30 neurons. The neural network A is trained with 120 iterations, neural network B is trained with 250 iterations, neural network C is trained with 150 iterations, and neural network D is trained with 300 iterations. A graph with error function calculated for each neural network is shown in Fig.31. The error graph shows after some iteration the amplitude falls to zero in such a way that the desired output is recognized. Different types of emergency signals like ambulances, siren, police, and fire-fighting vehicles have been trained to the neural
networks.

Fig. 30 shows the flow chart to find the error.
Fig. 31: Error graph
9. Conclusion and Future work

9.1 Conclusion:

This work is focused on the design and implementation of WBF based generalized side lobe canceller for the enhancement of emergency signal from the noisy atmosphere. The system has been implemented and its performance has been analyzed by considering an emergency signal and environmental noise (traffic noise) with output SNR and wiener beam former output evaluation of the proposed system.

Simulation results have been analyzed for different main source position’s and noise sources at three different cases. The results obtained from the algorithm outputs shows that origination of main signal and the noise source play a major role in effective cancelling of the noise. When the main signal is nearer to the microphone array than that of noise field, then both SNR at WBF 1 and at System output (e) is high and better when compared with, the noise fields which are nearer to the microphone array respectively.

The comparison of different algorithms has been carried out by showing the power spectral density in which NLMS and RLS algorithm has been found robust in the noise cancellation where as RLS is more complex than NLMS as the execution takes longer time compare to NLMS algorithm. Therefore, NLMS algorithm has been preferred.

9.2 Future work:

In this procedure we used two microphones for the continuous detection of the external signal for finding out the emergency signal and the entire procedure went on calculating the delay between the two microphones but in the future work there are areas that require further research, showing promise for improved results using a microphone array structure. However, in a real environment, multiple reflections leading to multiple target DOA’s can be implemented further. Additional microphones may improve overall performance, especially in the GSC. Apart from computer simulations, real time implementations are also necessary to check the beam formers performances.
10. Bibliography

REFERENCES:


[12] Fa-Long Luo and Bernard Widrow "Microphone arrays for hearing aids: An overview " Packard Electrical Engineering Building, Room 273, Stanford University, 350 Serra Mall, Stanford, CA 94305, USA b Quicksilver Technology, Inc., 6640 Via Del Oro, San Jose, CA 95119, USA.


