3-2-2021

Underwater Localization in a Confined Space Using Acoustic Positioning and Machine Learning

Amir Horri
University of Windsor

Underwater Localization in a Confined Space

Using Acoustic Positioning and Machine Learning

By

Amir Horri

A Thesis

Submitted to the Faculty of Graduate Studies

through the Department of Electrical and Computer Engineering

in Partial Fulfillment of the Requirements for

the Degree of Master of Applied Science

at the University of Windsor

Windsor, Ontario, Canada

2020

Underwater Localization in a Confined Space

Using Acoustic Positioning and Machine Learning

by

Amir Horri

APPROVED BY:

_____

J. Ahamed

Department of Mechanical, Automotive & Materials Engineering

_____

S. Erfani

Department of Electrical and Computer Engineering

_____

M. Saif, Co-Advisor

Department of Electrical and Computer Engineering

_____

S. Alirezaee, Co-Advisor

Department of Electrical and Computer Engineering

December 17, 2020

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Localization is a critical step in any navigation system. Through localization, the vehicle can estimate its position in the surrounding environment and plan how to reach its goal without any collision. This thesis focuses on underwater source localization, using sound signals for position estimation. We propose a novel underwater localization method based on machine learning techniques in which source position is directly estimated from collected acoustic data. The position of the sound source is estimated by training Random Forest (RF), Support Vector Machine (SVM), Feedforward Neural Network (FNN), and Convolutional Neural Network (CNN). To train these data-driven methods, data are collected inside a confined test tank with dimensions of 6m x 4.5m x 1.7m.

The transmission unit, which includes Xilinx LX45 FPGA and transducer, generates acoustic signal. The receiver unit collects and prepares propagated sound signals and transmit them to a computer. It consists of 4 hydrophones, Red Pitay analog front-end board, and NI 9234 data acquisition board. We used MATLAB 2018 to extract pitch, Mel-Frequency Cepstrum Coefficients (MFCC), and spectrogram from the sound signals. These features are used by MATLAB Toolboxes to train RF, SVM, FNN, and CNN. Experimental results show that CNN archives 4% of Mean Absolute Percentage Error (MAPE) in the test tank. The finding of this research can pave the way for Autonomous Underwater Vehicle (AUV) and Remotely Operated Vehicle (ROV) navigation in underwater open spaces.

DEDICATION

I would like to dedicate my work to my family, relatives and friends. A special feeling of gratitude to my loving parents, Samad Horri and Masomeh Kalhori whose words and their existence are the source of encouragement that inspires me throughout my life. Without their blessing and sacrifices, I would not have made it to this stage.

I dedicate my work to my loving wife, Mahsasadat Seyedbarhagh for her support and love. Thank you so much for being engaged with all struggles and discomfort that I may given you during my studies, your practical and emotional support as a role of wife and who was also a full-time student is truly appreciated and your unique help to complete my work will never be forgotten. Thank you for everything you do for me. I love you.

I wish to give my special thanks to my brother, Ali Horri for his support and assistance by providing me with many constructive guidance, directions and feedbacks throughout my life.

There are friends, there is family and then there are friends that become family. I would like to thank Arash Akhtari, Luca Saeid, Dr. Abdolrazagh Hashemi Shahraki, Ben Verdezoto, Dionte Williams and Heidi Clark for their encouragement and exceptional support. Your constant support and encouraging words have always uplifted me.

ACKNOWLEDGMENT

I would like to sincerely thank my co-supervisor, Dr. Shahpour Alirezaee, for his guidance and support in successfully completing my thesis. I am deeply grateful for his involvement, guiding, mentoring and providing any help that I needed to complete my degree. It is an honor to have worked under his supervision.

I am grateful to my co-supervisor, Dr. Mehrdad Saif, for his support and valuable comments which helped in completing this thesis.

I would also like to thank my committee members, Dr. Shervin Erfani and Dr. Jalal Ahamed for their encouragement, constructive comments and positive criticism which in fact, improved my ideas and solutions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS/SYMBOLS

Abbreviations/Symbols	**Description**
AUV	Autonomous Underwater Vehicle
GPS	Global Positioning System
INS	Inertial Navigation System
LBL	Long Baseline
NN	Neural Networks
ROV	Remotely Operated Vehicle
SBL	Short Baseline
ToA	Time of Arrival
TDOA	Time Difference of Arrival
TOF	Time of Flight
USBL	Ultra Short Baseline
UV	Underwater Vehicle
RF	Random Forest
SVM	Support Vector Machine
FNN	Feedforward Neural Network
CNN	Convolutional Neural Network
INS	Inertial Navigation System
LTI	Linear Time Invariant
BPSK	Binary Phase Shift Keying
GRNN	Generalized Regression Neural Network
MAPE	Mean Absolute Percentage Error
DNNs	Deep Neural Networks

ReLU	Rectified Linear Units
MLP	Multi-layer Perceptron
RNN	Recurrent Neural Network
SVR	Support Vector Regression
ML	Machine Learning
WLS	Weight Least Squares
API	Application Programming Interface
DAQ	Data Acquisition
ADC	Analog to Digital Converter
AI	Analog Input

CHAPTER 1

Introduction

Oceans play a pivotal role in the life of our planet. Covering more than 70% of the surface area of the earth, oceans influence weather and climate, represent rich sources of food, renewable energy, oil, and minerals [1]. Ocean engineering is a multidisciplinary field aimed at exploring and discovering oceans to preserve underwater ecosystem, harness oceans green energy resources, and take advantage of their food supply.

Underwater vehicles are being used to better understand the ocean. Spectrum of technologies such as Remotely Operated Vehicle (ROV) and Autonomous Underwater Vehicle (AUV) are available for object identification and vessel hull inspections [1]. ROV is an unoccupied underwater robot, as shown in Figure 1, that is remotely navigated by a human operator through connected cables. These are often used as an alternative for hydrographic diver investigations when diving by humans is either impractical or dangerous. Unlike ROV, an AUV conducts its mission without an operator intervention (Figure 2).

Figure. 1.1: Remotely Operated Vehicle [1]

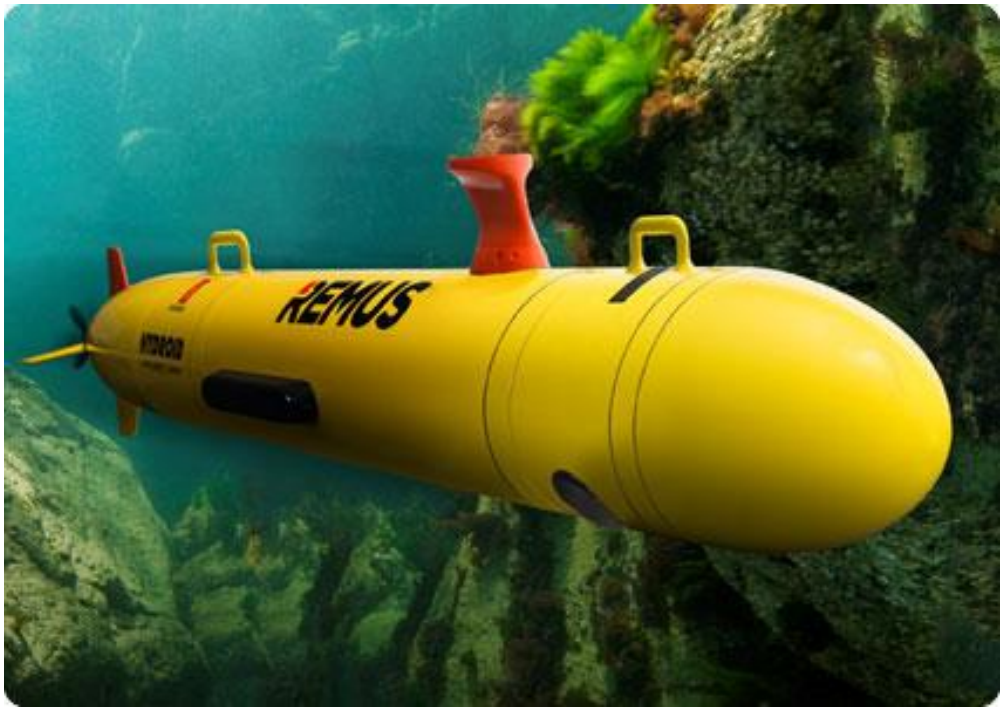


Figure. 1.2: Autonomous Underwater Vehicle [1]

1.1 Underwater Vehicle Navigation

Navigation is an important task for any type of mobile robot to avoid collisions and reach a goal location. Accuracy of navigation is highly related to the quality of the gathered data, which is critical in some application such as mine countermeasures or navigation in confined spaces.

A robot should be able to perform the following major steps to navigate in its surrounding environment [1].

- **Localization** refers to the robot's ability to determine its own location according to fix reference points. It is the cornerstone of navigation process. If robot fails to localize itself precisely within the frame of reference, it will not be able to reach the destination.
- **Path planning** can be regarded as a computational problem to find a sequence of movements to proceed from source to destination. An autonomous robot is expected to either construct or use a map and localize itself in the map for further movement.
- **Map building** can be performed in metric framework or topological framework. In metric framework, which is more perceivable by humans, objects are considered in two-dimensional space. Being sensitive to noise, this representation is difficult to be used for precise distance measurement. The topological framework represents the environment as a graph with several nodes and arcs corresponding to places and paths between them, respectively.
- **Map interpretation** is the process of using a map to figure out the best path from the current location to the destination.

While many mobile robots rely on Global Positioning System (GPS) to find their position, underwater vehicles cannot make use of GPS signals which have high attenuation in underwater environment. In contrast to GPS signal, sound propagates efficiently in the water due to low

acoustic absorption in this medium. So, vast majority of underwater vehicles (UVs) are based on Sonar (sound navigation ranging) systems to navigate or communicate in underwater. Acoustic navigation systems can be classified into passive and active methods. While passive sonar is based on listening for the sound made by other objects, active sonar is based on emitting pulses of sounds and listening for echoes.

1.2 Underwater Vehicles Applications

Underwater vehicles (UV) have several applications ranging from hobby to research [2]. Hobby applications of UVs have gained a lot of public interest recently. People enjoy participating in competitions to accomplishing objectives using their homemade UVs. The oil and gas industries often utilize UVs to make seabed maps before installing pipelines, cutting their infrastructure costs, and minimizing their disruption to the environment. UVs can be used by researchers to study lakes and the ocean floor to investigate marine-life extinction, fish stocks population decline, etc. In addition, it can be useful to explore the natural disaster alerts such as tsunami and tropical cyclones.

1.3 Objectives and Motivations

The main objective of this research is to find accurate position of an underwater vehicle (UV) inside of a confined environment (such as a tank), using active sonar for acoustic positioning. The proposed method relies on sound propagation and distances measurement between a transmitter (beacon) and a set hydrophone. Four hydrophones were installed alongside the tank to estimate 3D position of an UV accurately using Time of Arrival (TOA) of an acoustic signal transmitted from the beacon.

Since accuracy is a critical step in navigation systems, we were motivated to develop a novel acoustic positioning based on machine learning techniques to pinpoint the sound source location. In addition, UV localization in small tanks can be regarded as a starting point for inspection and navigation in more realistic confined spaces like wells, caverns, or sunken ships.

1.4 Challenges

Sound penetration depth and propagation pattern in water depends on several factors such as temperature, salinity, and depth. These parameters cause refraction, bending of acoustic rays, which in turn brings several challenges [3]. Due to refraction, the transmitted sound may propagate from beacon to hydrophone through several paths. Moreover, hydrophone may receive reflected sound signals from boundaries and objects. This phenomenon called multipath propagation, is aggravated in a tank in which water is confined with several walls, ceiling, and floor. Reverberation (echo) is another concern in UV navigation in confined spaces, which occurs when hydrophone receives several sound signals reflected from tank boundaries [4]. Since in this project the transmitter is mounted on the mobile UV, noise and doppler effect should also be taken into consideration. Doppler effect that causes frequency shifts, stems from relative movement between the transmitter and the hydrophone.

To address aforementioned issues, machine learning methods are leveraged in this thesis to obviate the need for prior knowledge of underwater parameters and characteristics. Analytical approaches for underwater localization, on the other hand, take underwater acoustic channel into consideration, which needs accurate estimation of channel parameters. The characteristics of the channel such as sound propagation speed, attenuation and noise are mainly influenced by internal

factors including water salinity, depth and temperature as well as external factors like seabed profile, and reflective objects within the sea [5, 6].

1.5 Contributions

The contributions of this thesis are summarized as follows:

(1) The generalization power of machine learning techniques is used to improve localization accuracy by having limited number of collected data. This obviates the need for model-generated fields and exact characteristics of underwater environment beforehand [7,8].

(2) Most of previous methods attempt to solve source localization problem by regression approaches [9]. In this study, we consider source localization from classification point of view. For this purpose, the whole space of the confined tank is regarded as a set of finite positions. To estimate the location of the sound source, the output of the machine learning methods is matched to the nearest predefined position.

(3) Apart from well-developed machine learning techniques [10], such as RF, SVM, and FNN, emerging CNN [11] with promising results was adopted and modified to be applicable of one-dimensional data.

1.6 Outline of Thesis

The structure of this thesis is as follows:

- **Chapter 2** explains in detail underwater localization and its background. It defines the underwater channel and time delay estimation, required for UV acoustic localization. This chapter also reviews the state-of-the-art techniques relevance to underwater localization.

- **Chapter 3**, the proposed acoustic localization method, which is based on machine learning techniques is presented.
- **Chapter 4** explains the hardware setup including transmitter and receiver.
- **Chapter 5** presents experimental results and discusses the effect of different data acquisition parameters and classifiers hyperparameters on the localization accuracy.
- **Chapter 6** concludes the thesis by summarizing our contributions and making suggestions for future research

CHAPTER 2

State of the Art

This chapter first presents various underwater navigation systems. Then, acoustic localization method is presented, and different topologies currently being employed in this field will be compared. Acoustic channel and phenomena affecting acoustic signals propagation in underwater environments will also be studied in this chapter. Finally, State-of-the-art techniques for acoustic localization will be reviewed at the end of this chapter.

2.1 Underwater Navigation Systems

All underwater vehicles (UVs), including Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs), rely on navigation systems to perform their missions. Navigation systems should have high accuracy and fast convergence to find the shortest path and avoid collision.

Accuracy is the chief criterion in navigation systems, and positioning error is usually defined according to the size of UV and the underwater environment. As UVs and their reference points are floating in the water, their positions are constantly changing. To cope with this situation, navigation systems should be fast enough to converge rapidly before determined position being outdated. These systems should also be energy efficient to rely on battery powered devices and for long missions. Accuracy of estimated position in underwater navigation systems is highly dependent on the number of reference points. This is due to the fact that the high propagation delay and high-power attenuation of sound signal in underwater environment make localization systems error prone. In such systems, positions of hydrophones, which are relatively expensive, are usually considered as reference points. Consequently, to reach a good compromise between accuracy and cost, underwater localization systems should depend on as few reference points as possible.

Because of the absence of global positioning system (GPS), underwater navigation is a challenging task. These systems should employ other sensors to navigate UVs in the water

environment. The most common sensors are compass, gyroscopes, camera, transceivers, beacon, hydrophone, and depth sensor [12].

Being inexpensive and low-power, compass is a basic navigation sensor used in the most underwater vehicles. It provides three-dimensional position of the local magnetic field. However, compass based navigation systems should be carefully calibrated when vehicle's area of operation changes. Gyroscope measures changes in vehicle orientation according to physical laws. Accurate navigation systems, however, require high performance gyroscope sensors which are costly. An *Inertial* Navigation System (INS) utilizes three gyroscopes, mounted perpendicularly to each other.

Unlike Inertial navigation systems, *geographical* positioning methods need sensors to capture and compare acquired information with previously generated datasets, characterizing geographical locations. Optical systems use camera to obtain underwater images. Underwater environments suffer from inadequacy of lighting, resulting in low quality images, especially in high depth areas. Reduced range of the camera limits the application of optical systems to shallow water navigation. To tackle this problem, some geographical positioning systems make use of acoustic signals instead of optical images. These sonar systems rely on acoustic equipment to emit and receive sound signals. While beacon (Pinger) emits a predefined sound signal periodically, hydrophone (underwater microphone) receives emitted signal. Some acoustic equipment such as transponder can both receive and emit signals. When transponder receives a signal with specific frequency, it sends a signal in response. Transducer is another emitter and receiver which transforms electric signal into acoustic signal and vice versa. Figure 2.1 shows different technologies for UV localization and navigation [13].

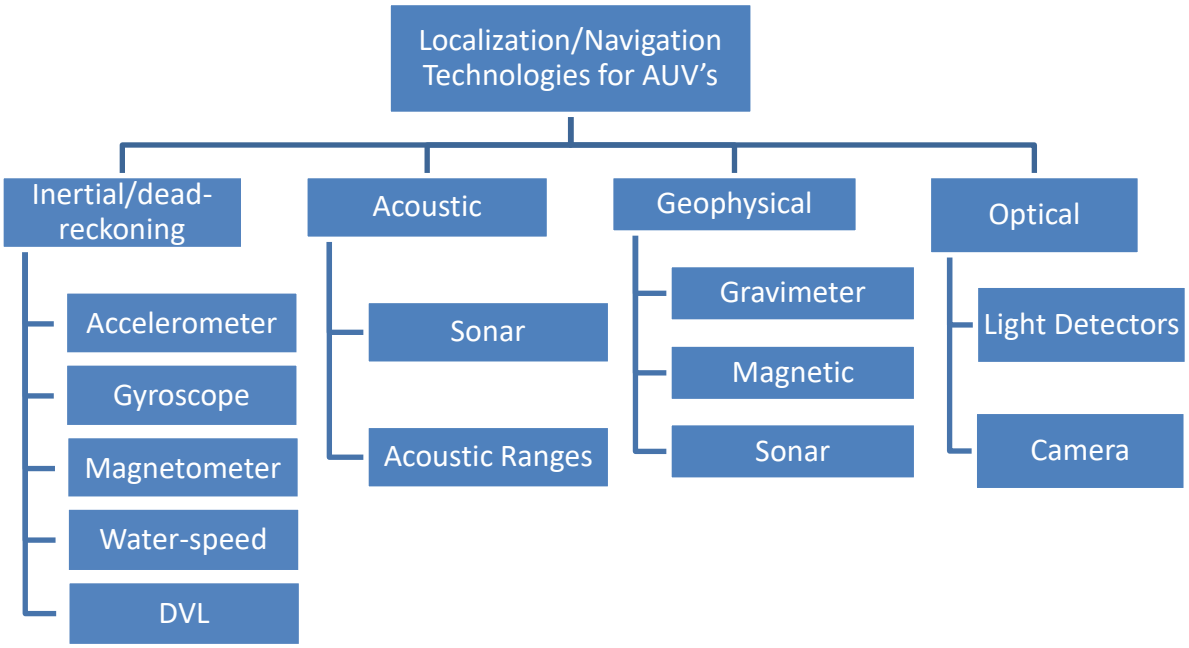


Figure. 2.1: Overview of underwater localization and navigation techniques [13]

2.2 Acoustic Localization

As electromagnetic energy cannot propagate effectively in water, acoustic signal is used frequently for underwater localization and communication. Acoustic localization aims to use sound signal to determine the distance and direction of its source. Location of UVs can be estimated either actively or passively. In *active* sonar, the sound source generates acoustic signal to produce an echo from the UV, which is then analyzed to determine the location of the object of interest. This method has been used by scientists to localize marine animals such as whales and dolphins. In the presence of several moving objects, active sonar receives a number of echoes, making localization process more challenging. A confined space with several reflections from walls, bottom and top also makes active sonar localization less effective.

To address this issue, *passive* acoustic method has been proposed which relies on a sound generator (beacon) and several sound receivers (hydrophone). The beacon is mounted on the UV and hydrophone arrays are either towed or independently deployed. Assume towed array configuration (Figure 2.2) including some hydrophones. By measuring the Time Difference of Arrival (TDOA) between the direct arrivals at the hydrophones, it is possible to estimate the location of beacon [15].

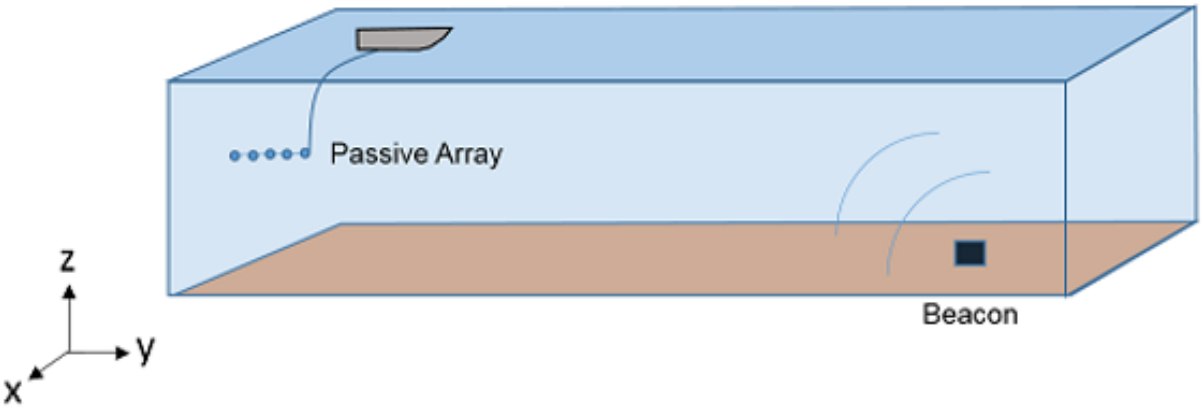


Figure. 2.2: Active sound source localization using hydrophones array.

2.3 Positioning Systems

Underwater acoustic positioning systems have been used for oil and gas exploration, marine science and archaeology as well as security operations [14]. In these systems, location of an UV is estimated by a set of transponders, called baseline, which are installed on a mothership. Based on the distance between transponders, these systems can be categorized into: Long Baseline (LBL), Short Baseline (SBL) and Ultra-Short Baseline (USBL) [15].

2.3.1 Long Baseline (LBL)

This method is based on some transponder beacons and one transducer. Beacons are installed on the seabed with a typical distance of 50 to 2000 meters, and the transducer is fixed to the UV [15]. Figure 2.3. shows LBL configuration.

First, the transducer on the UV sends an acoustic signal. After receiving this signal, the transponder beacons transmit another signal in response. Then, the UV receives the reply pings which have different frequencies to prevent any interference. The distance between the UV and each transponder can be calculated, based on the time difference between the transmitted signal from transducer and the reception of its response from each beacon. To find the position of the UV based on triangulation of acoustic signals, at least three transponders are required. In order to provide redundancy and quality check, fourth transponder might also be deployed. After finding the position of UV relative to the ship, its exact position can be obtained by having a priori knowledge of the ship position.

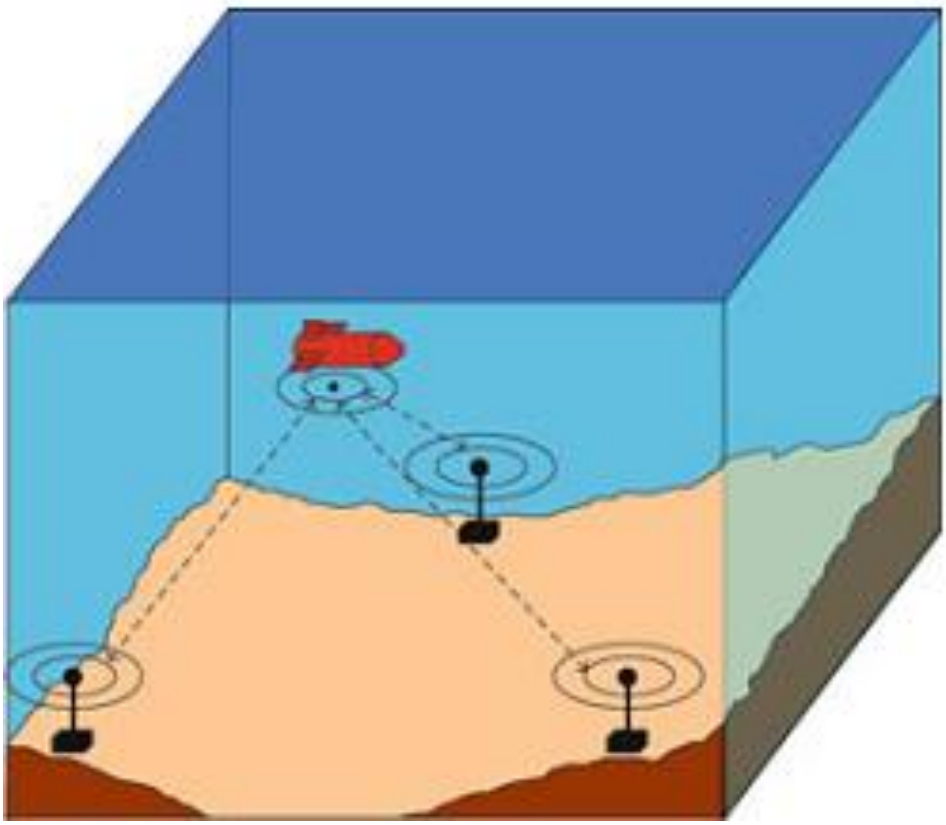


Figure. 2.3: Long baseline positioning system [15]

2.3.2 Short Baseline (SBL)

Unlike LBL, transponders are mounted on floating platforms like a ship in SBL configuration (Figure 2.4), and the distance between them ranges from 20 to 50 meters [15]. There is a positive

correlation between the transponders distance and localization accuracy; the longer the distance, the higher the accuracy.

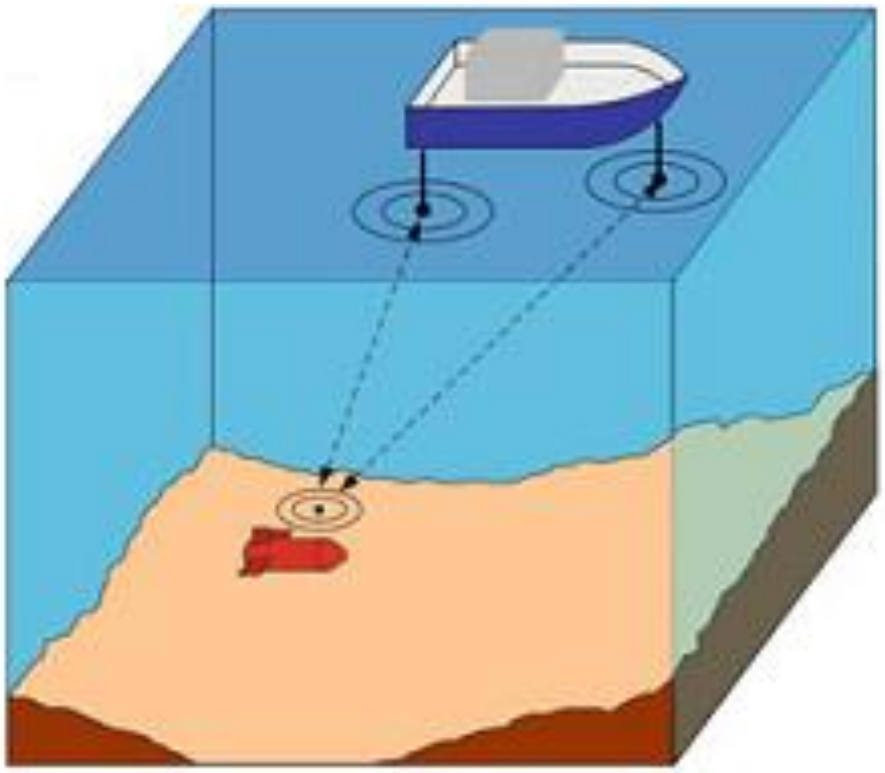


Figure. 2.4: Short baseline positioning system [15]

2.3.3 Ultra Short Baseline (USBL)

This method is composed of a single transponder, mounted to the object to be tracked, and an array of transducers, closely spaced with the approximated distance on the order of 10 cm (Figure 2.5). Same as the other two previous methods, USBL is also based on the Time of Flight Estimation of an acoustic pulse and its response to measure the distances. Due to proximity of transducers, however, phase-differencing method should be employed in conjunction with trilateration approach in order to calculate the angle to the transponder and increase localization accuracy [15].

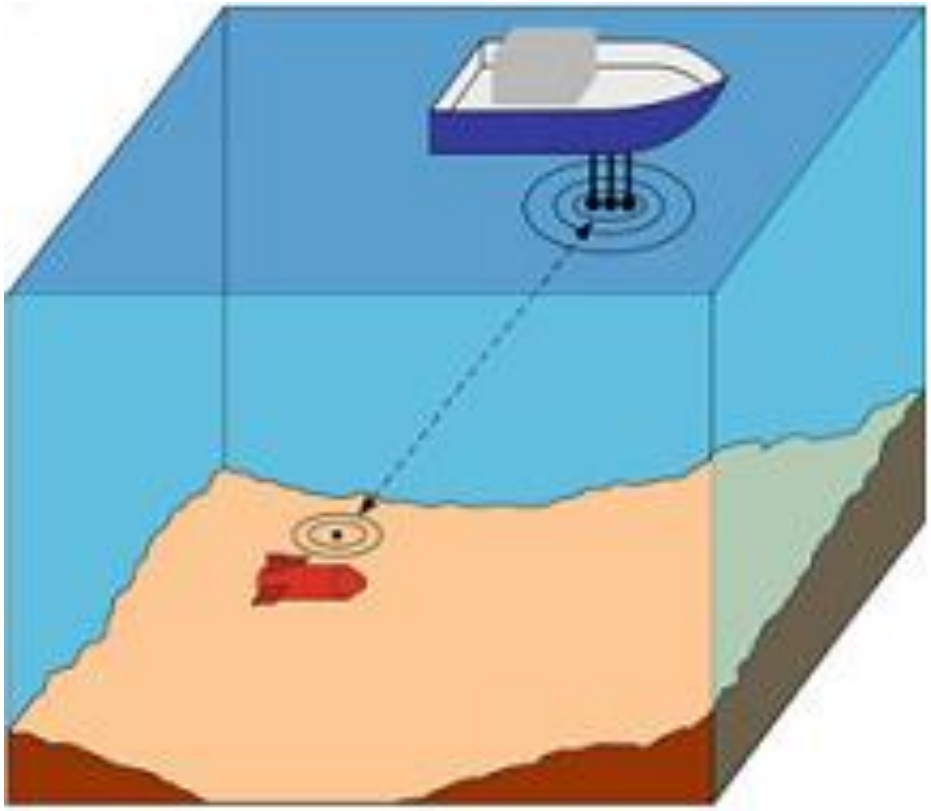


Figure. 2.5: Ultra short baseline positioning system [15]

2.3.4 Comparison between Positioning Systems

Each of the positioning methods has its own merits and disadvantages [15]. The main advantage of LBL is its high localization accuracy which is in the range few centimeters over a wide area. However, since transponder beacons should be installed on the seabed, this method is costly and time-consuming to set up.

Both SBL and USBL methods rely on a floating platform to install transponders which is not practical in confined spaces. Localization accuracy of SBL depends on the dimensions of the floating platform; the accuracy seemingly declines when the size of floating platform is short.

Because of easy installation and usage, USBL is the most popular positioning system, but in comparison with LBL and SBL methods, USBL has higher localization error.

2.4 Acoustic Channel Modeling

Since focus of this work is on acoustic underwater localization systems, it is beneficial to study acoustic channel model and the way sound signal travels in underwater environment. The characteristics of the channel are influenced by various factors which make underwater localization a challenging task.

2.4.1 Sound Speed

The speed of sound in water usually varies between 1440 m/s to 1550 m/s, depending on the pressure, salinity, and temperature of water. The increase of local pressure and density reduce propagation delay. In the ocean, water temperature first decreases with the increase of depth, and then it becomes stable. Considering these parameters, the sound speed can be estimated using the Equation (2-1) [16]

$$C = 1449.2 + 4.6T + (1.34 - 0.01T)(S - 35) + 0.016z \quad (2\text{-}1)$$

Where C is the sound speed in seawater, T is the temperature in Celsius, S is the salinity in parts per thousands and z is the depth of the water.

2.4.2 Attenuation and Noise

An acoustic signal traveling in underwater environment is attenuated, resulting in path loss due to the degradation of the amplitude of the signal. Ambient noise also affects the strength of the acoustic signal passing through channel. Acoustic noise is usually considered as a Gaussian noise with power spectral density decaying at around 18 dB per decade [16].

2.4.3 Doppler Effect

Frequency of transmitted sound is altered due to motion of either beacon or hydrophone, according to Doppler effect. This frequency shifts aggravated by the low speed of the sound. If beacon and hydrophone move with the u and v velocities, respectively, Doppler effect can be expressed as [17]:

$$f' = f \frac{c \pm u}{c \mp v'} \quad (2\text{-}2)$$

Where f and f' are the frequencies of transmitted sound by the beacon, and the received one by the hydrophone, respectively. C is the propagation velocity of the wave in a stationary medium.

2.5 Propagation Models

To measure the distance between beacon and hydrophone for localization, propagation speed of the sound signal should be multiplied by the delay time. In order to estimate the delay time of the sound, signal model should be determined to describe the propagation mechanism of the sound signal in acoustic environment. In the following, three main propagation models will be described [18].

2.5.1 Single-path Mode

In its simplest form, it can be assumed that there is only one direct path between the beacon and hydrophone in which the sound signal travels. Assume we have N hydrophones. The received signal by the nth hydrophone ($x_n[k]$) is the delayed and attenuated ($\propto n$) versions of the source signal (s[k]), affected by additive noise ($w_n[k]$).

$$x_n[k] = \alpha_n s[k - t - f_n(\tau)] + w_n[k], \;\; n = 0,1, \dots, N-1 \qquad (2\text{-}3)$$

Where $f_n(\tau)$ is the relative delay between sensors 0 and n. This model can be used in deep ocean environment when the beacon and hydrophone are far from the sea surface and sea bottom.

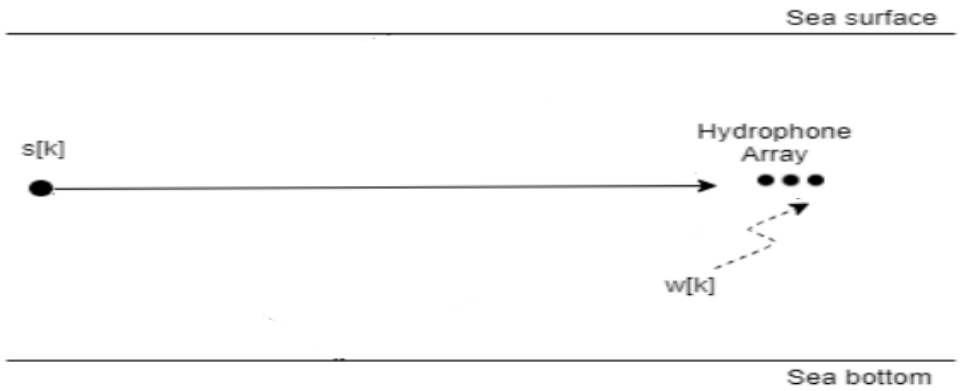


Figure 2.6: Single-path propagation model [18]

2.5.2 Multi-path Mode

In a more realistic situation, the hydrophone may receive sound signals from multiple paths, caused by reflection from the seabed, sea surface as well as direct path. This model is often utilized in shallow environments.

Assume there are M paths between the beacon and the hydrophones. If attenuation coefficient and delay time of the path between nth hydrophone and beacon through mth path are denoted by $\propto nm$ and τ_{nm}, then hydrophone n receives a signal ($x_n[k]$) which is the sum of m signals, contaminated by an additive noise ($w_n[k]$) [18]:

$$x_n[k] = \sum_{m=1}^{M} \alpha_{nm}\, s[k - t - \tau_{nm}] + w_n[k], \;\; n = 0,1, \dots, N-1 \qquad (2\text{-}4)$$

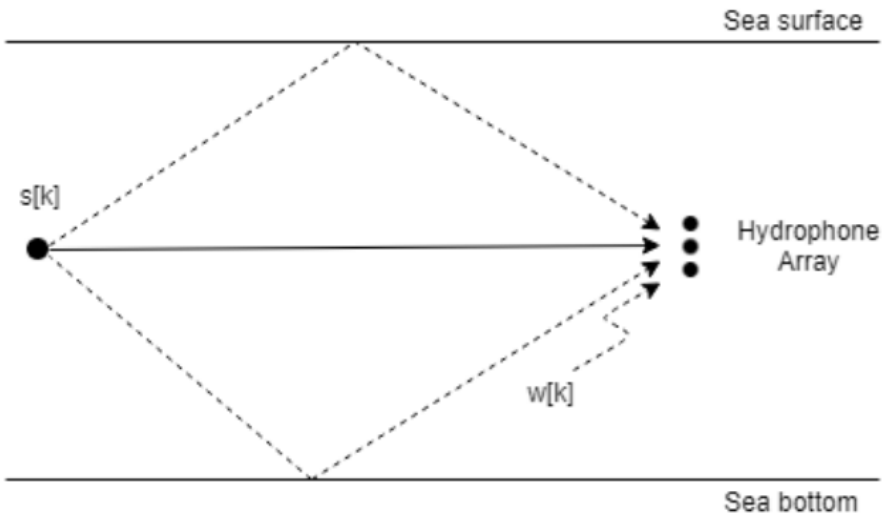


Figure 2.7: Multi-path propagation model [18]

2.5.3 Reverberation Mode

This model can be regarded as the extended version of multi-path mode. In a confined space, the hydrophone receives signals from numerous paths, resulting from reflections from the room boundaries, such as walls, ceiling, and floor. Since parameters which affect the sound speed such as temperature and pressure are constant in a confined space, this environment can be described by a linear time invariant model (LTI) [18]. So, in the reverberation mode with an impulse response of h_n for the nth hydrophone, the received signal for hydrophone n is [18]:

$$x_n[k] = h_n * s[k] + w_n[k] \quad (2\text{-}5)$$

Where * denotes convolution operation and w represents the additive noise. However, estimating all N channel impulse responses would be a challenging task for utilizing this model.

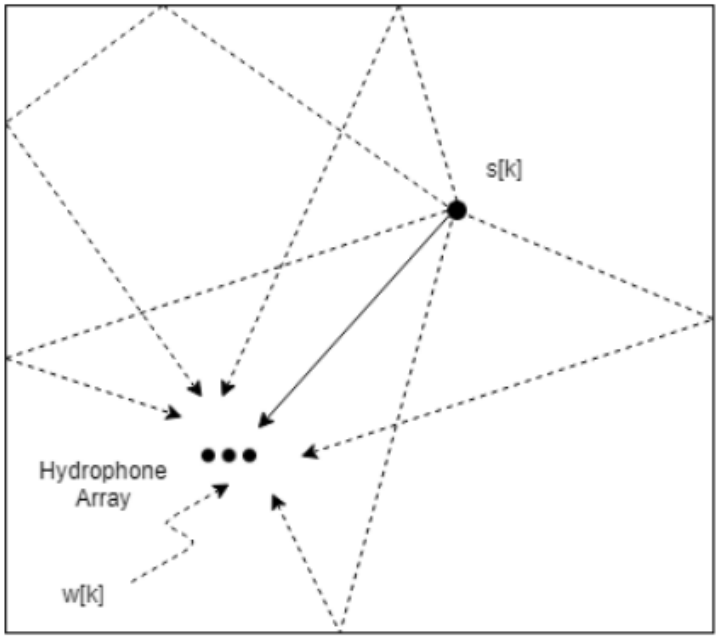


Figure 2.8: Reverberation propagation model [18]

2.6 Time Delay Estimation

After estimating the speed of sound (section 2.4.1) and determining its propagation mechanism (section 2.5), distance between sound receiver and the signal source can be calculated through following methods:

2.6.1 Time of Arrival (TOA)

TOA is the delay between the transmission of a sound signal at the beacon and its reception at the hydrophone. To measure TOA, sound receiver should be synchronized with the sound source to know the exact starting time of the sound propagation [19]. Upon receiving the acoustic signal, the receiver calculates the delay. Then, the distance to the sound source can be measured knowing the estimated underwater sound speed and Time of Arrival. As mentioned in section

2.4.1, several factors may affect the sound speed in water, causing this method error prone. Synchronization between sound source and receiver can also lead to accuracy problem.

2.6.2 Time of Flight (TOF)

To obviate the need of synchronization between the sound transmitter and the receiver, Time of Flight method has been proposed. TOF method requires that both nodes are capable of both sending and receiving acoustic signals. First, transmitter emits a sound signal. Upon receiving this signal, receiver sends another sound signal in response. Finally, the transmitter calculates the distance according to the round-trip-time [19].

2.6.3 Time Difference of Arrival (TDOA)

Another method to eliminate the synchronization requirement is Time Difference of Arrival. TDOA is based on pairs of spatially separated hydrophones in which one of the hydrophones is considered as the master (reference), while the remaining are the slaves (auxiliary). cross-correlation is one of the most common methods to measure TDOA [20].

When the hydrophones are in close vicinity to each other, it can be assumed that the contaminating noise is identical for the received signal by these sensors. As the signals received by the hydrophones are very similar, they have a high cross-correlation. So, cross-correlation can be utilized to estimate TDOA.

2.7 Localization Estimation

To localize the exact position of an underwater sound source, not only the distance but also the direction of the source is required. The most common location estimation technique is trilateration, which is based on determining TDOA [21].

For the sake of simplicity, assume three hydrophones are available on a 2D plane. After finding TDOA, each hydrophone can measure the distance of the sound source. The potential position of the sound source can be on the circumference of a circle with radius equals to the measured distance. According to Figure 2.9, the intersection of three circles corresponding to each hydrophone, determines the correct position of the sound source. After estimating the location of

the source of interest across 2D plane, pressure sensors can be utilized to measure the third dimension, which is depth.

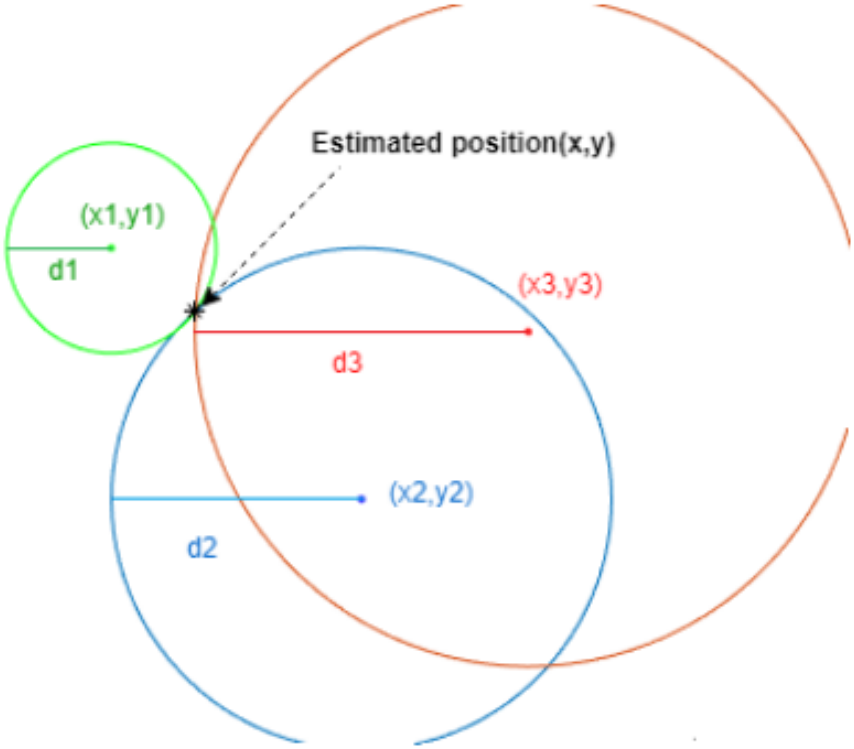


Figure 2.9: Triangulation method [20]

2.8 Literature Review

This section surveys the state-of-the-art techniques pertaining to underwater acoustic localization. We review the major acoustic channel models, including sound transmission and propagation mechanisms. Subsequently, conventional underwater localization papers based on Long Baseline (LBL), Short Baseline (SBL), and Ultra Short Baseline (USBL) methods will be reviewed. Finally, novel approaches that leveraged machine learning methods for underwater localization will be presented.

2.8.1 Acoustic Channel Model

Accurate description of the underwater channel plays a pivotal role in underwater communication and sound source localization. Doppler shift effect is important to consider because of the relative low propagation speed of acoustic waves in underwater medium and motion of underwater Vehicle.

In [22], the impact of the use of speed measurement in a localization algorithm was explored. The Doppler effect in communication from the anchor to the UV, induced by the movement of the UV, was utilized to measure the speed of the UV while communicating and ranging.

2.8.2 Conventional Underwater Localization

Underwater localization algorithms, relying on Time of Arrival (ToA) and triangulation method, assume that the acoustic signal travel path is a straight line. However, underwater acoustic speed diversities can transform the straight lines into refracted trajectories. The refraction phenomenon was considered in [3] to determine the positions of UV.

Uncertainties in sound propagation speed and time synchronization in measuring Time of Arrival (ToA) were taken into consideration in [23]. The anchors measure the ToAs of the signals from the other anchors to estimate the sound propagation speed. The agents measure the ToAs of the signals broadcast by the anchors and combine the ToAs measured in two consecutive intervals to estimate the clock skews. Then, the weighted Least Squares (WLS) algorithm was used to calculate the agents positions and clock offsets.

An acoustic positioning system, capable of estimating, in real time, the position of an object inside a confined test tank was proposed in [18]. It was based on a periodic transmission and reception of modulated sequences. Time of Arrival was estimated through the cross correlation between the received signal and a reference signal. An array of 4 hydrophones was used, and the ToA estimated through each one is conjugated to obtain the position of the acoustic source.

To improve the ToA estimation of an acoustic signal, the use of pseudo-random binary sequences modulated in Binary Phase Shift Keying (BPSK) was proposed [19]. The proposed

algorithm for improving the correlation peak, considerably increased the precision of the localization system.

2.8.3 Machine Leaning based Underwater Localization

Maneuvering vehicles can cause error in Long Baseline (LBL) acoustic localization systems. This is due to vehicle motion between the time the vehicle sends signal and the times of reception for the acoustic replies from the various transponders. A motion-compensated model for vehicle localization was developed based on Bayesian inference algorithm which includes travel-time corrections for all receptions as unknown parameters [24]. In a similar work [25], the Bayesian localization method introduced in [24] was tested in both shallow and deep water. This method used the time difference of direct and surface-reflected arrivals of pulsed signals at two hydrophones of known depths. Uncertainties in measured quantities such as TDOAs, hydrophone depths, and sound speed profile were considered by the Bayesian approach to localize the sound source more precisely.

The potential of machine learning for underwater source localization through a fluctuating ocean was studied in [7]. Conventional methods rely on knowledge of the environmental parameters, which is not available in a random and fluctuating underwater channel. Requiring only training data without the need for environmental characteristics makes machine learning techniques a suitable candidate for localization task in fluctuating channels. Performances of Kernel regression as well as the local linear regression were compared for sound localization in fluctuating environment.

In [9], source localization is regarded as a supervised learning regression problem and is solved by generalized regression neural network (GRNN). Machine learning framework and the acoustic propagation model were combined to obtain the training data of GRNN from the acoustic propagation model.

More details of machine learning methods for acoustic signal processing and application can be found in [10].

2.9 Summary

This chapter reviewed different underwater localization and navigation techniques. Then, underwater channel and various parameters affecting sound propagation were studied. Finally, state-of-the-art underwater source localization methods, especially machine learning based approaches, were reviewed.

CHAPTER 3

Machine Learning based Underwater Localization

This chapter presents the proposed method for underwater sound source localization which is based on Machine Learning (ML) technique. First, we briefly review machine learning techniques which will be used in this project. Then, supervised machine learning algorithms utilized for acoustic localization will be explained. Unlike analytical methods, ML approaches learn directly from data, obviating the need for underwater channel parameters estimation [26]. In other words, ML methods only rely on acoustic data and do not need any propagation models of the known environment to predict source location. Figure 3.1 depicts the main methods for acoustic source localization [27].

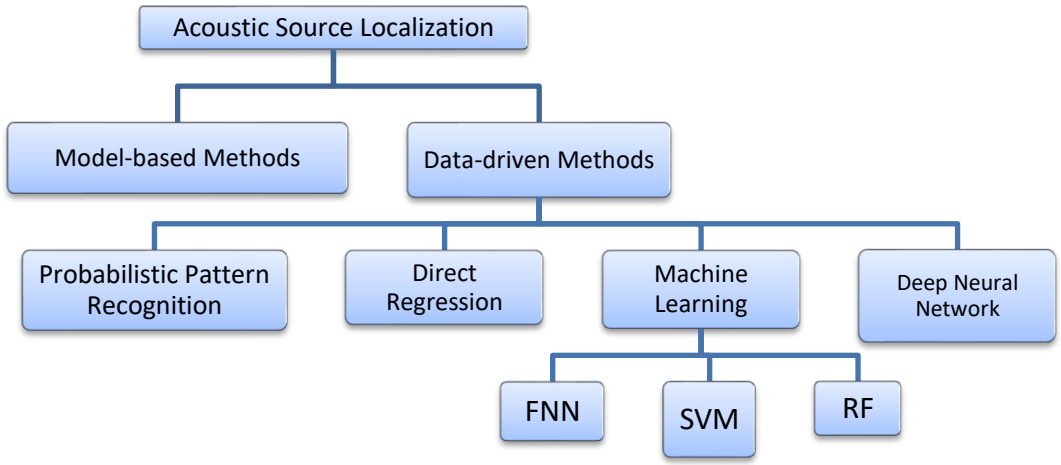


Figure 3.1: Overview of acoustic source localization methods [27]

3.1 Machine Learning Principals

Acoustic data have been frequently used in various underwater engineering projects including mammal vocalizations, source localization, as well as seabed imaging and map creation. In these applications, data analysis is challenging due to data corruption, reverberation, and large data volumes. To address these acoustics challenges, machine learning techniques have been proposed.

ML is a family of techniques for automatically detecting and utilizing patterns in data. The extracted patterns can be used later to estimate data labels based on measured attributes. Take

source localization as an example; based on recorded acoustic data, a label can be assigned to each data which indicates the position of the sound source. For this purpose, ML methods are expected to gain implicit knowledge from the data and learn how to assign labels. Learning algorithms are based on a set of data, called training set, with predefined labels, and statistical methods to cope with uncertainty in data collection and measurements. ML methods are often divided into two major categories [10]: supervised and unsupervised learning. In *supervised* learning, the goal is to learn a predictive mapping from inputs to outputs using training set with known labels. In *unsupervised* learning, however, no labels are available, and the task is to extract complex and subtle patterns within the data.

Although ML methods have provided compelling solutions for practical engineering applications, they have their own drawbacks [10]. Since ML is data-driven, quality of the predicted labels of extracted patterns highly correlated to the quality of collected data for ML training. That is, ML models require significant amounts of training data to perform accurately and reliably. Training data collection is a tedious and time-consuming task, especially in large and dynamic environments such as oceans. Further, if the working environment changes, the data collection process and training step should be repeated again. A good illustration for this is utilizing sound localization in a confined task which was trained in an opens underwater space. Another major shortcoming of an ML model is their complexity which hinders model interpretation. In other words, ML models are involved black-box models, and hence, no physical insight into them is possible.

3.2 Supervised Learning

The aim of supervised learning is to learn a predictive mapping from inputs to outputs given that the training data set consists of labeled input and output pairs. In the case of sound localization, for instance, the input is the received sound signals to the hydrophones while 3D position of the sound source regarded as the output. Supervised learning is the most widely used ML category, ranging from simple methods such as nearest-neighbor classifiers to more sophisticated algorithms like Support Vector Machine (SVM) and Neural Network (NN) [10]. In the following, four supervised methods used in this thesis will be briefly presented.

3.2.1 Support Vector Machines

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification and regression tasks. Assume classification challenge with two classes in which the aim is to find the hyper-plane that differentiates the two classes with minimum error rate. Instead of taking all training samples into account, SVM only considers closest samples to the desire hyperplane, which are called support vectors. SVM is a maximum-margin classifier. In order to cope with noisy samples which may cause error, SVM endeavors to maximize the distances between support vectors and hyperplane, called margin. Consider Figure 3.2 with three potential hyperplanes (A,B, and C). Among them hyperplane C has the maximum margin.

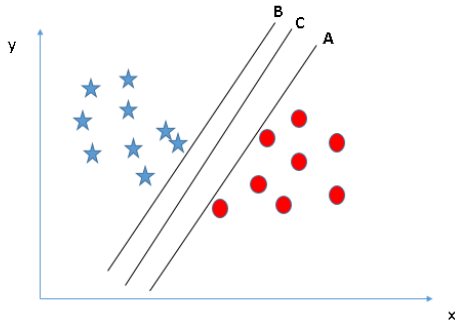


Figure 3.2: Two class sample classification with three different hyperplanes [28]

In the case of linear separable classes, SVMs can be easily formulated as follow [10]:

$$y = Xw + w_0 \tag{3-1}$$

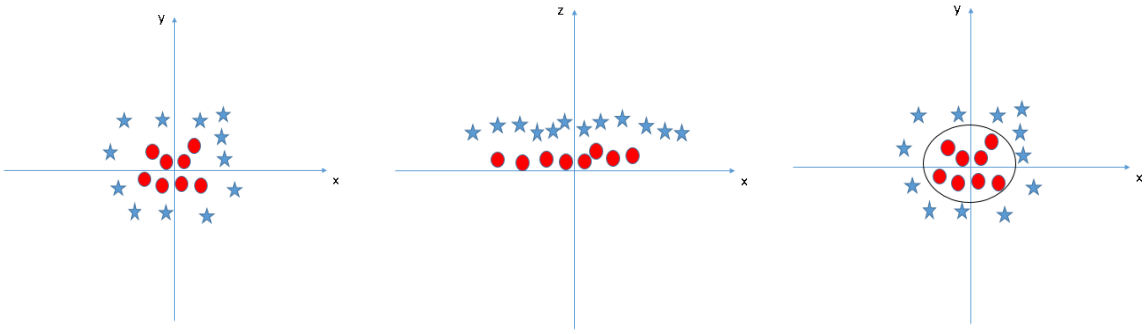
where w and w_0 the weights and biases. A decision hyperplane satisfying $Xw + w_0 = 0$ is used to separate the classes. If the calculated output y_m is above the hyperplane ($y_m > 0$), the estimated class label is $s_m = +1$, otherwise $s_m = -1$. Since the margin between two classes is $\frac{2}{\|w\|}$ we can minimize $\frac{1}{2}\|w\|$ to find the weights. So, the weights w and w_0 are estimated by the following quadratic program [10]:

$$\min_{\boldsymbol{w},w_0} \frac{1}{2} \|\mathbf{w}\|_2^2$$

$$\text{Subject to } s_m(\mathbf{w}^T\mathbf{x}_m + w_0) \geq 1 \forall m \tag{3-2}$$

The first formula in Eq. 3-2 is called objective function, while the second formula is the constraint that implies all training samples should be correctly classified.

When classes are linearly non-separable, as in Figure 3.3 (a), the SVM algorithm should map low dimensional input space into a higher dimensional space. This technique is called the *kernel trick* and it is used to convert not separable problem to separable problem. In this case, if we use a kernel $z = x^2 + y^2$, data samples will be differentiated by a linear hyperplane in z space (Figure 3.3 (b)). After converting this linear hyperplane to lower dimensional input space, x and y, a non-linear hyperplane can be obtained (Figure 3.3 (c)).



(a) (b) (c)

Figure 3.3: kernel trick. (a): Non-separable classes. (b): high order data transformation. (c): non-linear hyperplane

For non-linear classification problems, Equation 3-2 can be kernelized (k is the kernel function) to make the data linearly separable in a non-linear space. The most common kernels include polynomial, Gaussian radial basis function, and Hyperbolic tangent. Equation 3-3 can be solved using the Lagrangian dual method [28].

$$L(\mathbf{a}) = \sum_{i=1}^{M} a_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} a_i a_j s_i s_j k(\boldsymbol{X}_i, \boldsymbol{X}_j) \tag{3-3}$$

$$\text{Subject to} \qquad 0 \leq a_i \leq C$$

$$\sum_{i=1}^{M} a_i s_i = 0$$

As mentioned before, SVM can also be used for regression. The first version of SVM for regression was proposed in 1996 and is called support vector regression (SVR) [33].

$$\text{minimize } \frac{1}{2}\|\mathcal{W}\|^2$$

$$\text{Subject to } |y_i - \langle \mathcal{W}, \mathcal{X}_i \rangle - b| \leq \varepsilon \tag{3-4}$$

Where x_i is the i[th] training sample and y_i is its corresponding target value. Weights and bias are presented by w, b. Parameter ε is a free parameter that serves as a threshold to determine acceptable error value. According to the constraint of Equation 3-4, the difference between target value and predicted value that is inner product of input and weight plus bias should be less than the predetermined error value. Figure 3.4 shows predictions of SVR with different thresholds ε. As it can be seen, the prediction becomes less sensitive to errors when ε increases.

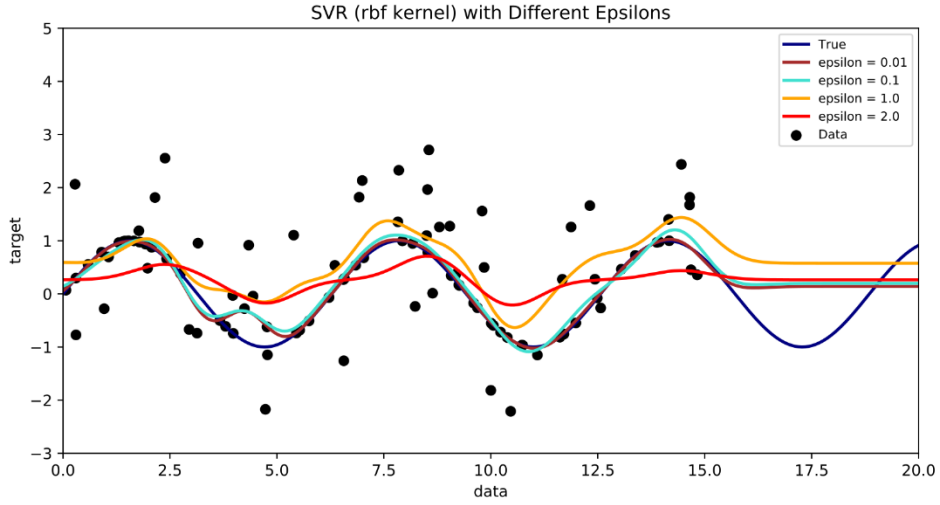


Figure 3.4: The effect of threshold in the constraint in regression SVM [28]

3.2.2 Neural Networks

In this section, we review two different types of Neural Networks (NNs): Multi-Layer Perceptron (MLP), and Convolutional Neural Network (CNN).

3.2.2.1 Multi-layer Perceptron

Although linear models, like linear SVM, can efficiently be fit for linear separable problems, they are unable to model non-linear functions. To tackle this problem, as stated in section 3.2.1, SVM utilizes kernels to transfer the features into a more useful non-linear space. However, the number of these kernels are limited, and they are general functions with moderate performance for many tasks.

To overcome the limitation of SVM, Neural Networks (NNs) have been proposed. NNs provide the algorithmic way to learn the non-linear mapping of the inputs directly from specific training data. In this thesis, we will use feed forward NNs (FNN), also called multi-layer perceptron (MLPs), to approximate mapping functions. In FNNs, unlike recurrent NN (RNN), information flows only from the features to the labels, without any feedback loop. NNs are composed of a series of layers: input layer, hidden layer, and output layer. Figure 3.5 shows a fully connected FNN with one input layer (x), two hidden layers $(z^{(1)}, z^{(2)})$, and one output layer (y). The number of hidden layers plus the output layer is called the NN depth that affects the capacity and performance of NNs.

The hidden and output layers utilize a non-linear function, called activation function, to transform the inputs to the outputs. Softmax, sigmoid, hyperbolic tangent and rectified linear units (ReLU) are the most common activation functions [10]. Each unit in the hidden and output layers computes the weighted sum of the received signals and pass it through an activation function. For instance, assume the q^{th} unit in the first hidden layer that receives signal from N neurons in the input layer. If weight and bias between the input layer and the first hidden layer are represented by $w_{nq}^{(1)}, w_{q0}^{(1)}$, the input of the q^{th} unit can be obtained as follow [10]:

$$a_q = \sum_{n=1}^{N} w_{nq}^{(1)} x_n + w_{q0}^{(1)} \tag{3-5}$$

The output of the hidden unit $z_q^{(1)} = g_1(a_q)$, with $g_1$is the activation function of the neurons in the first layer.

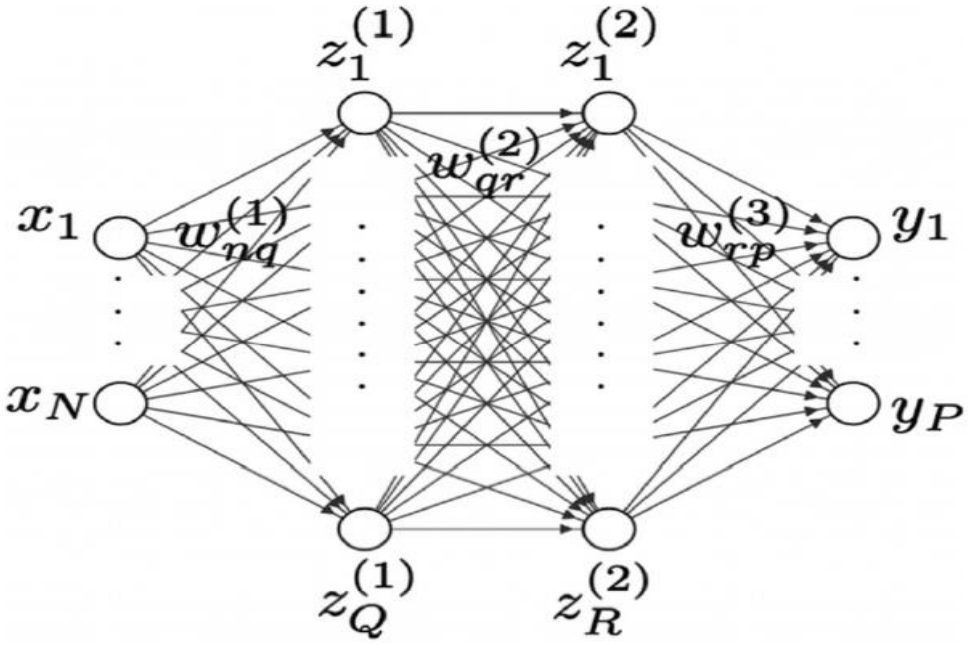


Figure 3.5: A feedforward neural network architecture [28]

To obtain weights and biases of the network, NN training is needed that is based on a loss function and its gradients. A typical loss function, L, for classification is cross-entropy. Given the target values S and input features X, the average cross-entropy L and weight estimate are given by:

$$L_{(w)} = -\frac{1}{P}\sum_{P=1}^{P}\sum_{m=1}^{M} S_{Pm} \ln y_{Pm}$$

$$\widehat{w} = \arg\min_{w} L(\boldsymbol{w}) \tag{3-6}$$

To find the optimum value of the weights, $\widehat{w}$, that results in minimum loss, the gradient of the objective, $\nabla L(w)$, is obtained via backpropagation. It uses the derivative chain rule to find the gradient of the cost with respect to the weights at each NN layer. The simplest weight update is obtained by taking a small step in the direction of the negative gradient [28]:

$$\boldsymbol{W}^{new} = \boldsymbol{W}^{old} - \eta \nabla L\,(\boldsymbol{W}^{old}) \tag{3-7}$$

With η called the learning rate, which controls the step size.

3.2.2.2 Convolutional Neural Networks (CNN)

In recent years, Deep Neural Networks (DNNs) have gained immense popularity due to their excellent performance in various applications. Convolutional NNs (CNNs) are a class of DNN used mainly for analyzing visual imagery. They are designed to diminish the number of weighted links in a conventional NN and memory requirements. In fully connected NNs, each neuron is connected to every neuron in the previous layer, resulting to excessive number of weights especially for large networks. In CNNs, on the other hand, each neuron is connected only with subsets of neurons in the former layer, called filter size. To reduce the number of parameters, for a given filter, the same weights are used for all receptive fields, called weight sharing. Local connectivity and weight sharing not only significantly reduce number of weights but also offer other beneficial properties. Limiting the local receptive field can capture spatially correlated features within an image and consequently results in better representation and recognition. Weight sharing brings about shift invariance property because a filter must model well signal content that is shifted in space. Figure 3.6 shows a deep neural network consists of two main parts: feature hierarchy and classifier.

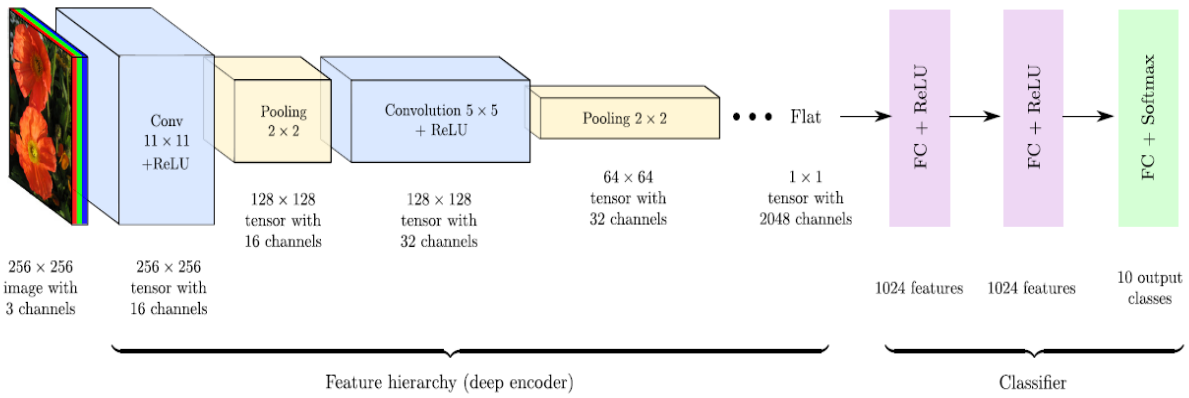


Figure 3.6: Architecture of a CNN [10]

3.2.3 Random Forests

The random forest (RF) classifier is an extension of the decision tree model, which tries to predict classes through series of yes/no questions. Assume N samples (x_n) are available to train

the decision tree. The input data can be partitioned into two regions by defining a cutoff value, c along the ith dimension.

$$\boldsymbol{X}_n \in X_{left} \quad If\ X_{ni} > c$$

$$\boldsymbol{X}_n \in X_{right} \quad If\ X_{ni} \leq c \tag{3-8}$$

Where c is the cutoff value and x_{ni} is the ith dimension of nth sample x_{left} and x_{right} are the left and right regions, respectively. To find the optimum vale of the cutoff in the training process, a cost function, G, is defined as follow:

$$c^* = \arg\min_c G(c) \tag{3-9}$$

$$G(c) = \frac{n_{left}}{N} H\left(x_{left}\right) + \frac{n_{right}}{N} H\left(x_{right}\right)$$

where n_{left} and n_{right} are the numbers of points in the regions left and right regions. H(.) is an impurity function. For the classification problem, the Gini index is usually utilized as the impurity function. It is a measure of homogeneity from 0 (homogeneous) to 1 (heterogeneous).

$$H(x_m) = \frac{1}{n_m} \sum_{x_n \in x_m} I\left(t_n, \ell_m\right) \left[1 - \frac{1}{n_m} I\left(t_n, \ell_m\right)\right] \tag{3-10}$$

where n_m is the number of points in region x_m and l_m represents the assigned label for each region, corresponding to the most common class in the region:

$$\ell_m = \arg\max_{r_k} \sum_{X_{n \in X_m}} I\left(t_n, r_k\right) \tag{3-11}$$

Where r_k are the source range classes and t_n is the label of point X_n in region m, and:

$$I\left(t_n, r_k\right) = \begin{cases} 1 & if\ t_n = r_k \\ 0 & otherwise \end{cases} \tag{3-12}$$

Figure 3.7 Shows the decision tree classifier that partitions the samples into three regions with the two cutoff values.

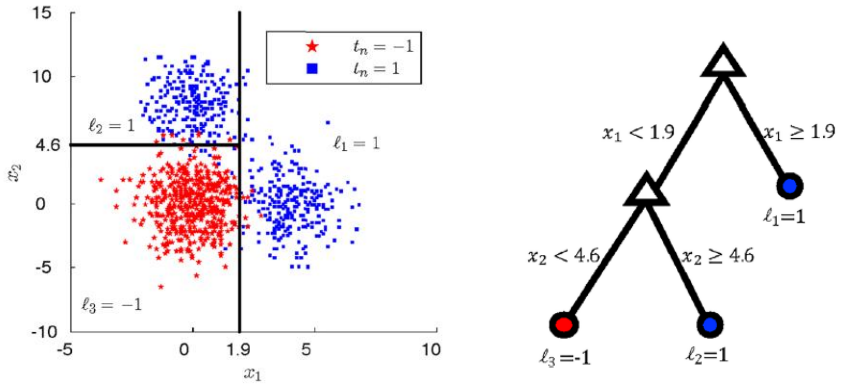


Figure 3.7: Random forest classifier [28]

In order to use RF for regression challenge rather than classification problem, mean of the true class, r_n, for all points in the region should be utilized to obtain the estimated class, l_m, for each region. In addition, the mean squared error is used as the impurity function.

$$\ell_m = \frac{1}{n_m} \sum_{x_n \in x_m} r_n \tag{3-13}$$

$$H\,(x_m) = \sum_{x_n \in x_m} (\ell_m - r_n)^2$$

3.3 Source Localization Algorithm

The proposed method for underwater source localization relies on the machine learning techniques. Unlike model-based methods which require the environmental parameters, data-driven approaches are based on training data to extract subtle information and model the sound propagation underwater. Therefore, quality of collected data plays a pivotal role in this thesis. After data acquisition by the hydrophones, some preprocessing steps such as noise removal should be performed to enhance the quality of the acoustic data. Then, these data are divided into three disjoint parts: training, validation, and test data, to train machine learning models and evaluate their performances. Details of each step will be presented in the following:

3.3.1 Data Acquisition

In order to estimate the location of a moving object based on acoustic positioning method, a sound source is mounted on the ROV and several hydrophones are installed in different positions

in the tank. The sound source, also called beacon or transducer, generates acoustic signals with predefined frequency. The hydrophones convert the received sound signal to electric signal. Figure 3.8 Shows the schematic setup for data acquisition in a tank. Thanks to the confined underwater space, we can move the sound source in finite positions and record the received hydrophone signals. Stationary conditions of the tank guarantee accurate environment perception, using enough training data.

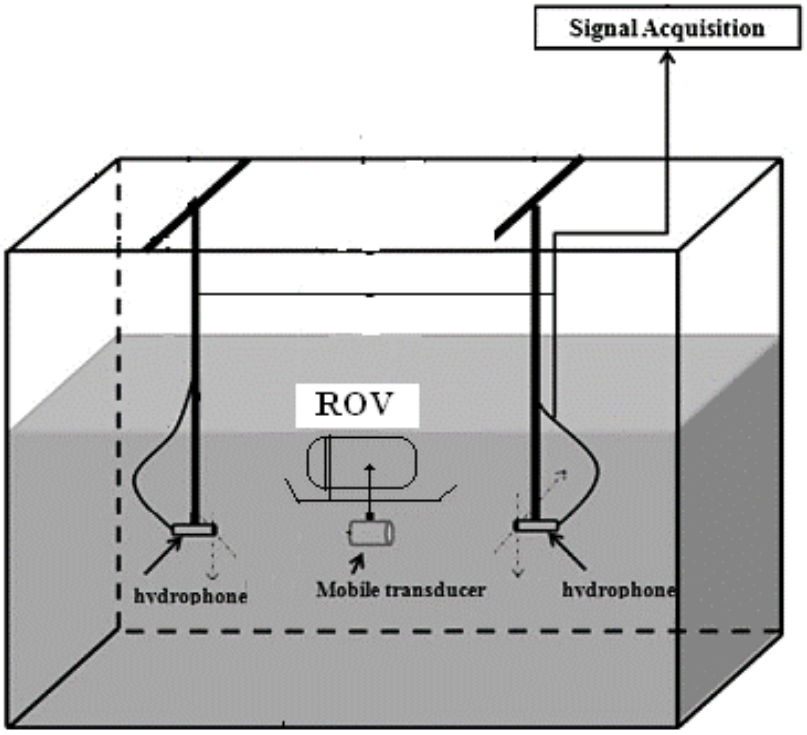


Figure 3.8: Data acquisition in the test tank

Assume that the beacon sends sound signals periodically over time. Figure 3.9 shows the oscillogram that presents the waveform and amplitude of the sound over time. The Spectrogram of the source which is one the most common time-frequency representations is depicted in Figure 3.10 [28]. Since the beacon generates a sharp signal in the time domain, this impulse signal contains a wide range of frequencies, as shown in Figure 3.11.

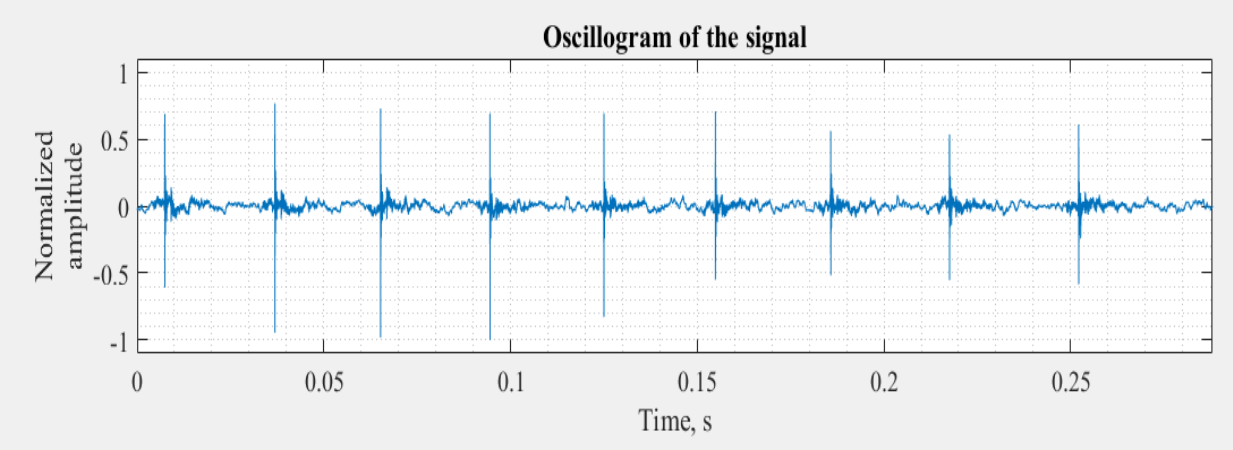


Figure 3.9: Oscillogram of the generated sound signal

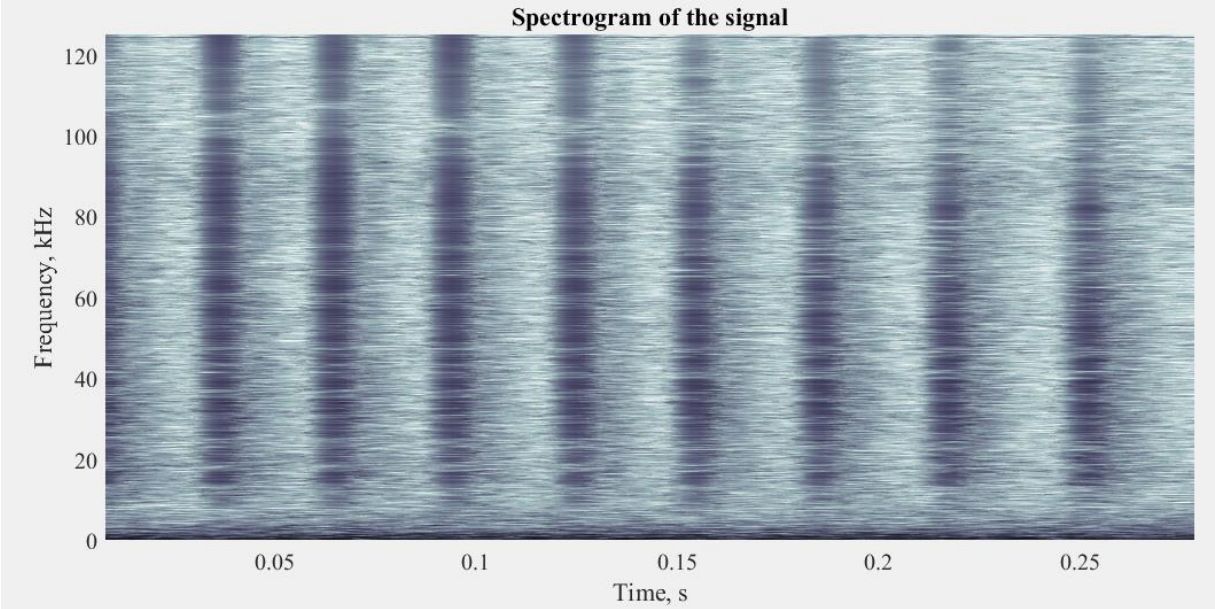


Figure 3.10: Spectrogram of the signal represented in Figure 3.9

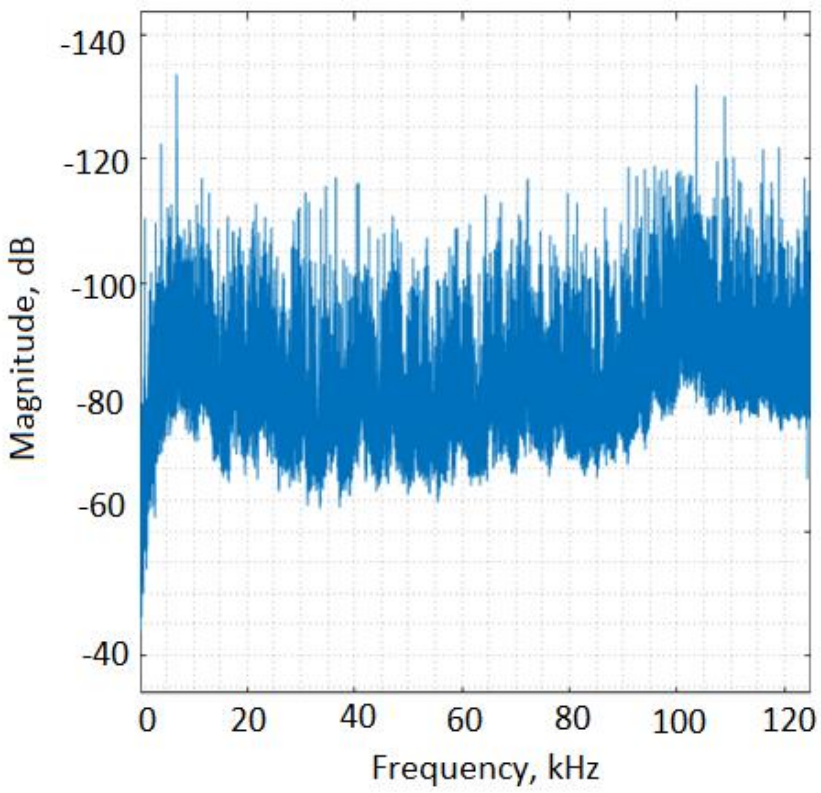


Figure 3.11: Spectrum of the signal represented in Figure 3.9

To analyze the quality of a signal, a chart of correlation statistics (called correlogram) is usually utilized [10]. Figure 3.12 shows the auto-correlogram of the source signal. As it can be seen, with a small shift in the time domain, the autocorrelation coefficient was sharply declined. This means that the sound signal transmitted by the beacon is not similar to itself even after a small-time variance. It should also be noted that the highest the correlation peak, the more precise the TOA, and consequently the more accurate localization results.

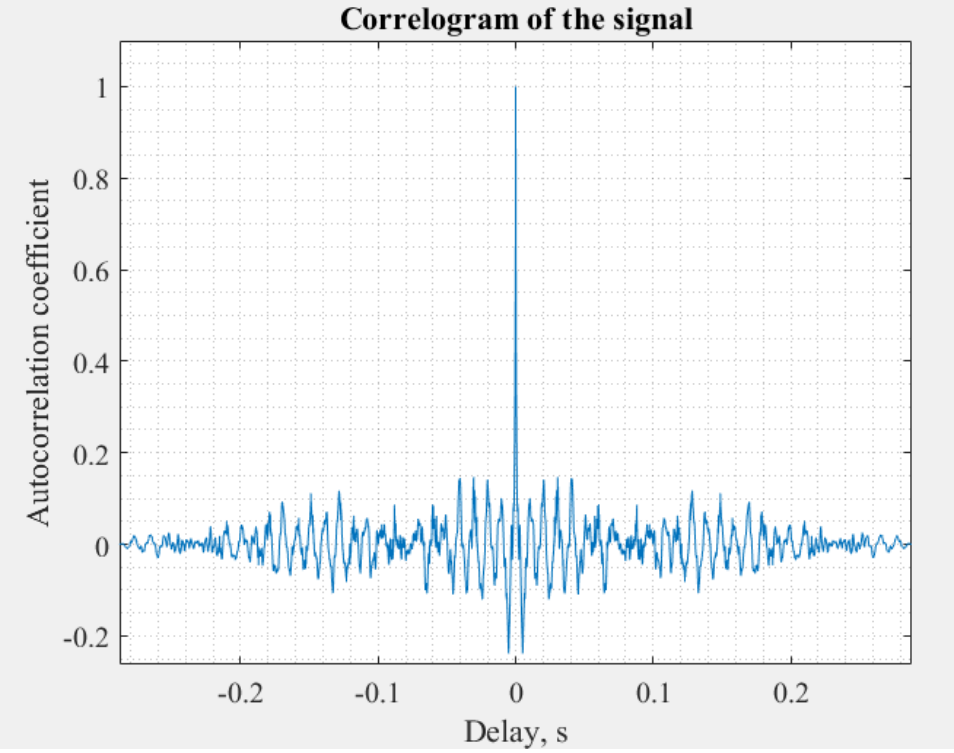


Figure 3.12: Correlogram of the signal represented in Figure 3.9

3.3.2 Data Preparation

There is a consensus among data scientists that there is no general algorithm to perform best on every problem. Different circumstances such as the type of problem, the amount of training, and the cost functions determine the best algorithm for the particular problem. In the other words, by optimizing the parameters of an algorithm to fit the given problem, it can outperform other algorithms.

To select the optimum model, the data should be divided into *three* disjoint sets: training, validation and test sets. The given dataset is usually divided equally to create these subsets. Take Neural Networks (NN) as an example; the training set is used to optimize the parameters of the classifier (the weights) based on the learning algorithm. The validation set is used to optimize the hyperparameters of the algorithm (number of hidden units, learning rate). Finally, the test set is utilized to estimate the true error rate of the model.

Figure 3.13 shows the procedure of using training, validation and test sets for model selection and error estimation. The approach is managed in the following steps:

(i) Divide the available data into training, validation, and test data

(ii) Select the architecture and training parameters

(iii) Train the model using the training set

(iv) Evaluate the model using the validation set

(v) Repeat steps (ii)–(iv) using different architectures and training parameters

(vi) Select the best model and train it using data from the training and validation sets

(vii) Assess the model using the test set

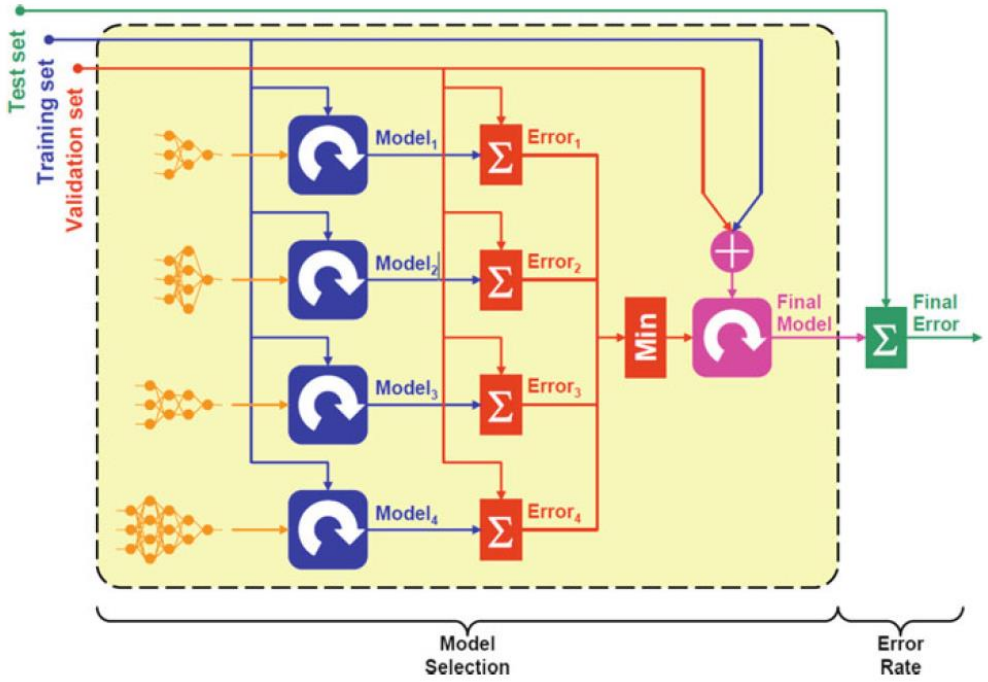


Figure 3.13: The use of training, test, and validation sets in supervised machine learning methods

3.3.3 Error Estimation

To measure the error rates for the given source localization task, mean absolute percentage error (MAPE) over N samples is used as [28]:

$$E_{MAPE} = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{R_{pi} - R_{gi}}{R_{gi}} \right| \quad (3\text{-}14)$$

where R_{pi} and R_{gi} are the predicted range and the ground truth range, respectively. MAPE is frequently used as an error measure in localization applications because it considers both the magnitude of error and the frequency of correct estimates.

3.4 Summary

This chapter briefly reviewed supervised machine learning techniques needed in this thesis. We use Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), Support Vector Machine (SVM), and Random Forest (RF) for underwater sound source localization. Data collection, the keystone of supervised machine learning algorithms, was explained in detail. Finally, Mean Absolute Percentage Error (MAPE) was introduced as a criterion to evaluate the localization methods. Experimental results of the proposed localization method will be explored in the chapter 5.

CHAPTER 4

Implementation

This chapter describes the hardware and experimental setup used in this project. The hardware consists of two main parts: transmitter and receiver.

4.1 Transmitter

In order to generate appropriate acoustic signals, three main modules should be connected as shown in Figure 4.1. FPGA is responsible to generate sound signals, which will be amplified by the transducer driver. Then, the amplified sound signal is fed to the piezoelectric transducer to be transmitted in the underwater environment. In the following each of these blocks will be explained in more details.

Figure 4.1: Schematic of the transmission unit

4.1.1 FPGA

Xilinx LX45 FPGA is used as the transmitter in this project to generate sound signals. In this manner, amplitude and frequency of the sound signal can be controlled. ATLYS board includes this FPGA and PMOD port, which enables us to connect the FPGA to the transducer driver. Figure 4.2 shows ATLYS board used in this project.

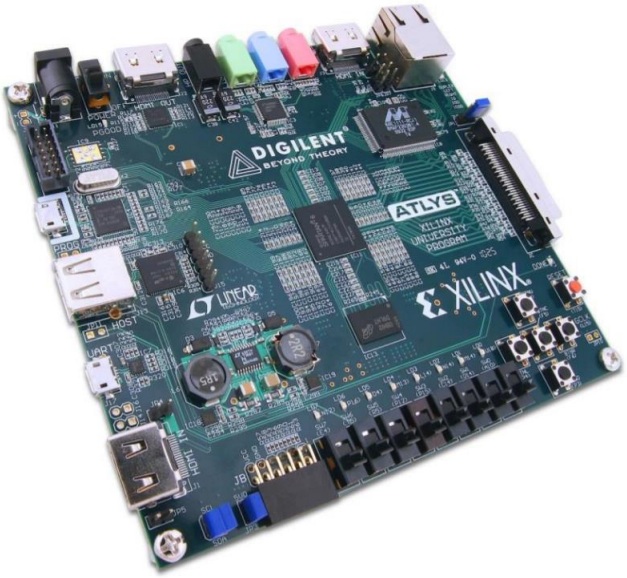


Figure 4.2: ATLYS board for sound generation [29]

The board was designed in a way that allows it to be useful for a variety of applications. According to the top-level diagram of the FPGA implementation, shown in Figure 4.3 the transmitter has two operating modes. These modes can be selected by External Selection Input (the blue command line). If this input is High, the FPGA can communicate with an external PC through an UART interface to receive parameters from the user. Otherwise, the FPGA reads messages that has been previously uploaded into a RAM.

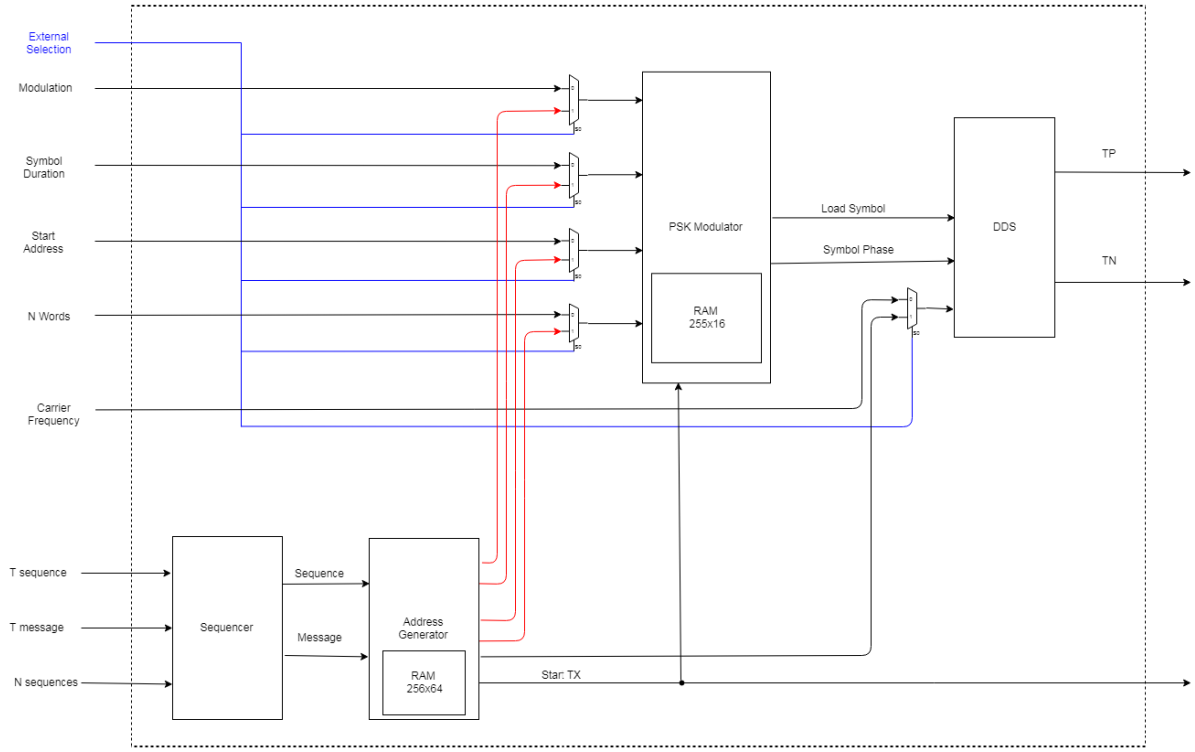


Figure 4.3: Block diagram of the transmitter unit

The FPGA has four modules: Sequencer, Address Generator, PSK Modulator and DDS. The *sequencer* module provides inputs for the *Address Generator* module, according to its inputs. Figure 4.4 shows the output of the *sequencer* for the given inputs, $T_{sequence}$, $T_{messege}$ when $N_{sequence}$ is set to 3 to generate three samples. For underwater localization system in a confined space, $T_{sequence}$, $T_{messege}$ should be set in a manner to guarantee that the generated signal and its reverberation are received by the hydrophones before generating another sound signal.

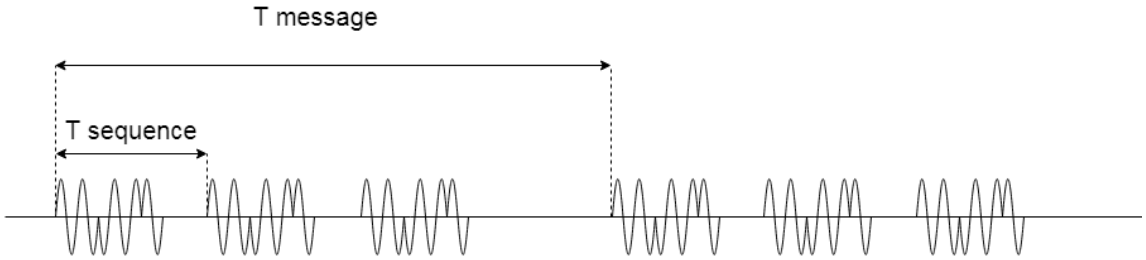


Figure 4.4: Generated signal with 3 samples per each period

The second module, *Address Generator*, receives the *Sequence* and *Message* signals from Sequencer. Whenever the *Sequence* input is high, the *Start TX* output is set to high to inform the *PSK Modulator* that a sequence with the stored parameters in a 256x64 RAM, should be transmitted. However, if the *Message* input is set to high, the parameters relative to the new message are considered to transmit a signal. This module also has four outputs that are connected to the inputs of the *PSK Modulator.*

The *PSK Modulator* performs modulation based on either its internal RAM contents or external parameters (*Modulation*, *Symbol Duration*, *Start Address*, and N_{words}). When Start TX is set to high, it starts reading N_{words} words from its RAM, starting at *Start Address*. The type of modulation (BPSK or QPSK) and the duration of each symbol are determined by *Modulation*, *Symbol Duration* inputs.

The final module in the FPGA is *Direct Digital Synthesis* (DDS) that relates each phase value with the logical values to be input into the *Transducer Driver* to generate a sinusoidal signal. This operation is performed by a Look Up Table (LUT) which is implemented on a 128x2 RAM. Whenever *Load Symbol* is active, *Symbol Phase* and the carrier's period are added to the current phase. The resulting phase is looked up in the LUT to generate TP and TN outputs.

4.1.2 Transducer Driver

The transducer driver utilized in this project consists of two transistors and a transformer (Figure 4.5). This block has two inputs (TP, TN) that receive signals from the *Direct Digital Synthesis* module in FPGA. The two transistors are driven by TP and TN to allow current to flow through the transformer in opposite directions. For example, when TP=1 and TN=0, the top transistor is on and a positive signal is generated at the output. Figure 4.6 shows the output signal of the transducer driver for different input values. By alternating the logic values of the inputs, a semi-sinusoidal signal can be generated at the output.

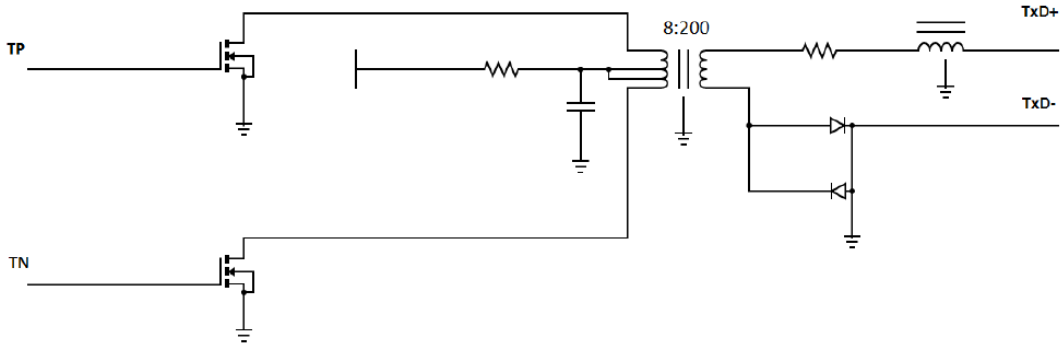


Figure 4.5: Transducer driver circuit

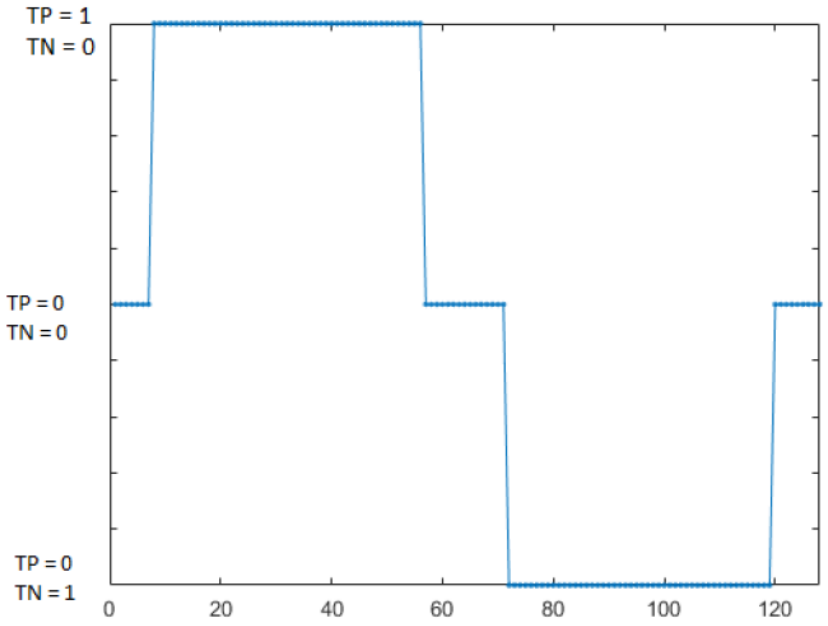


Figure 4.6: Transducer output for different command values

4.1.3 Transducer

After generating electrical signal by the driver, transducer should be utilized to convert the electrical signal to sound signal. We used Neptune Sonar company transducer (Figure 4.7) which can transmit signals with frequencies ranging from 16kHz and 30 kHz and has a maximum transmission power of 400W.

Figure 4.7: Transducer [30]

4.2 Receiver

The reception block is responsible for filtering and amplifying the received signal, and converting it to digital format using analog to digital convertor (ADC) for signal processing and storage. Figure 4.8 depicts the three main blocks of the receiver.

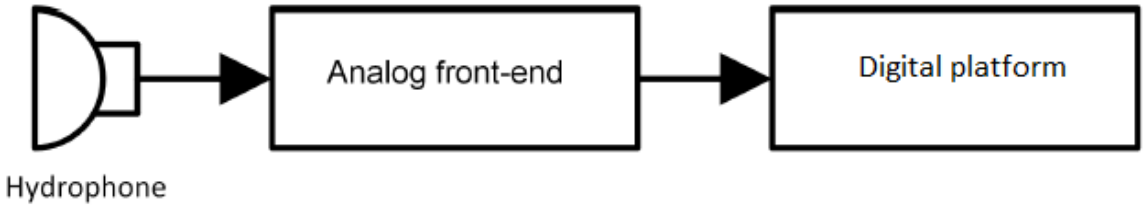


Figure 4.8: Schematic of the receiver unit

4.2.1 Hydrophones

In this project four hydrophones of the model TC4013 made by Teledyne company were utilized to acquire acoustic signals [31]. They have a frequency range between 1 Hz and 170 kHz and a high sensitivity relative to its size. Figure 4.9 shows one of these hydrophones. The overall characteristics of this omnidirectional hydrophone are shown in Figure 4.10.

Figure 4.9: Hydrophone [31]

TECHNICAL SPECIFICATIONS	
Usable Frequency range:	1Hz to 170kHz
Receiving Sensitivity:	-211dB ±3dB re 1V/µPa
Transmitting Sensitivity:	130dB ±3dB re 1µPa/V at 1m at 100kHz
Horizontal Directivity Pattern:	Omnidirectional ±2dB at 100kHz
Vertical Directivity Pattern:	270° ±3dB at 100kHz
Nominal capacitance:	3.4nF
Operating depth:	700m
Survival depth:	1000m
Operating temperature range:	-2°C to +80°C
Storage temperature range:	-40°C to +80°C
Weight (in air):	75g
Cable length:	Standard length 6m Optional cable lengths available on request
Encapsulating material:	Special formulated NBR

Figure 4.10: Technical details of the hydrophone [31]

4.2.2 Analogue Front-end

In order to convert sensor output to a proper input signal for an ADC, analog front-ends are required. They usually consist of Op-Amps and filters, and other necessary signal processing circuits needed for best performance. In this project, hydrophones are connected to an analog front-end board consisting of a two-stage amplifier, followed by a low-pass filter (Figure 4.11). The first stage of the amplifier has a fixed gain (10x), while the gain of the second stage can be adjusted in the range of (0.1x-2500x). A low pass filter has the cut-off frequency of 250kHz.

Since the Red Pitay board (Figure 4.12) has only two analog inputs, two of them are required to handle four hydrophones: a primary and a secondary one. The inputs of the hydrophones H_0 and H_2 are multiplexed into the channel A of the board, while H_1 and H_3 are multiplexed into channel B.

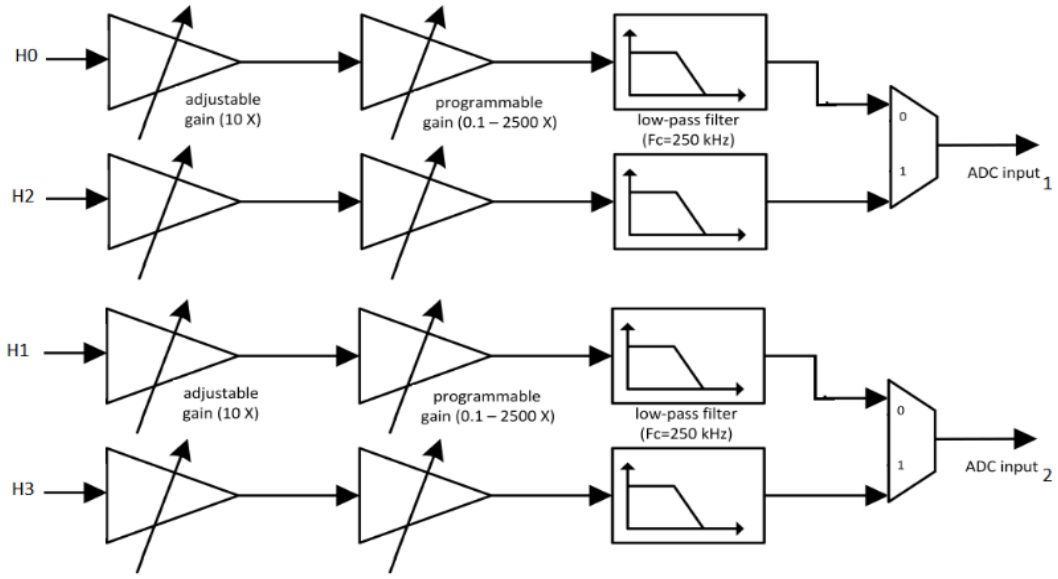


Figure 4.11: Schematic of the analog front-end board

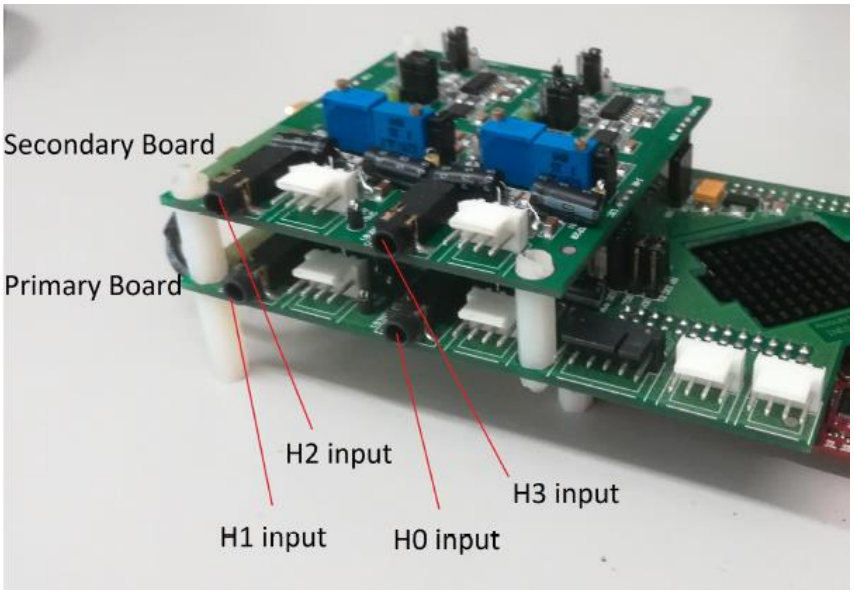


Figure 4.12: The Red Pitay board

4.2.3 Digital Platform

Received sound signals are converted to electric signals by the hydrophones. Then, the electric signals are amplified and filtered by an analog front-end module. The final module is a digital platform that is required to sample the processed analog signals and convert them into digital numeric values to be read by. This process is called Data Acquisition (DAQ) and enables us to manipulate and store the digital signals by a computer.

Figure 4.13 shows main blocks in a typical DAQ board which is an interface between real world and a computer. Signal conditioning circuitry transforms noisy real-world signals into appropriate forms needed for the next stage. For instance, in an analog-to-digital converter (ADC) application, signal voltage limiting, and anti-aliasing filtering can bring about more accurate measurement. The second block is ADC that converts real-world analog data into digital signal required for computers. DAQ relies on computer bus to transmit data to a computer. The most common computer buses are USB, PCIe, or Ethernet. Figure 4.14 shows the effect of the different DAQ submodules on the input analog input signal.

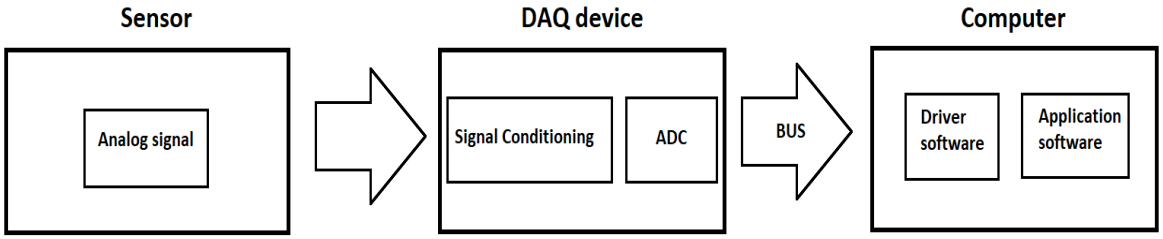


Figure 4.13: Schematic of the data acquisition system

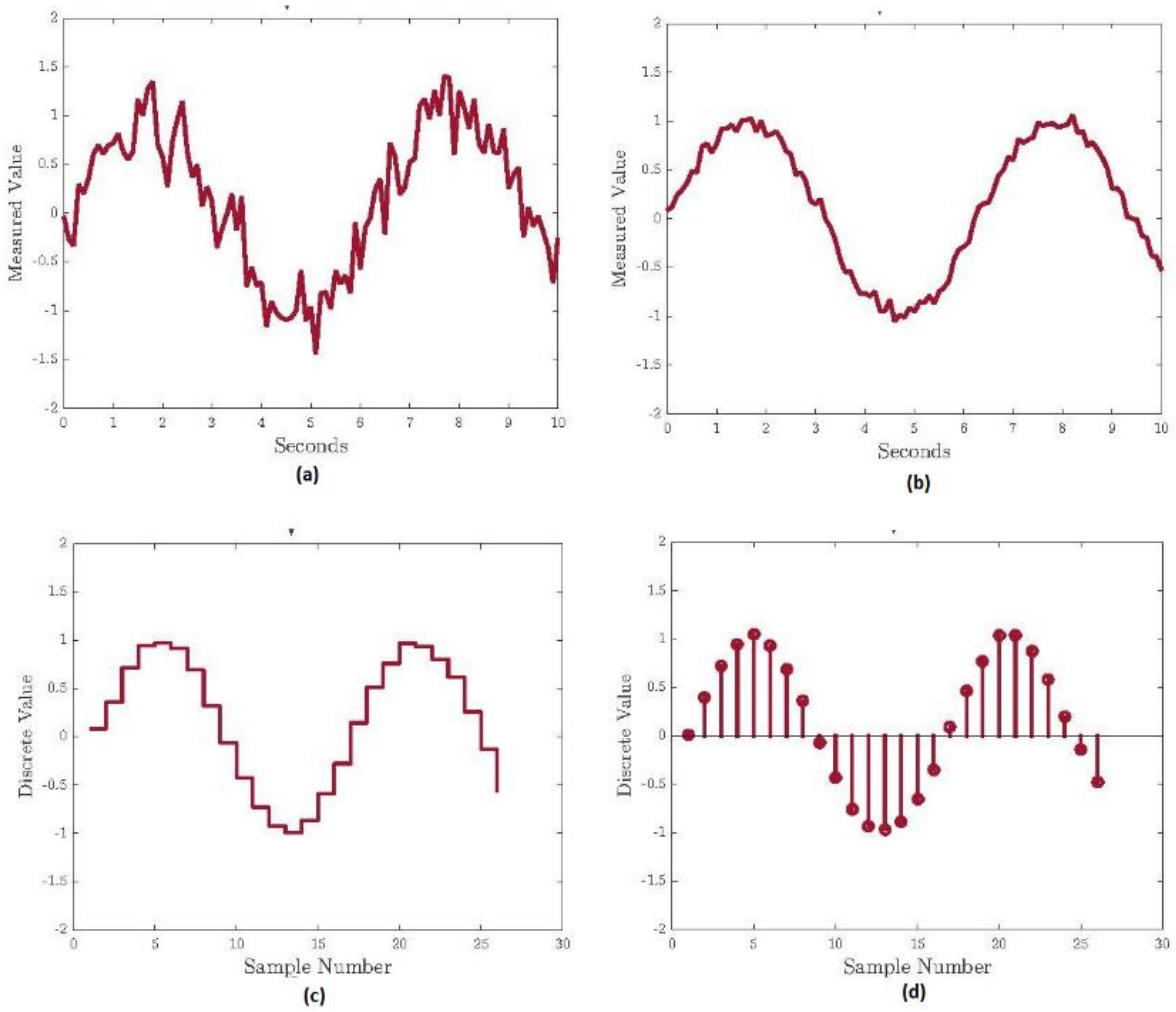


Figure 4.14: Different signals demonstration

(a) Analog input signal. (b) Conditioned signal. (c) Digitized signal. (d) Sampled signal.

To perform different tasks such as visualization, processing, and storage on the measurement data, DAQ software should be installed on the computer. DAQ software includes *driver software* and *application software*. Through driver software, application software can control the DAQ device with menu-based configuration or an Application Programming Interface (API). National Instrument (NI) company [32], for example, has developed NI-DAQmx driver software which is compatible with NI data acquisition hardware. NI-DAQmx enables users to reduce development time and take full advantage of their data acquisition applications. NI company also provided FlexLogger application software to facilitate building data-logging systems with NI DAQ

hardware, without any programming. Apart from driver software and application software, users can utilize *programming environment* to develop their own application to acquire, analyze, and present data. Programming environment, for instance, enables users to have access to low-level timing and triggering information as well as synchronize with multiple devices. By using libraries APIs, users can access and control their DAQ device in a more flexible way. LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a programming environment based on the graphical language, named G, for a visual programming from National Instruments [32]. Figure 4.15 shows a compact DAQ with embedded LabVIEW software [32].

Figure 4.15: NI cDAQ 9134 [32]

In this project we used NI 9234 data acquisition module and NI-DAQmx driver software, both developed by National Instruments [32]. The main reason for selecting NI 9234 module is that it

is a four-channel dynamic signal acquisition module and enables us to acquire information from four hydrophones simultaneously. This data acquisition module delivers 102 dB of dynamic range and signal conditioning at 2 mA constant current for hydrophones. The maximum sampling rate of it is 51.2 kS/s, which can be automatically adjusted by built-in anti-aliasing filters. Figure 4.16 shows the NI 9234 data acquisition module.

Figure 4.16: NI 9234 DAC [32]

Figure 4.17 shows the circuitry of the NI 9234 data acquisition module. Signal conditioning is performed on each analog input signal (AI) through prefilter and differential amplifier block. Filtering is performed by a set of analog and digital filters that differentiate in-band and out-of-band signals according to the frequency range, or bandwidth of the signals. A Delta-Sigma ADC converts the conditioned signal into a digital signal. The analog input is connected to chassis ground through two parallel diodes and a 50 Ω resistor for safety measures and over-voltage protection. Each channel can be configured for AC or DC coupling by turning the Integrated Electronics Piezoelectric (IEPE) excitation current on or off via driver software.

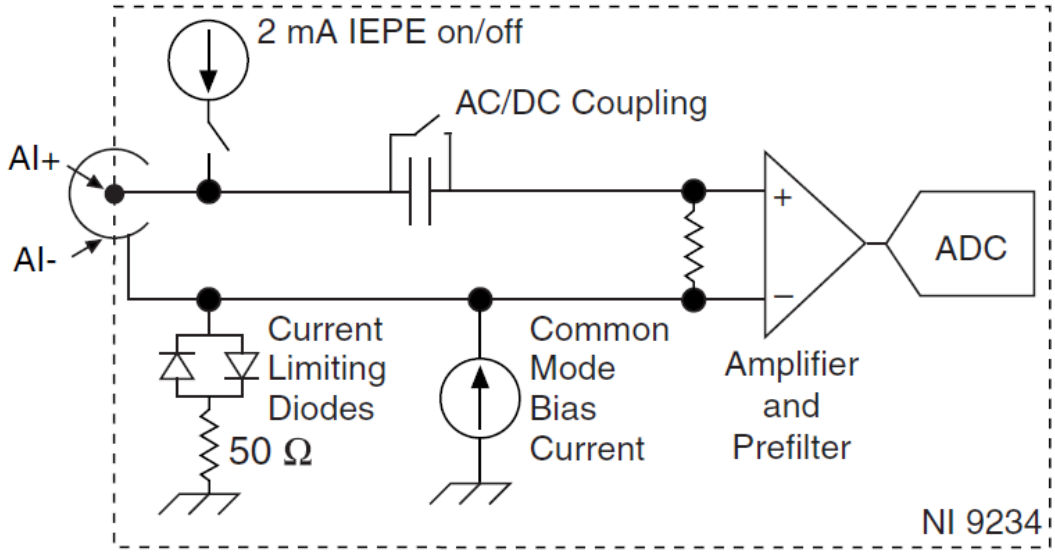


Figure 4.17: Schematic of the NI 9234 board [32]

Sampling rate frequency (f_s) is determined by frequency of a master time base (f_M), which is 13.1072 MHz in NI 9234. Equation 4.1 provides the available data rates:

$$f_s = \frac{f_M \div 256}{n} \tag{4-1}$$

where n is any integer from 1 to 31. Depending on the value of n, sampling frequency can increase from 1.652 kS/s to 51.2 kS/s. Table 4.1 Indicates the NI 9234 overall specifications.

Table 4.1: Technical details of NI 9234 board [32]

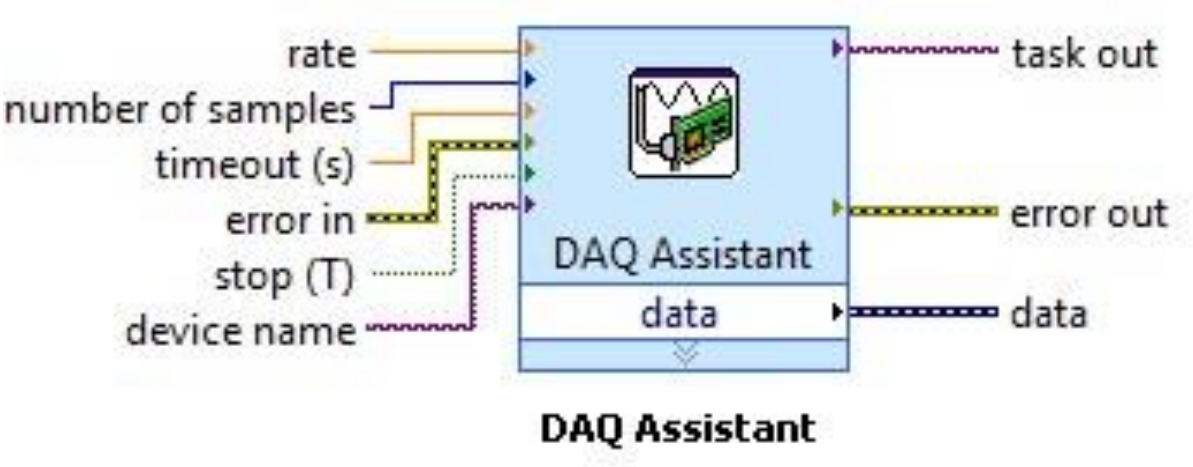
Input Characteristics	Number of channels	4 analog input channels
	Input coupling	AC/DC (software-selectable)
	Input range	±5 V
	Overvoltage protection	-6 V to 30 V
	Input impedance	305 kΩ
Sampling Characteristics	ADC resolution	24 bits
	Type of ADC	Delta-Sigma
	Sampling mode	Simultaneous
	Internal master timebase	13.1072 MHz
	Minimum data rate	1.652 kS/s
	Maximum data rate	51.2 kS/s
Accuracy	Gain Error	0.05%, ±0.005 dB
	Offset Error	±0.04%, 2.3 mV
	Total Harmonic Distortion (THD)	-95 dB
Power Requirements	Active mode	900 mW
	Sleep mode	25 μW
Physical Characteristics	Weight	173 g

4.3 Software

As mentioned earlier, to transfer measured data into computer for processing and storage, driver software and application software are required.

4.3.1 Driver Software

In this project we used NI-DAQmx driver software [32] to take full advantage of many features of NI 9234 data acquisition hardware. DAQ Assistant is one of the main features in NI-DAQmx that facilitates development experience through obviating the need for programming. Figure 4.18 shows DAQ Assistant graphical interface. User can utilize this interface to create a virtual channel consisting of a physical channel on a DAQ device and the configuration information. Acquisition or generation parameters including timing and triggering information, input range, and custom scaling can be set by DAQ Assistant (Figure 4.18).



DAQ Assistant

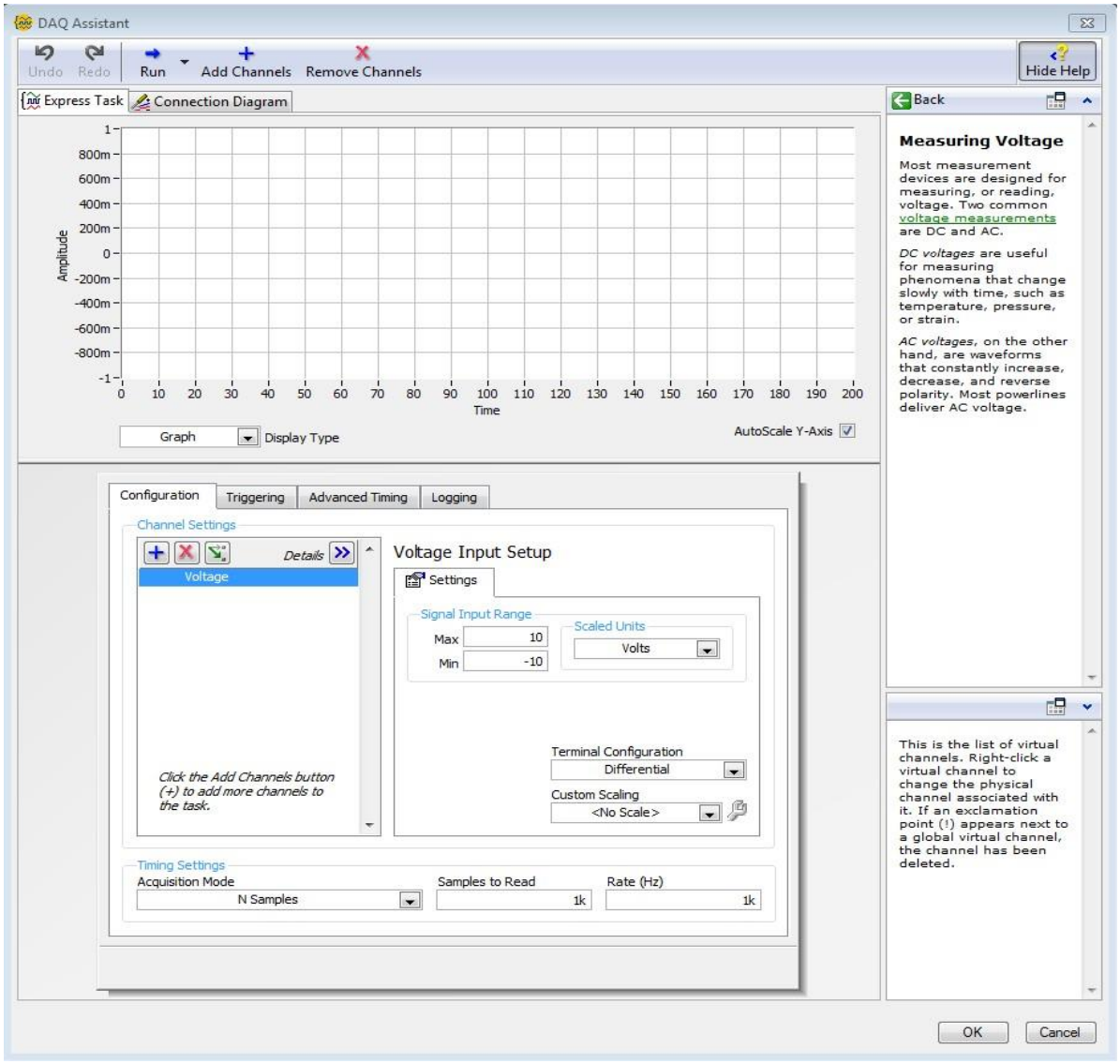


Figure 4.18: DAQ assistant graphical interface

DAQ Assistant can also generate code to configure and use the task in an application program such LabVIEW. In this manner specified acquisition or generation tasks can be performed with higher flexibility. Figure 4.19 displays how LabVIEW code can be generated automatically and also shows the resulting generated LabVIEW code.

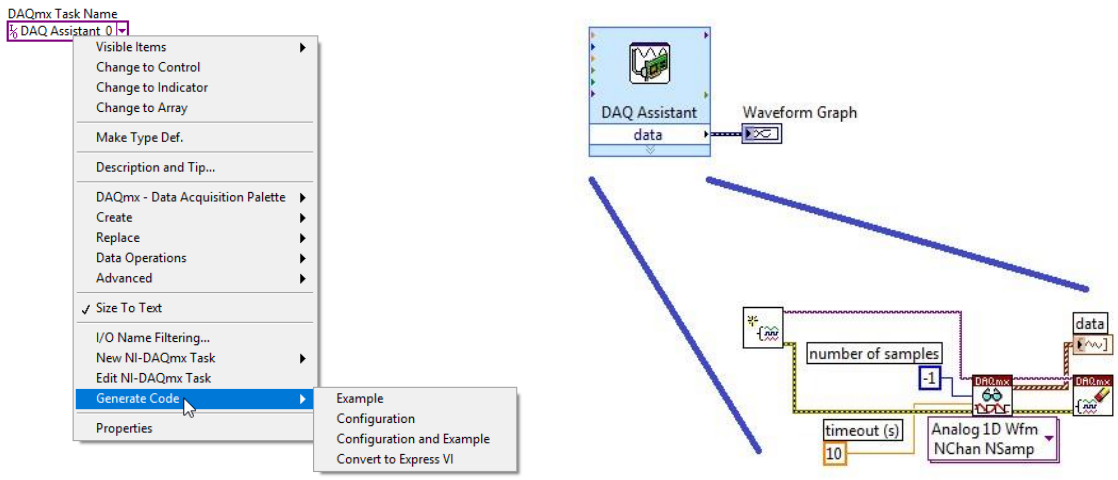


Figure 4.19: Automatic LabVIEW code generation

4.3.2 Application Software

Data acquisition applications are usually controlled by using various general-purpose programming languages. LabVIEW and MATLAB are two common programming languages which have been used frequently to visualize, process, and store measured data by a hardware. MATLAB *Data Acquisition Toolbox* is used in this project to monitor, analyze, and save sound signals for underwater localization application [33]. With this toolbox we can configure NI 9234 data acquisition hardware and transfer its received sound data into MATLAB software for localization.

In addition to receiving data from DAQ devices, user can send out data over device-specific analog and digital output channels. Several functions are available for controlling analog and digital input/analog, counter/timer in a DAQ device. Data Acquisition Toolbox can synchronize multiple devices to acquire signals from several DAQ devices.

The toolbox supports well-established DAQ vendors including National Instruments. Table 4.2 shows the list of vendors supported in the MATLAB.

Table 4.2: List of DAQ vendors supported by MATLAB [33]

Support Package	Vendor	Earliest Release Available	Last Release Available
Analog Devices Hardware	Analog Devices®	R2015b	Current
Digilent Analog Discovery Hardware	Digilent®	R2014a	Current
Measurement Computing Hardware	Measurement Computing™	R2017a	Current
National Instruments NI-DAQmx Devices	National Instruments®	R2014a	Current
Windows Sound Cards	Microsoft®	R2014a	Current

After conncting a hydrophone to channel 0 of NI 9234 data acquistion hardware, and attaching DAQ device to the computer via USB port, we can discover list of supported devices by the *daqlist* command in MATLAB. Figure 4.20 shows the output of the MATLAB command. For the sake of simplicity, other DAQ devices such as Microsoft internal sound card are not shown in this figure.

```
d = daqlist("ni")
```

```
 DeviceID                Description                        Model            DeviceInfo
____________    ___________________________________    _____________    ____________________
"cDAQ1Mod1"     "National Instruments NI 9234"         "NI 9234"        [1×1 daq.DeviceInfo]
```

```
deviceInfo = d{1, "DeviceInfo"}
```

```
deviceInfo =

ni: National Instruments NI 9234 (Device ID: 'cDAQ1Mod1')
   Analog input supports:
      -5.0 to +5.0 Volts range
      Rates from 1000.0 to 51200.0 scans/sec
      4 channels ('ai0','ai1','ai2','ai3')
      'Voltage','Accelerometer','Microphone','IEPE' measurement types
```

Figure 4.20: Recognition of the connected DAQ device by MATLAB

The next step is to create a data acquisition and add a channel with microphone measurement type (Figure 4.21). Parameters of the microphone channel such as sensitivity property can be set to the value specified in the sensor's data sheet.

```
dq = daq("ni");
ch = addinput(dq, "cDAQ1Mod3", "ai0", "Microphone");
```

```
ch.Sensitivity = 0.037;
ch
```

```
ch =

    Index    Type      Device       Channel     Measurement Type             Range                 Name
    _____    ____    __________    _______    __________________    ____________________    _____________

      1      "ai"    "cDAQ1Mod1"     "ai0"     "Microphone (Diff)"    "-200 to +200 Pascals"    "cDAQ1Mod1_ai0"
```

Figure 4.21: Creating a channel to receive hydrophone signals

We can set the acquisition scan rate (scans per second) and use *read* command to acquire four seconds of data. To play back the acquired microphone signal *audioplayer* command can be utilized.

```
dq.Rate = 51200;
tt = read(dq, seconds(4));
t = tt.Time;
data = tt.cDAQ1Mod3_ai0;
```

```
p = audioplayer(data, dq.Rate);
play(p);
```

Figure 4.22: MATLAB commands for recording and playing hydrophone sound signals

The acquired data can be presented in time or frequency domains. It should be noted that by non-blocking commands, user can work in the MATLAB command window during the acquisition process. This is called *background* acquisition. On the other hand, using *foreground* acquisition causes MATLAB to wait for the entire acquisition to complete before executing next command.

4.4 Experimental Setup

All experiments were performed in a confined underwater environment containing fresh water. The test tank has a length of 6 m, a width of 4.5 m and a depth of 1.7 m. The effects of reverberation and multi-path are considerable due to small dimensions of the pool. The beacon

was mounted at the top of the ROV and moved to the different positions within the tank. Hydrophones were located at the different positions inside the confined tank (hydrophone layouts will be discussed in chapter 5). In the next chapter, accuracy of the proposed underwater localization method will be examined as a function of beacon and hydrophones positions.

4.5 Summary

This chapter focused on the practical aspects and experimental setup of this project. Hardware and software units of the transmitter and receiver blocks were explained in detail. Then data acquisition board required to convert captured data into computer readable format was explained. Finally, underwater test tank, used to collect data and evaluate the performance of the proposed localization system, was described.

CHAPTER 5

Experimental Results

This chapter begins with the description of acoustic data set generation to perform underwater source localization based on machine learning techniques. For this purpose, feature extraction methods should be applied on the received acoustic data, and the results should be stored in a data base. In this project, three different feature extraction approaches were deployed: pitch of the acoustic signal, the Mel-Frequency Cepstrum Coefficients (MFCC), and Speech Spectrogram (SS) [34]. The prepared data set is divided into train, validation, and test subsets, as discussed in chapter 3. Then, Machine Learning (ML) techniques will be utilized to train and finetune classifiers using the training and validation sets, respectively. Performance of the proposed underwater ROV localization method will be evaluated by the validation set.

5.1 Feature Extraction

Data-driven ML approaches heavily rely on feature extraction phase to reduce dimensionality of their input patterns for better data representation. From pattern recognition point of view, feature extraction refers to the process of finding the most discriminative information from raw data which maximizes class separation. In acoustic signal processing, feature extraction methods can broadly fall into three main groups [34]: (i) temporal features, (ii) spectral features, and (iii) spectro-temporal features. Since the *temporal* features are defined in time domain, they can be easily extracted and have easy physical interpretation. Examples of temporal features include: the energy of signal, zero crossing rate, maximum amplitude, and minimum energy. On the other hand, *spectral* features are frequency-based features. Fourier Transform converts the time domain signal into the frequency domain. Fundamental frequency, frequency components, and power spectral density are the most common spectral features used in speech recognition. While temporal features and spectral features are one-dimensional (1D) features, *spectro-temporal* features can be regarded as two-dimensional (2D) features. Spectrogram, for instance, is a visual representation of the spectrum of frequencies of a signal as it varies with time. The main reason for using 2D features is to leverage convolutional neural networks which have shown satisfactory results for image classification.

5.1.1 One-dimensional Features

This subsection briefly describes *pitch* and Mel-Frequency Cepstrum Coefficients (*MFCC*) as one-dimensional features.

- Pitch [34]

Sound tones can be described by several auditory attributes such as pitch, duration, loudness, and timbre. Pitch is a perceptual attribute that determines if a sound is "higher" or "lower". Since only sounds with clear and stable frequency have pitch, this attribute can be used to differentiate *voice* from *noise*. Figure 5.1 shows time-domain representation of the pronounced word "two". It consists of two parts: the consonant “T” and the vowel “WO”. While the consonant part can be regarded as noise sound (t<2.7s), the vowel segment (t>2.7s) has a strong fundamental frequency, so it can be considered as voiced sound.

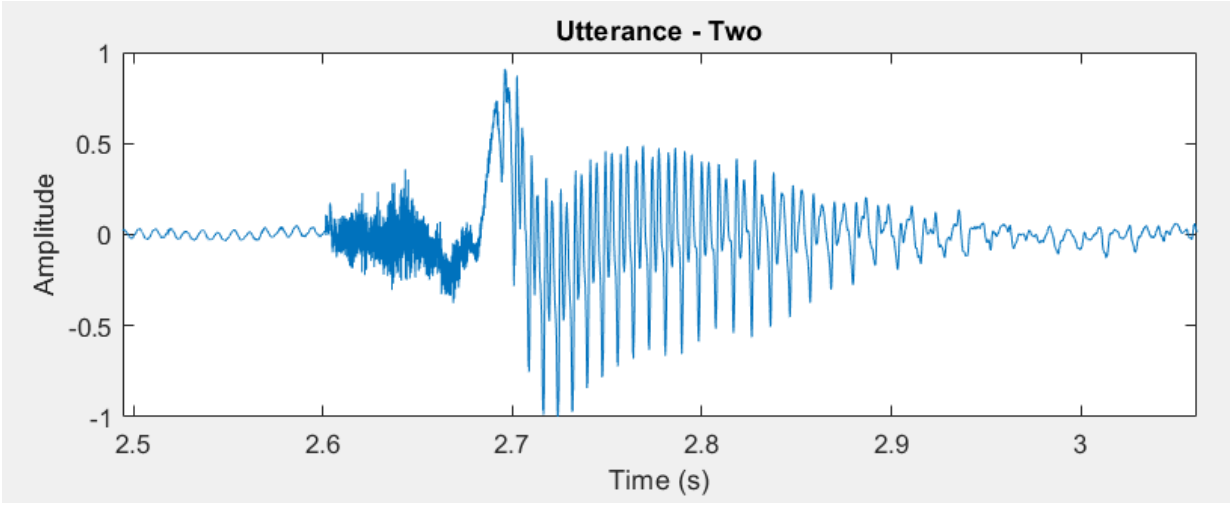


Figure 5.1: Time-domain representation of a sound signal

To measure the pitch, the sound signal is regarded as a set of overlapping windows. The sound signal in each window is assumed to be stationary without considerable change over time. In this example window size is set to 30 ms with a 25 ms overlap. Figure 5.2 shows pitch counter of the sound in Figure 5.1.

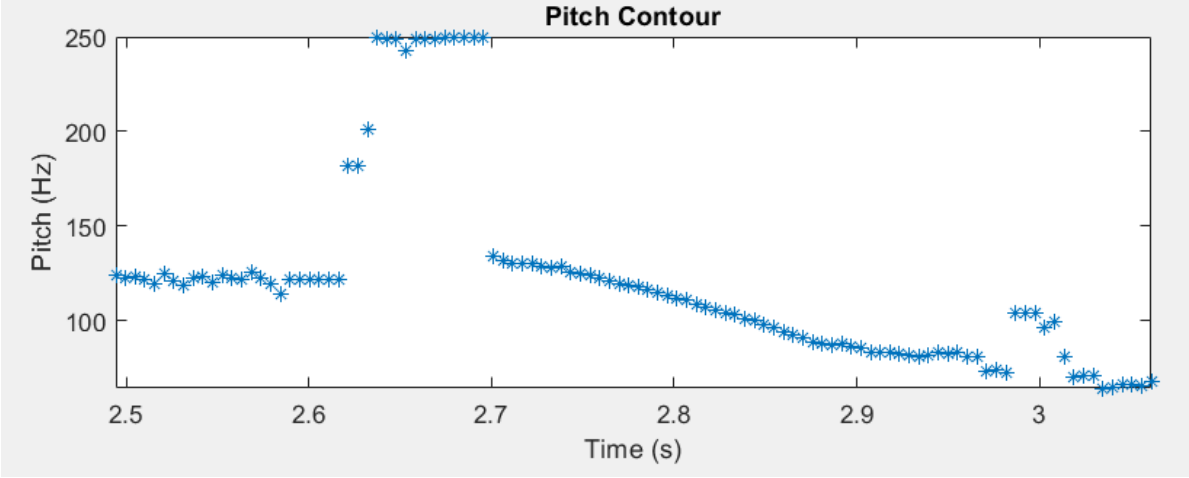


Figure 5.2: Pitch counter of the sound in Figure 5.1

To differentiate voiced and noise sounds, zero crossing rate (ZCR) can be used. ZCR is defined as the rate at which the signal changes from positive to negative or vice versa. ZCR above a given threshold implies that the sound is noise-like, while small ZCR shows that the sound has a dominant frequency and can be considered as a voiced sound. Figure 5.3 shows the voiced signal after applying a threshold on ZCR.

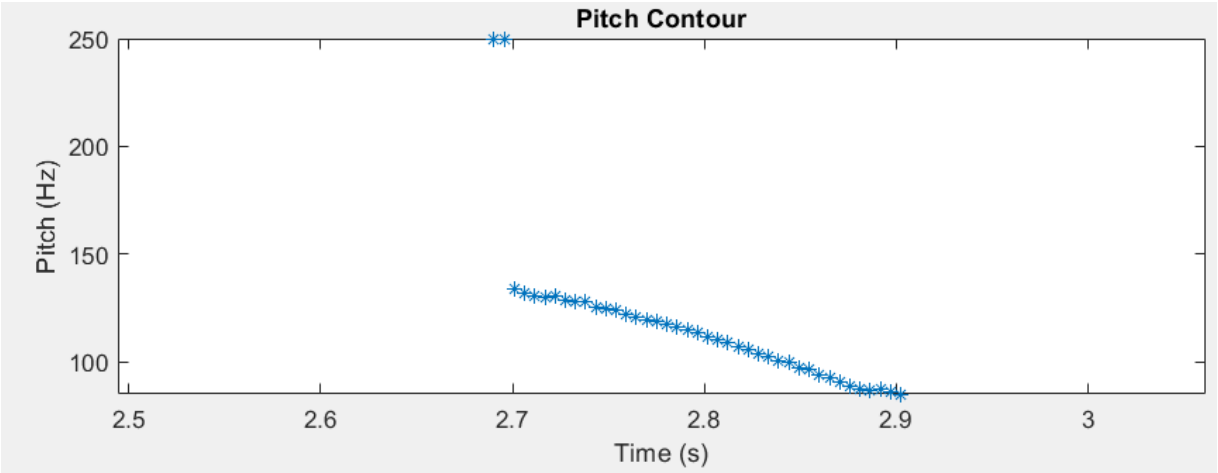


Figure 5.3: Pitch counter after thresholding Figure 5.2 with threshold value of 300

- Mel-Frequency Cepstrum Coefficients (*MFCC*) [34]

In this project we extract MFCC as a spectral feature vector from underwater sound signals. Short-term power spectrum of a sound can be represented by Mel-frequency cepstrum (MFC).

MFCCs are coefficients that collectively make up an MFC. Figure 5.4 illustrates how MFCC can be derived for the speech signal. Figure 5.5 shows 13 MFCCs of the sound signal. This process can be performed as follows [34]:

- Take the Fourier transform of a signal in a window.
- Map the powers of the spectrum obtained above onto the Mel scale, using triangular overlapping windows.
- Take the logs of the powers at each of the Mel frequencies.
- Take the discrete cosine transform of the list of Mel log powers.
- The MFCCs are the amplitudes of the resulting spectrum.

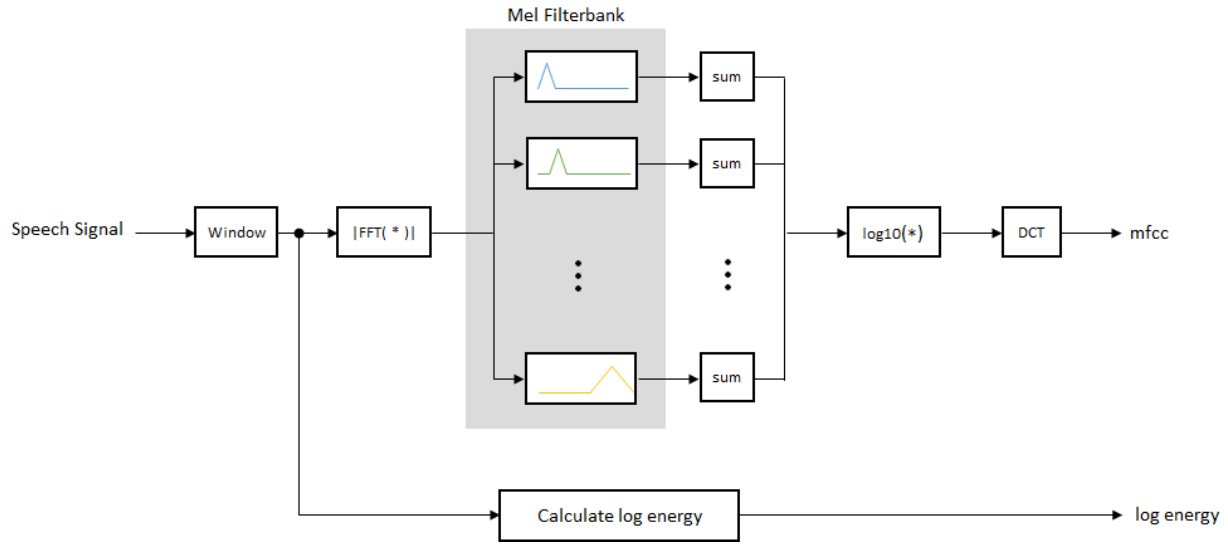


Figure 5.4: MFCC extraction process

Pitch	MFCC1	MFCC2	MFCC3	MFCC4	MFCC5	MFCC6	MFCC7	MFCC8	MFCC9	MFCC10	MFCC11	MFCC12	MFCC13
237.44	-4.4218	3.3816	0.73331	0.98626	0.47093	0.13808	-0.083348	0.069072	0.2345	0.3403	-0.14417	-0.15685	0.022186
242.42	-4.3104	4.7899	0.80432	0.7148	0.46027	0.032963	-0.28647	0.38366	0.1449	0.0093271	-0.2559	-0.17832	-0.11693
231.88	-3.6432	5.0192	0.74801	0.58299	0.50475	-0.014551	-0.32653	0.39201	0.20982	-0.20282	-0.25637	-0.20576	-0.27675
230.89	-3.0934	5.132	0.46794	0.57104	0.64546	-0.085145	-0.22453	0.55408	0.14131	-0.17966	-0.17135	-0.22111	-0.22027
112.49	-2.9718	5.3249	0.48934	0.66976	0.56446	-0.14691	-0.26824	0.4536	0.31515	-0.21356	-0.34067	-0.21872	-0.14108
111.89	-2.6202	5.2746	0.53966	0.55468	0.50989	0.012264	-0.26755	0.3318	0.32108	-0.18096	-0.44212	-0.21208	-0.21385
111.11	-2.6138	5.0492	0.68513	0.40281	0.36792	0.13352	-0.07321	0.25863	0.25314	-0.1787	-0.51149	-0.14679	-0.077431
110.1	-2.4483	5.5192	0.64449	0.44857	0.25178	0.25716	0.042426	0.32466	0.17774	-0.194	-0.70127	-0.16868	-0.041083

Figure 5.5: One-dimensional features values including pitch and 13 MFCCs

5.1.2 Two-dimensional Features

Apart from one-dimensional features described in section 5.1.1, we utilize spectrogram which is a spectro-temporal feature. Spectrogram shows the presence of a particular frequency at a specific time. For the sake of simplicity, amplitude of each frequency is proportional to the color of each point in the spectrogram. Figure 5.6 shows spectrogram of a sound signal. Such two-dimensional way of representing the signal enables us to leverage convolutional neural networks which have shown promising results in visual imagery applications. More details of spectrogram calculation can be found in [34].

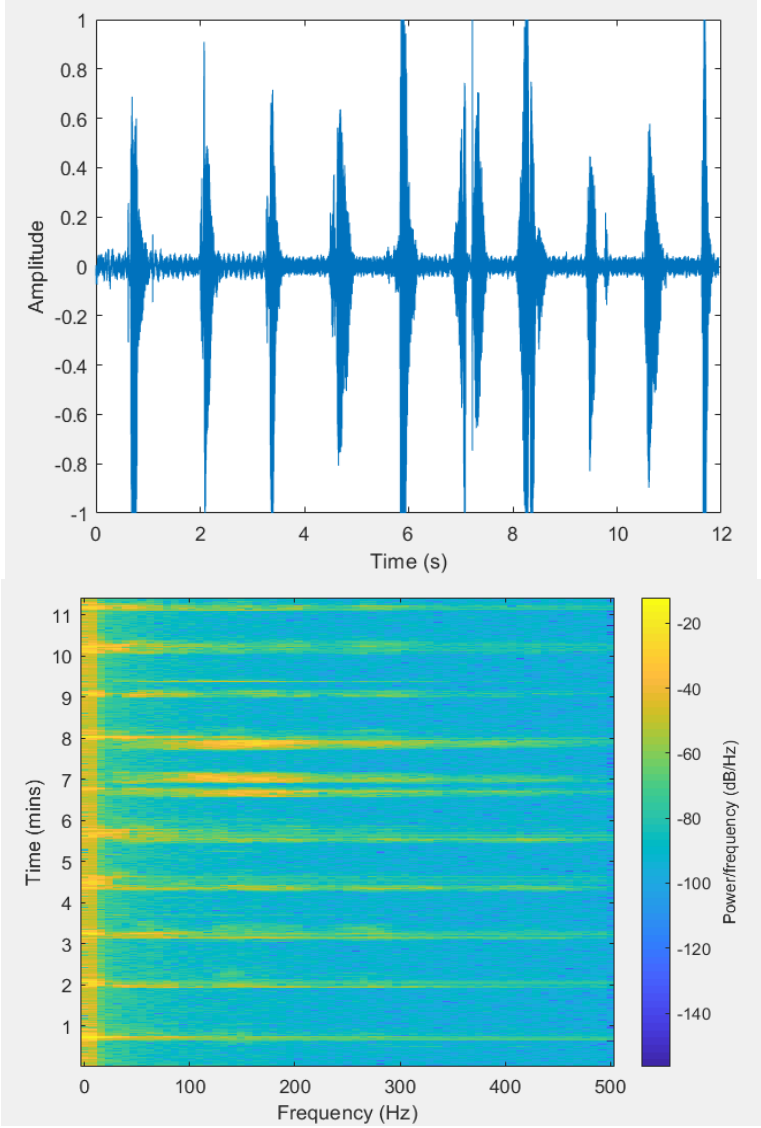
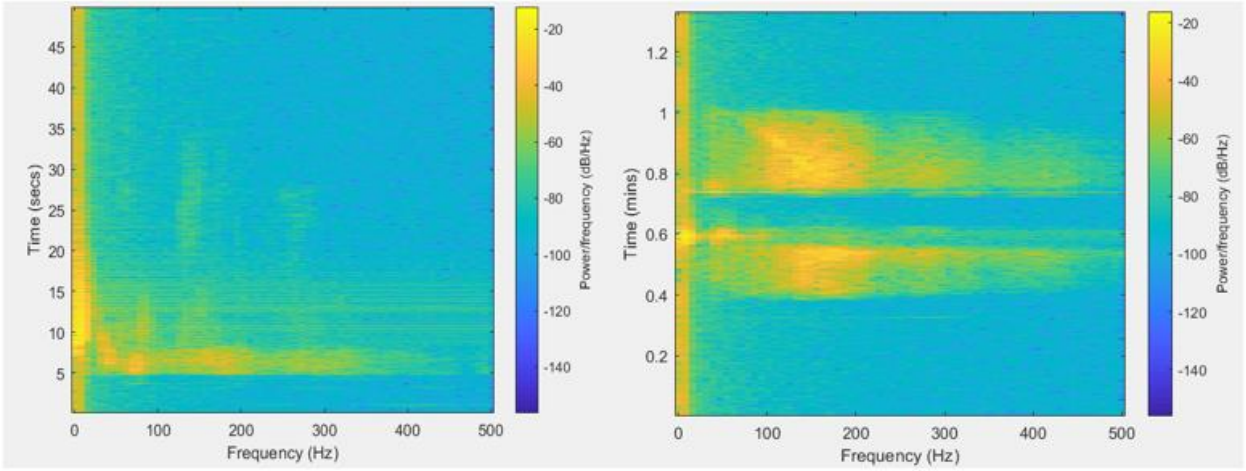
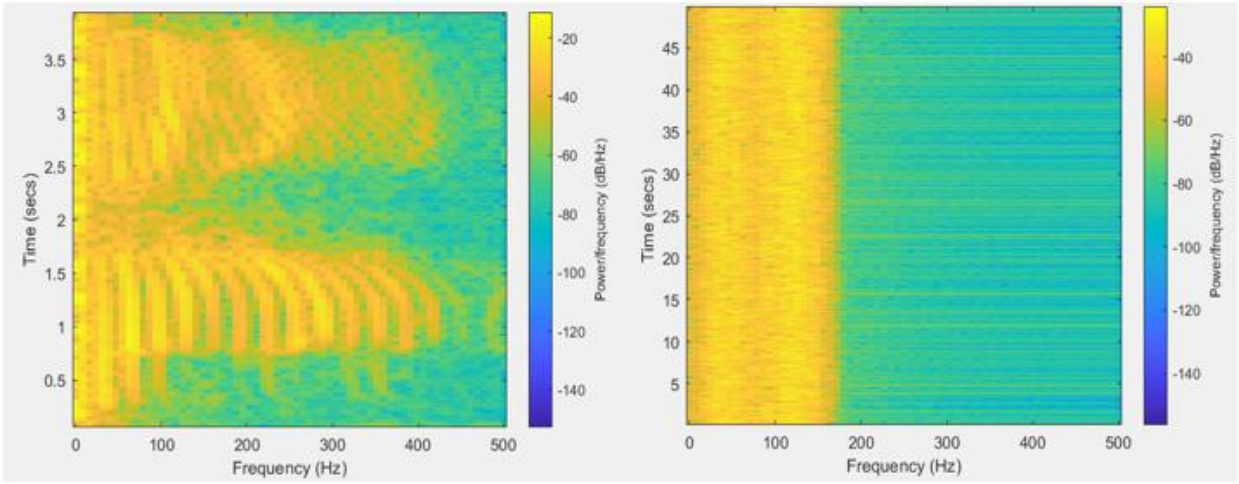


Figure 5.6: Time-domain and spectrogram of a sound signal

Figure 5.7 shows spectrogram of various sounds. As it can be seen, different sound signals have distinct spectrograms which makes it easy for the classifier to learn the differences and differentiate the corresponding signals.



(a) : Spectrogram of “two” (b) : Spectrogram of”six”



(c) : Spectrogram of ”Matlab” (d) : Spectrogram of “noise”

Figure 5.7: Spectrogram of different sounds

5.2 Preprocessing

After data collection, preprocessing step should be performed to make the data suitable for machine learning purposes. The most common preprocessing technique is data normalization which makes the data zero-mean and unit-variance. This can be simply achieved by subtracting the mean from the data and dividing the result by standard deviation.

$$x' = \frac{x - \bar{x}}{\sigma} \tag{5-1}$$

Where x is the raw data, $\bar{x}$ is the mean of the data values, and σ is the standard deviation.

Normalization step should be performed on each feature individually because the pitch and MFCCs are not on the same scale (Figure 5.8). It is worth mentioning that in this project, other preprocessing tasks, such as noise removal, sound smoothing, and echo cancelation are not performed because experimental results show that the machine learning methods can learn these nonideal situations.

Pitch	MFCC1	MFCC2	MFCC3	MFCC4	MFCC5	MFCC6	MFCC7	MFCC8	MFCC9	MFCC10	MFCC11	MFCC12	MFCC13
0.90535	-1.8778	0.11469	0.25866	-0.41449	0.97803	-0.34062	-0.22379	-0.031962	0.62995	0.81708	-0.29036	-0.47	-0.04532
0.98367	-1.8143	1.2196	0.32634	-0.65383	0.9623	-0.52854	-0.61983	0.70533	0.40654	-0.066892	-0.60354	-0.53907	-0.51924
0.81806	-1.4342	1.3996	0.27266	-0.77005	1.0279	-0.61349	-0.69793	0.72491	0.56842	-0.6335	-0.60484	-0.62735	-1.0637
0.8025	-1.1209	1.4881	0.0057061	-0.78058	1.2356	-0.7397	-0.49906	1.1048	0.39759	-0.57164	-0.36655	-0.67672	-0.87127
-1.0579	-1.0516	1.6394	0.026102	-0.69355	1.116	-0.85012	-0.58429	0.86925	0.83105	-0.66218	-0.84113	-0.66903	-0.60151
-1.0674	-0.85121	1.5999	0.074074	-0.79501	1.0355	-0.56555	-0.58294	0.5838	0.84584	-0.57511	-1.1255	-0.64767	-0.8494
-1.0797	-0.8476	1.4231	0.21274	-0.92891	0.82602	-0.34877	-0.20402	0.41231	0.67644	-0.56908	-1.3199	-0.43764	-0.38468
-1.0955	-0.75325	1.7918	0.174	-0.88856	0.65463	-0.12772	0.021446	0.56706	0.48842	-0.60995	-1.8518	-0.50805	-0.26085

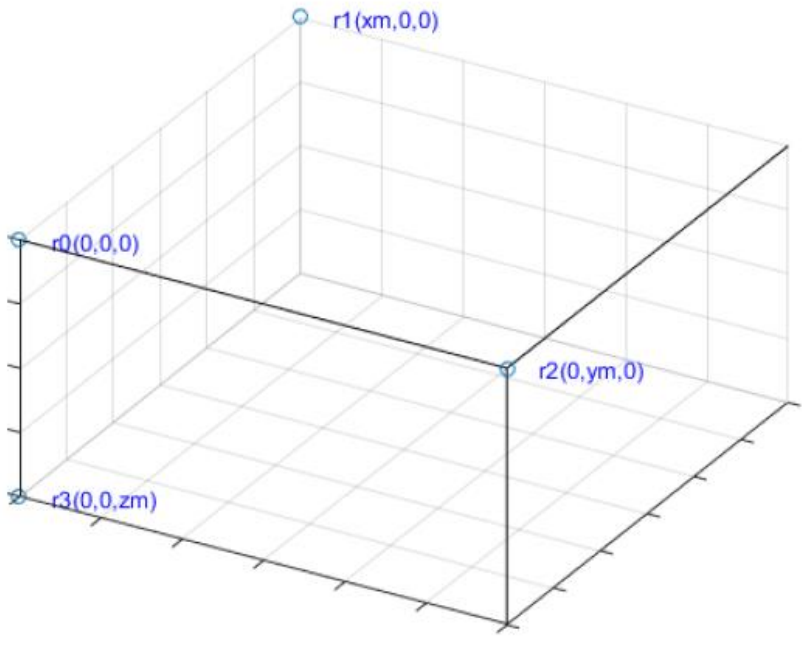
Figure 5.8: Normalizing One-dimensional features values

It should also be noted that the length of the received audio signal varies according to the distance between the sound signal and the hydrophone. The more the distance, the wider the received signal. This is mainly due modal dispersion and multipath propagation which are intensified by distance. Since the number of MFCCs depends on the signal length, we select the width of windows and overlapping period between consequent windows adaptively to extract a fix number of FCCs. Thirteen MFCCs were extracted in this project because experimental results show that 10 to 20 coefficients can accurately describe the signals. A fixed number of features is

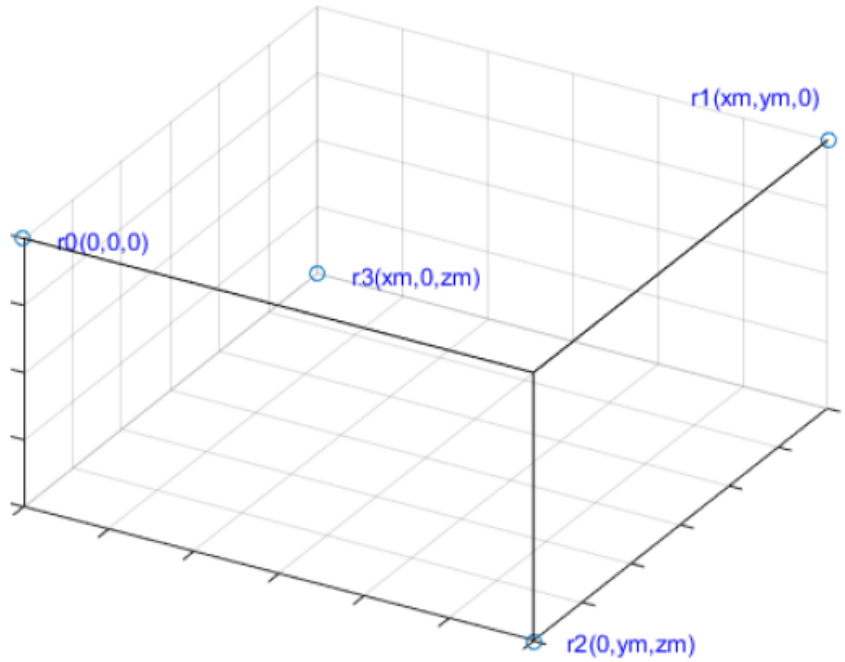
needed when utilizing those machine learning techniques, such as neural networks, with fix parameters, like number of neurons in input layer.

5.3 Hydrophone Layouts

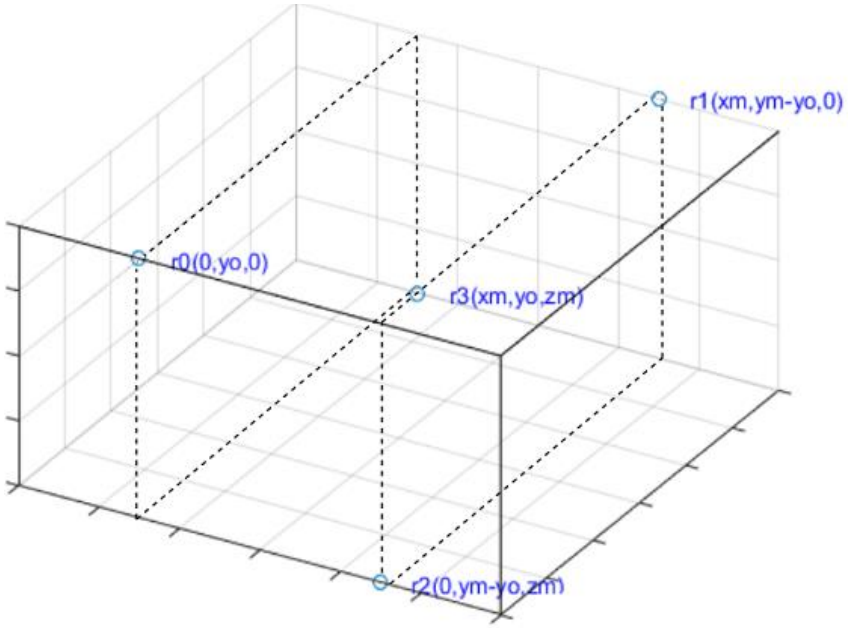
Performance of machine learning methods strongly depend on the quality and quantity of the training data set. Data collection, however, is a challenging and time-consuming task. It involves several steps including collecting, preprocessing, analyzing, and visualizing. After performing all these steps, feature extraction can be performed. To the best of our knowledge, there is no public data set available for underwater sound localization in a confined space. So, in this project we generated the training set manually. For this purpose, the hydrophones were installed at different locations within the water tank and the beacon mounted at the top of an ROV which was moved manually. Figure 5.9 shows three different hydrophone layouts.



(a)



(b)



(c)

Figure 5.9: Different hydrophone layouts

In *layout A*, three hydrophones were installed on the surface of the tank and only one hydrophone was at the bottom. Experimental results show that this layout does not have a good coverage. When the beacon is on the surface, far away from the hydrophones, localization error increases. In *layout B*, two hydrophones were installed on the tank surface while two other hydrophones were installed on the floor. To have a wide coverage, hydrophones were located at the opposite corners at each plane. However, *layout* B is not able to capture sound reverberations from test tank walls. To address this issue, in *layout C* the hydrophones were located far from the frontal and dorsal walls. It should be noted that other hydrophone layouts can be proposed for sound localization, which may require more training data to produce reliable results.

5.3.1 Uniform Sampling

In the simplest form, the beacon can be located at different positions, equally spaced in the test tank. Number of collected data depends on the spatial resolution. Since the rectangular water tank has dimensions of 6.0 m x 4.5 m x 1.7 m, uniform sampling with 10cm resolution results to 45,900 samples. Manually collecting such immense data set is a tedious and time-consuming task. In response to this issue, we reduced the spatial sampling resolution to 50cm and collected samples at 342 positions uniformly. Taking the length of ROV into consideration (approximately 60cm), 50cm resolution can be acceptable in noncritical applications. Figure 5.10 shows the uniform sampling process.

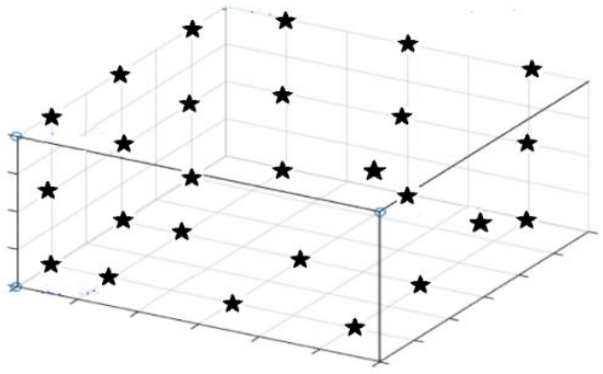


Figure 5.10: uniform sampling process

5.3.2 Nonuniform Sampling

Uniform sampling requires significant amounts of time to produce high quality data resulting adequate localization accuracy. To reduce the total acquisition time, nonuniform sampling method can be deployed in which a smaller number of samples are collected in the areas with unambiguous interpretation to speed up the experiment. In other words, while the number of acquired samples is reduced in the central area of the tank to minimize the collection time, the sampling spatial resolution is elevated at critical areas such as tank corners to maintain localization accuracy. Figure 5.11 shows nonuniform data acquisition. In this experiment, 100 samples were collected nonuniformly. Most of the data were gathered from the tank floor, ceiling, and corners which highly experience multi-path and reverberation phenomena. After positioning the beacon at the predefined location within the tank, the received signals by the hydrophones (H_1,H_2,H_3,H_4) and (x,y,z) of that point are recorded as the data and its corresponding label. This location is called *ground truth* and it will be used later to train machine learning methods.

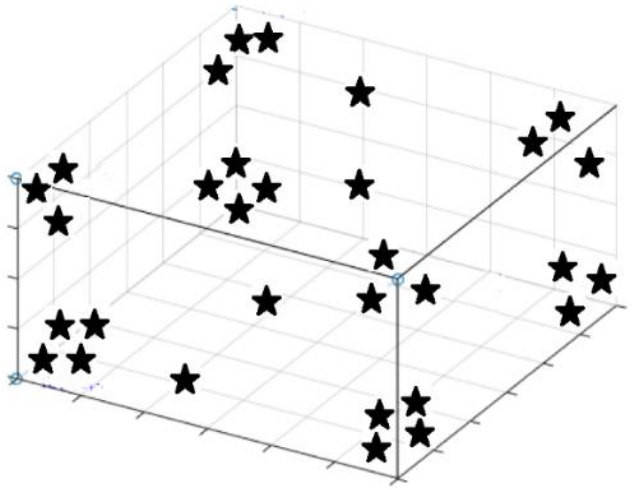


Figure 5.11: Nonuniform data acquisition process

5.4 Classification Results

As stated in Chapter 3, based on the type of estimated range, source localization can be regarded as classification or regression. While output of the classification module is discrete, the

regression module represents the estimated range as a continuous value. This section illustrates the results of MLP, CNN, SVM, and RF machine learning techniques in the classification domain.

- Range resolution

In the classification regime, unlike regression approach, the position of the sound signal is quantized. The number of classes corresponds to the number of these quantized values which are determined by the resolution of range steps. As stated in Section 5.2.1, to alleviate data collection burden, the maximum number of classes was set to 342 in uniform sampling method, corresponding to 50cm range resolution. The number of classes determines the accuracy of the localization algorithm and affects the time of the machine learning algorithms. In case of non-uniform sampling, Section 5.2.2, the maximum number of classes is limited to 100.

- Snapshots

When the beacon is positioned at predetermined locations, sound recording process should be repeated for several times. The number of repeating this experiment is called *snapshots*. Since the beacon is mounted at the top of the ROV and sound of the ROV motors are stochastic, the propagated sound of the beacon is contaminated by different noises. Moreover, in a confined space, not only the location of the beacon but also its direction affects the received sound of the hydrophones. Assume the beacon is located near to the left wall of the water tank. If the beacon sends sound signals to the left, the hydrophones on the right receive signals with high reverberation caused by the left side wall reflection. But if the beacon is aligned to the right, the hydrophones receive signal with less reverberation. Therefore, data collection should be repeated several times at each position. Increasing the number of snapshots can make the localization system more robust to noise and nonideal phenomena, such as reverberation.

- Dataset

Each hydrophone receives a sound signal with the length of 40ms to 100ms, according to its distance from the beacon and acoustic condition. These samples are divided into frames of 20ms with an overlap of 50%. For the sake of simplicity, assume the average of received signals is 60ms. Therefore, five frames are extracted for each signal. Having four hydrophones, locating the beacon in 342 locations in uniform sampling scenario, and propagating the sound signal in five different directions (snapshot), we will have 34,200 ($5\times4\times342\times5=34200$) frames (training

samples). Then, one-dimensional features, including pitch and MFCCs, or two-dimensional feature, spectrogram, are extracted from these frames and concatenated to build the dataset. In other words, in the case of uniform sampling and one-dimensional features, the dataset has 34,200 rows and 14 columns (pitch and 13 MFCCs). When non-uniform sampling is utilized, the dataset has 10,000 rows (5×4×100×5=10000) and 14 columns. In case of two-dimensional feature, we will have 34,200 and 10,000 spectrogram images for uniform and non-uniform sampling scenarios, respectively.

5.4.1 Feed Forward NNs (FNN)

As stated in Section 5.1, pitch and MFCC coefficients were extracted from the received sound signals. Since the feature vector consists of one pitch parameter and 13 MFCCs, the neural network with 14 neurons in the input layer was deployed. In the FNN, the number of neurons at the output layer is three, corresponding to (x,y,z) of the estimated location. Performance of the proposed localization system is evaluated when number of hidden layers and number of neurons in each layer vary. Neural networks with one and two hidden layers are utilized because increasing the number of hidden layers exponentially increase the number of links in the network, which requires more training data. We also investigate the number of neurons in each hidden layer. Usually, the number of hidden neurons is power of two, because they are more effective than other numbers. This is due to fact that training neural networks is based on matrix multiplication, which is performed in batch sizes that are power of 2. Therefore, selecting number of hidden neurons as power of two can elevate the network training speed. In this research the number of hidden neurons was increased from 32 to 2048.

We used *MATLAB 2018 Neural Network Toolbox* for training the FNN. The learning algorithm is gradient descent and learning rate is set to 0.01. According to Figure 5.12, the best validation performance occurs at epoch 70 which corresponds to 9% MAPE (see section 3.3.3 for MAPE definition).

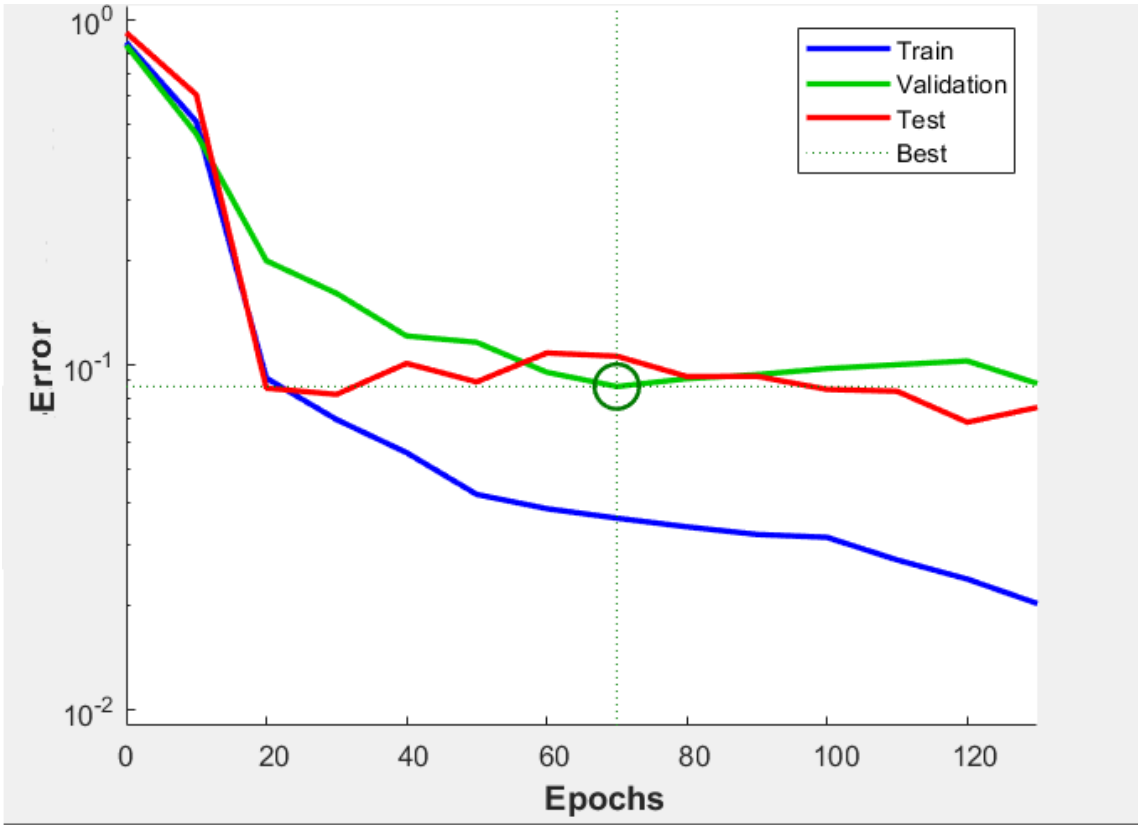


Figure 5.12: Localization error on train, validation and test data sets

Table 5.1 shows the effect of different parameters on the FNN result. Although increasing the number of hidden neurons enables the network to capture more of the variance in the data, it raises the computational burden. Experimental results show that accuracy of the localization system does not improve by increasing the number of hidden layers in FNN. So, FNNs with one and two hidden layers were investigated in this research. As shown in Table 5.1, non-uniform sampling for training data collection results to more accurate localization. Minimum of MAPE of 7% (non-uniform) and 10% (uniform) is obtained when FNN has one hidden layer with 1024 neurons. Shortage of training data may account for performance decline when number of hidden layers or hidden neurons are increased.

In next step, we explore the effect of spatial resolution, number of classes, on the localization accuracy. Moreover, performance of the localization system is investigated when number of snapshots vary. For the sake of simplicity, number of hidden layers and number of hidden neurons are set to the values which yield the minimum MAPE in the previous experiment (Table 5.1). As Table 5.2 shows, by increasing the number of classes, the localization error decreases.

Performance of uniform sampling is always inferior to non-uniform sampling method. With 300 classes, uniform sampling result is still less than non-uniform outcome with 100 classes. Performance of the proposed method is tested using 1, 5, and 10 snapshots. According to Table 5.2, increasing the number of snapshots makes the localization system more robust to the input data variation. However, experimental results show that using 5 snapshots makes the system robust enough and 10 snapshots does not considerably affect the MAPE of the localization system. It should be noted that maximum number of classes for non-uniform sampling is limited to 100. So, in Table 5.2 for class numbers of 200 and 300, MAPE cannot be estimated (NA). The minimum MAPE was achieved with 5 snapshots with 100 classes using non-uniform sampling.

Table 5.1: Sensitivity of FNN classifier to number of hidden layer and hidden neuron

				MAPE	
No. of hidden layers	No. of hidden neurons	No. of classes	No. of snapshots	Uniform sampling (%)	Non-uniform sampling (%)
1	64	100	5	14	10
1	128	100	5	15	10
1	256	100	5	14	9
1	512	100	5	13	9
1	1024	100	5	**10**	**7**
1	2048	100	5	11	8
2	64	100	5	11	9
2	128	100	5	12	10
2	256	100	5	12	10
2	512	100	5	13	11
2	1024	100	5	12	9
2	2048	100	5	12	10

Table 5.2: Sensitivity of FNN classifier to number of classes and snapshots

				MAPE	
No. of hidden layers	No. of hidden neurons	No. of classes	No. of snapshots	Uniform sampling (%)	Non-uniform sampling (%)
1	1024	10	1	19	16
1	1024	10	5	16	13
1	1024	10	10	17	13
1	1024	20	1	16	12
1	1024	20	5	13	10
1	1024	20	10	13	10
1	1024	50	1	12	10
1	1024	50	5	11	9
1	1024	50	10	10	8
1	1024	100	1	12	9
1	1024	100	5	10	**7**
1	1024	100	10	10	7
1	1024	200	1	9	NA
1	1024	200	5	**8**	NA
1	1024	200	10	8	NA
1	1024	300	1	13	NA
1	1024	300	5	11	NA
1	1024	300	10	12	NA

5.4.2 Support Vector Machine (SVM)

Gaussian radial basis function (RBF) kernel is used in the SVM classifier. This classifier has two hyperparameters, C and γ [Eq. 3-3] which were tested over $[10^{-3} \quad 10^{-1}]$ and $[10^{1} \quad 10^{3}]$, respectively. Experimental results show that $C=10^{-2}$ and $Gama=10^{1}$ result to the minimum error. Since localization problem is a multiclass problem and SVM is a binary classifier, we used *ClassificationECOC* classifier in MATLAB. This classifier performs a multiclass learning task by reducing it to multiple binary SVM classifiers. It uses the one-vs-one approach which splits the dataset into one dataset for each class versus every other class. For K classes, the one-vs-one

method requires K(K − 1)/2 binary classifiers. Number of localization classes is increased from 10 to 300. We also increase the number of snapshots from 1 to 10. The results are in consensus with previous experiment which reports that non-uniform sampling is always superior to uniform sampling in terms of localization accuracy. Increasing the number of classes reduces the MAPE error, except for 300 class case. Table 5.3 presents the MAPE of SVM classifier. As it can be seen, the minimum MAPE of SVM is 5% using non-uniform sampling with 5 snapshots and 100 classes.

Table 5.3: Sensitivity of SVM classifier to number of classes and snapshots

		MAPE	
No. of classes	No. of snapshots	Uniform sampling (%)	Non-uniform sampling (%)
10	1	18	17
10	5	15	15
10	10	14	15
20	1	15	14
20	5	13	11
20	10	13	12
50	1	14	9
50	5	10	6
50	10	11	7
100	1	12	8
100	5	9	**5**
100	10	9	5
200	1	9	NA
200	5	**7**	NA
200	10	8	NA
300	1	11	NA
300	5	8	NA
300	10	7	NA

5.4.3 Random Forest (RF)

Random forest is an ensemble classifier, consisting of a several decision trees. Each individual tree is trained on a randomly subset of the training set. After each tree makes its own prediction, the final prediction is made by majority voting method. In other words, the class which was selected by most of the tress is considered as the final decision of the random forest. This method of classifier combination is called bagging. In comparison with FNN, RF has less hyperparameter to choose. The minimum samples per leaf and the number of decision trees are the most important RF hyperparameters. In this project, minimum samples per leaf were increased linearly from 1 to 100. The number of decision trees was doubled from 10 to 1000. Experimental results show that when 500 trees were bagged to create the RF with a minimum of 50 samples per leaf results to minimum MAPE. The lowest MAPE achieved by the RF is 9% using non-uniform sampling. RF also reaches 12% MAPE for uniform sampling.

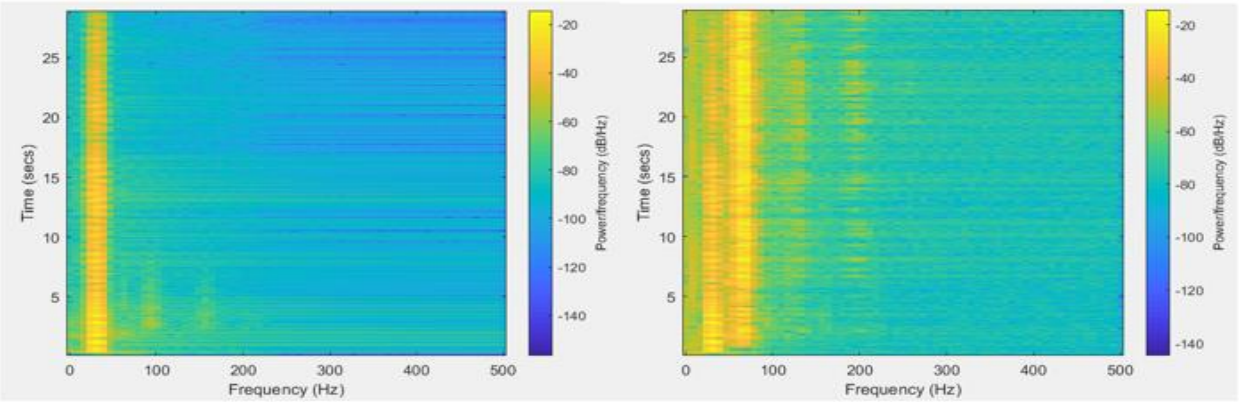
Table 5.4: Sensitivity of RF classifier to number of classes and snapshots

		MAPE	
No. of classes	No. of snapshots	Uniform sampling (%)	Non-uniform sampling (%)
10	1	24	15
10	5	23	13
10	10	23	13
20	1	25	14
20	5	22	11
20	10	23	12
50	1	22	12
50	5	20	10
50	10	20	10
100	1	21	12
100	5	18	**9**
100	10	19	8
200	1	17	NA
200	5	14	NA
200	10	14	NA
300	1	16	NA
300	5	**12**	NA
300	10	13	NA

5.4.4 Convolutional Neural Network (CNN)

Two types of one-dimensional features (pitch and MFCCs), were extracted from the received sound signal and they were fed into different classifiers to localize an underwater sound source. Promising results of CNNs visual recognition on computer vision applications has motivated researchers to leverage CNN for sound processing applications. Since sound signal is one-dimensional and CNNs require images which are two-dimensional, spectrogram of the signal is usually utilized.

Figure 5.13 (a) shows spectrogram of the beacon sound when it is far away from tank walls and when ROV motor is off. In this case, there is no reverberation, and the signal was not contaminated by the ROV noise sound. Figure 5.13 (b) shows spectrogram of the beacon sound when ROV motor is on.

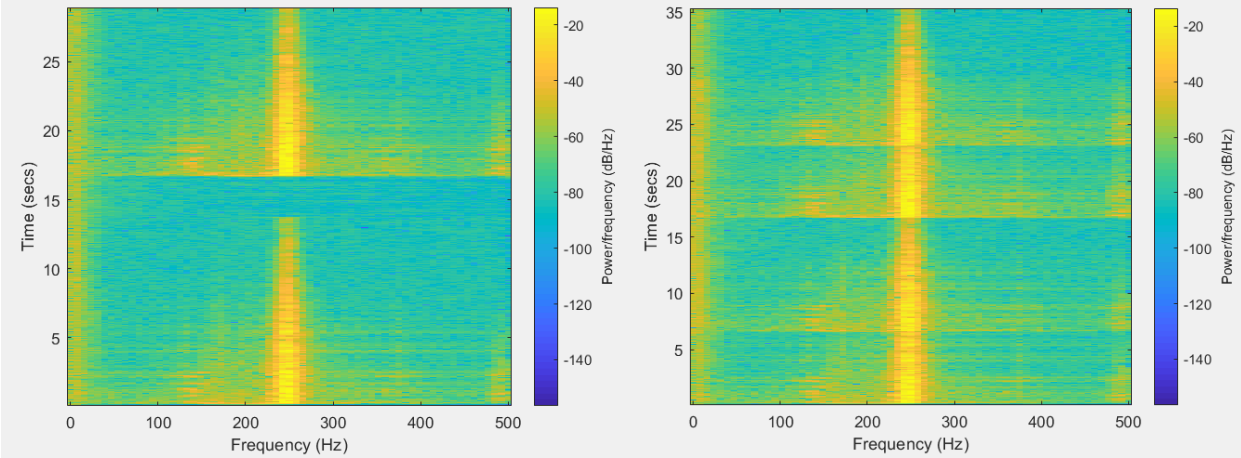


(a) (b)

Figure 5.13: The effect of motor noise on spectrogram of the received signal

(a): ROV motors are off. (b): ROV motors are on.

To see the effect of reverberation on the spectrogram, the beacon is located near to the tank wall in a perpendicular direction. Then, the beacon sends two sound signals (Figure 5.14 (a)). Because of the reverberation produced by the tank wall, four acoustic signals were received by the hydrophone (Figure 5.14 (b)).



(a) (b)

Figure 5.14: The effect of reverberation on spectrogram of the received signal

(a): Low reverberation. (b): High reverberation.

As mentioned earlier in Section 5.3, at each location several samples were collected to include maximum variations in the samples. In this project maximum of 12 snapshots were used. In other words, at each position the beacon direction was changed to East, West, South, North, Up, and Down. At each direction two samples were recorded. Having 4 hydrophones and 12 snapshots, 48 samples can be collected at each location. Considering non-uniform sampling method with 100 classes, data set with the size of 4800 samples is available for training and evaluation of the CNN. The amplitudes of the collected spectrograms were scaled to range of [0,1] and spectrogram images are resized to 400x144.

Figure 5.15 shows the CNN architecture used in this research. It includes five convolution layers. The convolution layers are followed by a batch normalization layer, a rectified linear unit (ReLU) activation layer, and a max pooling layer. The output layer is a fully connected layer with 3 neurons, followed by softmax activation needed for classification. Figure 5.16 shows the learning parameters.

```
1   ''   Image Input             400x144x1 images
2   ''   Convolution             16 10x10 convolutions with stride [1  1] and padding 'same'
3   ''   Batch Normalization     Batch normalization
4   ''   ReLU                    ReLU
5   ''   Max Pooling             10x10 max pooling with stride [2  2] and padding [0  0  0  0]
6   ''   Convolution             32 5x5 convolutions with stride [1  1] and padding 'same'
7   ''   Batch Normalization     Batch normalization
8   ''   ReLU                    ReLU
9   ''   Max Pooling             10x10 max pooling with stride [2  2] and padding [0  0  0  0]
10  ''   Convolution             32 5x5 convolutions with stride [1  1] and padding 'same'
11  ''   Batch Normalization     Batch normalization
12  ''   ReLU                    ReLU
13  ''   Max Pooling             10x10 max pooling with stride [2  2] and padding [0  0  0  0]
14  ''   Convolution             32 5x5 convolutions with stride [1  1] and padding 'same'
15  ''   Batch Normalization     Batch normalization
16  ''   ReLU                    ReLU
17  ''   Max Pooling             5x5 max pooling with stride [2  2] and padding [0  0  0  0]
18  ''   Convolution             32 5x5 convolutions with stride [1  1] and padding 'same'
19  ''   Batch Normalization     Batch normalization
20  ''   ReLU                    ReLU
21  ''   Average Pooling         2x2 average pooling with stride [2  2] and padding [0  0  0  0]
22  ''   Fully Connected         3 fully connected layer
23  ''   Softmax                 softmax
24  ''   Classification Output   crossentropyex
```

Figure 5.15: The CNN architecture

```
options = trainingOptions('adam', ...
    'ExecutionEnvironment','gpu',...
    'MiniBatchSize',128, ...
    'MaxEpochs',30, ...
    'InitialLearnRate',1e-2, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropFactor',0.1, ...
    'LearnRateDropPeriod',10, ...
    'Shuffle','every-epoch', ...
    'Verbose',false, ...
    'Plots','training-progress');
```

Figure 5.16: The CNN parameters in learning phase

After defining the network architecture and learning parameters, the CNN was trained using *trainNetwork* in MATLAB. Performance of the localization system is shown in Figure 5.17. The network reached a 4% MAPE on test set after 30 epochs.

Figure 5.17: Performance of the CNN classifier

In the following the effect of number of convolutional layers and filter size on the accuracy of the proposed CNN-based localization system will be investigated. For the sake of simplicity, number of classes is set to 100 and 12 snapshots were utilized in all experiments. The consensus about deep neural network architecture is that by increasing the depth of convolutional layers and reducing spatial size of feature maps, better results can be obtained [35]. Following this rule of thumb, several CNN architectures were explored in Table 5.5. As it can be seen, by increasing the number of convolutional layers the localization error reduces. However, MAPE suddenly rises when 6 convolutional layers are deployed. This can be contributed to the shortage of training spectrogram images.

Table 5.5: Parameter sensitivity of CNN classifier

No. of conv. layers	No. of hidden neurons	Size of filter	MAPE (%)
1	8	5	14
1	8	10	13
1	16	5	12
1	16	10	14
1	32	5	15
1	32	10	15
2	16-16	5-5	12
2	16-32	5-10	13
3	16-32-32	10-5-5	9
4	16-32-32-32	10-5-5-5	7
5	16-32-32-32-32	10-5-5-5-5	**5**
6	16-32-32-32-32-32	10-5-5-5-5-5	8

5.5 Comparative Study

In this chapter, performance of four different machine learning algorithms was evaluated for underwater sound source localization application. Each classifier was examined with various parameters including number of classes, number of snapshots, and model hyperparameters. In this section, only the best performances of these machine learning techniques will be compared. All experiments have been conducted on MATLAB 2018b installed on a laptop with Windows 10 operating system, 12 GB memory, and Intel Core i5 @ 1GHz CPU. Table 5.6 compares performance of RF, SVM, FNN, and CNN in terms of training time and MAPE. As it can be seen, RF is the fastest classifier, but its performance is the worst. CNN achieves the lowest MAPE with 4%, however, its training time is considerably longer than other machine learning techniques. The lengthy training process hinders examining various network architecture and a wide range of hyperparameters to achieve the minimum localization error. One possible solution would be to utilize GPU (Graphics Processing Unit) rather than CPU (Central Processing Unit).

Performing matrix operations in parallel makes GPUs more effective for image and video processing applications.

Table 5.6: Comparison between FNN, SVM, RF, and CNN classifiers

Classifier	Training Time	MAPE (%)
Random Forest	**65** s	9
Support Vector Machine	120 s	5
Feedforward Neural Network	150 s	7
Convolutional Neural Network	35 m	**4**

CHAPTER 6

Conclusion

6.1 Summary

The main goal of this thesis was to develop a sound source localization system for underwater vehicles positioning inside a confined test tank. The proposed system was based on underwater acoustic and machine learning techniques. A beacon, installed on a ROV, sends acoustic signals in the water which were received by 4 hydrophones, located at different positions inside of the tank. Acoustic signals were generated by Xilinx LX45 FPGA board. The collected signals by the hydrophones (Teledyne TC4013) were amplified and filtered by an analog front-end device (RedPitay board). The preprocessed analog signals were converted to digital format by the NI 9234 data acquisition module and NI-DAQmx driver software. We used MATLAB 2018 to process the digital acoustic signals in a computer and utilized them for machine training. One of the main challenges in harnessing the power of machine learning techniques is training data collection, which was performed manually in this project. After locating the ROV in each predefined position, (x,y,z) of that point as well as received signals by the hydrophones were recorded in the dataset. Performance of four common machine learning methods were evaluated for source localization. While Random Forest (RF) has the lowest training time (65s), Convolutional Neural Network (CNN) reached the minimum localization error (MAPE 4%). Support Vector Machine (SVM) and Feedforward Neural Network (FNN) were moderate both in terms of time and accuracy.

6.2 Contributions

The novelties of this thesis are as follows:

- Utilizing machine learning techniques. We leveraged the generalization power of machine learning techniques to improve localization accuracy. By collecting a limited number of data and without knowing the exact characteristics of underwater environment, neural networks, support vector machine, and random forest techniques were used for underwater localization.

- Classification instead of regression. Unlike previous methods which attempt to solve source localization problem by regression approaches, we consider source localization from classification point of view. To this end, the whole space of the confined tank is regarded as a set of finite positions. To estimate the location of the sound source, the output of the machine learning methods is matched to the nearest predefined position.
- Utilizing deep neural networks. Convolutional neural network (CNN), which has shown promising results in various applications, was modified, and adopted for sound localization application. For this purpose, one-dimensional sound signals were converted to two-dimensional images, using time-frequency technique. Then, the spectrogram images were considered as the inputs to train the CNN for localization task.

6.3 Conclusion

This thesis investigates the potential of machine learning approaches for underwater source localization in a confined space. The experimental results show that underwater vehicles localization problem can be solved by supervised learning approaches with high accuracy. These promising results can pave the way for utilizing machine learning techniques for source localization in rivers and oceans. However, large scale dataset preparation remains as a challenge for practical applications in open spaces. We found that spectrogram images and CNN can accurately approximate the location of an underwater sound source (MAPE 4%). To fully take advantage of CNNs potential, however, novel learning techniques should be utilized. For instance, fine-tunning methods can enhance CNN accuracy and deploying GPUs (Graphical Processing Unit) can diminish its training time.

6.4 Future Work

The following functionalities can be added to the developed system in the future:

- Performing localization task in online scenario. The current study addressed localization problem in offline framework. However, in real-world applications, localization should be performed online which requires more emphasis on acoustic sound generation timing and hydrophones synchronization.

- Introducing more effective methods for training sample collection. Working in a confined underwater space such as a test tank, enabled us to collect data manually which was still tedious and time-consuming. Manual data gathering is impractical in underwater open spaces such as oceans. Semi-automatic data collection methods based on combination of underwater sound propagation models and real data will lighten training set preparation burden.
- Introducing novel two-dimensional (2D) features. In this thesis, we used spectrogram images as 2D features and fed them to CNN to localize the sound source. During converting 1D sound signal to 2D images some information may not be well presented. Combinational techniques for 1D and 2D feature fusion can boost performance of the localization system.
- Adopting recurrent deep neural networks. CNN was utilized in this research which is not completely matched with 1D sound signals. The use of Recurrent Neural Networks (RNN), such as long short-term memory (LSTM), can be explored in the future study.
- Open space underwater localization. UV localization in confined spaces is completely different from open space aquatic environments positioning. Due to lack of external infrastructure, low illumination levels, high turbidity, and a lack of salient features in confined spaces, traditional localization methods cannot be utilized for such environments. Localization in confined space is challenging and has various applications in liquid storage tanks and sewers, sunken ships, underwater caves, and flooded tunnels. However, it is believed that applications of confined space exploration are limited, in comparison with open space environments. Combining localization methods in open space and confined aquatic environments is a great research topic. This thesis paved the way for utilizing machine learning methods for confined spaces which can be modified and combined with traditional localization methods for open environments. One possible solution is to collect a limited number of real data and leverage generalization power of machine learning techniques to make existing sound propagation models more accurate.

REFERENCES/BIBLIOGRAPHY

[1]- Dhanak, Manhar R., Xiros, Nikolas I, “Handbook of Ocean Engineering”, Springer, 2016.

[2]- G Griffiths,” Technology and applications of autonomous underwater vehicles”, CRC Press, 2002.

[3]- Jiajun Shen, Xueli Fan and Qixin Wang,” WiP Abstract: Underwater AUV Localization with Refraction Consideration”, ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), pp. 333-334, 2018.

[4]- David C. Baumann, Jacob M. Brendly, Drake B. Lafleur, Patrick L. Kelley, Robert L. Hildebrand, Edoardo I. Sarda, “Techniques for Scaled Underwater Reverberation Measurements”, OCEANS MTS/IEEE SEATTLE, 2019.

[5]- Meisam Naderi, Matthias Pätzold, Rym Hicheri, Néji Youssef,” A Geometry-Based Underwater Acoustic Channel Model Allowing for Sloped Ocean Bottom Conditions”, IEEE Transactions on Wireless Communications, vol. 16, no. 4, pp. 2394-2408, 2017.

[6]- B. V. Menna, G. G. Acosta and S. A. Villar, "Underwater acoustic channel model for shallow waters," 3rd IEEE/OES South American International Symposium on Oceanic Engineering (SAISOE), pp. 1-7,2016.

[7]- Riwal Lefort, Gaultier Real, Angélique Drémeau,” Direct regressions for underwater acoustic source localization in fluctuating oceans”, Applied Acoustics, 116, pp. 303–310, 2017.

[8]- Zhaoqiong Huang, Ji Xu, Zaixiao Gong, Haibin Wang, and Yonghong Yan,” Source localization using deep neural networks in a shallow water environment”, The Journal of the Acoustical Society of America 143, 2922, pp. 2922–2932, 2018.

[9]- Yun Wang, and Hua Peng,” Underwater acoustic source localization using generalized regression neural network”, The Journal of the Acoustical Society of America 143, 2321, pp. 2321–2331, 2018.

[10]- Michael J. Bianco, Peter Gerstoft, James Traer, Emma Ozanich, Marie A. Roch, Sharon Gannot, and Charles-Alban Deledalle,” Machine learning in acoustics: Theory and applications”, The Journal of the Acoustical Society of America 146, 3590, pp. 3590–3628, 2019.

[11]- Wei Ma and Xun Liu,” Phased microphone array for sound source localization with deep learning”, Aerospace Systems, volume 2, pp. 71–81, 2019.

[12]- Asgeir J. Sørensen, Martin Ludvigsen, Petter Norgren, Øyvind Ødegård, Finlo Cottier,” Sensor-Carrying Platforms”, POLAR NIGHT Marine Ecology, pp 241-275, 2020.

[13]- Josué González-García, Alfonso Gómez-Espinosa, Enrique Cuan-Urquizo, Luis Govinda García-Valdovinos, Tomás Salgado-Jiménez and Jesús Arturo Escobedo Cabello, “Autonomous Underwater Vehicles: Localization, Navigation, and Communication for Collaborative Missions”, Appl. Sci. 10, 1256, 2020.

[14]- K. G. Kebkal and A. I. Mashoshin, ”AUV acoustic positioning methods”, Gyroscopy and Navigation volume 8, pp. 80–89, 2017.

[15]- Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li,” AUV Navigation and Localization: A Review”, IEEE JOURNAL OF OCEANIC ENGINEERING, VOL. 39, NO. 1, pp. 131-149, 2014.

[16]- M. W. Khan, Y. Zhou, and G. Xu,” Modeling of acoustic propagation channel in underwater wireless sensor networks”, International Conference on Systems and Informatics (ICSAI 2014), pp. 586–590, 2014.

[17]- A.-M. Ahmad, J. Kassem, M. Barbeau, E. Kranakis, S. Porretta and J. Garcia-Alfaro,” Doppler Effect in the Acoustic Ultra Low Frequency Band for Wireless Underwater Networks”, Mobile Networks and Applications, volume 23, pp. 1282–1292, 2018.

[18]- Afonso Mateus Bonito,” Acoustic system for ground truth underwater positioning in DEEC’s test tank”, University Do Porto, Master Thesis, 2019.

[19]- João Miguel Fernandes Magalhães,” Improving Time of Arrival Estimation Using Compensated Acoustic Signals”, University Do Porto, Master Thesis, 2018.

[20]- Peng Wu, Shaojing Su, Zhen Zuo, Xiaojun Guo, Bei Sun, and Xudong Wen,” Time Difference of Arrival (TDoA) Localization Combining Weighted Least Squares and Firefly Algorithm”, Sensors (Basel). 19(11): 2554, 2019.

[21]- Matsna Nuraini Rahman, M. T. Ir. Ahmad Tri Hanuranto, S. T. M. T. Ratna Mayasari,” Trilateration and iterative multilateration algorithm for localization schemes on Wireless Sensor Network”, International Conference on Control, Electronics, Renewable Energy and Communications (ICCREC), pp. 88-92, 2017.

[22]- C. Aubry, P. Forjonel, P. Bouvet, A. Pottier and Y. Auffret, "On the use of Doppler-shift estimation for simultaneous underwater acoustic localization and communication," OCEANS IEEE, pp. 1-5, 2019.

[23]- Chaofeng He, Yiyin Wang, Cailian Chen and Xinping Guan, "Underwater acoustic localization with uncertainties in propagation speed and time synchronization," OCEANS MTS/IEEE Monterey, pp. 1-5, 2016.

[24]- Z. Li, S. E. Dosso and D. Sun, "Motion-Compensated Acoustic Localization for Underwater Vehicles," IEEE Journal of Oceanic Engineering, vol. 41, no. 4, pp. 840-851, 2016.

[25]- Emmanuel K. Skarsoulis, George Piperakis, Michael Kalogerakis, Emmanuel Orfanakis, Panagiotis Papadakis, Stan. E. Dosso and Alexandros Frantzis, “Underwater Acoustic Pulsed Source Localization with a Pair of Hydrophones”, Remote Sens. 10, 883, 2018.

[26]- R. Jiang, X. Wang, S. Cao, J. Zhao and X. Li, "Deep Neural Networks for Channel Estimation in Underwater Acoustic OFDM Systems," IEEE Access, vol. 7, pp. 23579-23594, 2019.

[27]- Jianyun Yangzhou and Zhengyu Ma,” A deep neural network approach to acoustic source localization in a shallow water tank experiment”, The Journal of the Acoustical Society of America, 146, 4802, pp. 4802–4811, 2019.

[28]- Haiqiang Niu, Emma Reeves, and Peter Gerstoft,” Source localization in an ocean waveguide using supervised machine learning”, The Journal of the Acoustical Society of America 142, 1176, pp. 1176–1188, 2017.

[29]- Digilent atlys board reference page: https://reference.digilentinc.com/reference/programmable-logic/atlys/start.

[30]- Neptune Sonar Ltd: http://www.neptune-sonar.co.uk/

[31]- Teledyne Technologies: http://www.teledyne.com/

[32]- National Instruments: https://www.ni.com/en-ca.html

[33]- MATLAB: https://www.mathworks.com/products/data-acquisition.html

[34]- Francesc Alías, Joan Claudi Socoró and Xavier Sevillano,” A Review of Physical and Perceptual Feature Extraction Techniques for Speech, Music and Environmental Sounds”, Appl. Sci., 6, 143, 2016.

[35]- Anamika Dhillon and Gyanendra K. Verma,” Convolutional neural network: a review of models, methodologies and applications to object detection”, Progress in Artificial Intelligence, volume 9, pp. 85–112, 2020.

VITA AUCTORIS

NAME: Amir Horri

PLACE OF BIRTH: Tehran, Iran

YEAR OF BIRTH: 1984

Education:

Master of Applied Science (Electrical Engineering)	December 2020
University of Windsor	Windsor, ON
Bachelor of Engineering Technology (Mechatronics Stream)	September 2018
University of Windsor	Windsor, ON
Mechanical Engineering Technology – (Robotics and Automation)	September 2011
Conestoga College	Kitchener, ON