A PROJECT REPORT

ON

"Performance Analysis of Modulation Techniques in Underwater Channels"

Submitted to

BITS Pilani K.K. Birla Goa Campus

BY

Ishaant Agarwal	**2016B5A30103G**
Mayank Kumar	**2016B1AA0624G**
Srijan Nikhar	**2016B5AA0474G**

Under the Guidance of
Prof. Sarang C. Dhongdi

EEE/ECE F366
Department of Electrical and Electronics Engineering
BITS Pilani K.K. Birla Goa Campus

2019-2020

Acknowledgements

We are profoundly grateful to **Prof. Sarang Dhongdi** for his expert guidance and continuous encouragement throughout to see that this project is true to its target since its commencement to its completion.

We would like to express our deepest appreciation towards **Archit Saxena** for generously sharing his project report, that proved to be instrumental for the simulations and was one of the founding stones for this project.

At last we must express our sincere and heartfelt gratitude to all the staff members and fellow students of the EEE Department who helped us directly or indirectly during the course of this project.

Ishaant Agarwal
Mayank Kumar
Srijan Nikhar

ABSTRACT

In recent years there has been an increased interest in underwater acoustic communications because of its applications in marine research, oceanography, marine commercial operations, the offshore oil industry and defense. However, the research field is still nascent and many foundational investigations are yet to be carried out. In this project we conduct a theoretical analysis of the underwater channel for acoustic communications. We simulate expected results for the specifications of the hardware in our group and attempt to verify these results using UnetStack3. This analysis should help to accelerate easy deployment of our setup in the water test-bed

Keywords: transducer, hydrophone, UnetStack, SNR, modulation, BER

Contents

List of Figures

Chapter 1

Introduction

Acoustic communications are the typical physical layer technology in underwater networks. Constrained by the physical characteristics of water, communication media other than acoustic waves suffer from severe propagation loss and refraction distortion. Radio waves propagate at long distances through conductive sea water only at extra low frequencies (30 - 300 Hz), which require large antennae and high transmission power. Optical waves do not suffer from such high attenuation but are affected by scattering. Only sound waves can propagate in water over large distances.

Practically all kinds of telemetry, communication, location, and remote sensing of water masses and the ocean bottom use sound waves.However, high-speed communication in the underwater acoustic channel has been challenging because of limited bandwidth, extended multipath, refractive properties of the medium, severe fading, rapid time variation and large Doppler shifts. Compared to terrestrial communication, underwater communication has low data rates because it uses acoustic waves instead of electromagnetic waves. These disadvantages make the underwater acoustic channel one of the most difficult channels to use. The propagation loss depends on the signal frequency and transmission distance. As a result, it is impossible to simultaneously achieve high data rates and long communication distances.

The past three decades have seen a growing interest in underwater acoustic communications because of its applications in marine research, oceanography, marine commercial operations, the offshore oil industry and defense. Continued research over the years has resulted in improved performance and robustness as compared to the initial communication systems. Despite the ongoing efforts to improve the performance of underwater acoustic communications and network protocols, it is still a challenging area.

Chapter 2

Objectives and Outline

This project has four main main objectives:

1. Theoretical analysis of the underwater channel and evaluation of various modulation techniques on it.
2. Simulation of results for the ratings of our hardware equipment
3. Verification of the calculations using a simulation environment on UnetStack.
4. Implementation of this work on hardware in the water test-bed.

Due to the COVID-19 lockdown a large part of the project was disrupted and we were unable to go ahead with the main hardware implementation and testing part of the project. However, we refined and completed the theoretical part instead.

We started off with a thorough investigation and study of the underwater acoustic channel and communication schemes. After getting a fair idea of basic foundational concepts we moved on to exploring UNETsim and UNETAudio, hoping to use the interface as a starting point for our hardware tests. Just before we would start work on the hardware, the lockdown was declared and our plans got disrupted. At this point, we decided to focus on the simulation and theoretical aspects of the project. So, we started with writing out the loss and noise expressions for underwater signals, performed calculations to understand what results we could expect with our hardware and if it would match our needs. We use this to deice on a range of best operating parameters for our successors. Then, we verify these ratings using UNETsim. After verification, we investigate the applicability and utility of different modulation schemes for our signal.

Chapter 3

Theoretical Analysis of UAC

Underwater acoustic communications are mainly influenced by path loss, noise, multi-path, Doppler spread, and high and variable propagation delay. All these factors determine the temporal and spatial variability of the acoustic channel, and make the available bandwidth of the Underwater Acoustic Channel (UAC) limited and dramatically dependent on both range and frequency. Long-range systems that operate over several tens of kilometers may have a bandwidth of only a few kHz, while a short-range system operating over several tens of meters may have more than a hundred kHz bandwidth. In both cases these factors lead to low bitrates. Underwater acoustic communication links can be classified according to their range as very long, long, medium, short, and very short links. Usually in oceanic literature, shallow water refers to water with depth lower than 100m, while deep water is used for deeper oceans.

Category	Range (kms)	Bandwidth(kHz)
Very Short	<0.1	>100
Short	0.1-1	20-50
Medium	1-10	≈ 10
Long	10-100	2-5
Very Long	>100	<1

Table 3.1: Typical bandwidths and ranges for UAC

We analyzed the factors that influence acoustic communications:

3.1 Transmission loss

Transmission loss is caused mainly by attenuation and geometric spreading during the propagation of acoustic signals in water. Attenuation is generally caused by absorption when acoustic energy is transferred into heat. A distinguishing property of acoustic channels is that the absorptive loss for acoustic waves will increase with both propagation distance and frequency. In underwater acoustic communication, whose carrier frequencies are less than 50 kHz, the absorption coefficient α can be expressed using Thorp's empirical formula (Thorp, 1967) with respect to frequency f in kHz as

$$\alpha(f) = \frac{0.11f^2}{1+f^2} + \frac{44f^2}{4100+f^2} + \frac{2.75}{10^4}f^2 + \frac{3}{10^3} \tag{3.1}$$

It gives α in dB/km. It is obvious that the absorption coefficient increases rapidly with the frequency. Hence, the available frequency for transmission is limited for underwater acoustic communication.

Geometric spreading loss is caused by the spreading of acoustic energy into a larger area as a consequence of acoustic wave expansion. Generally, there are two types of geometric spreading loss: spherical and cylindrical. Spherical spreading occurs when the source is omnidirectional and acoustic waves spread spherically, and it is usually applied for deep-sea acoustic communication. Cylindrical spreading occurs when acoustic waves spread horizontally and it is applicable for shallow water acoustic communications. In practical underwater channels, geometric spreading is a hybrid of spherical and cylindrical spreading. The geometric spreading is dependent on only propagation distance and it is frequency-independent.

Most papers then give an equation for Transmission Loss as follows:

$$10 \log \mathrm{TL}(d, f) = k \cdot 10 \log d + d \cdot \alpha(f) \tag{3.2}$$

However, in this equation the first d (in the geometric loss term) is in units of metres while the d in the attenuation term is in units of kms. Therefore strictly speaking, the following equation should be followed:

$$10 \log \mathrm{TL}(d, f) = k \cdot 10 \log d + d \cdot \alpha(f) \times 10^{-3} \tag{3.3}$$

where d is distance in metres.

3.2 Ocean environment noise

Acoustic noise in the underwater communication channel can be either ambient noise or man-made noise. Man-made noise is caused mainly by machinery. Even in the quiet deep sea, ambient noise still exists. There are four main sources for ambient noise in the ocean: turbulence, shipping, waves, and thermal noise. Because of the multiple sources, the ambient noise can be approximated as a non-white Gaussian variable. The level of underwater ambient noise may also vary based on time and location. The power spectral density of these four noise components is given by an empirical formula (Stojanovic, 2006b) in dB re μPa/Hz (the sound pressure at reference sound power per Hz) as a function of frequency in kHz as

$$\begin{aligned}
10 \log N_t(f) &= 17 - 30 \log f \\
10 \log N_s(f) &= 40 + 20(s-5) + 26 \log f - 60 \log(f + 0.03) \\
10 \log N_w(f) &= 50 + 7.5\sqrt{w} + 20 \log f - 40 \log(f + 0.4) \\
10 \log N_{th}(f) &= -15 + 20 \log f
\end{aligned} \tag{3.4}$$

Turbulence noise influences only the very low frequency region, f < 10 Hz. Noise caused by distant shipping is dominant in the frequency region 10 Hz -100 Hz, and it is modeled using the shipping activity factor s, ranging from 0 to 1 for low and high activity, respectively. Surface motion, caused by wind-driven waves (w is the wind speed in m/s) is the major factor contributing to the noise in the frequency region 100 Hz - 100 kHz (which is the operating region used by the majority of acoustic systems). Finally, thermal noise becomes dominant for $f > 100$ kHz. The power spectral density of ambient noise relative to f is given by:

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f) \tag{3.5}$$

Remember that this equation is in the linear domain, while the above p,s,d expressions are in the log domain. We will need to do the necessary conversions before adding them.

Using the tranmission loss TL(d, f) and the noise p.s.d. $N(f)$ one can evaluate the signal-to-noise ratio (SNR) observed over a distance d when the transmitted signal is a tone of frequency f and power P. The narrow-band SNR is given by:

$$\mathrm{SNR}(d,f) = \frac{P/\mathrm{TL}(d,f)}{N(f)\Delta f} \tag{3.6}$$

where P and f are the power and frequency of the transmitted signal, respectively, and Δf is the receiver noise bandwidth in Hz

In decibels this corresponds to

$$\mathrm{SNR}(dB) = P - (TL + N + 10\log(\Delta f)) \tag{3.7}$$

where all quantities except for Δf are in dB.

Chapter 4

Hardware Specifications

4.1 Basic Implementation

Our setup for underwater acoustic communication can be represented as follows:

I. Message Generator

II. Preamplifier

III. Transducer

IV. Hydrophone

V. Power Amplifier

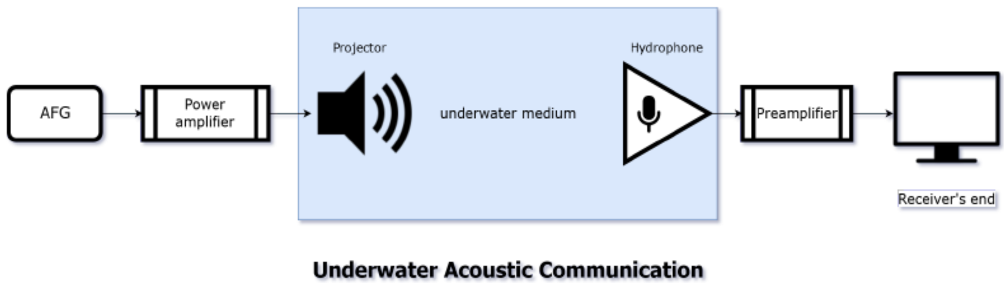


Figure 4.1: End to End block model of underwater acoustic communication

The communication process can be thought of as comprising the following steps:

i. A message is generated/inputted and modulated in some scheme to form an analog signal.

ii. The signal is sent to the preamplifier for amplification.

iii. The amplified signal is converted by the transducer and transmitted across the medium(in our case, water) as an acoustic wave.

iv. At the receiver side, the hydrophone picks up the acoustic signal, and converts t into an electrical signal which is sent to the amplifier.

v. The output of the amplifier is the received signal which can then be demodulated and decoded for the final message.

4.2 Equipment Overview

4.2.1 Power Amplifier (BII 5011)

A power amplifier is an electronic amplifier designed to increase the magnitude of power of a given input signal. The power of the input signal is increased to a level high enough to drive loads like RF Transmitters, Headphones or in our case, Transducers. The given input signal will be fed from some DSP or laptop. The specification sheets given by manufactures dictate usage characteristics and parameters of the device. The model being used is the Benthowave model BII-5011. It is a 7-watt, linear wideband amplifier, which offers low distortion and low power consumption to battery-powered underwater acoustic system. Other important characteristics:

- **Source Level Capability**: 177 + DI (dB re μPa)
- **Input Impedance**: 10kΩ
- **Gain**: 26 dB

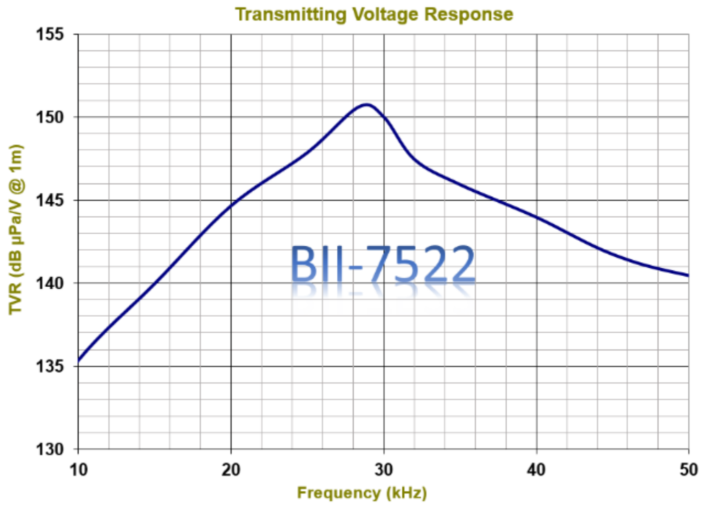
4.2.2 Transducer (Transmitter) (BII 7522)

A transducer converts some sort of energy to sound (source) or converts sound energy (receiver) to an electrical signal.Because the field of transducers is large by itself, we concentrate in this section on some very practical issues that are immediately necessary to either convert receive voltage levels to pressure levels or transmitter excitation to pressure levels. Practical issues about transducers (and hydrophones) deal with the understanding of specification sheets given by the manufacturer. Among those, we will describe on a practical example the definition and use of the following quantities:

- Transmitting voltage response
- Transmitting beam patterns at specific frequencies.
- Resonant frequency, maximum voltage and maximum source level

Part Number:	BII-7522
Signal Type:	Pulsed SINE, Chirp, PSK, FSK, etc.; Pulsed Square Waveform
Resonant Frequency fs:	#1: 27.8 kHz; #2: 28.4 kHz.
Quality Factor:	#1: 4.6; #2: 4.2.
Transmitting Voltage Response:	Refer to TVR Graph
Free-field Voltage Sensitivity:	-190.0 dB V/µPa @ fs
-3dB Beam Width:	Omnidirectional
Beam Pattern:	Refer to Beam Pattern Graph

(a)



(b)

Figure 4.2: Extract from specification sheet of our underwater acoustic transponder

Figure 4.2 (b) corresponds to the transmitting sensitivity versus frequency. The units are in dB re μPa/V @ 1m, which means that, at the resonant frequency 27.8 kHz for example, the transducer excited with a 1V amplitude transmits at one meter a pressure P_t such that $20\log_{10}\left(\frac{P_t}{1\times 10^{-6}}\right) = 151$ dB i.e. $P_t \approx 35.48$ Pa

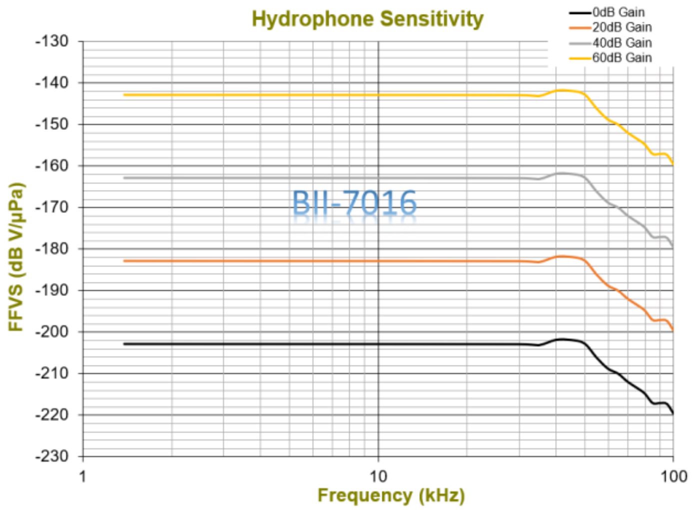
4.2.3 Hydrophone (Receiver) (BII 7016) and Pre-amplifier (BII 1092)

Hydrophones are usually described with similar characteristics as transducers but they are designed to work in reception. To this goal, hydrophones are usually connected to a pre-amplifier with high input impedance to avoid any loss in the signal reception. Hydrophones usually work on large frequency bandwidth since they don't need to be adjusted to a resonant frequency. Like transducers, specification sheets have important quantities that dictate their usage:

- Free-field Voltage Response (FFVS)
- Transmitting beam patterns at specific frequencies.
- Pre-amplifier gain, maximum voltage and frequency range

Part Number:	BII-7016
Free-field Voltage Sensitivity:	Refer to Graph of **FFVS vs. Frequency**.
Usable Frequency:	1 Hz to 70 kHz
Bespoke Preamp Gain (dB):	Programmable Gain Preamp: 0, 20, 40, 60 dB.

(a)



(b)

Figure 4.3: Extract from specification sheet of our hydrophone

Fig. 4.3 (b) shows the receiving sensitivity versus frequency. The units are now in dB re 1V/μPa which means that, at 28 kHz for example, with no preamp gain, the hydrophone converts a 1μPa amplitude field into a voltage V$_r$ such that $20\log_{10}\left(\frac{V_r}{1}\right) = -202$ dB i.e. V$_r$ $\approx 7.9 \times 10^{-11}$V.

Chapter 5

UnetStack

5.1 Introduction

UnetStack is an agent-based network stack used in the UNET project(The Underwater Networks Project). Developed by the ARL in NUS, it is used to develop and test underwater networks. There is also an inbuilt simulator which can be used to mimic the behaviour of nodes and the underwater channel. These networks can also be deployed on UNET-compatible modems allowing for real-world analysis of networks. There is also UNETAudio, which uses the laptop soundcard as a modem, to generate soundwaves.

5.2 Working

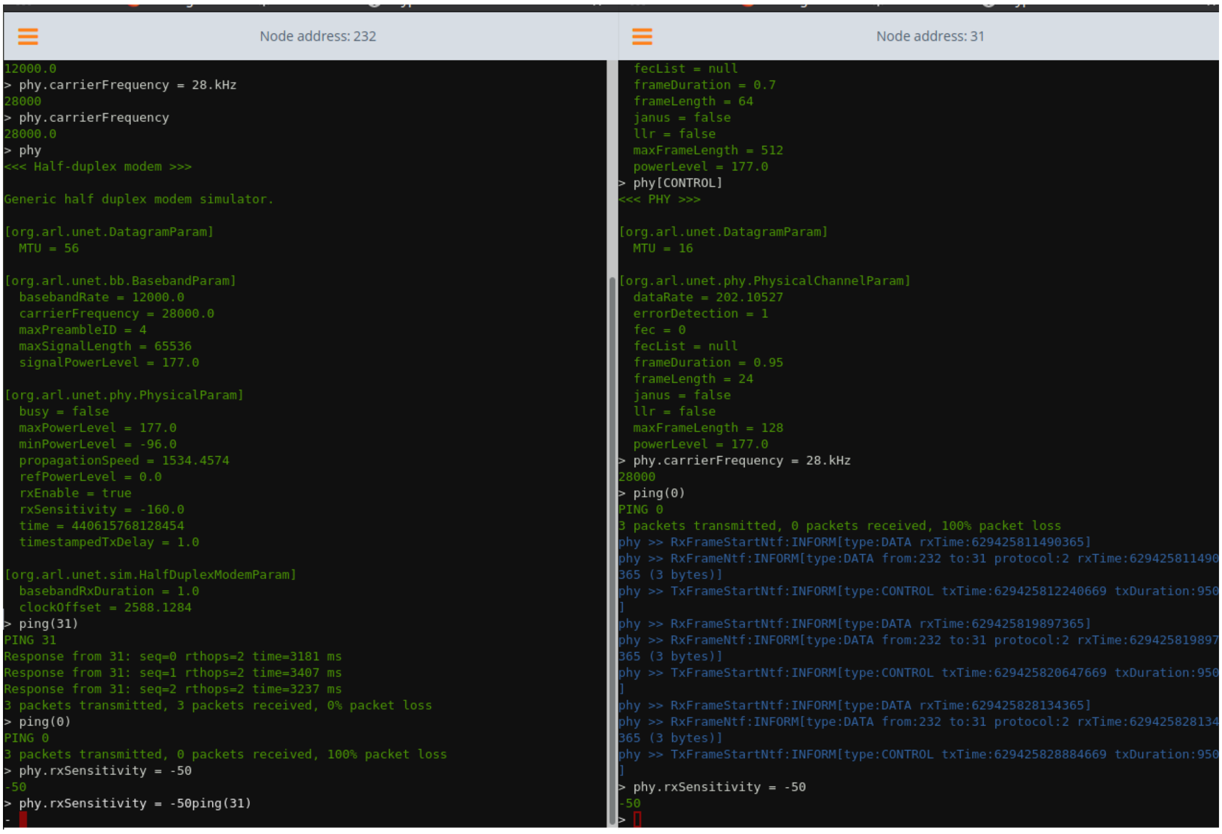


Figure 5.1: Code used for simulation of a 2 node network in UnetSim

We started using the UNETStack software, by running the basic 2-node-network. In this simulation there are two nodes about a kilometre apart. Each node is loaded with default protocols for routing, mac, transport, etc. We were able to move the nodes location, and test out different power levels of transmission along with their effects on communication.

We then tried out the UNETAudio package, which uses the laptops soundcard to send and receive soundwaves. Using this we were actually able to use our laptops as nodes, moving it around and sending signals from one to another. Any signal would be generated as a high frequency wave which could be heard by us. We could also change the power level of transmission to see its effects. There is an inbuilt method in UNETAudio to send a standard signal and check the BER in the received signal. We used this to try different distances, power levels and frequencies to check it's BER.

5.3 UnetStack Model

The next step was to use the UNETSim to simulate the model and check our calculations. The aim was to capture the underwater channel model and match it's BER with our calculated values. We used the values from the hardware datasheet to model our nodes. The underwater channel model has already been explored in previous chapters. To do this we modified the 2-node-network model. An inbuilt class, *BasicAcousticModel*, was available which could be used to model the Urick model. The parameters used are shown in the code. We tried different power levels and distances in this network. Commands executed in the shell of 2_node_network-

i. **node**: This command allowed us to see the node characteristics such as mobility, location.

ii. **node.location**: This is used to get/set the current location of the node. It is what we used to vary distance between nodes.

iii. **phy**: This command shows us the characteristics of the modem we are simulating.

iv. **phy.refPowerLevel**: This is the reference value of transmission power settings. The final transmitting power is calculating using *plvl* referred to this value.

v. **phy.maxPowerLevel**: The maximum allowable power level which can be set for plvl.

vi. **phy.minPowerLevel**: The maximum allowable power level which can be set for plvl.

vii. **plvl**: This allowed us to change transmission power level. This value is in reference to the *refPowerLevel* parameter.

viii. **phy.carrierFrequency**: This allows to change the carrier Frequency.

ix. **phy.rxSensitivty**: The name is misleading as this is NOT the receiver threshold value to check for incoming transmissions. Changing this value did not seem to affect reception in any way we could see.

x. **ping(*dest*, *no*)**: This command sent a ping to the destination node given by *dest*. The node was pinged *no* number of times and the packet loss was calculated on the sender node. A packet is defined as a loss when either there is a timeout or there is a bit error in the frame sent/received.

We can only check BER only in UNETAudio mode. The commands to be typed are:

```
>phy[DATA].threshold = 0
>phy[CONTROL].test = true
>phy[CONTROL].fec = 0
> 10.times { phy << new TxFrameReq(); delay(2000); } // This is groovy syntax
//This command sends out 10 transmissions with 2 seconds delay
```

The output of this would be

```
phy >> TxFrameNtf:INFORM[type:CONTROL txTime:204359766]
phy >> RxFrameNtf:INFORM[type:CONTROL rxTime:204385187 rssi:-28.9 cfo:0.0 ber:0/144 (18 bytes)]
phy >> TxFrameNtf:INFORM[type:CONTROL txTime:205578432]
phy >> RxFrameNtf:INFORM[type:CONTROL rxTime:205603853 rssi:-28.4 cfo:0.0 ber:0/144 (18 bytes)]
phy >> TxFrameNtf:INFORM[type:CONTROL txTime:207567766]
phy >> RxFrameNtf:INFORM[type:CONTROL rxTime:207589186 rssi:-28.5 cfo:0.0 ber:0/144 (18 bytes)]
phy >> TxFrameNtf:INFORM[type:CONTROL txTime:209583766]
phy >> RxFrameNtf:INFORM[type:CONTROL rxTime:209609187 rssi:-28.2 cfo:0.0 ber:0/144 (18 bytes)]
phy >> TxFrameNtf:INFORM[type:CONTROL txTime:211573099]
phy >> RxFrameNtf:INFORM[type:CONTROL rxTime:211594519 rssi:-28.3 cfo:0.0 ber:0/144 (18 bytes)]
phy >> TxFrameNtf:INFORM[type:CONTROL txTime:213589099]
phy >> RxFrameNtf:INFORM[type:CONTROL rxTime:213614520 rssi:-28.1 cfo:0.0 ber:0/144 (18 bytes)]
phy >> TxFrameNtf:INFORM[type:CONTROL txTime:215578432]
phy >> RxFrameNtf:INFORM[type:CONTROL rxTime:215599853 rssi:-28.5 cfo:0.0 ber:0/144 (18 bytes)]
phy >> TxFrameNtf:INFORM[type:CONTROL txTime:217594432]
phy >> RxFrameNtf:INFORM[type:CONTROL rxTime:217619853 rssi:-28.2 cfo:0.0 ber:0/144 (18 bytes)]
phy >> TxFrameNtf:INFORM[type:CONTROL txTime:219583766]
phy >> RxFrameNtf:INFORM[type:CONTROL rxTime:219605186 rssi:-28.0 cfo:0.0 ber:0/144 (18 bytes)]
phy >> TxFrameNtf:INFORM[type:CONTROL txTime:221599766]
phy >> RxFrameNtf:INFORM[type:CONTROL rxTime:221625187 rssi:-27.7 cfo:0.0 ber:0/144 (18 bytes)]
```

Unfortunately we could not find BER as easily as we did while using UNETAudio, we used the *ping* command to check packet loss between the two nodes. The next problem we had was to actually set the Receiver Threshold. This was not available directly as a setting we could control and we would have to manually code it up.

```
//setting up channel properties
channel.model = org.arl.unet.sim.channels.BasicAcousticChannel

channel.carrierFrequency = 28.kHz       //f
channel.bandwidth = 4096.Hz             //B
channel.spreading = 2                   //α
channel.temperature = 25.C              //T
channel.salinity = 35.ppt               //S
channel.noiseLevel = 0.dB               //N0
channel.waterDepth = 20.m               //d

channel.ricianK = 10
channel.fastFading = true
channel.pfa = 1e-6
channel.processingGain = 0.dB

modem.dataRate = [2400, 2400].bps              // arbitrary data rate
modem.frameLength = [2400/8, 2400/8].bytes     // 1 second worth of data per frame
modem.headerLength = 0                         // no overhead from header
modem.preambleDuration = 0                     // no overhead from preamble
modem.txDelay = 0                              // don't simulate hardware delays
```

Failing this we tried to make our own script which we could use to actually calculate BER using the channel and communication model with the parameters set according to our hardware. This required going into the documentation of the API and finding out how everything was implemented. Unfortunately our script failed and we could not debug it as no error messages were being shown explicitly.

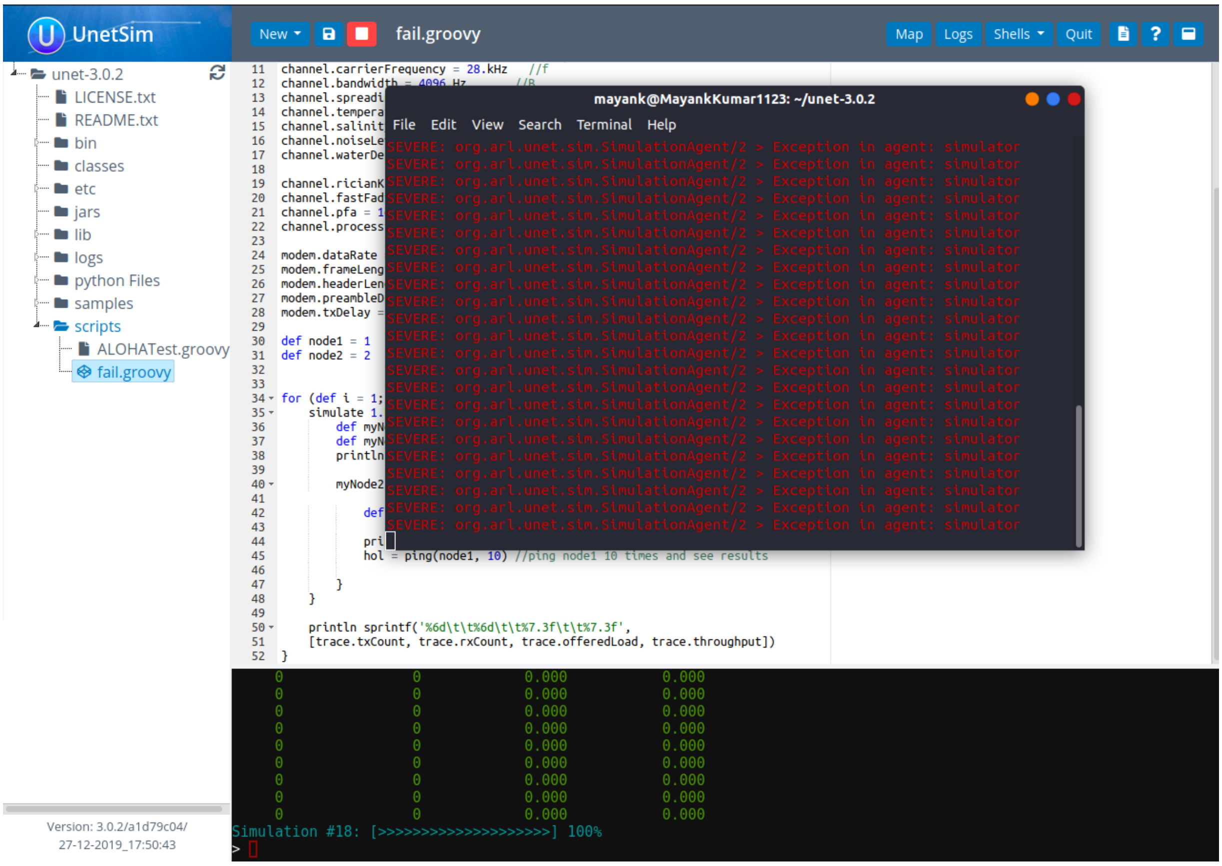


Figure 5.2: Displayed error messages

Chapter 6

Methods

6.1 Theoretical Analysis

I coded up a simulation to find out SNR for a signal in waterbed as well as ocearnic environment. I assumed that the noise in a waterbed will be caused mostly due to thermal processes since the other factors do not come into factor as much.

6.2 Autocorrelation function Code

channel_model.m

```
%% Parameters
f=28E3 ; % Transmission frequency
L=100*1E-3; %Length of path in km
% PtW=150 %Transmission power
Pt=150 %10*log10(PtW);

envr=1; %Put envr =1 for indoor tank and envr =2 for sea simulation
bw=1; %KHz
Gt=26; %Transmission Amplifier Gain
f_k=f/1000 %in KHz
A0=1; %Scaling Constant
k=1.5; %spreading factor; k = 2 for spherical spreading, k = 1 for cylindrical spreading, and k
    = 1.5 for so-called practical spreading
%% Path Loss

a=0.11*(f_k^2)/(1+(f_k^2))+44*(f_k^2)/(4100+(f_k^2))+2.75E-4*f_k^2+0.003; %Absorption
    coefficient in db/km

% Path Loss in dB
A =10*log10(A0)+k*10*log10(L*1E3)+L*(a);
%% Noise

s=0; %Shipping acticvity coefficient 0-min, 1-max
w=0; %Speed of wind in m/s

% Noise in dB
Nt=17-30*log10(f_k);  %Turbulence
Ns=40+20*(s-0.5)+26*log10(f_k)-60*log10(f_k+0.03); %Shipping
Nw=50+7.5*sqrt(w)+20*log10(f_k)-40*log10(f_k+0.4); %Surface motion due to wind driven waves
Nth=-15+20*log10(f_k); %Thermal Noise
bw_dB=10*log10(bw);

if (envr==2)
%{
%     switch true
%         case f<10
%             N=Nt
```

```
%           case f<100
%               N=Ns
%           case f<100E3
%               N=Nw
%           case f>100E3
%               N=Nth
%       end
%}
    N=10*log10(10^(Nt/10)+10^(Ns/10)+10^(Nw/10)+10^(Nth/10))+bw_dB
else

    N=-15+20*log10(f_k)+bw_dB %Only thermal noise in indoor tanks

end


% N=30;
%%

SNR=Pt+Gt-A-N
```

code/channel_model.m

The commented part in the noise section actually replaces the summation of noise with only the dominant contributing factor. Putting envr=2 gives an oceanic channel noise model and envr=1 gives a water bed simulation. I also put a non-negative constraint on Path Loss because the expression would result in amplification instead of attenuation over short distances and that is not physically possible.

This function will give the resultant SNR according to the situation parameters as output. All other quantities will also be available in the workspace.

To automate the generation of SNR value for multiple starting parameters (like distance) this script can easily be changed to a function format and run. An example of this applied to distance follows:

main.m and sim_au_snr.m

```
clear all
for i=1:61
    L(i)=10^((i-1)/15);
    SNR_temp=sim_au_snr(28E3,L(i)*1E-3);
    SNR(i)=SNR_temp;
    if (SNR_temp<0)
        break
    end
end

semilogx(L,SNR,'-o')
title('SNR vs Distance for sea')
ylabel('SNR in db')
xlabel('Distance (in m)')
grid on
```

code/main.m

```
function SNR=sim_au_snr(f,L)
%% Parameters
% f=28E3 ; % Transmission frequency
% L=0.5; %Length of path in km
% PtW=150 %Transmission power
Pt=150; %10*log10(PtW);
bw=1; %Bandwidth in KHz

envr=2; %Put envr =1 for indoor tank and envr =2 for sea simulation
```

```

Gt=26; %Transmission Amplifier Gain
f_k=f/1000 %in KHz
A0=1; %Scaling Constant
k=1.5; %spreading factor; k = 2 for spherical spreading, k = 1 for cylindrical spreading, and k
    = 1.5 for so-called practical spreading
%% Path Loss

a=0.11*(f_k^2)/(1+(f_k^2))+44*(f_k^2)/(4100+(f_k^2))+2.75E-4*f_k^2+0.003; %Absorption
    coefficient in db/km

% Path Loss in dB
A =10*log10(A0)+k*10*log10(L*1E3)+L*(a);
A=A*(A>0);

%% Noise

s=0; %Shipping acticvity coefficient 0-min, 1-max
w=0; %Speed of wind in m/s

% Noise in dB
Nt=17-30*log10(f_k);  %Turbulence
Ns=40+20*(s-0.5)+26*log10(f_k)-60*log10(f_k+0.03); %Shipping
Nw=50+7.5*sqrt(w)+20*log10(f_k)-40*log10(f_k+0.4); %Surface motion due to wind driven waves
Nth=-15+20*log10(f_k); %Thermal Noise
bw_dB=10*log10(bw);

if (envr==2)
%{
%     switch true
%         case f<10
%             N=Nt
%         case f<100
%             N=Ns
%         case f<100E3
%             N=Nw
%         case f>100E3
%             N=Nth
%     end
%}
    N=10*log10(10^(Nt/10)+10^(Ns/10)+10^(Nw/10)+10^(Nth/10))+bw_dB
else

    N=-15+20*log10(f_k)+bw_dB %Only thermal noise in indoor tanks

end


% N=30;
%%

SNR=Pt+Gt-A-N
end
```

code/sim_au_snr.m

Chapter 7

Results

7.1 Theoretical Analysis

7.1.1 Signal Variation with distance:

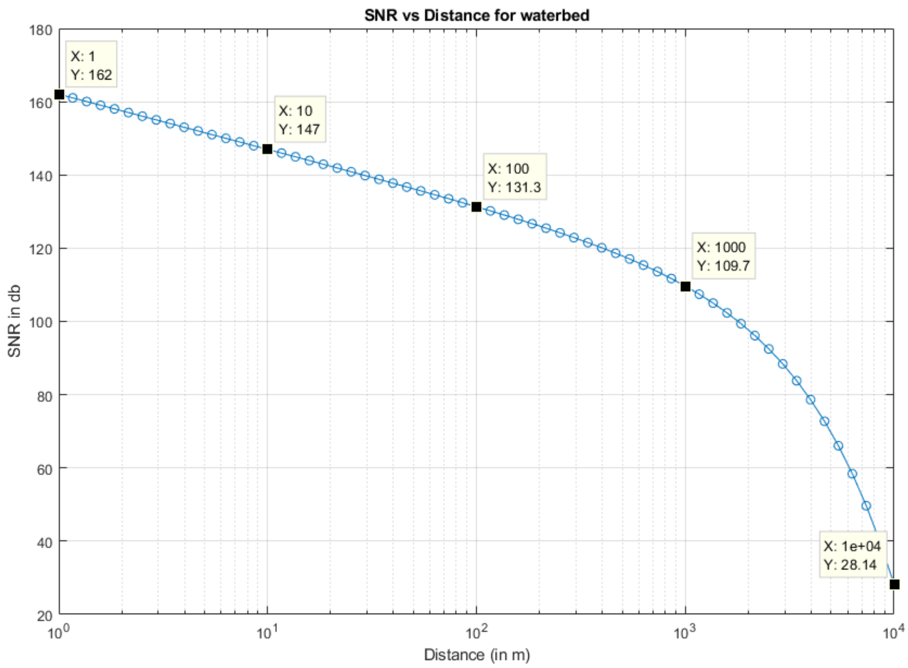


Figure 7.1: SNR trend with distance in a waterbed environment

Distance	SNR	Received Voltage for different preamp gains			
		0 db	20 db	40dB	60dB
0 m	162 db	0.001 V	0.01 V	1 V	100 V
10 m	147 db	3.16E-6 V	3.16E-4 V	3.16E-2 V	3.16 V
100 m	131.3 db	8.51E-8 V	8.51 E-6 V	8.51E-4 V	8.51E-2 V
1000 m	109.7 db	5.88E-3 V	5.88E-8 V	5.88E-6 V	5.88E-4 V

Table 7.1: Estimated Voltage output of the hydrophone with various preamp settings and source positions in waterbed environment

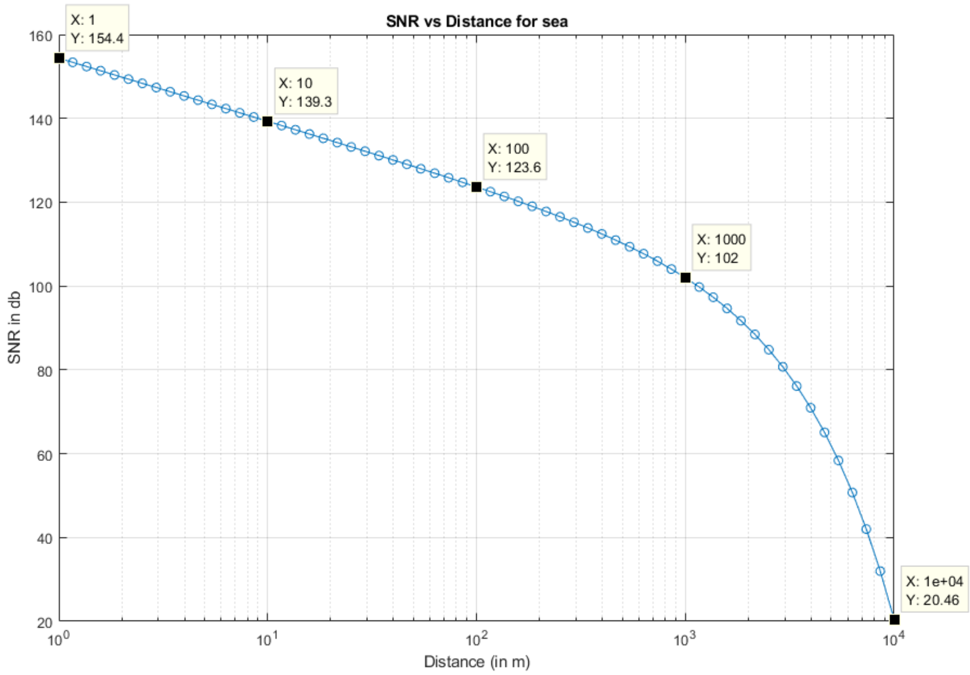


Figure 7.2: SNR trend with distance in a sea-like environment

Distance	SNR	Received Voltage for different preamp gains			
		0 db	20 db	40dB	60dB
0 m	154.4 db	1.73E-5 V	1.73E-3 V	0.173	17.37 V
10 m	139.3 db	5.37E-7 V	5.37E-5 V	5.37E-3 V	0.537V
100 m	123.6 db	1.44E-8 V	1.44E-6 V	1.44E-4 V	1.44E-2 V
1000 m	102 db	1E-10 V	1E-8 V	1E-6 V	1E-4 V

Table 7.2: Estimated Voltage output of the hydrophone with various preamp settings and source positions in sea

The two graphs show how the SNR falls as the source-receiver distance is increased. There is an almost linear fall in SNR with log(distance) upto 1 km and then the signal quality starts falling drastically. This is because the absorption loss which is linearly proportional to the distance starts dominationg the spreading loss, which has only a logarithmic dependence on distance. The accompanying tables show what receiving voltage we might find for 1V input at the transducer. Even the SNR is calculated considering 1V input. The hydrophone receiving response is calculated as

$$V_{out} = 10^{\frac{\text{SNR-RS+G}}{10}} \times V_{in} \tag{7.1}$$

where G refers to the preamp gain in dB and RS is receiver sensitivity in dB re 1V/μPa

7.2 Modulation Schemes

Here we talk about different modulation schemes which can be used. One of the basic ones is PSK(Phase Shift Key), in which different symbols are represented by different phases of the same carrier wave. For Binary Phase Shift Key(BPSK), the phases are 180° apart. The maximum rate of modulation of BPSK is 1 bit per symbol. QPSK(Quadrature Phase Shift Keying) is another form of PSK, where 2 bits are represented by one symbol. Here the phases are 90° apart.
Another form of modulation is QAM(Quadrature Amplitude Modulation). This can transmit two signals in parallel, by using ASK on the signals. Each signal would have a carrier wave with same frequency as the other signal, but a 90° phase shift. The final signal would be the summation of these two carriers.
Shown below is a comparison of the different digital modulation schemes as BER vs SNR graph. From the figure we can see that BPSK hsa the lowest BER values compared to QPSK and QAM. However, we must remember that BPSK also offers the least data rate at the same bandwidth compared to QPSK and QAM.

The BER formulae are well known for FSK and QPSK modulation techniques (Rappaport,1996), which require the Energy per Bit to Noise psd, $\frac{E_b}{N_o}$, that can be found from the SNR by:

$$\frac{E_b}{N_o} = SNR(r) \times \frac{B_c}{R_b} \tag{7.2}$$

where R_b is the data rate in bps and B_c is the channel bandwidth. Equation 7.3 and 7.4 are the uncoded BER for BPSK/QPSK and FSK respectively:

$$QPSK: \quad \text{BER} = \frac{1}{2}\,\text{erfc}\left[\frac{E_b}{N_o}\right]^{1/2} \tag{7.3}$$

$$FSK: \quad \text{BER} = \frac{1}{2}\,\text{erfc}\left[\frac{1}{2}\frac{E_b}{N_o}\right]^{1/2} \tag{7.4}$$

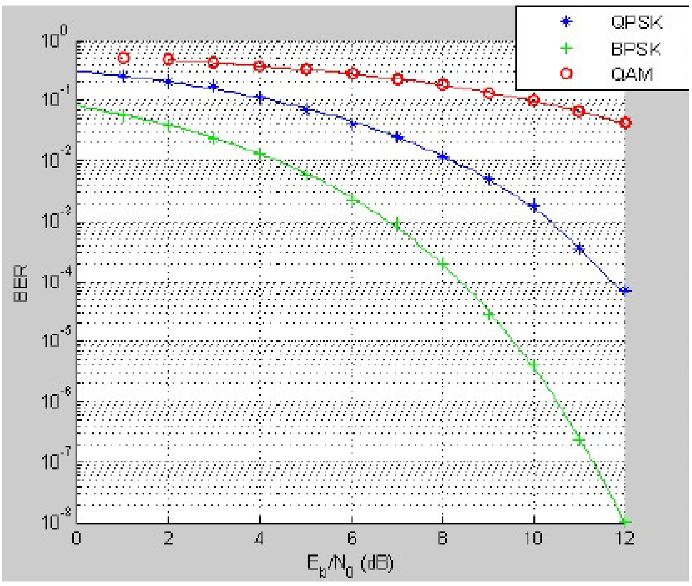


Figure 7.3: Comparison of different modulation schemes in a Rayleigh channel

7.3 UNET Simulations

We used UNETStack to first get the minimum power required to successfully transmit and receive for different distances between two nodes. The channel model used is the Urick model with a rxSensitivity of -200 dB.

Distance (m)	Min Power(dB)
100 m	110 dB
200 m	116 dB
300 m	119 dB
400 m	122 dB
500 m	124 dB
600 m	126 dB
700 m	128 dB
800 m	130 dB
900 m	130 dB
1000 m	130 dB

Also, using this model and a source power level of 177 dB, we found the maximum distance a packet can travel without errors is 4027m.

Chapter 8

Conclusion and Future Scope

In this project, we studies various tools available to simulate underwater communications. Focussing on UNETstack and custom scripts, we simulated transmission and receival of underwater acoustic signals. Important quantities such as range and output voltage were calculated for different scenarios. We hope that the information compiled and provided in this report will help future researchers in hardware deploement and can serve as a quickstart guide.

Going forward, the calculations has to be confirmed through actual hardware tests. The actual modems along with a waterbed for testing can be used for this. This is only for sending and receiving a standard signal across the waterbed. One could further this by actually sending meaningful custom messages across and trying to decode it at the other end. This would require the use of some sort of processor on both ends, to modulate/demodulate the signal and get the original message. Different modulation schemes can be explored and a performance analysis can be carried out to evaluate optimal setups for different applications.

References

[1] M. Stojanovic, On the relationship between capacity and distance in an underwater acoustic channel, Vol. 11, 2006, pp. 41–47. doi:10.1145/1161039.1161049.

[2] M. A. Chitre, R. Bhatnagar, W.-S. Soh, Unetstack: An agent-based software stack and simulator for underwater networks, 2014 Oceans - St. John's (2014) 1–10.

[3] M. Chitre, S. Shahabudeen, M. Stojanovic, Underwater acoustic communications and networking: Recent advances and future challenges, Marine Technology Society Journal 42 (2008) 103–116. doi:10.4031/002533208786861263.

[4] L. M. Brekhovskikh, Y. P. Lysanov, Fundamentals of ocean acoustics (3rd edition), The Journal of the Acoustical Society of America 116 (4) (2004) 1863–1863. doi:10.1121/1.1792644.

[5] J. Huang, H. Wang, C. He, Q. Zhang, L. Jing, Underwater acoustic communication and the general performance evaluation criteria, Frontiers of Information Technology Electronic Engineering 19 (2018) 951–971. doi:10.1631/FITEE.1700775.

[6] I. Akyildiz, D. Pompili, T. Melodia, Challenges for efficient communication in underwater acoustic sensor networks, ACM SIGBED Review 1. doi:10.1145/1121776.1121779.

[7] U. Qureshi, F. Shaikh, Z. Aziz, S. Shah, A. Sheikh, E. Felemban, S. Qaisar, Rf path and absorption loss estimation for underwater wireless sensor networks in different water environments, Sensors 16 (6) (2016) 890. doi:10.3390/s16060890.

[8] W. Kuperman, P. Roux, Underwater acoustics, Springer Handbook of Acoustics, ISBN 978-0-387-30446-5. Springer-Verlag New York, 2007, p. 149 -1 (2007) 149. doi:10.1007/978-0-387-30425-0_5.

[9] G. Burrowes, J. Khan, Short-Range Underwater Acoustic Communication Networks, 2011. doi:10.5772/24098.

[10] E. M. Sözer, Underwater Acoustics: A Brief Introduction, MIT Sea Grant College Program.
URL https://dspace.mit.edu/bitstream/handle/1721.1/74140/2-017j-spring-2006/contents/lecture-notes/05_3uap_notes.pdf

[11] A. Saxena, Analysis of modulation techniques on underwater acoustic modem, BITS Goa Undergraduate Thesis BITS F421T 2019.