

ABSTRACT

GOVILKAR, SAMIR R. Simulation of communication for power constrained embedded systems. (Under the direction of Assistant Professor Alexander G. Dean).

The aim of this thesis is to aid in the development of the RAPTEX project. The RAPTEX project deals with designing efficient communication protocols for underwater telemetry and sensor networks.

Underwater telemetry is used to track crabs, initially in shallow water and later in marine environments. Ultrasonic waves are used for such underwater telemetry. The ultrasonic waves are affected by various phenomena as they pass through water. These phenomena include attenuation, multi-path fading, noise, etc. A small microcontroller based transmitter will be placed on the back of the crab allowing the telemetry data to be transmitted. The microcontroller will directly drive an ultrasonic transducer through a Digital to Analog Converter to output a modulated carrier wave. It is necessary to simulate the communication between the microcontroller based transmitter and the receiver before testing this system in the real-world environment.

The purpose of the simulation is two-fold. It helps to validate the design and it also helps in verifying the end-to-end communication of the system. This becomes important as more and more higher layer protocols are added to the RAPTEX project. Since the simulation environment has control over all the parts of the communication system; viz. the transmitter (based on the AVR microcontroller), the channel and the receiver, it is possible to inspect the flow of data at any point by adding suitable taps.

Simulation of communication for power constrained embedded systems

by

Samir R. Govilkar

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Master of Science

Computer Engineering

Raleigh

2006

Approved By:

Dr. Suleyman Sair

Dr. Mihail L. Sichiuiu

Dr. Alexander G. Dean
Chair of Advisory Committee

Dedicated to my parents and brother ...

Biography

Samir Govilkar was born on July 14, 1982 in Gujarat, India. He grew up in Mumbai and graduated with a Bachelor's degree in Electronics Engineering from the University of Mumbai. During his undergraduate days he worked with Philips Semiconductors India Ltd. to prototype a contactless smart card reader.

He then went on to pursue his Master of Science in Computer Engineering at North Carolina State University. While working on his degree there he worked as an intern at Intel Corp. in Raleigh, North Carolina doing embedded software development. He worked on his thesis under the guidance of Dr. Alex Dean at the North Carolina State University.

Acknowledgements

I am thankful for the active part that Dr. Alexander Dean has taken in guiding me through the research for this thesis. He always encouraged and appreciated my work, whilst focusing my efforts in the correct direction. It would have been really difficult to finish this thesis without his support.

I would also like to thank Dr. Suleyman Sair and Dr. Mihail Sichiuiu for being on my thesis committee.

My roommates and friends, Viraj Datar, Trushant Kalyanpur and Manan Shah made my stay at NC State a pleasure and were always supportive and helpful during the time we lived together. I thank them for the fantastic time we had. Kirtesh Patil, my roommate for the last six months also deserves mention for being a great guy.

Gregory Parsons was always helpful and I enjoyed our long discussions on topics ranging from technology to the history of the United States.

I owe a debt of eternal gratitude to the best parents one can hope for, Rajshekhar and Shubhada Govilkar. My work and studies here at NC State University would have been impossible without their support. My younger brother, Mihir always believed in me and encouraged and pestered me to try and do things that I would never otherwise would have attempted. A big thank you to them.

Contents

List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 The RaPTEX Project	2
1.2 Studying crabs using acoustic biotelemetry	4
2 The Problem	6
2.1 A statement of the problem	6
2.2 Considerations in underwater acoustic communication	7
2.2.1 Propagation Losses	8
2.2.2 Multiple Paths	8
2.2.3 Noise	9
2.3 Requirement for a simulation environment	10
3 Related Work	12
3.1 Avrora - AVR Simulator	12
3.2 IT++ - A Signal Processing Library	14
3.3 Other related work	15
4 Theory	17
4.1 Modelling of Propagation Losses	17
4.1.1 Spreading Losses	18
4.1.2 Absorption Losses	20
4.2 Noise Modelling	23
4.2.1 Ambient Noise	23
4.2.2 Intermittent Noise	24
4.3 Fading Channel	26
4.4 Sampling Rate Conversion	28
4.4.1 Calculation of rational factors	30
4.4.2 Design of FIR low pass filters	30

5	Implementation	32
5.1	Overview of the implementation	32
5.2	Embedded System Simulator	34
5.3	Water Channel Simulator	37
5.4	Receiver Simulator	39
5.5	Visualization Module	41
6	Case study	44
6.1	Amplitude Shift Keying Theory	44
6.2	Implementation	45
6.3	Simulation	46
6.4	Results	47
7	Conclusion	51
7.1	Observations	51
7.2	Future Work	51
	Bibliography	53

List of Figures

1.1	RaPTEX Plan of Work	3
2.1	Multiple paths for acoustic waves from transmitter to receiver.	9
4.1	Spreading Losses	18
4.2	Tapped Delay Line Model for simulating a frequency selective channel . . .	27
4.3	Sampling rate conversion block diagram	29
5.1	System Block Diagram	33
5.2	A block diagram of the Embedded System Simulator (ESS)	35
5.3	Packet format for the output of the Embedded System Simulator (ESS) . .	36
5.4	Water Channel Simulator Block Diagram	37
5.5	Block Diagram of the Receiver Simulator	40
5.6	Visualization Module GUI	42
5.7	Visualization Module Graph Window	43
6.1	Simulation Speed Comparison for the Channel Simulator	48
6.2	Sample input output waveforms for the Water Channel Simulator(WCS) . .	49
6.3	Sample input output power spectra for the Water Channel Simulator(WCS)	50

List of Tables

6.1 Underwater acoustic channel profiles used for the case study	46
--	----

Chapter 1

Introduction

The progress of science relies on the confluence of its different streams; on the intermingling of its practitioners, pushing each other to advance together for a better understanding of the world we live in. Biology and engineering are two fields which benefit from each other tremendously. Engineers try to emulate the efficient systems found in nature to solve problems they encounter, while biologists benefit from the advances in engineering that give them devices allowing them to observe the dance of life. It is this symbiotic relationship shared by these two fields that serves as a foundation for this thesis.

Observing nature presents unique challenges for the biologist as well as the engineer trying to develop a system for doing so. The biologist has to learn about engineering in order to know what is possible and the engineer has to learn about the biological application in order to design a good system. This thesis forms a part of a larger project aimed at allowing the engineering non-specialists like the biologist to concentrate on the science that they are most comfortable with while allowing the engineer to present multiple viable solutions,

without the non-specialist ever having to delve too deeply into the intricate details of such a design.

1.1 The RaPTEX Project

RaPTEX[7] stands for Rapid Prototyping Tool for Embedded Communications Systems. The goals of RaPTEX are to design and implement a framework for developing embedded communication systems.

There exist many communication protocols for almost every conceivable communication need. However, trying to find the right protocol stack is extremely difficult, especially for researchers from other fields who just want to employ an embedded communication system for their research. There are trade-offs to be made and deciding on the most optimal configuration needs an extensive background in the design of communication systems.

RaPTEX aims to simplify this process by providing a tool which can be used to estimate the performance of various protocols for different conditions. The tool would perform cycle accurate simulations for a variety of protocols running on a different microcontrollers.

RaPTEX will have a Graphical User Interface (GUI) which would allow the researcher to develop an embedded communication system and obtain performance estimates. The backend of the GUI will consist of the GCC tool chain with compiler backend Thrint for optimizations like Software Thread Integration (STI) [8], static timing analysis and power estimation, microcontroller simulators, and communication channel simulators.

The microcontroller and communication channel simulators are an important part of the project since it will enable the researcher to estimate the relevant performance pa-

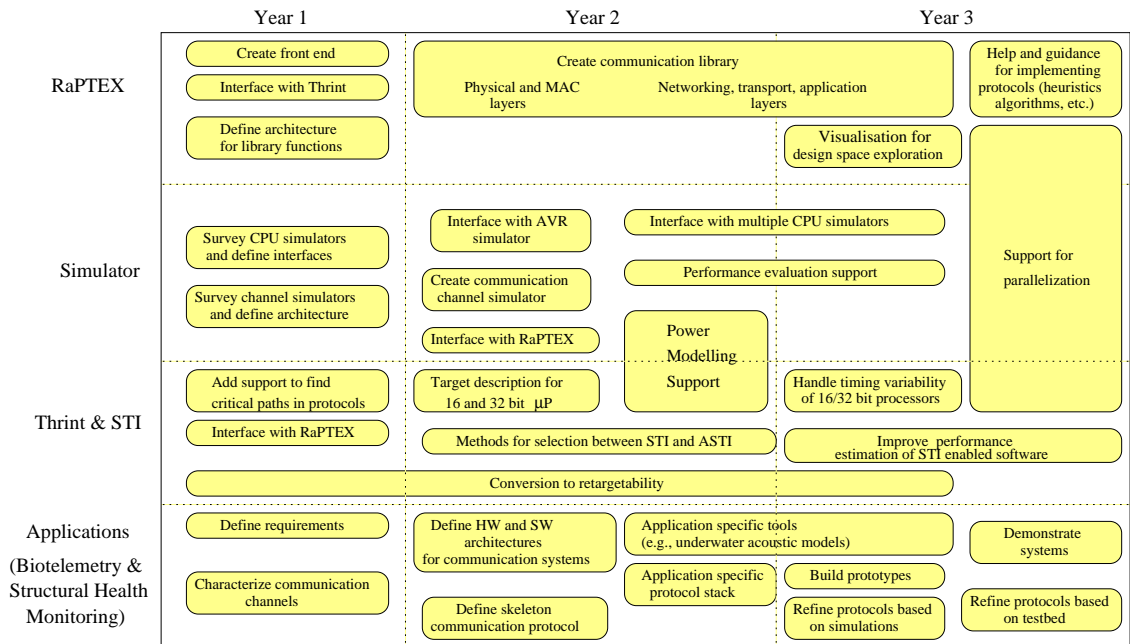


Figure 1.1: RaPTEX Plan of Work[7]

rameters, the resource requirements and allow for debugging of the communication systems.

The scope of this thesis as a part of the RaPTEX project can be defined as the work on the Simulator encompassing the development of the Communication Channel Simulator along with the definition of its architecture and interfaces and the interface with the AVR microcontroller simulator.

The RaPTEX project is being targeted at two applications 1) The study of crabs using acoustic biotelemetry and 2) The health monitoring of bridges using wireless sensors. These two applications will serve as an evaluation platform for the RaPTEX project.

It should be noted that the purpose behind targeting these applications is to demonstrate the utility of RaPTEX for a variety of purposes and should not be construed as narrowing the scope of this project to the exclusion of other applications. Apart from

-serving as a demonstration of the versatility of RaPTEX, these applications also provide a useful focus for the entire project.

1.2 Studying crabs using acoustic biotelemetry

The study of crabs, particularly the blue crab, *Callinectes sapidus*, is well-suited for biotelemetry. The crabs are robust enough to carry transmitters, allowing for a study of various physiological and behavioural parameters.

The work of Wolcott[5] and others demonstrates that the transmitters embedded on the crabs should weigh less than 2% of the body weight of the crabs in order to minimize side effects, like changed behaviour and increased energy expenditure. This places constraints on the amount of electronic circuitry that can be built into the transmitter, forcing the use of a smaller battery. This in turn, requires the transmitter to be highly power efficient, allowing the transmitter to be used for a meaningful amount of time for observation.

It is not sufficient to only have a power efficient transmitter circuit. It is also necessary to have the proper communication protocol stack to ensure that the maximum possible amount of data is transmitted before the battery runs down. This is precisely where RaPTEX would play its part by providing various options regarding the protocols to be used, along with the power and resource requirement analyses, thus enabling researchers to design systems which work within a stringent power and energy budget.

With such low power transmitters, ultrasonic acoustic waves cannot travel more than 1 km with standard designs due to various effects that affect acoustic waves as described

in the following chapter. Consequently, the researcher has to follow the crabs in order to collect data. This is highly time consuming and results in much lesser amount of data collected. A solution to this problem is to embed data loggers with receivers on the bottom of the water body, forming a network. Such a network then allows for the precise tracking of the crabs. The researchers can then retrieve the data at their own convenience by accessing the data loggers via satellite or by visiting them.

RaPTEX would be ideal for simulating such a network with multiple nodes, since RaPTEX aims to provide multiple options for protocols at every layer of the OSI stack from the physical layer to the network layer.

Chapter 2

The Problem

This chapter aims to define the problem that that was set out to be solved. It explores the reasoning behind the thesis and the factors that were taken into account when attempting a solution.

In the final section of this chapter, a tool is proposed which can help considerably in solving the problem.

2.1 A statement of the problem

Water does not provide a good medium for the transmission of electromagnetic waves. The reason for this is twofold. Water, as present in most of the water bodies in nature and especially in the oceans, is highly conductive. This leads to the quick dissipation of electromagnetic radiation at any frequency. The second reason is that electromagnetic radiation at the wavelengths used for RF communication or the wavelengths of visible light are easily scattered by the particles present in the water. As a result of this, it becomes

necessary to explore some other forms of transmission.

Acoustic waves provide a good method for accomplishing underwater communication. Since, acoustic waves are mechanical vibrations set up in the medium; they do not suffer from dissipation due to the conductivity of the water like electromagnetic waves. The wavelengths of the acoustic waves used for underwater communication are also such that particles in the water do not give rise to significant scattering. The propagation characteristics of acoustic waves are thus much more suited to the purpose of underwater communication than electromagnetic waves.

Acoustic waves travel at around 1500 m/s in water and can be used for communication over hundreds of kilometres underwater. These properties of acoustic waves make them an ideal candidate to be used for underwater telemetry. [16]

Designing a system to estimate the location of crabs in a water body is an application of underwater telemetry and hence, uses acoustic waves to accomplish the goal.

2.2 Considerations in underwater acoustic communication

Although acoustic waves are well suited for transmission through water, there exist a few considerations which need to be taken into account when designing a system based on the transmission of acoustic waves through water.

These considerations can be enumerated as follows:

2.2.1 Propagation Losses

The attenuation of acoustic waves in water can be attributed to three major factors, viz. spreading losses, attenuation losses and scattering losses.

Spreading losses also known as divergence losses take place because the amount of energy per area decreases as the wave propagates further and further away from the source. This process is just an extension of the principle of conservation of energy. The spreading losses are more significant near the source.[16]

Absorption losses take place because of the dissipation of energy in the water. The reasons for this dissipation of energy are the viscosity of the water and molecular relaxation. Molecular relaxation of Magnesium Sulphate ($MgSO_4$) and Boric Acid ($B(OH)_3$) is the chief contributor to the attenuation of sound underwater. [16, 18]

Scattering Losses are caused by the scattering of energy by bubbles of gas generated from underwater vegetation or surface actions like motion of ships. Scattering losses can also be caused by marine bodies like shoals of fish, plankton, etc. [18]

2.2.2 Multiple Paths

When acoustic waves travel through water they follow many distinct paths. The source of these distinct paths is the reflections that take place from the bottom of the water body and the water-air interface. The result is that the receiver receives a signal from the direct path i.e. without any reflections and that signal is then followed by echoes of diminished amplitudes based on the number of reflections that each path has gone through.

The number of paths contributing significantly to the signal varies widely from

situation to situation. However, these paths generally lead to a degradation of the signal. This degradation can be of two types. For signals whose duration is less than the duration between the arrivals of two different paths at the receiver, the effect is of multiple echoes following the main signal. For signals whose duration is longer than the duration between inter-arrival times of different paths, an interference pattern results.[16]

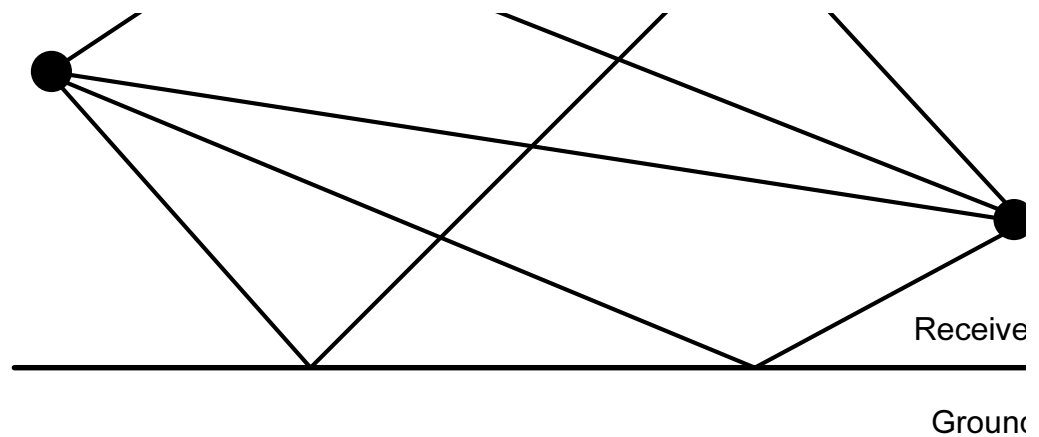


Figure 2.1: Multiple paths for acoustic waves from transmitter to receiver.

2.2.3 Noise

There are many different sources of underwater noise that can interfere with the acoustic waves used to transmit telemetry data and cause a degradation of the signal.

Ambient noise in the high frequency region, is chiefly the effect of two processes.

Surface Agitation which is caused by wind dissolves small amounts of air in the water at the air-water boundary. This dissolved air creates bubbles which burst due to the motion of the surface of the water. The reason for this motion is, once again, the action of wind on the water surface. The bursting of these bubbles gives rise to noise.[16]

Thermal Noise is caused by the random motion of molecules. This type of noise

begins to dominate over the surface agitation noise as the frequency increases and is the chief source of noise over 100 kHz.[25]

Intermittent Noise in the high frequency region can be attributed to the following factors.

Snapping Shrimp cause a lot of noise when they snap their claws. This is the chief source of intermittent noise in the high frequency region in water bodies where these shrimp are found.[25]

Rain causes noise because of the impact of the rain drops on the surface of the water and the oscillations that are set up on the bubbles formed out of the air carried below the water surface by the rain drops. This noise can dominate the high frequency region if the rate of rainfall is high enough.[25]

2.3 Requirement for a simulation environment

As stated earlier, the main goal of the RAPTEX project is rapid prototyping of communication protocols for embedded systems. An important step in achieving this goal is to recognise that, the physical level protocols need optimization when the various phenomena that come into play for a particular medium are taken into account.

When exploring the design space during the development of a system, it is important to evaluate the different designs before finalizing on the implementation. A major part of this evaluation resides in being able to extensively test a particular design.

In the case of underwater acoustic communication, the considerations that need to be taken into account when evaluating designs were enumerated in the previous section.

It is obvious that a water body environment is necessary to carry out such testing. This places restrictions on the amount and scope of the testing that can be carried out.

A simulation environment, as developed for this thesis, goes a long way in complementing the physical testing that needs to be carried out in a water body. It helps in weeding out simpler design errors and serves as a good estimation for the conditions that are encountered in the real world.

After incorporating data about the target environment, which in this case is a water body, into the simulator, the simulator can significantly cut down the number of trips that need to be made to the target environment.

Thus, a simulation environment provides significant advantages for the design of underwater acoustic communication systems and allows the designer to concentrate on optimizing the protocols. In the case of the RaPTEX project, it will help in evaluating the performance of the system being prototyped.

Chapter 3

Related Work

This chapter deals with the related work that has already been done on the subject of this thesis. It aims to detail the major components on which the work of this thesis is based. These components are Avrora, which is a microcontroller simulator and IT++ which is a signal processing library. Both these components were extensively modified and used for the purposes of this thesis.

Also noted in the last section of this chapter are some related papers and books which deal with the process of simulation of communication systems and in particular with underwater communication systems. These served as a starting off point for the research conducted herein.

3.1 Avrora - AVR Simulator

Avrora [24] was designed to be a fast, highly scalable, cycle accurate simulator to simulate sensor networks of up to 10,000 nodes. It was designed to be used with a variety

of AVR based microcontrollers with most of the on-chip devices supported.

The Avrora simulator allows new "devices" externally connected to the microcontroller and other devices on a "platform". This ability of the simulator is highly desirable when work on different configurations while developing simulations for embedded systems.

The speed achieved by the Avrora simulator can be attributed to the event queue model that is used. Sensor nodes and other power constrained embedded systems, make use of the sleep mode in the AVR microcontroller to conserve power. This sleep mode can be exited only by a timer based interrupt either on the microcontroller or on the platform. Each such interrupt is put in the event queue. This allows the simulator to simulate a lot of time with very little processing, needing only to handle the time based interrupt at the head of the queue when required.

This speedup translates to an AVR microcontroller running at around 25 MHz on 3.06 GHz Intel Xeon processor based system. Extrapolating from that, the Avrora simulator is capable of almost real time simulation for an AVR microcontroller running at around 20 MHz. The Avrora simulator performs much better than previous cycle accurate AVR microcontroller simulators, in cases going up to twice as fast. [24]

The Avrora simulator is written in the Java programming language and the extensibility stems from the well documented and clean interface classes. This allows for custom devices to be attached externally to the microcontroller. These devices can be synchronized using the clock used by the microcontroller. This allows a lot of flexibility as regards to the development of simulation systems based on the Avrora simulator.

The use of Java also gives the advantage of platform independence in running the

simulations. A very important effect of this choice on the part of the developers of Avrora, is that it allows future work, like the simulator implemented for this thesis, to achieve the same platform independence and not be tied to a particular operating system.

The Avrora simulator also provides the option of allowing other programs to "connect" to it via TCP/IP sockets. The serial port of the microcontroller is connected to the TCP/IP socket and the data and commands can be fed to the embedded system and also received from it. This allows outside programs to interact with the simulator without changing any of the source code.

Some other features of the Avrora simulator are given below:

- Energy consumption analysis
- Stack performance analysis
- Profiling utilities
- Monitoring capability for all internal registers
- Monitoring of all interrupts
- Program trace

The Avrora simulator however does not simulate the effect of clock drift caused by temperature, aging and power supply to the embedded system.

3.2 IT++ - A Signal Processing Library

IT++ [20] is an open source library written in C++ which provides functionality for mathematics, signal processing, speech processing and communication. The library is

based on vector and matrix template classes.

IT++ makes use of several open-source library packages provided by Netlib.¹. Notable among them are BLAS (Basic Linear Algebra Subprograms) [15], CBLAS (C interface for BLAS) and LAPACK (Linear Algebra Package)[1] and also the FFTW library [13].

IT++ was originally a project of the Information Technology department of the Chalmers University of Technology, Gothenburg, Sweden.

The IT++ library has many features for signal processing like filtering and filter design functions, and functions for various transforms like FFT, DCT, DFT, etc. There are also other classes which handle protocol simulation, speech processing and basic mathematical operations on vectors and matrices.

In the realm of communications it has COST 207, COST 257 and ITU channel profiles which are pertinent to this thesis. They are communication channel profiles for multipath channels. These classes and their support classes were extensively modified and then used for the simulating underwater acoustic channels.

3.3 Other related work

Simulation of multipath fading for underwater acoustics is a well studied field and the work by Falahati, Woodward and Bateman [10] describes how a channel model simulation can be carried out. Other papers [9, 2] also deal with simulation of underwater acoustic channels for communication .

A method of simulating multipath fading channels using tapped delay lines is

¹Netlib is a repository of software for scientific computing. (<http://www.netlib.org>)

given in the work of Silva, Suoto, et al.[23]. Although it deals with RF communication, the principles behind simulation of fading channels remain the same.

The work of Law, Yeung, Bradbeer and Wu[14] explores the use of multicarrier modulation of underwater acoustic communication.

Considerations regarding limitations of the acoustic channel and the challenges in the design and analysis of systems for such acoustic channels can be found in the works of Catapovic and others.[3, 4].

Chapter 4

Theory

This chapter deals with the theoretical basis for the simulation of various phenomena that affect the propagation of acoustic waves in underwater environments. The first three sections deal with propagation losses, noise and multipath fading.

The last section of this chapter deals with the theory behind sampling rate conversion. This section pertains to one of the features of the simulation system allowing the use of data gathered from the field, which may be sampled at different rates.

4.1 Modelling of Propagation Losses

When an acoustic wave travels through water it encounters a loss in intensity because of two reasons. One of the reasons is *spreading losses* or *divergence losses* while the other reason is that the energy of the acoustic wave is dissipated in the water, which is known as *Attenuation Loss*.

4.1.1 Spreading Losses

Spreading losses are an extension of the law of conservation of energy. If we consider a point source for the acoustic wave, then the envelope of the wave as it travels away from the source can be thought of as an expanding sphere. As a result of this the amount of energy contained within the sphere and travelling outwards stays constant, if we disregard attenuation losses for the moment. Hence, the amount of energy passing through a unit area on the surface of the sphere goes on decreasing as the distance from the source increases.

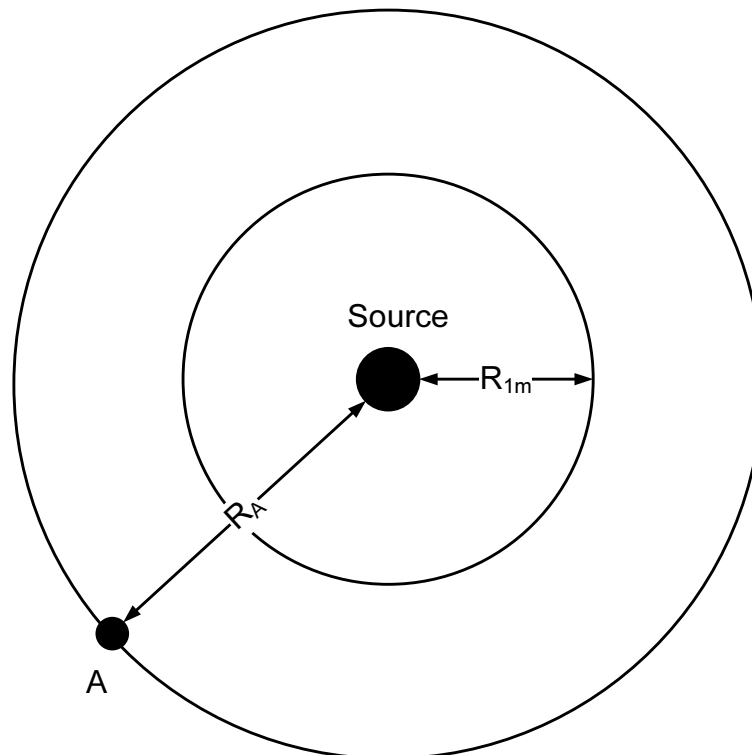


Figure 4.1: Spreading Losses

The average amount of acoustic energy passing through a unit area over a unit

amount of time in a particular direction is defined as the *intensity* of the acoustic wave at a given point. Since, we are considering a uniform sphere, all of the energy will be directed perpendicular to the surface of the unit area. Thus, it will suffice to only consider the magnitude of the intensity.

The amount of energy passing through the unit surface in unit time is nothing but the power flux through the unit area. Also, since the total energy in the sphere at any radius is always constant, the power flowing out of the sphere at any given instant can be written as.

$$P = 4\pi r^2 I \quad (4.1)$$

where,

P is the power flux through the sphere,

r is the radius of the sphere and

I is the intensity of the acoustic wave at radius r

Let I_{1m} be the acoustic intensity at any point at a distance of 1 metre from the point source. Let this distance be represented by R_{1m} . Also, let the acoustic intensity at a point A, which is at a distance R_A from the source, be I_A . Hence, substituting in Eqn. (4.1), we get

$$P = 4\pi R_A^2 I_A = 4\pi R_{1m}^2 I_{1m} \quad (4.2)$$

Rearranging the terms, we get

$$\frac{I_{1m}}{I_A} = \frac{R_A^2}{R_{1m}^2} \quad (4.3)$$

Expressed as spreading transmission loss, in terms of dB we get

$$TL_{spreading} = 20 \log_{10} \frac{R_A}{R_{1m}} \quad (4.4)$$

4.1.2 Absorption Losses

Absorption losses in water bodies are caused by two dominant factors, viz. 1) Viscosity of pure water as regards to both shear and bulk viscosity and 2) Chemical molecular relaxation.[16, 17]

Viscosity of pure water contributes to absorption losses is due to physical relaxation. This just means that the fluid takes some time to return to its normal state after the acoustic pressure wave has passed. This time is called as the *relaxation time* and the process is referred to as *relaxation*. [17]

Chemical relaxation occurs because of the dissociation of ionic compounds like Boric Acid ($B(OH)_3$) and Magnesium Sulphate ($MgSO_4$). The relaxation time in this case refers to the time required by the molecules to recombine. The ionic dissociation is activated by a condensation and relaxation occurs during a rarefaction. The relaxation time depends on the temperature and pressure of the water which in turn depends on the depth at which this process is occurring. [17]

Relaxation frequency is given as,

$$f_r = \frac{1}{2\pi\tau_r} \quad (4.5)$$

where τ_r is the relaxation time.

At very low frequencies of the acoustic wave, the molecules relax during the condensation phase itself, causing constructive interference whereas in the case of very high frequencies the molecules don't dissociate because the condensation passes by too quickly. However, when the frequency of the acoustic wave is near the relaxation frequency of the molecule, then the molecule is activated during the condensation phase and relaxes during the rarefaction phase. This causes destructive interference, causing attenuation. [16, 17]

The work of Fracois and Garrison[11, 12] gives us a model that can be used to calculate the absorption, α at a given frequency of the acoustic wave, based on various parameters.

The model can be given as follows

$$\alpha = \frac{A_1 P_1 f_1 f^2}{f^2 + f_1^2} + \frac{A_2 P_2 f_2 f^2}{f^2 + f_2^2} + A_3 P_3 f^2 \quad (4.6)$$

where α is the absorption coefficient in dB/km. The other coefficients are described in terms of D (depth in metres), T (temperature in °C), S (salinity in practical salinity units or parts/1000) and the relaxation frequencies of Boric acid, f_1 and Magnesium Sulphate, f_2 in kHz.

Pure water contribution:

For $T \leq 20^\circ\text{C}$,

$$A_3 = 4.937 \times 10^{-4} - 2.59 \times 10^{-5}T + 9.11 \times 10^{-7}T^2 - 1.50 \times 10^{-8}T^3 \text{ dB km}^{-1} \text{ kHz}^{-1} \quad (4.7)$$

For $T > 20^\circ\text{C}$,

$$A_3 = 3.964 \times 10^{-4} - 1.146 \times 10^{-5}T + 1.45 \times 10^{-7}T^2 - 6.5 \times 10^{-10}T^3 \text{ dB km}^{-1} \text{ kHz}^{-1} \quad (4.8)$$

Pressure dependence factors

$$P_1 = 1, \quad (4.9)$$

$$P_2 = 1 - 1.37 \times 10^{-4}D + 6.2 \times 10^{-9}D^2, \quad (4.10)$$

$$P_3 = 1 - 3.83 \times 10^{-4}D + 4.9 \times 10^{-9}D^2 \quad (4.11)$$

Magnesium Sulphate contribution

$$f_2 = \frac{8.17 \times 10^{(8 - \frac{1990}{\theta})}}{1 + 0.0018(S - 35)} \text{ kHz}, \quad (4.12)$$

$$\theta = T + 273, \quad (4.13)$$

$$A_2 = 21.44 \left(\frac{S}{c} \right) (1 + 0.025T) \text{ dB km}^{-1} \text{ kHz}^{-1}, \quad (4.14)$$

$$c = 1412 + 3.21T + 1.19S + 0.0167D \text{ m/s} \quad (4.15)$$

where c is the speed of sound in water.

Boric Acid contribution

$$f_1 = 2.8 \left(\frac{S}{35} \right)^{0.5} \times 10^{(4-1245/\theta)} \text{ kHz} , \quad (4.16)$$

$$\theta = T + 273, \quad (4.17)$$

$$A_1 = \frac{8.86}{c} \times 10^{(0.78pH-5)} \text{ dB km}^{-1} \text{ kHz}^{-1} , \quad (4.18)$$

$$c = 1412 + 3.21T + 1.19S + 0.0167D \text{ m/s} \quad (4.19)$$

where c is the speed of sound in water.

4.2 Noise Modelling

4.2.1 Ambient Noise

The modelling of ambient noise is based on the work done at the University of Washington Applied Physics Laboratory (UW-APL)[19] for ambient noise due to surface agitation whereas the modelling for ambient thermal noise is based on the model given by Urlick[25].

Surface Agitation Noise

The noise caused by the bursting of bubble of dissolved air at the air-water interface, gives rise to noise which is chiefly dependant on wind speed.

$$NL_{wind} = 20.5 + 22.4 \log U \quad \Delta T < 1^\circ C \quad U \geq 1 \text{ m/s} \quad (4.20)$$

$$NL_{wind} = 20.5 + 22.4 \log U - 0.26(\Delta T - 1.0)^2 \quad \Delta T \geq 1^\circ C \quad U \geq 1 \text{ m/s} \quad (4.21)$$

where $\Delta T = T_{air} - T_{sea}$, U is the wind speed in metres/second at a reference height of 10 metres above the surface of the water and NL_{wind} is the ambient noise level in dB re $1\mu\text{Pa}$ due to the effect of the wind. The equation with the temperature difference term is important only in windy coastal areas and enclosed seas in summer.

The frequency dependant ambient noise level due to surface agitation can hence be given as

$$NL_{surf} = NL_{wind} + 20.7 - 15.9 \log f \quad U \geq 1 \text{ m/s} \quad (4.22)$$

where f is in kHz and NL_{surf} is in dB re $1\mu\text{Pa}$.

Thermal Noise

The noise due to the thermal excitation of the medium can be given as

$$NL_{thermal} = -15 + 20 \log f \quad (4.23)$$

where f is in kHz and $NL_{thermal}$ is in dB re $1\mu\text{Pa}$.

4.2.2 Intermittent Noise

Intermittent noise as the name suggests is not always present, but needs to be factored in because of the relatively high noise levels produced by these intermittent noise sources.

Noise due to Snapping Shrimp

There is no theoretical model for modelling the noise due to snapping shrimp. Moreover, there is no data collected by direct observation for frequencies above 50 kHz. However, we can extrapolate using the curves available in the work of Urick[25].

These curves have a slope of -25 dB/decade in the region of 10 - 50 kHz. Thus, the curves due to various observations give us an empirical extrapolated formula as,

$$NL_{shrimp} = 115 - 25 \log f \quad f \geq 10kHz \quad (4.24)$$

where f is in kHz and NL_{shrimp} is in dB re $1\mu\text{Pa}$.

Rain Noise

The noise created by water droplets striking the water surface create noise through two mechanisms, viz. the impact and the oscillations of the air bubbles that are created when the water droplet carries some air when it impacts the water. It was found that the chief source of noise is the oscillations of the air bubbles.

Consequently, the noise level for rain is a function of the size and velocity of the water droplets when they hit the water surface. For rain both of these depend on a single parameter which is the rate of rainfall.[25]

The noise due to rain is a combined effect of the wind speed and the rain rate.

This noise can be given as,[19]

$$NL_{rain} = b + a \log RR \quad RR \geq 10^{-2} mm/hr \quad (4.25)$$

$$a = 25.0, \quad U \leq 1.5 \quad (4.26)$$

$$= 5.0 + 5.7(5.0 - U), \quad 1.5 < U < 5.0$$

$$= 5.0 \quad U \geq 5.0$$

$$b = 41.6, \quad U \leq 1.5 \quad (4.27)$$

$$= 50.0 + 2.4(5.0 - U), \quad 1.5 < U < 5.0$$

$$= 50.0 \quad U \geq 5.0$$

where RR is the rain rate in millimetres/hour, U is the wind speed in metres/second and NL_{rain} is in dB re $1\mu\text{Pa}$.

4.3 Fading Channel

Frequency Selective Channels are those channels for whom the propagation delays between the acoustic waves reaching the receiver after having followed different paths cannot be ignored. This happens because the propagation delays are significant as compared to the symbol time of the communication system.

Such a frequency selective channel can be modelled as a tapped-delay-line (TDL) model. A tapped-delay-line model is simply a filter with time varying coefficients.

Since, there will be a strong line-of-sight (LOS) component in the underwater acoustic communication system, the choice of using the Rice process for obtaining the

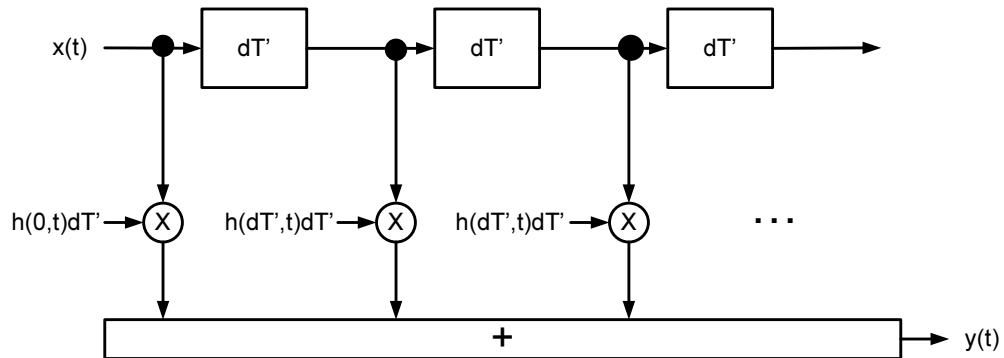


Figure 4.2: Tapped Delay Line Model for simulating a frequency selective channel[21]

coefficients for the TDL was made. In the TDL channel model the first tap is used to model the LOS component.

The Rice process is simulated by using the Rice Method. It uses a finite number, N_i of harmonic functions. The deterministic Rice process is given by[21]

$$\tilde{\xi}(t) = |\tilde{\mu}_\rho(t)| = |\tilde{\mu}(t) + m(t)| \quad (4.28)$$

$$\tilde{\mu}(t) = \sum_{n=1}^{N_i} c_{i,n} \cos(2\pi f_{i,n}t + \theta_{i,n}) \quad (4.29)$$

where $m(t)$ is the LOS component, and $\tilde{\mu}(t)$ is the component due to multiple paths. The parameters, $c_{i,n}$, $f_{i,n}$ and $\theta_{i,n}$ are called as the Doppler coefficients, discrete Doppler frequencies and Doppler phases respectively. These parameters are calculated

using the Method of Exact Doppler Spread(MEDS) as follows,[21]

$$c_{i,n} = \sigma_0 \sqrt{\frac{2}{N_i}} \quad (4.30)$$

$$f_{i,n} = f_{max} \sin \left[\frac{\pi}{2N_i} \left(n - \frac{1}{2} \right) \right] \quad (4.31)$$

$$f_{max} = \frac{v}{c} f_c \quad (4.32)$$

where f_{max} is the maximum Doppler frequency, v is the relative velocity between the transmitter and the receiver, c is the speed of sound in water, f_c is the carrier frequency and $\theta_{i,n}$ is a random variable with a uniform distribution in the interval $(0,2\pi]$.

The tap spacing is given by,

$$T_s = \frac{1}{f_s} \quad (4.33)$$

where f_s is the sampling frequency and the first tap simulates the LOS component with no delay.

4.4 Sampling Rate Conversion

Sampling rate conversion takes place in the receiver module. The receiver module is capable of handling input from sources with different sampling rates because of this sampling rate conversion. This is needed because the bandpass filters for the carriers in the receiver have been designed for a specific sampling rate.

The sampling rate conversion by a rational factor (L/D) where L is the upsampling

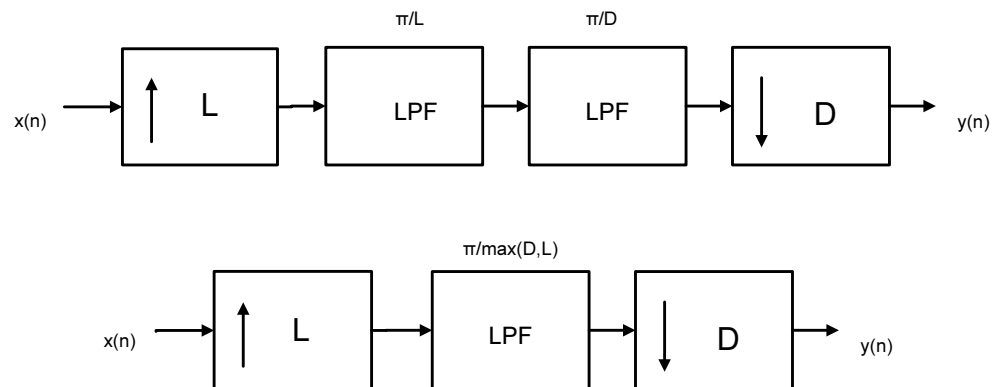


Figure 4.3: Sampling rate conversion block diagram[6]

factor and D is the downsampling factor. This leads to a sampling rate $F_y = (L/D)F_x$.

In the process of decimation or downsampling by a factor of D , we choose every D^{th} sample. In this case all frequencies in the original signal above π/D cause aliasing. Hence, it is necessary to filter out these higher frequencies using a low pass filter with a cutoff frequency of π/D before the downsampling takes place[6].

In the process of interpolation or upsampling by a factor of L , we add $(L-1)$ zeros between two samples of the original signal. This causes image frequencies above the frequency of π/L . Hence, it is necessary to filter out these image frequencies using a low pass filter with a cutoff frequency of π/L after the upsampling takes place[6].

When converting the sampling rate by a rational factor of (L/D) we perform the downsampling after the upsampling because downsampling results in a loss of information. Since the two low pass filters are now in series with each other we can combine the two low pass filters into a single filter with a cutoff frequency given by $\pi/\max(D, L)$ [6].

4.4.1 Calculation of rational factors

The rational factors L and D are calculated using a recursive algorithm till the required precision is reached. This is required because L and D need to be approximated from a floating point number, where L and D are reasonably small integers.

The algorithm ¹ can be expressed as

$$\frac{L}{D} = d_1 + \frac{1}{d_2 + \frac{1}{\left(d_3 + \dots + \frac{1}{d_k}\right)}} \quad (4.34)$$

where $d_1, d_2 \dots d_k$ are obtained by recursively assigning the integer part in a given iteration to a 'd' and passing the reciprocal of the fractional part to the next iteration.

The algorithm can exit when a desired precision is reached.

4.4.2 Design of FIR low pass filters

Since this low pass filter has to be designed dynamically we calculate the coefficients of an FIR filter using a rectangular window. The unit sample response of such a filter can be given as,[22]

$$h(n) = \frac{\sin \omega_c \left(n - \frac{M-1}{2}\right)}{\pi \left(n - \frac{M-1}{2}\right)} \quad 0 \leq n \leq M-1 \quad n \neq \frac{M-1}{2} \quad (4.35)$$

where M is the length of the filter.

¹This algorithm is used in the *rat* and *rats* MATLAB functions and was detailed in the MATLAB function reference. (<http://www.mathworks.com>)

If M is odd then the unit impulse response is,

$$h(n) = \frac{\sin \omega_c n}{\pi} \quad n = \frac{M-1}{2} \quad (4.36)$$

A rectangular window gives a sharper transition band but a little bit of ringing in the pass band because of *Gibb's phenomenon*. On the other hand, other windows like Hamming, Hanning, etc. eliminate the ringing but at a cost of a wider transition band.

Chapter 5

Implementation

This chapter starts out with the applicability of the theoretical matter presented in the previous chapter. It gives an overview of the implementation for the simulation system and then proceeds to describe in detail each of the modules that make up the simulation system.

5.1 Overview of the implementation

The simulation system aims to simulate the whole communication system right from the embedded system which generates the signal, to the water channel through which the acoustic waves pass through, to the receiver. It can be said that the embedded system simulator is the producer of the data, the water channel simulator processes the data and the receiver is the consumer of that data.

The simulation system consists of

- The *Embedded System Simulator (ESS)* which has at its heart the Avrona AVR

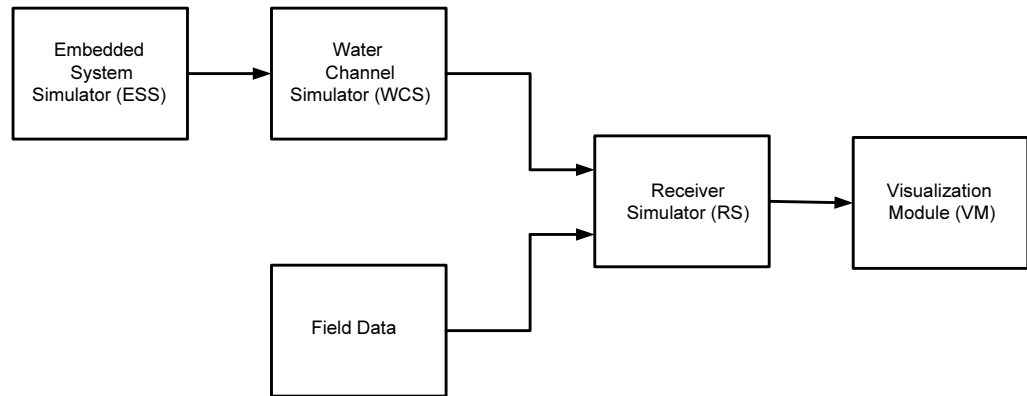


Figure 5.1: System Block Diagram

microcontroller simulator along with extensions to it forming a platform that is used to generate and transmit an acoustic signal. This signal then passes on to the analog channel simulator.

- The *Water Channel Simulator (WCS)* is, by design and necessity, based on mathematical models of physical phenomena that occur in the transmission of ultrasonic acoustic waves travelling through water. As a result, the channel simulator is the core of the simulation system. It processes the data received from the embedded system simulator, passing it through various modules internally to simulate the different effects that water has on an acoustic wave travelling through it. The data is then passed on to the receiver simulator.
- The *Receiver Simulator (RS)* receives the data from the water channel simulator and separates out the multiple carriers in the signal, demodulates each of the carriers to retrieve the data and then passes on the separated carriers and the demodulated data to the visualization module. The Receiver Simulator is also capable of handling data

from a field data capture unit.

- The *Visualization Module (VM)* is a module capable of displaying a graphical output for each of the carriers passed to it along with the demodulated data on each carrier. This is helpful when trying to analyze the correctness of the demodulated data.

5.2 Embedded System Simulator

The AVR microcontroller is the design choice for the RAPTEX project. Hence, the Embedded System Simulator(ESS) is based on the AVR microcontroller simulator called *Avrora*[24]. The *Avrora simulator* has a number of useful features as mentioned in section 3.1; the most important among them being the ability to extend the simulator to create a platform.

These platforms can be thought of as a circuit board with different devices connected to the microcontroller. For the purposes of the Embedded System Simulator two devices were added on to a platform to create an embedded system. These two devices are the Digital to Analog Converter (DAC) and an ultrasonic transducer to convert the output of the DAC into acoustic waves.

This platform can output a stream of data based on the program being running on it. The platform listens to incoming TCP connection requests on a given port and then streams the data over that connection to whichever program initiates the connection; i.e. this platform behaves like a server waiting for a program to connect to it before starting the run of the program loaded into it. The platform can currently support only one such program connecting to it. The main reason for this approach is to allow for future work in

the simulation system with multiple such platforms and a single channel simulator talking to all of them; thus having a more elaborate simulation scenario.

The Avrora simulator supports three file formats for the input program, which are supported by the ESS. These are the Atmel Assembler syntax (.asm), the GNU Assembler or GAS (.s) and the output of the avr-objdump facility (.od).

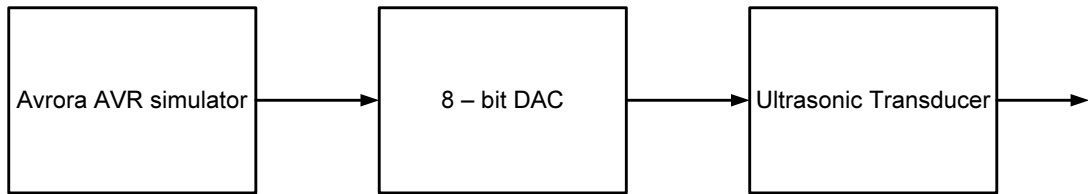


Figure 5.2: A block diagram of the Embedded System Simulator (ESS)

The *Digital to Analog Converter (DAC)* has an 8-bit input and can be connected to any of the general purpose ports of the AVR microcontroller. The DAC is synchronized with the clock used by the microcontroller itself. This allows the DAC to send out data along with the current clock cycle every time data changes at the general purpose port to the transducer. The maximum output voltage for the DAC can be passed a parameter to the DAC.

The *ultrasonic transducer* simulates the conversion of the signal from the electrical domain to the acoustic domain. For every ultrasonic transducer, a parameter known as the Transmission Sensitivity of a ultrasonic transducer defines its "gain". This "gain" can be expressed as,[16]

$$SV = 20 \log \left(\frac{p_{1V}}{p_{ref}} \right) \quad (\text{in dB re } 1\mu\text{Pa}/1 \text{ m}/1 \text{ V}) \quad (5.1)$$

where p_{1V} is the acoustic pressure 1 m away from the transducer, for an electric voltage of 1 V and p_{ref} is the reference acoustic pressure ($1 \mu\text{Pa}$). It is usually given in the direction of the maximum amplitude.

This parameter is used to model the output of the transducer in terms of an acoustic signal given a particular input voltage from the DAC.

The final output of the Embedded System Simulator is the output of the transducer. This output consists of the acoustic pressure levels in μPa at a distance of 1 m from the embedded system, assuming the direction in which the output power of the embedded system is the maximum. Each such acoustic pressure level value is accompanied by the cycle count of the microcontroller when the acoustic pressure level value changed to the current value. This gives a basis for the channel simulator with which it calculates the timing of the acoustic pressure level values based on the knowledge of the operating frequency of the microcontroller.

Hence the format of a given data packet that is the output of the embedded system can be shown as, where 'Acoustic Pressure in μPa ' is a 64-bit double precision

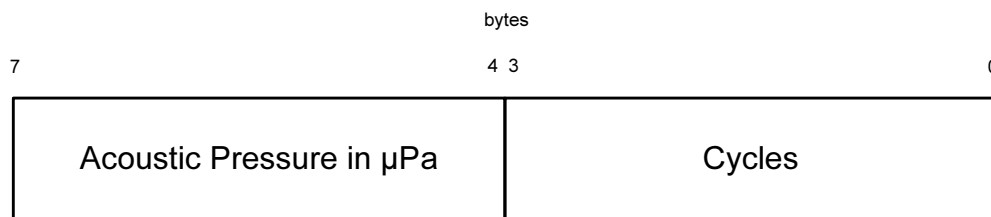


Figure 5.3: Packet format for the output of the Embedded System Simulator (ESS)

floating point value as defined in IEEE 754 and 'Cycles' is a 64-bit long integer.

These packets are then passed on to the Water Channel Simulator or any other

program which connects to this platform at a given TCP port.

5.3 Water Channel Simulator

The Water Channel Simulator(WCS) attempts to simulate the various phenomena that occur when an acoustic wave passes through water. The most important of these phenomena are propagation losses, noise and multipath fading.

The WCS uses the data passed to it by the Embedded System Simulator. This data is nothing but the acoustic pressure levels along with the timing information. The WCS calculates the sampling time period for each of the acoustic pressure level samples passed to it based on the cycle count associated with the sample. This allows the WCS to sample the incoming data at a constant rate of 5 MHz irrespective of the rate at which data is produced by the ESS. This use of a consistent sampling rate for all the Digital Signal Processing operations in the subsequent blocks, greatly simplifies the design of the simulator.

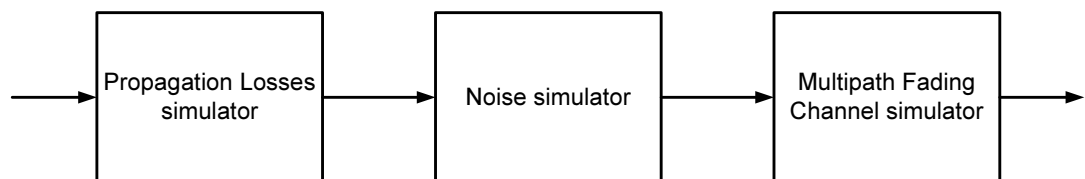


Figure 5.4: Water Channel Simulator Block Diagram

As can be seen from the block diagram, the data from the ESS is first subjected to propagation losses and then to noise before being passed to the multipath fading simulator.

Propagation Losses are a function of various parameters like the distance between

the transmitter and the receiver, the salinity of the water, the pH of the water, the temperature of the water and the depth at which the acoustic wave travels.

The modelling for propagation losses is done based on the set of equations given in section 4.1. This is the first block in the water channel simulator and the different carrier frequencies are separated out using filters and are attenuated as per the propagation loss models.

These different attenuated carriers are then added back into a single signal before passing the signal on to the next block in the Water Channel Simulator.

The level of most of the *noise* in water affecting the transmission of an acoustic wave is a function of the frequency of the acoustic carrier wave. Hence, it becomes necessary to filter White Gaussian noise through a filter and add it to the particular carrier at the required level.

The parameters governing the noise levels are chiefly the temperature of the water, the wind speed over the surface of the water and the rate of rainfall. These can be built up into profiles to signify various conditions using the Beaufort scale for wind and rain.

The explanation for the various losses and the equations governing the noise models can be found in section 4.2. The signal received from the block dealing with the attenuation of the signal is once again split into the component carrier frequencies before adding the filtered noise at the appropriate level to the signal.

The carriers are then recombined before passing them on to the block dealing with multipath fading.

Multipath Fading is a phenomena governed by very complex models. The basis

of the model is a tapped delay line with Rice processes used to generate the time varying coefficients at each of the taps. The Rice processes are simulated using the Rice Method whose parameters are calculated using the Method of Exact Doppler Spread (MEDS) as mentioned in section 4.3. As we are simulating a Rician fading channel we use the first tap to model the Line of Sight (LOS) component.

The implementation for the simulation of the Rice processes and the Tapped Delay Line channel was done using a modified version of the IT++ library[20]. All the computations within this block are done using complex numbers.

The parameters required for the effective simulation of multipath fading channels are specified using a profile, which specify the normalized maximum Doppler frequency, the Doppler frequency of the LOS component, the relative amplitude of the LOS component and the delays and amplitudes for each of the taps. These parameters can be combined into a convenient profile to be passed to the block handling the multipath fading.

5.4 Receiver Simulator

The Receiver Simulator(RS) consists of three building blocks, viz. the Sampling Rate Converter, the Receiver Filter array and the Demodulator array.

The *Sampling Rate Converter* is the first block in the RS and is responsible for converting the sampling rate of the input data which is received either from a field data capture unit or from the Water Channel Simulator to a particular sampling frequency. The sampling frequency used for this implementation is 5 MHz. The sampling rate converter consists of rational fraction approximation module which tries to compute the upsampling

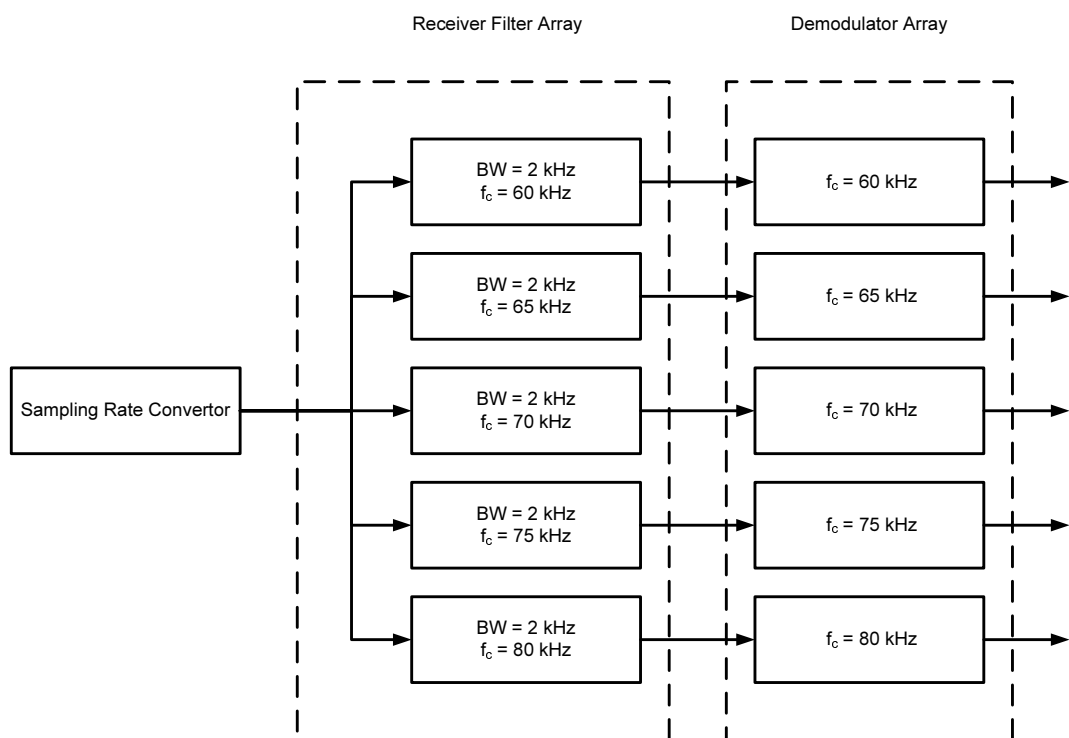


Figure 5.5: Block Diagram of the Receiver Simulator

and downsampling factors, the upsampler and a low pass filter with variable cutoff frequency followed by a downsampler. The upsampler adds $(L-1)$ zeros between two consecutive samples whereas the downsampler throws away every D^{th} sample. The theoretical basis for the sampling rate converter is given in section 4.4

The *Receiver Filter Array* is responsible for splitting the received signal into the five carrier frequencies at 60, 65, 70, 75 and 80 kHz. The data is then passed on to the Demodulator Array. The filters are elliptic IIR filters of order 6 with a bandwidth of approximately 2 kHz centred around the carrier frequencies. The filters were designed using the sptool in MATLAB.

The *Demodulator Array* can be of any kind based on the modulation scheme employed. The interfaces for this block are kept simple in order to facilitate the easy substitution of the default demodulator with a more sophisticated one. The default modulation scheme used is Amplitude Shift Keying (ASK).

The demodulated binary data and the individual carriers modulated with the data are the outputs of the Receiver Simulator which are then passed on to the Visualization Module.

5.5 Visualization Module

The Visualization Module (VM) is intended as a tool to quickly evaluate the correctness of the demodulated data and can aid in adjusting the receiver for optimum performance. These adjustments include but are not limited to changing the noise threshold, increasing or decreasing the bandwidth of the receiver filters, etc.

The VM consist of a Graphical User Interface (GUI) which can be launched as an individual application or can be launched by specifying a command line switch to the Receiver Simulator. This approach allows the data generated by the receiver to be immediately viewed using the VM or the VM can display data from a file which has been generated by the RS and has been loaded into the VM using the GUI to display the data.

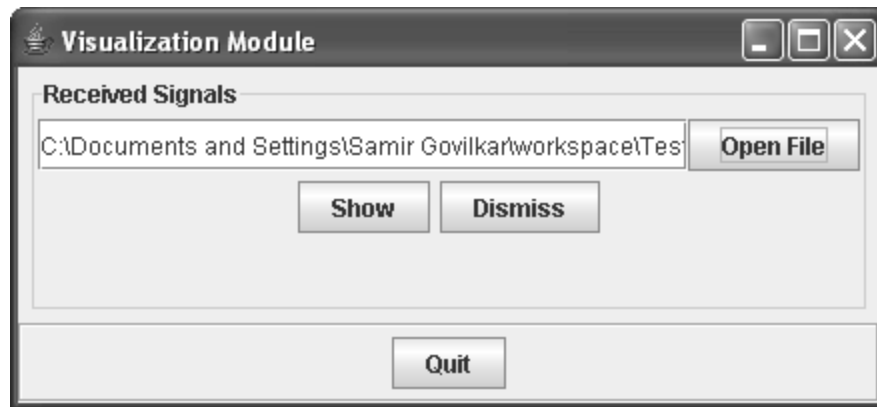


Figure 5.6: Visualization Module GUI

The VM is flexible enough to display multiple 2D graphs just by passing it a variable length array of objects that represent a two dimensional array of points for the graphs. The Visualization module can also read in information directly from a file whose filename is passed to it at the command line, through the RS or through the file selection option on the VM GUI. The JPlot ¹ library made by Dr. Lee was used to display the graphs.

The VM takes advantage of the fact that the carriers are highly oversampled. Although, the carriers lie within the range of 60 to 80 kHz they have been sampled and processed at 5 MHz. For display purposes such a high sampling rate is unnecessary. Hence,

¹Dr. Jan van der Lee, Ecole des Mines de Paris - Centre de Géosciences Equipe Hydrodynamique et Réactions - Reactive Hydrodynamics Group, <http://jplot.sourceforge.org>

we resample all the carriers at a sampling rate of 500 kHz which gives a reasonably good approximation of the carriers and the modulated data. This allows the VM to cut down on the data to be displayed by 10 times. Apart from reducing the file loading and processing times, it results in a faster display of the graphs. This feature is especially useful when multiple simulation results need to be evaluated consecutively.

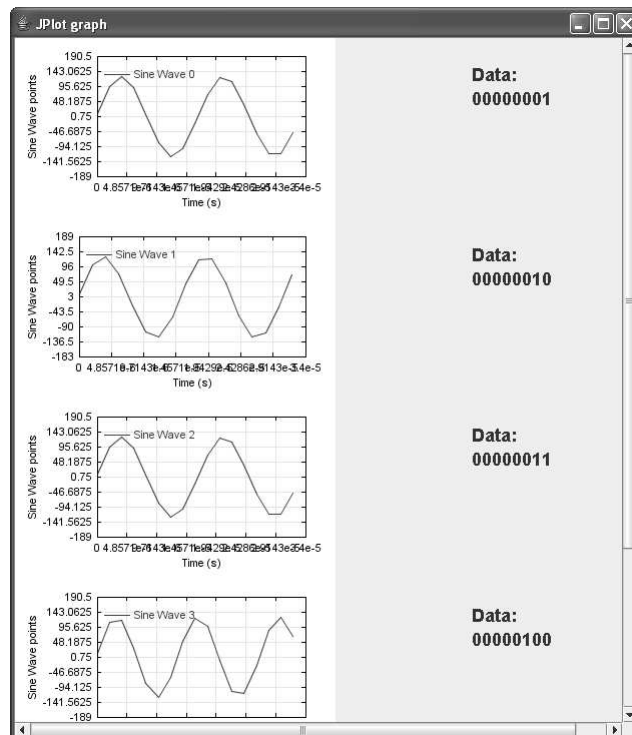


Figure 5.7: Visualization Module Graph Window

Thus, the VM which is the final module in the simulation system terminates the data flow that began in the ESS and passed through the WCS and the RS.

Chapter 6

Case study

This chapter deals with a particular simulation using Amplitude Shift Keying as the modulation scheme. The purpose of this chapter is to illustrate how to set up the simulation system and obtain results.

6.1 Amplitude Shift Keying Theory

Amplitude Shift Keying (ASK) is a simple modulation scheme that uses the amplitude of the carrier to code the binary data. A special case of ASK is On-Off Keying (OOK) which uses just two levels to signify a binary '1' and '0'. Typically these amplitude levels are the presence and absence of a carrier to signify binary '1' and '0' respectively.

OOK is spectrally inefficient due to the sharp transitions caused by the abrupt changes of the amplitude of the carrier wave. However, this can be compensated by adjusting the rise and fall rates of the carrier wave. This is possible for the low symbol rate being used for this case study.

OOK is particularly susceptible to noise, which is one reason for choosing this modulation scheme for the case study. As the effects of noise are more pronounced, OOK gives a good idea about performance of the simulation system.

Another reason for choosing OOK was that it is the simplest modulation scheme and the simplicity allows for easier analysis of the results obtained from the simulation system. This in turn allows for easier debugging and testing of the component modules.

6.2 Implementation

The implementation for the transmitter is based on the generation of arrays of the carrier wave values, which can be added together based on whether the carrier is present during a particular symbol period. The symbol period for this case study was chosen to be 20 cycles of the 60 kHz carrier.

The carrier waves are 60-80 kHz sine waves spaced at 5 kHz with a length equal to the time period of 20 cycles of the 60 kHz carrier wave. Thus, these carriers are selectively added before being transmitted.

The transmit routine is based on a loop which cycles through the symbol array till it is interrupted by a timer. It is written in assembly to ensure that the microcontroller writes values to the port connected to the DAC at precisely 5 MHz.

There are various configuration settings that need to be handled in the Water Channel Simulator before doing the simulation run. These are 1) wind speed, 2) rain conditions, 3) water profile, 4) delay and amplitude profiles for the tapped delay line channel and 5) the distance between the Transmitter and the Receiver or range. For this case study,

the wind speed was set to a 'Light Breeze' (1.6-3.3 m/s), the rainfall rate was set to 'Drizzle' (0.25 mm/hr), the water profile was set as follows. The temperature was set to 10 °C, the salinity was chosen to be 'Lake' (0.5 p.s.u.), the pH was set to 'Lake' (7) and the depth of the transmitter in water at 10 metres. The range was set to 100 metres.

The delay and amplitude profiles need to be experimentally setup for each unique environment to get the best possible simulation results. However, for the purposes of demonstration standard profiles were chosen.

Table 6.1: Underwater acoustic channel profiles used for the case study

Underwater 1				
Taps	1	2	3	4
Delay(ms)	0	2	4	6
Power(dB)	0	-20	-30	-40
Underwater 2				
Taps	1	2	3	4
Delay(ms)	0	0.2	0.4	0.6
Power(dB)	0	-20	-30	-40
Underwater 3				
Taps	1	2	3	4
Delay(ms)	0	0.02	0.04	0.06
Power(dB)	0	-20	-30	-40

The demodulator for ASK is a simple integrator which integrates over the symbol period and makes a decision of a binary '1' or '0' based on the noise threshold. Thus, it is important to set the noise threshold before running the simulation.

6.3 Simulation

The procedure for simulation starts off with loading a program compiled for the AVR microcontroller and converted to the appropriate file format into the ESS as mentioned in section 5.2. The ESS output can then be redirected to a file or it can wait for the WCS

to connect to it.

The WCS is then started up in a mode which allows it connect to the ESS, which then goes ahead and processes the data coming out of the ESS. However, the WCS can also be instructed to recorded the incoming data so as to run the WCS later with different profiles without re-running the ESS. This is useful when evaluating the performance of the system under different operating conditions.

Once the WCS completes its run, its output is redirected to a file which can be used by the RS. The RS is then with or without the command line switch to start the VM. If the RS is started without the command line switch to start the VM then the output of the RS is redirected to a file.

The VM, as mentioned earlier, can be started independently of the RS too. In this mode, the output of the RS needs to be loaded into the VM via the GUI to display the output.

6.4 Results

The results in this case study pertain to the speed of the simulation under different operating conditions and with different options.

The 'live' and 'recorded' descriptions in the figure 6.1 refer to the mode in which the WCS is run. When in the 'live' mode the ESS and the WCS are run simultaneously while in the recorded mode, the output of the ESS is stored in a file and is then later used by the WCS for performing simulations. The 'Symbols Only' and 'Full Data' refer to the simulation speeds for certain symbols within the sample data being used as input to the

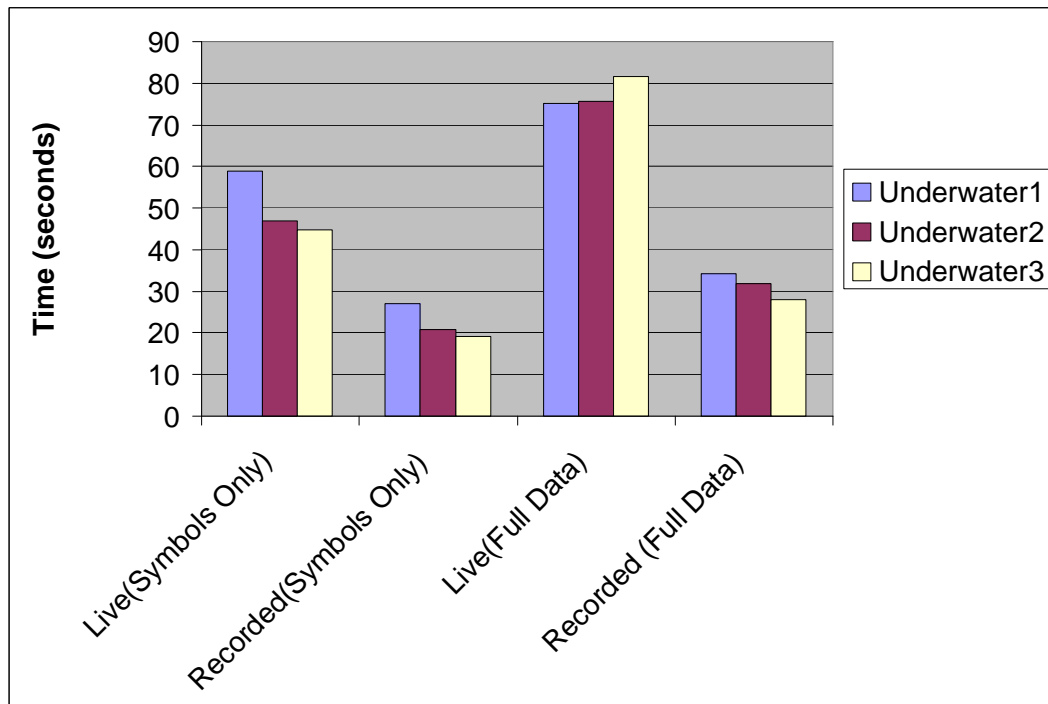


Figure 6.1: Simulation Speed Comparison for the Channel Simulator

WCS and the entire sample data respectively.

As can be seen from the graph 6.1, there is a clear advantage when the 'recorded mode' of the WCS is used. Not only is the simulation faster when it is run independently, multiple simulations can be performed one after another with varying configurations. The difference in the speeds stems from the fact that the WCS does not have to wait for the ESS to produce its output.

The graph6.1 also shows that while a clear correlation exists between the simulation speeds for the three profiles and the computational complexity of the underwater profiles; 'Underwater 1' being the computationally hardest, this correlation is blunted when the 'live mode' of the WCS is used.

Sample input output waveforms and power spectra are shown in figures 6.2 and 6.3 respectively. The plots 6.2 and 6.3 demonstrate how the propagation losses, noise and multipath fading affect the waveform and the power spectrum. In this case the symbols being transmitted are $[1\ 1\ 1\ 1\ 1]$ throughout. These symbols are transmitted in order to get the greatest power on each of the carrier frequencies for the purpose of demonstrating the effects stated above.

For the simulation conditions mentioned in section 6.2, it can be seen that all the factors cause the signal to noise ratio to decrease from 60 dB to about 30 dB.

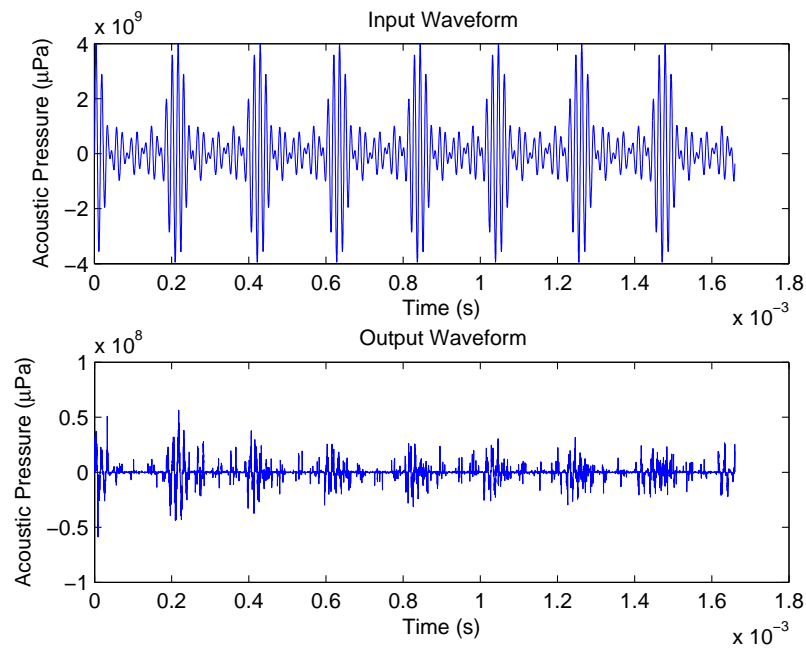


Figure 6.2: Sample input output waveforms for the Water Channel Simulator(WCS)

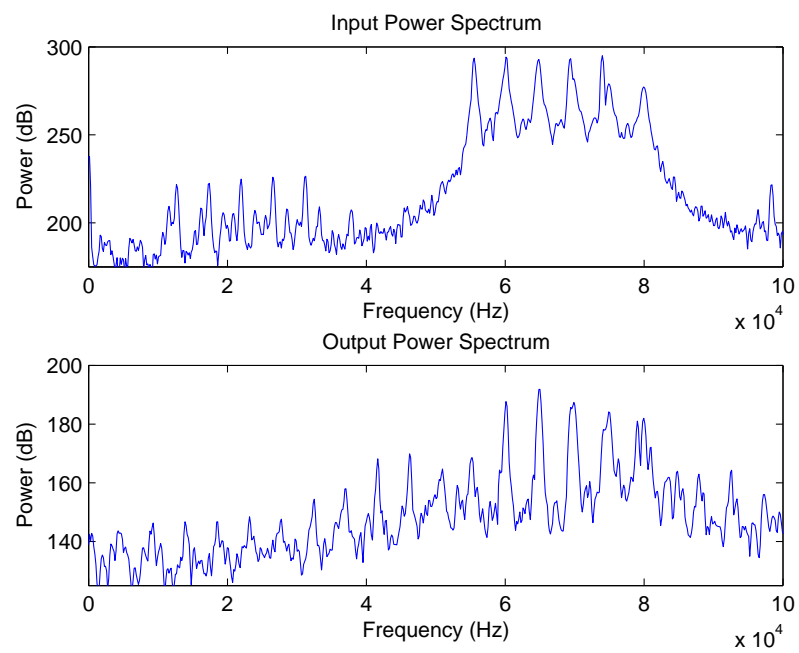


Figure 6.3: Sample input output power spectra for the Water Channel Simulator(WCS)

Chapter 7

Conclusion

7.1 Observations

The aim of this thesis was to provide a solution for the simulation of a underwater acoustic communication system. Various phenomena that affect the such a system were explored and the latest models based on recent research were used to accurately simulate them, thus providing a base on which the RaPTEX project can build on.

A case study was undertaken to establish that the simulation system is indeed a viable solution.

7.2 Future Work

There is a lot of scope for further work on the simulation system. Some of the issues are listed below.

Integration with RaPTEX needs to be performed in order to use this system effi-

ciently. The input and output interfaces have been deliberately left to be text files for ease of connection to the rest of the RaPTEX project.

Water body profiles need to be built up by performing measurements of the relevant parameters in the different water environments which RaPTEX plans to target.

The *Visualization Module* can be improved to include more information about the received signal, based on the modulation scheme used.

Support for multiple modulation schemes can be added to the receiver, in order to evaluate their pros and cons.

Support for a network of ESS platforms simultaneously talking to a single WCS.

Bibliography

- [1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.
- [2] C. Bjerrum-Niese, L. Bjorno, M.A. Pinto, and B. Quellec. A simulation tool for high data-rate acoustic communication in a shallow-water, time-varying channel. *Oceanic Engineering, IEEE Journal of*, 21(2):143–149, April 1996.
- [3] J. Catipovic, A. Baggeroer, K. Von Der Heydt, and D. Koelsch. Design and performance analysis of a digital acoustic telemetry system for the short range underwater channel. *Oceanic Engineering, IEEE Journal of*, 9(4):242–252, October 1984.
- [4] J. A. Catipovic. Performance limitations in underwater acoustic telemetry. *Oceanic Engineering, IEEE Journal of*, 15(3):205–216, July 1990.
- [5] Steven J Cooke, Scott G. Hinch, Martin Wikelski, Russel D. Andrews, Thomas G. Kuchel, Louise J. Wolcott, and Patrick J. Butler. Biotelemetry: a mechanistic approach to ecology. *Trends in Ecology & Evolution*, 19(6):334–343, June 2004.

- [6] Roberto Cristi. *Modern Digital Signal Processing*. Brooks/Cole-Thomson Learning, 2004.
- [7] Alexander G. Dean, Mihail L. Sichitiu, and Thomas G. Wolcott. Raptex proposal. North Carolina State University, Raleigh, NC, USA.
- [8] Alexander Guimarães Dean. *SOFTWARE THREAD INTEGRATION FOR HARDWARE TO SOFTWARE MIGRATION*. PhD thesis, CARNEGIE MELLON UNIVERSITY, 2000.
- [9] A. Essebbar, G. Loubet, and F. Vial. Underwater acoustic channel simulations for communication. *OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.'* *Proceedings*, 3:495–500, September 1994.
- [10] A. Falahati, B. Woodward, and S.C. Bateman. Underwater acoustic channel models for 4800 b/s qpsk signals. *Oceanic Engineering, IEEE Journal of*, 16(1):12–20, January 1991.
- [11] R. E. Francois and G. R. Garrison. Sound absorption based on ocean measurements: Part i: Pure water and magnesium sulfate contributions. *The Journal of the Acoustical Society of America*, 72(3):896–907, September 1982.
- [12] R. E. Francois and G. R. Garrison. Sound absorption based on ocean measurements. part ii: Boric acid contribution and equation for total absorption. *The Journal of the Acoustical Society of America*, 2(6):1879–1890, December 1982.
- [13] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3.

- Proceedings of the IEEE*, 93(2):216–231, 2005. special issue on "Program Generation, Optimization, and Platform Adaptation".
- [14] T. M. Law, L F. Yeung, R.S. Bradbeer, and Angus K M Wu. Acoustic modem for underwater mobile robots: A new wavelet based modulation technique for fast changing channels. *Proceedings 10th IEEE International Conference on Mechatronics and Machine Vision in Practice*, pages 195–202, December 2004.
- [15] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic Linear Algebra Subprograms for Fortran usage. *ACM Transactions on Mathematical Software*, 5(3):308–323, September 1979.
- [16] Xavier Lurton. *An Introduction to Underwater Acoustics : Principles and Applications*. Praxis Publishing Ltd, Chichester, UK, 2002.
- [17] Herman Medwin and Clarence S. Clay. *Fundamentals of Acoustical Oceanography*. Academic Press, 1300 Boylston Street, Chestnut Hill, MA 02167, USA, 1997.
- [18] Herman Medwin and colleagues. *Sounds in the Sea : From Ocean acoustics to Acoustical Oceanography*. Cambridge University Press, 2005.
- [19] University of Washington Applied Physics Lab. Apl-uw high-frequency ocean environmental acoustic models handbook. Technical report, Applied Physics Laboratory, University of Washington, Seattle, WA 98105-6698, USA, 1994.
- [20] Tony Ottosson, Adam Piatyszek, et al. It++. <http://itpp.sourceforge.net/>.

- [21] Matthias Pätzold. *Mobile Fading Channels*. John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012, USA, 2002.
- [22] John G. Proakis and Manolakis Dimitris G. *Digital Signal Processing : Principles, Algorithms and Applications*. Pearson Education (Singapore) Pte. Ltd., Indian Branch, 482 F.I.E. Patparganj, Delhi 110 092, India, 3 edition, 1996.
- [23] Joo Carlos Silva, Nuno Souto, Francisco Cercas, and Cezar Bruma Amrico Correia, A. Rodrigues. Efficient simulation of tapped delay channel models using converted discrete time channel models.
- [24] B.L. Titzer, D.K. Lee, and J. Palsberg. Avrora: scalable sensor network simulation with precise timing. *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium*, pages 477–482, 2005.
- [25] R. J. Urick. *Ambient Noise in the Sea*. Naval Sea Systems Command, Department of the Navy, Washington, D.C. 20362, 1984.