ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Autonomous Systems Lab
Prof. Roland Siegwart

Semester-Thesis

Design of a High Speed, Short Range Underwater Communication System

Part I - Electronic Concept and Simulation of the Acoustic Underwater Channel

Spring Term 2009

Supervised by:
Jérôme Maye
Andreas Breitenmoser

Author:
Elias Hagmann

Contents

Abstract

The present report describes the first part of the development of a high-speed, short range underwater communication system to be used by the NARO fish. It is found that the acoustic communication approach matches best the given requirements of the NARO project. Constitutive parts of this first stage of the project are thus on one hand the design of the needed electro-acoustic circuitry and on the other hand the development of a powerful but highly specialised simulation of the acoustic underwater channel.

Chapter 1

Introduction

Two systems developed in the tensions of the cold war illustrate like nothing else the technical challenges involved with underwater communication - the american Sanguine and the sowjet ZEVS system. Both systems use electromagnetic waves in the ELF (extremely low frequency) domain to transfer information to submerged submarines up to a depth of around 300m. Waves situated in the ELF spectrum with frequencies between 30Hz and 300Hz have the ability to penetrate in a great measure the electrically conducting seawater, explaining this large possible depth. But there are huge drawbacks related with waves in this low frequency region. On one hand only very low data rates in the range of a few bits per minute are possible. On the other hand these frequencies have wavelengths λ of between 100km and 100000km and since a good transmitting antenna has a length of a multiple of $\lambda/4$, gigantic structures are needed to transmit signals in the ELF spectrum. The ZEVS system for example uses a transmitting ground dipole of an alleged length of 106km!

The present project is focused on the development of a communication system that enables the use of considerable higher data rates and smaller communication structures than the two systems described above - with the large advantage that its transmission range is allowed to be as small as a few hundred metres. In a first phase of the project the involved electronic units were designed and a simulation of the underwater channel was developed. In a second project phase the focus will lie on the implementation of the electronics and on the development of powerful signal processing algorithms to increase the maximum possible data rates as much as possible. The present report describes the first project phase.

Chapter 2

Target Application

The present work focusses on the development of a high speed, short range underwater communication system for shallow water environments. The selection of this application field is not occasional but follows the main operational area of the parent project of this work - the NARO fish.

2.1 NARO - Nautical Robot

The project NARO - NARO stands for nautical robot - is concerned with the development of an artificial fish. The main goal of the project lies in the imitation of the natural fish movement, which has a locomotion efficiency that excels by far all human built devices down to the present day. The design of the NARO fish is oriented by the shape and motion of a tuna fish - a fish that has an average locomotion speed of 20km/h and can reach velocities up to 70km/h on short distances! Figure 2.1 shows a concept study of the artificial fish. The partition of the fish in six distinct segments is visible, which should ensure that the fish could use a nature like motion although tuna fishes generate most of their acceleration by the tail fin.

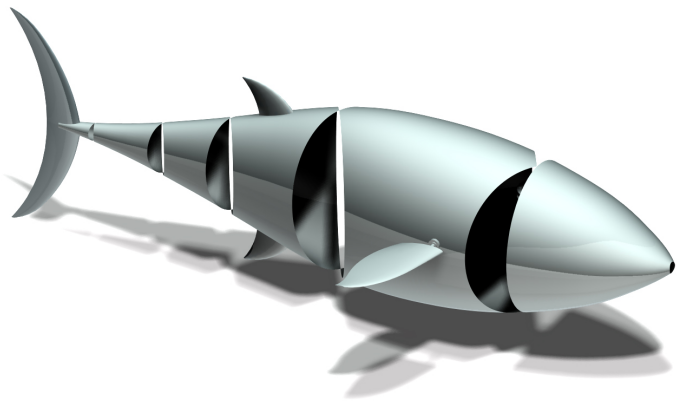

Figure 2.1: Concept of the NARO fish.

2.2 Requirements on the Communication System

The underwater communication system should provide two basic functionalities: On one hand it should be able to transmit control signals from the remote control to the fish, since the fish will not act autonomously but will follow human control commands. On the other hand, a video signal should be transmitted back from the fish to the remote station. Thus, the overall system should provide a full duplex communication link with low-speed, low-probability-of-error communication in one direction and high-speed communication in the other direction without that strict requirements on the bit error rate.

2.2.1 Mandatory Features

This section describes the minimal set of requirements the communication system has to fulfil in all cases. They can be partitioned in requirements on speed, size, power consumption and the interface to other electronic units inside the fish.

Minimum Speed Requirements

The most important part of the communication system that has to be implemented under all circumstances is its ability to transmit control commands to the fish. A calculation of a member of the NARO team led to the following minimum speed requirements:

$$\begin{aligned} & 4 \times \text{proportional controller} \times 10 \text{ bit resolution} \times 64\text{Hz} \\ & + 2 \times \text{switches} \times 1 \text{ bit} \times 16\text{Hz} \\ & = 2.592\text{kb/s} \, . \end{aligned} \tag{2.1}$$

Thus, in the minimal case where the video transmission is left away the system has to be able to transmit data reliably at a data rate of 2.592kb/s.

Maximum System Size

In order to be able to implement the system in the fish, the mechanical dimensions of the system part that has to be placed inside the fish are not allowed to exceed 20cm × 10cm × 10cm. Since the power supply of the communication system is provided by the fish electronics, it could be excluded from the space calculations. On side of the remote control, no restrictions on the system size occur.

Maximum Power Consumption

The time the NARO fish could stay underwater largely depends on the power consumption of the used electronic units. Thus a low power communication system is desired. On the other hand, the larger the used transmission power is, the higher gets the achievable data rate and signal range. Therefore one needs to balance these two contradictory needs. To have an upper bound in these balance deliberations, the maximum allowed power dissipation of the communication system part situated in the fish is set to 10W.

Interface

The communication system has to provide an interface to the other electronic units in the fish in order to be able to deliver the control commands and to receive the video signals. This interface is set to be a pulse width modulated (PWM)

line. PWM uses two states of a physical quantity - like the voltage on a line - and a sampling frequency f_s. Signal values are now encoded by the ratio of the occurence times of the two states of the physical quantity during one sampling period $T_s = 1/f_s$. Demodulation can be done for example by means of a low pass filter that acts like an integrator on the incoming pulse.

2.2.2 Desired Features

The previous section dealt with the mandatory features and the main differences between the mandatory and the desired features lie on one side on the occurence of a video transmission system affecting mainly the needed data rate and on the other side the minimization of the system's power dissipation.

Video Transmission System

To be able to transmit video signals, a high data rate is needed. Since the higher the data rate of the communication system is the better the image quality of the transmitted video signal could be, there is no optimal data rate for video transmission. In fact one can make the very general statement: The higher the maximum data rate gets, the better the system is rated. Thus only a rough estimate of a lower bound on the data rate to enable video transmission is provided here. We think that a video image should consist of at least 200×100 black/white pixels and should be actualised at least once per second not leading to a fluent video image but to a sequence of single pictures. Neglecting the fact that there is a load of video compression algorithms we thus calculate the minimum required bit rate br to:

$$br = 200\text{pixel} \times 100\text{pixel} \times 1\text{bit} \times 1\text{Hz} = 20\text{kbit/s} \, . \tag{2.2}$$

Desired Power Dissipation

The desired power dissipation of the communication system is set to be 1W - a value hard to reach.

Chapter 3

State of the Art in Underwater Communication

Since during and after the second world war submarines gained more and more importance in the naval warfare, research in the area of underwater communication was strongly intensified. In the last decades, applications with sensor networks for oceonagraphic research and for the offshore oil and gas industry also strengthened the demand after fast and reliable underwater communication links. Through these application fields, many different approaches to communicate underwater were invented and pursued in the last century. In the following, the different kind of approaches are presented together with their advantages and disadvantages.

3.1 Different Approaches to Communicate Underwater

There are basically three completely different ideas, how information could be transfered through the underwater channel. The first approach uses general electromagnetic waves and in special optical waves. The second approach tries to communicate via electric current waves without making use of the electromagnetic energy conversion and the third approach uses acoustic waves to transfer signals.

3.1.1 Electromagnetic Waves

The reasons why mainly electromagnetic waves are used to transfer information in the classic wireless air channel lie in their fast propagation speed, in the wide usable frequency spectrum and in the small environment noise compared for example with acoustics, factors that all lead to high possible data rates. Furthermore, the electromagnetic wave has the ability to propagate without a carrier medium and the electric-magnetic field conversion enables in general very large communication ranges.

But in water - especially in seawater - things get different. Solving Maxwell's equation to predict the propagation of electromagnetic waves for the case of a linearly polarised plane wave travelling in z-direction, we get for the electric field strength E_x and the magnetic field strength H_y [1]:

$$E_x = \hat{E} e^{j\omega t - \gamma z} \tag{3.1}$$

$$H_y = \hat{H} e^{j\omega t - \gamma z} \tag{3.2}$$

where $\hat{E}$ and $\hat{H}$ are the amplitudes of the electric and the magnetic field wave and

$$\gamma = j\omega\sqrt{\epsilon\mu - \frac{j\sigma\mu}{\omega}} = \alpha + j\beta \; . \tag{3.3}$$

γ is the propagation constant expressed in terms of the permittivity ϵ, the permeability μ and the conductivity σ of the material. α is the attenuation and β the phase factor of the wave.

As Siegel and King [2] showed, α is very large in seawater and thus the electromagnetic waves are affected by strong attenuation. This explains the need for large transmission power and the short possible communication distances when using electromagnetic waves underwater.

Optical Waves

Most of the electromagnetic waves in the visible spectrum are not able to propagate underwater over larger distances due to a high absorption of waves in this frequency region. But there is an exception for visible electromagnetic waves in the blue-green region with a corresponding wavelength of about 470nm $\sim$ 570nm. These waves experience small attenuation compared with other optical waves with attenuation constants $e^{\alpha} \in [0.155, 0.5]$dB/m . This "Window Effect" called phenomenon enables optical waves in the blue-green spectrum to propagate over large distances and are therefore usable in underwater communications. The biggest advantages of a communication system using blue-green Lasers are the very small propagation delay due to the transmission at light-speed and the high possible data rates. But there are some difficulties related with this communication method making it impractical in many situations. One problem lies in the need for environments that are not to heavily distorted by ambient light. And a major problem comes with the fact that a transmitting Laser always has to point exactly on the receiver to be able to transfer information. This problem can be circumvent by using not a directed but by applying a diffused Laser beam. If the beam angle is for example 30°, it is quite easy for the transmitter to hit the receiver. Unfortunately this method reduces the possible transmission range to several metres[3][4].

3.1.2 Electric Current

The electric current method uses the fact that seawater is a conductive medium. The modulated signal wave is applied to a pair of transmitting electrodes that launch a current field in the channel. If this current field is strong enough, the receiver - that also uses a pair of electrodes - could measure a potential difference and therefore receive the signal. Since electric current noise is extremely low in seawater, small current field amplitudes are sufficient to receive information and a large data rate is achievable. Nevertheless, large currents are needed on transmitter side to generate current fields strong enough to penetrate a sufficient distance in the water. Another problem with this method is that it completely fails if the medium is non-conductive like for example non-salty water[5][6].

3.1.3 Acoustic Communication

The most common method to communicate underwater is the acoustic approach. Here, a sound projector converts electric signals into pressure signals and a hydrophone receives the pressure signals and converts them back into electric signals. The sound projectors and hydrophones are often called transducers since they are able to act both as a hydrophone and a projector but in most cases a device is

matched to one of the two functionalities. Both hydrophones and projectors are mostly built by using piezoelectric material. Sound has by far the largest underwater propagation range compared to the used transmission power. But there are some large drawbacks associated with acoustic underwater communications, which are presented in the next section.

Problems with Acoustic Underwater Communication

In the following, the most important problems associated with acoustic underwater communications are presented[7][16]:

- Multipath Propagation: Reflections of the sound waves at the channel boundary give rise to the formation of different distinct paths between transmitter and receiver. These paths all have their own propagation delay, thus they arrive at the receiver at different times and may reduce the received signal energy by negative superposition or may even cause intersymbol interference (ISI).

- Formation of Shadow Zones: Sound celerity gradients bend the acoustic waves in a designated direction and can thus lead to the formation of shadow zones where no acoustic power is transfered to.

- Small Propagation Speed: Underwater acoustic waves have a propagation speed around 1500m/s which is very small compared to the propagation speed of electromagnetic waves that lies in the region of 3×10^8m/s. The small propagation speed leads to a large propagation delay.

- Noise: The communication link is strongly affected by ambient noise like shipping and animal sounds.

- Limited Bandwidth: The usable acoustic frequencies depend on distance since higher frequencies are highly attenuated. Even for very short distances frequencies above 1MHz can not be received. This bandwidth limitation also limits the maximum achievable data rate.

3.2 Current Commercial Solutions

Table 3.2 presents an overview over some of the commercial underwater communication systems currently available. It can be seen that the only solution that uses electromagnetic (EM) waves has by far the highest data rate. But at the same time this is the solution with the shortest range $\leq$10m. If we compare the acoustic devices among themselves, a strong dependency between range and maximum possible data rate can be identified: A shorter range in general leads to a higher possible data rate. The last device Develogic HAM.NODE is only usable for vertical acoustic communications. This can be seen as a completely different acoustic channel, that is why this device could hardly be compared with the other ones.

Name	Data Rate	Range	Power Consumption (Rec./Trans.)	Method
Wireless Fibre Seatooth	≤100kb/s	≤10m	4.56W/15.84W	EM
EvoLogics S2C R 48/78	≤28kb/s	1km	500mW/2.5W	Acoustic
LinkQuest UWM1000	≤19.2kb/s	350m	0.75W/1W	Acoustic
AquaComm Modem	480b/s	3km	25.2mW/252mW	Acoustic
CDL DATUM	480b/s	10km	1W/3.5W	Acoustic
Teledyne Benthos OEM	360b/s	2km	OEM	Acoustic
Develogic HAM.NODE	7kb/s[1]	-	≤3W/max. 500W	Acoustic

Table 3.1: Commercial solutions for underwater communications.

Chapter 4

Electronics

4.1 Carrier Choice

In chapter 3 four different signal carriers for underwater communication are presented:

- General electromagnetic waves
- Optical waves in the spectral region between 470nm and 570nm
- Electric current fields
- Acoustic waves

All of these carriers have their separate advantages and disadvantages thus the used signal carrier has to be selected carefully. Optical waves and current fields provide both large possible data rates. But optical signals are either hard to point in the right direction by using a high directivity Laser or will have a transmission range that is too small when using diffuse signal propagation. Current fields do not suffer from these drawbacks but raise the need after a highly conductive transmission medium and furthermore involve large transmission power. With that they are not usable in the present application. Therefore the general electromagnetic wave and the acoustic approach remain. The main drawback of electromagnetic waves underwater is their large attenuation. Admittely Al-Shamma'a et al. and Shaw et al. showed [8][9] that it is possible to build a short-range communication system using electromagnetic waves at MHz frequencies with sophisticated transmitting antennas. But compared with acoustic signal carriers this approach is extremely range-limited, requires substantial more transmission power and is only badly understood yet. Thus the acoustic signalling method is chosen despite of the large number of difficulties coming together with this approach.

4.2 Frequency Band Selection

Acoustic signals underly two basic restrictions which form constraints on the usable signal frequencies. The first restriction comes from ambient noise in the underwater environment, such as wave or shipping noise. Since underwater noise is high primarily for low frequencies, one is advised to choose a lower bound on the used signal frequencies high enough to avoid the noisy frequency bands. On the other hand, high frequency signals experience a much stronger attenuation on their propagation path than signals with low frequency do. This leads to an upper bound on

the highest usable frequency. Therefore, at least a rough model of signal attenuation and ambient noise in the underwater channel is needed to be able to make a reasonable decision upon which frequency band to use. In the following a noise and an attenuation model from [7] for the case of an open sea application are presented. This models are not expected to exactly match the case of a pool-like underwater environment - but since only rough estimates of noise and attenuation are needed at this stage, these models are considered to be sufficient.

4.2.1 Noise Model

The following empirical formulae give the power spectral densities of the four main noise sources turbulence, shipping, waves and thermal noise in dB re μPa per Hz as a function of frequency in kHz:

$$10 \log N_t (f) = 17 - 30 \log f \tag{4.1}$$

$$10 \log N_s (f) = 40 + 20 (s - 0.5) + 26 \log f - 60 \log (f + 0.03) \tag{4.2}$$

$$10 \log N_w (f) = 50 + 7.5\sqrt{w} + 20 \log f - 40 \log (f + 0.4) \tag{4.3}$$

$$10 \log N_{th} (f) = -15 + 20 \log f \tag{4.4}$$

It can be seen that in different frequency regions different noise sources dominate the overall noise. Turbulence noise affects mainly the frequency band $\prec 10Hz$. In the region between 10 Hz and 100 Hz, distant shipping - modeled through the shipping activity factor $s \in [0, 1]$ - is the dominating noise. Wave noise becomes important in the following band between 100 Hz and 100 kHz and is modeled through the wind speed w in m/s. Finally, for frequencies $f \succ 100kHz$, thermal noise becomes dominant. Figure 4.1 shows the overall noise $N (f) = N_t (f) + N_s (f) + N_w (f) + N_{th} (f)$ in dB re μPa in the case of shipping activity $s = 0$ and wind speed $w = 1$m/s.

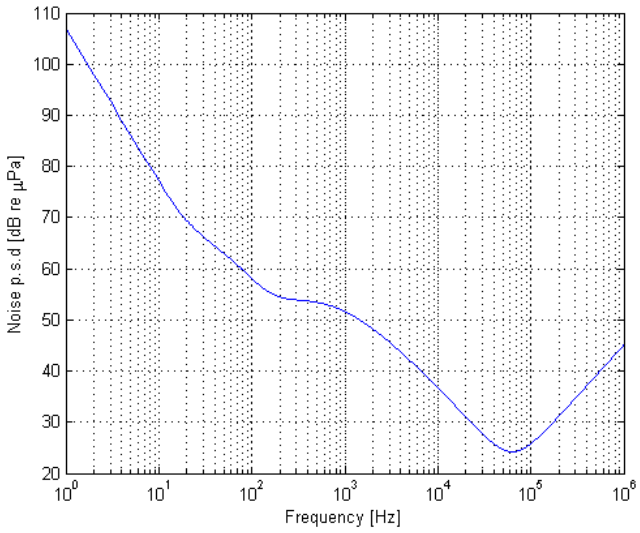

Figure 4.1: Power spectral density of the ambient noise $N (f)$ for $s = 0$ and $w = 1$m/s.

4.2.2 Attenuation Model

Acoustic signals that propagate underwater suffer from a degradation of amplitude called attenuation or path loss. One reason for this lies in the spatial spread of the sound waves, but an additional explanation can be found in the absorption of wave energy and transformation into heat. While the first effect depends only on the distance from the signal source to the receiver, the second phenomenom is besides its spatial dependency also frequency dependent. Combining both effects leads to the following law of attenuation with distance l and frequency f:

$$A(l, f) = l^k a(f)^l \tag{4.5}$$

where k stands for the spreading factor and $a(f)$ is the absorption coefficient. k represents the geometry of propagation; typical values are $k = 2$ for spherical spreading, $k = 1$ for cylindrical spreading and $k = 1.5$ for the so-called practical spreading. The absorption coefficient can be expressed using the Thorp's formula

$$10 \log a(f) = 0.11 \frac{f^2}{1+f^2} + 44 \frac{f^2}{4100+f^2} + 2.75 \cdot 10^{-4} f^2 + 0.003 \tag{4.6}$$

which gives $a(f)$ in dB/km for f in kHz. Figure 4.2 shows a plot of α in dB/km versus frequency.

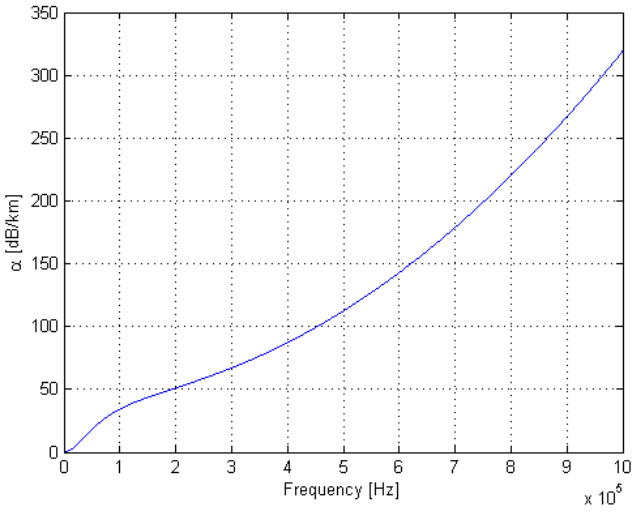

Figure 4.2: Attenuation factor α vs. frequency.

4.2.3 SNR and Band Selection

As in [7] we calculate the signal-to-noise ratio SNR depending on distance l and frequency f to

$$SNR(l, f) = \frac{P/A(l, f)}{N(f)\Delta f} \tag{4.7}$$

where Δf is the narrow receiver noise bandwidth around f. 4.7 can be separated in a constant term $\frac{P}{\Delta f}$ and a distance and frequency dependent term given by

$$SNR(l, f) \propto \frac{1}{A(l, f)N(f)} \, . \tag{4.8}$$

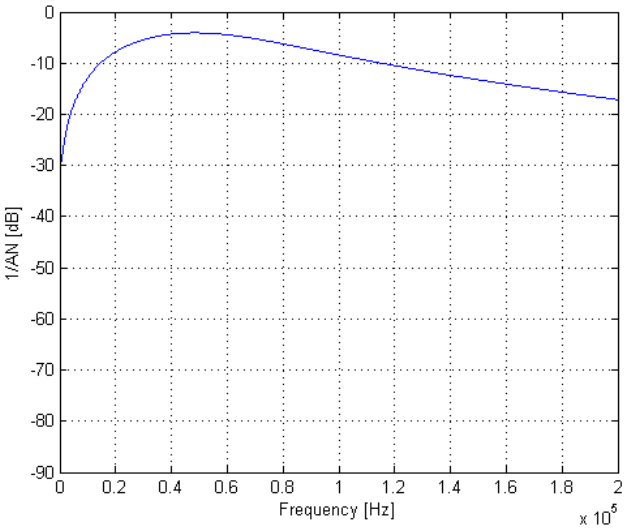

Figure 4.3: Plot of the frequency dependent part of the SNR with $l = 200$m and $k = 1.5$.

Figure 4.3 shows a plot of $\frac{1}{A(l,f)N(f)}$ with noise as in figure 4.1, $l = 200$m and $k = 1.5$ (practical spreading). It can be seen that the SNR has a global maximum around 50kHz and that in the band between 20kHz and 100kHz $\frac{1}{A(l,f)N(f)}$ is higher than -8dB. Thus the target frequency band for the communication system is set to be the frequency range between 20kHz and 100kHz with a bandwidth of 80kHz.

4.3 Functional Schematic of the Communication System

Figure 4.4 shows a high level schematic of a combined acoustic transmitter and receiver circuit. On transmitter side, the signal time function $s(t)$ is generated within the digital signal processor (DSP) and then converted into an analog step signal by the digital-to-analog (D/A) converter. In order to reduce transmission power and to minimize the amount of high frequencies arriving at the receiver, the signal is subsequently processed by an analog lowpass filter. Afterwards a power amplifier amplifies the signal to the desired power level and passes it on to the D/70 sound projector. The receiver starts by picking up the noisy convolution of the signal time function with the channel impulse response $r(t) = (s * h)(t) + n(t)$ using the TC4013 hydrophone. Afterwards an amplifier controlled by the dsp unit shifts the power level of $r(t)$ in a way that it is optimal for further processing. Then the receiving analog lowpass filter is used to supress high frequencies in order to avoid aliasing by the subsequent sampling using the analog-to-digital (A/D) converter. After that, the amplified and lowpass-filtered signal $r(t)$ is discretized in time and amplitude and handed over to the DSP unit to identify the transmitted symbols.

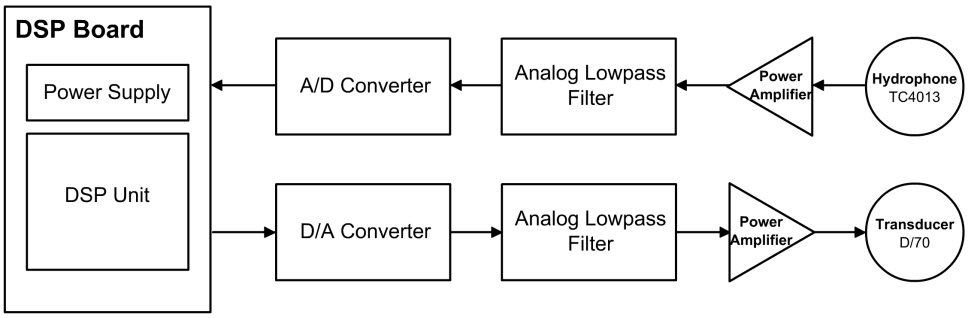

Figure 4.4: Functional schematic of a combined transmitter and receiver circuit.

4.4 DSP Unit

As section 4.3 shows, the core of the communication system is formed by the DSP unit. A digital signal processor is a processor that is specifically designed for signal processing purposes. It has among other things the following DSP-specific properties:

- Implementation of the processor using the Harvard-architecture that separates instruction memory from data memory. With that the instruction memory and the data memory could be accessed simultaneously and even multiple data memory accesses are possible during the execution of one instruction.
- Special synchronous serial interfaces for data input and output are available. This allows to receive and to transmit data without interrupting the processor's mathematical operations.
- A multiply-accumulate (MAC) instruction exists that enables parallel addition and muliplication during one instruction. This instruction has the form of $a' = a + bc$ and helps with calculating convolutions and FFT conversions.
- Availability of adress generation units (AGU) that enable the automatic generation of adresses during data fetch or store operations. This can be used for example to accelerate array handling since the processing of an array requires fetching values from consecutive memory locations. Furthermore a special adress generation that treats the bits in a register like they were in reversed order before incrementing the register value is used in FFT calculations.
- Presence of a large number of registers to store intermediate values. This is needed for example to implement time delays.

DSPs could be implemented either using a fixed or a floating point architecture. Fixed point architectures simply code values using the bit representation of these values. Thus they are easier to implement and less power consuming. But the problem with fixed point DSPs lies in the possible occurence of overflows if the number of bits to represent a value exceeds the processor's register size. Floating point architectures are more complex since values are represented using $m \times 2^e$ with m the mantissa and e the exponent. With that, floating point processors have an enourmous dynamic range but suffer from higher power consumption. Therefore - from a user point of view - the decision has to be made between algorithm complexity that is higher with fixed point architectures and power consumption that is higher with floating point architectures.

In the present work the Texas Instruments C6713 floating point DSP is used. It can perform up to 2400 million instructions per second (MIPS) by processing eight instructions per cycle with a 300MHz clock and is therefore a very fast signal processor. This is needed in order to be able to implement complex modulation and demodulation algorithms to handle the challenging acoustic underwater channel.

4.4.1 McBSP - Multichannel Buffered Serial Port

The C6713 DSP has a number of different interfaces to communicate with the external world. The interface used in the present work to access the receiving and transmitting electronic units is the Multichanel Buffered Serial Port (McBSP). Figure 4.5 shows a block diagram of the McBSP presenting its internal and external components [10]. On one side the McBSP uses seven pins to communicate with other devices and to provide or receive framing and clocking information. These are the triple buffered data receive (DR) input and the double buffered data transmit (DX) output pin, the input/output clock receive (CLKR) and clock transmit (CLKX) pins, the input/output frame receive (FSR) and frame transmit (FXR) synchronisation pins and the CLKS pin that allows an external clock to be used by the McBSP. On the other side the McBSP can be accessed by the C6713 CPU via the 32 bit wide peripheral bus.

The most important aspect in this context is the synchronisation of the receive and transmit data bits as well as the receive and transmit frames. The boundaries between the single bits are provided by the receiving and transmitting clock that is generated by the McBSP Sample Rate Generator sourced either by the internal clock or by an external signal via CLKS. Frames are used to distinguish individual data blocks. The C6713 McBSP provides the ability to configure the following parameters for data and frame synchronisation:

- Polarities of FSR, FSX, CLKX, and CLKR
- A choice of single- or dual-phase frames
- For each phase, the number of elements per frame
- For each phase, the number of bits per element
- Whether subsequent frame synchronization restarts the serial data stream or is ignored
- The data delay from frame synchronization to first data bit which can be 0-, 1-, or 2-bit delays
- Right or left justification as well as sign extension or zero filling for receive data.

All configurations can be done indepently for receiving and transmitting operation.

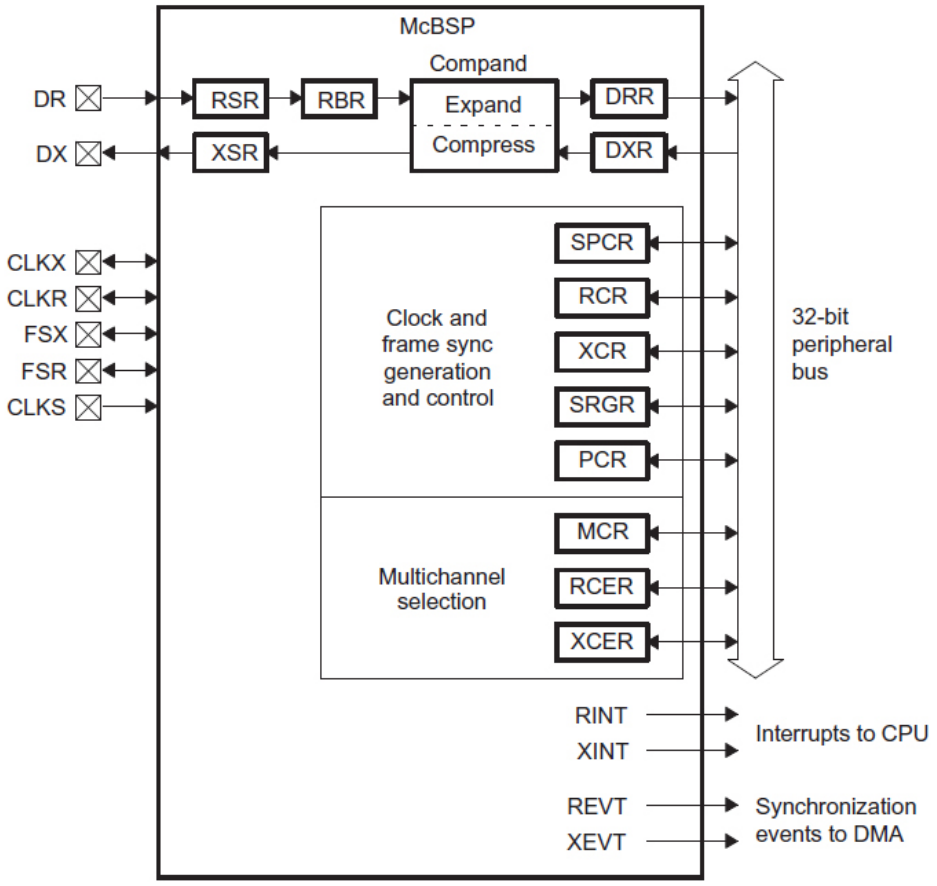

Figure 4.5: Block diagram of the C6713 McBSP.

4.5 Acoustic Transducer

The interface between the electronics and the underwater channel is formed by acoustic transducers that convert electrical energy into acoustic energy (sound projector) on one side and acoustic energy into electrical energy on the other side (hydrophone). These transducers are built in most of the cases using piezoelectric materials.

4.5.1 Piezoelectric Energy Transformation

The theory of piezoelectricity describes the phenomenon that on one hand the non-uniform deformation of designated materials produces an electric voltage along the line of deformation and on the other hand the inverse, that a voltage applied to those materials induces a deformation proportional to the voltage. Since the piezoelectric effect can be used either to generate or to measure pressure pulses in the medium around the piezoelectric material, it is possible to launch or to receive acoustic waves using piezoelectrocity.

4.5.2 Transducer Selection

In table 4.5.2 different acoustic transducers for underwater applications currently available on the market are presented. A preselection focussed mainly on the device

weight has been done in order to restrict the selection field to applicable devices.

Name	Max. Sensitivity Transmit [dB re μPa/V]/ Receive [dB re V/μPa]	Frequency Range [Hz]	Mechanical Dimensions [mm]
Aquarian H1A	none/-190	1-100k	46×25
Reson TC4013	130/-211	1-170k	63×10
Neptune Sonar D/70	148/-195	10-100k	45×34
Neptune Sonar D/140	143/-205	20-170k	30×20
Neptune Sonar B/200	none/-210	10-180k	24.5×9
ITC 1042	148/-200	10-100k	76.2×37.5

Table 4.1: Available transducers.

To be able to rate the different transducer one must understand the meaning of receive and transmit sensitivity. The transmit sensitivity is the measured pressure output of an acoustic projector for an input of 1Vrms (root mean square voltage value) in a distance of 1m away from the projector. Since the transmit sensitivity is given in decibel with a reference pressure value of 1 μPa, its unit is [dB re μPa/V]. The receive sensitivity is the measured open circuit rms voltage at the transducer terminals if a pressure of 1μPa is applied to it. The receive sensitivity is also measured in decibel with reference value 1μPa, thus its unit is [dB re V/μPa]. With that we see that both the receive and the transmit sensitivity is better for higher values.

With the above and taking into account restrictions on the delivery times of the Neptune transducer units, the selections fell on the Aquarian H1A as hydrophone and on the Reson TC4013 as sound projector. First tests showed that the transmit sensitivity of the TC4013 is not large enough to reach the desired transmission range. Thus for the future system the employement of the Neptune Sonar D/70 as sound projector is recommended.

4.6 Lowpass Filter Design

Section 4.3 showed that both on receiver and transmitter side an analog lowpass filter with equal cutoff frequency f_c is needed. Following 4.2, f_c is set to the maximum allowed signal frequency of 100kHz. This will also slightly attenuate the highest frequencies below f_c, but since it is not planned to transmit at frequencies higher than 90kHz, this choice of f_c has not much influence on the transmission behaviour of the system.

The lowpass filter is designed following the guidelines from Texas Instruments [11]. After [11], the general transfer function of a lowpass filter normalised to the cutoff angular frequency $\omega_c = 2\pi f_c$ calculates to:

$$H_{lp}(j\omega/\omega_c) = \frac{A_0}{[1 + a_1 j\omega + b_1(j\omega)^2][1 + a_2 j\omega + b_2(j\omega)^2] \dots [1 + a_o j\omega + b_o(j\omega)^2]} \tag{4.9}$$

where o is the filter order, A_0 is the direct current filter response and a_i and b_i are the filter coefficients. It holds that the higher the filter order is, the narrower the transition between pass- and stopband gets. By using different values for the filter coefficients we can choose among other from three types of lowpass filters. The first

type is the Bessel filter that results in a linear phase response up to ω_c, the second type is the Tschebyscheff filter which sharpens the transition between passband and stopband and the third type - the Butterworth filter - leads to a maximal flatness of the passband region. For the present circuit a fifth order Butterworth filter with the coefficients

$$a_1 = 1, a_2 = 1.6180, a_3 = 0.6180 \tag{4.10}$$

$$b_1 = 0, b_2 = 1, b_3 = 1 \tag{4.11}$$

is used. Figure 4.6 shows the impulse response of such a lowpass filter.

On the receiver side, there is a need after amplifying and lowpass filtering the received signal. This can be done at once by choosing A_0 in the filter response to be larger than 1. On transmitter side there is no need after amplifying the signal voltage but to increase the transmission power. Thus the amplification of the signal is done separately and the direct current response A_0 of the transmit lowpass filter is chosen to be 1. Figure 4.7 illustrates how a fifth order lowpass filter is built using resistors, capacitors and operational amplifier. The resistor and capacitor values can be determined using the filter coefficients, ω_c and A_0, more care has to be taken concerning the choice of the operational amplifier.

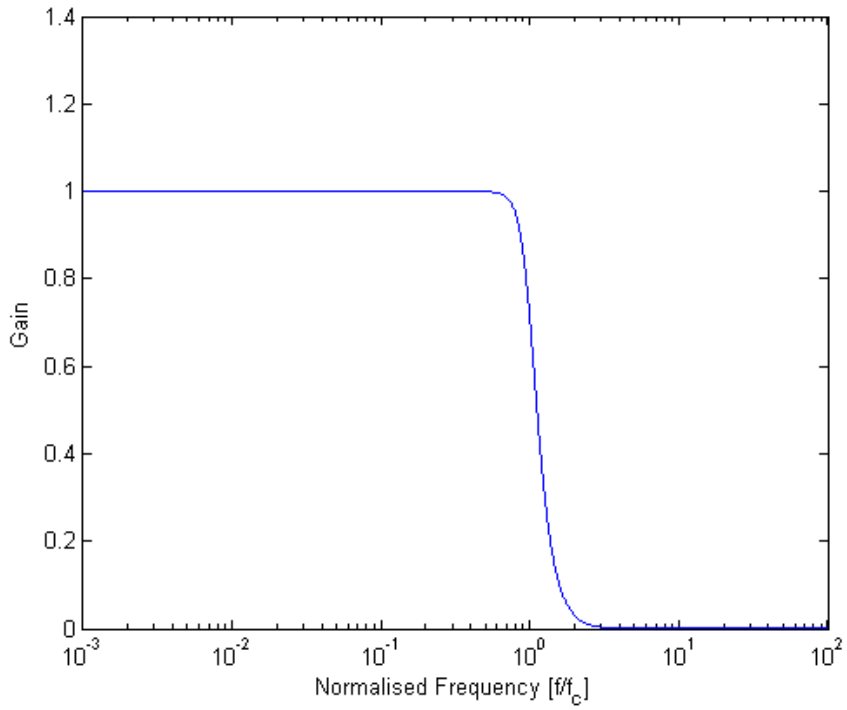

Figure 4.6: Impulse response of a fifth order Butterworth lowpass filter.

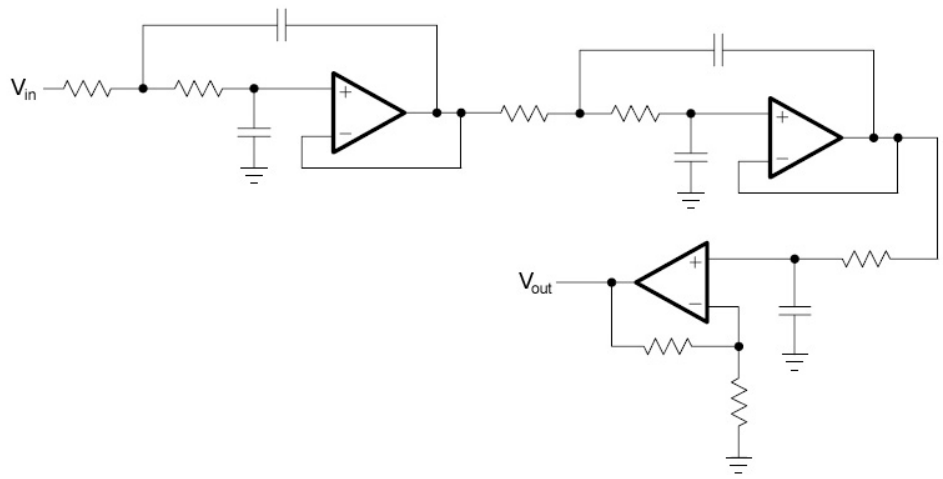

Figure 4.7: Fifth order lowpass filter.

4.7 Receiver Circuit Design

In the following, the design of the receiver circuit is illustrated.

4.7.1 Bipolar to Unipolar Signal Level Shift, Amplifying and Lowpass Filtering

As a first step, the incoming signal from the hydrophone is shifted from its original bipolar range $[-\hat{v}_s, +\hat{v}_s]$ to an unipolar voltage region $[GND, V_{DD}]$. This is achieved by using a special highpass filter as shown in figure 4.8. With that, the subsequent electronics can all be supplied by the GND - V_{DD} pair. The shifted signal is then presented to a butterworth lowpass filter with cutoff angular frequency $\omega_c = 2\pi \cdot 100kHz$ as presented in section 4.6, only with one slight difference. In order to be able to amplifiy the signal in a dynamic fashion, instead of fixed resistor values in the feedback loop of the first order filter stage (the last operational amplifier in 4.6) a digitally controllable potentiometer of type AD8400-10 is used. This potentiometer is connected to the McBSP interface of the C6713 DSP and allows an amplification of the signal up to the factor 100.

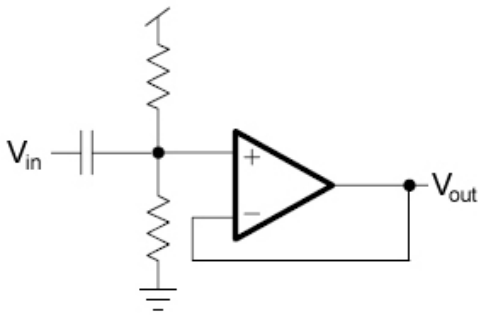

Figure 4.8: Bipolar to unipolar signal level converter.

4.7.2 Single-ended to Differential Driver

Since the used A/D converter, the 16 bit AD7693, has two inputs IN+ and IN- and converts the value computed by IN+ − IN-, it would be a waste of resources if a signal in the range of $[GND, V_{DD}]$ would be used as converter input. Thus the signal is converted from single-ended type to differential using the structure shown in figure 4.9.

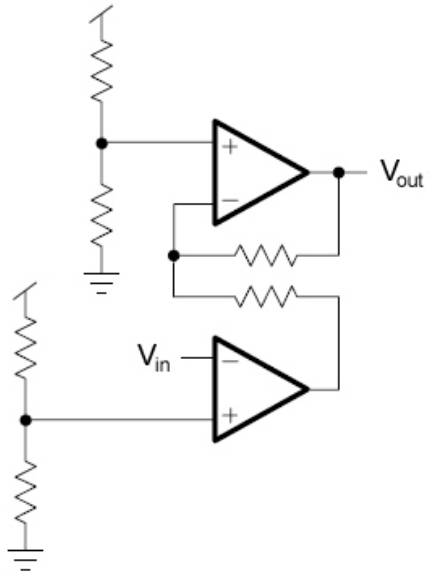

Figure 4.9: Single-ended to differential driver.

4.7.3 Analog to Digital Converter

As a last step the two differential signals are fed to the converter inputs via an analog passive lowpass stage formed by a resister together with a capacity. These lowpass filter - with large cutoff frequencies compared with the highest signal frequency - should minimise the noise presented to the converter. Figure 4.10 illustrates the converter input stage. At the end, the A/D comverter is digitally connected with the McBSP interface.

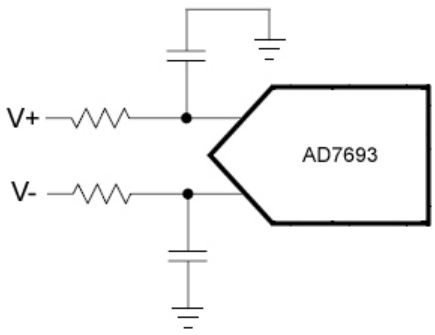

Figure 4.10: A/D converter input stage.

4.8 Transmitter Circuit Design

This section describes the design of the transmitter circuit.

4.8.1 Digital to Analog Converter

Following the signal path, the first thing to consider is the digital-to-analog conversion of the signal that is fed to the D/A converter using the McBSP pins DX (data), CLKX (bit clock) and FSX (frame synchronisation), where FSX triggers the particular conversions. The part thereby used is the AD5063, a 16 bit full-accurate D/A converter.

4.8.2 Lowpass Filter

After the D/A conversion of the signal it is fed to a unity gain butterworth lowpass filter with angular cutoff frequency $\omega_c = 2\pi \cdot 100\text{kHz}$ as described in section 4.6.

4.8.3 Signal Level Shift

Since a large direct current component in the signal could damage the sound projector, the transmitted wave should not be unipolar but bipolar with mean value GND. This can be reached by using a supply voltage with the size of $2 \cdot \text{V}_{DD}$ that is split in the middle to get three available voltages $-\text{V}_{DD}$, virtual GND and $+\text{V}_{DD}$. Virtual GND and $+\text{V}_{DD}$ are then used to source the D/A converter and the lowpass circuit while an operational amplifier with rail voltages $-\text{V}_{DD}$ and $+\text{V}_{DD}$ converts this unipolar signal to a bipolar signal. This procedure is illustrated in figure 4.11 where the operation amplifier on the left is responsible for providing the three different voltages and the OpAmp on the right shifts the level of the voltage $\text{V}_{in} \in [\text{GND}, +\text{V}_{DD}]$ to $\text{V}_{out} \in [-\text{V}_{DD}, +\text{V}_{DD}]$.

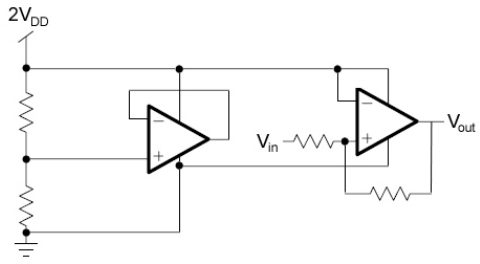

Figure 4.11: Unipolar to bipolar signal level shift.

4.8.4 Power Amplifier

Finally, a power amplifier ensures that the circuit is capable of sourcing enough current into the sound projector in order to get the desired transmission power.

Chapter 5

Simulation

5.1 Modelling the Acoustic Underwater Channel

5.1.1 Acoustics

Acoustics are concerned with the propagation of longitudinal waves in gases, fluids and solids. As particles in the different media are displaced from their equilibrium, they get closer to other particles and an elastic force arises which pulls them back to their original position. Due to the inertia, the particles overshoot their starting point and start oscillating back and forth. Since the displacement of one particle also displaces the next closest particle, the displacement propagates through the medium and an acoustic wave is generated.

Mathematical Fundaments

Sound is the propagation of local pressure variations in a medium. Thus to describe a sound field, one needs to find a pressure function $p(.,.)$ depending on position $\vec{s}$ and time t that solves the material equations under the given initial and boundary conditions. In the simple case of a plane one-dimensional wave in the free sound field we could use the basic hydrodynamic law for one dimension with particle velocity v, pressure p and material density ρ:

$$\frac{dv}{dt} = -\frac{1}{\rho}\frac{dp}{dx}. \tag{5.1}$$

Furthermore we have

$$\frac{dv}{dx} = -\frac{1}{K}\frac{dp}{dt} \tag{5.2}$$

where K is the compression module of the material. If we differentiate 5.1 and 5.2 by x and by t and combine this two equations, we get

$$\frac{\partial^2 p}{\partial t^2} = \frac{K}{\rho}\frac{\partial^2 p}{\partial x^2}. \tag{5.3}$$

5.3 has the form of the wave equation of d'Alembert where $c^2 = \frac{K}{\rho}$ and is solved by the general complex pressure wave function

$$p(x,t) = Ae^{j(\omega t - kx)} + Be^{j(\omega t + kx)} \tag{5.4}$$

with $\omega = |k|\, c$. This solution forms two waves, $Ae^{j(\omega t-kx)}$ traveling in forward and $Be^{j(\omega t+kx)}$ traveling in reverse direction. Using 5.1 we can calculate $v(.,.)$ as

$$v(x,t) = \frac{A}{\rho c} e^{j(\omega t - kx)} - \frac{B}{\rho c} e^{j(\omega t + kx)}. \tag{5.5}$$

If we concentrate on a forward traveling wave we could calculate the acoustic impedance Z as the ratio between local pressure and particle speed. We get

$$Z = \rho c. \tag{5.6}$$

As the above discussion holds only for the case of a free sound field where pressure and particle velocity are always in phase, 5.6 is also only valid under this special but common case. In general, the particle velocity $v(.,.)$ could have an arbitrary phase shift compared with the pressure $p(.,.)$. Therefore, in general Z has the form

$$Z(\omega) = \frac{\hat{p}(\omega)}{\hat{v}(\omega)} e^{\phi_p(\omega) - \phi_v(\omega)} \tag{5.7}$$

where $\phi_p(\omega) - \phi_v(\omega)$ is the phase shift between pressure and particle velocity depending on the angular frequency ω. Two important sonic characteristics are sound intensity I and sound power P of a sound source. They can be calculated employing the following equations:

$$I = \frac{1}{2} \hat{v} \hat{p} = \frac{\hat{p}^2}{2Z} \tag{5.8}$$

$$P = \int_S I dS \tag{5.9}$$

where $\hat{v}$ is the particle velocity amplitude, $\hat{p}$ the sound pressure amplitude and S is an arbitrary face around the sound source [12][13].

Transmission and Reflection

If a sound wave hits an interface between two media with different acoustic impedances Z the wave will be partially transmitted to the new medium and partially reflected back, as figure 5.1 illustrates. We set the $\rho_s = I_r/I$ and $\tau_s = I_t/I$ as the ratio between incoming and reflected sound intensity and incoming and transmitted sound intensity, respectively. The law of conservation of energy states that $\rho_s + \tau_s = 1$. With this we can calculate the reflection coefficient of an interface as

$$R = \frac{\hat{p}_r}{\hat{p}} = \frac{Z_2 - Z_1}{Z_2 + Z_1}. \tag{5.10}$$

This leads to a sound intensity reflection ratio of

$$\rho_s = \left(\frac{Z_2 - Z_1}{Z_2 + Z_1} \right)^2 \tag{5.11}$$

and a sound intensity transmission ratio (also called absorption coefficient) of

$$\tau_s = \frac{4 Z_1 Z_2}{(Z_2 + Z_1)^2} \tag{5.12}$$

Equation 5.11 shows that for $Z_2 \gg Z_1$ and $Z_2 \ll Z_1$ nearly the complete wave is reflected. In the first case the incoming wave hits the surface of a medium with higher acoustic impedance and the reflected wave gets phase shifted by π (R is positive). In the second case the reflection coefficient R is negative and the phase shift between incoming and reflected wave is 0 [12].

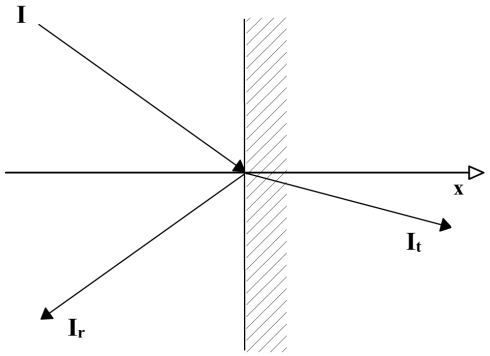

Figure 5.1: Partial reflection and transmission of an acoustic wave.

Refraction

If a wave of any kind is transmitted from one medium to another medium with a different wave propagation speed, the wave changes its propagation direction in the new medium according to Snell's Law. Formalised, Snell's Law states the following:

$$\frac{\sin \phi_1}{c_1} = \frac{\sin \phi_2}{c_2} \tag{5.13}$$

where $\phi_{1,2}$ are the propagation directions compared to the perpendicular of the interface between the two media and $c_{1,2}$ are the propagation celerities in medium 1 and medium 2 [14]. Figure 5.2 illustrates the refraction of a wave.

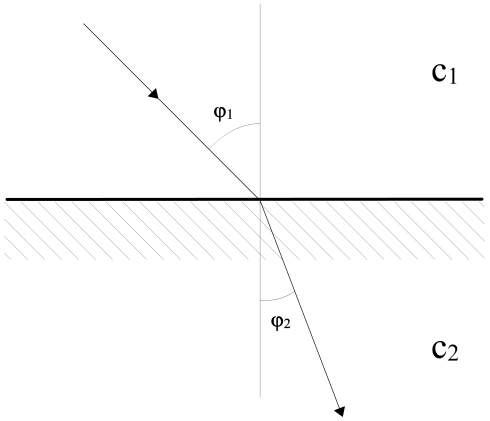

Figure 5.2: Refraction of an acoustic wave.

5.1.2 The Acoustic Underwater Channel

Ray Tracing Method

There exist different approaches to simulate the acoustic underwater (AUW) channel, including time expensive methods that solve the acoustic field equations on a mesh of points representing the area of interest. This work focuses on the point to point communication in a stable UWA channel with a small receiver compared to the wavelength of the involved waves, thus there is no necessity to know the values of the acoustic field anywhere else than on a single point representing the receiver. Under this conditions, a pretty simple simulation approach compared with solving the field equations can be used - Ray Tracing. As the name says, ray tracing uses

rays to find an expression of the acoustic field at exactly one point in the channel where the receiver is situated. To achieve this, the transmitter transmits a large number of rays uniformly distributed in all possible directions. The rays start to propagate through the channel and can be reflected by the channel boundary and refracted by sound celerity gradients. All rays that find their way to the receiver before a maximum number of reflections (or some other termination critera) is reached represent different paths between transmitter and receiver and are called eigenrays. Since all rays (and thus all eigenrays) are represented by a phasor with attenuated amplitude and phase shift both depending on the previous path, the complete set of found eigenrays can be used to calculate an approximation of the acoustic field at the receiver.

If we look at a single tone of frequency $f = \frac{\omega}{2\pi}$ transmitted with Power P, we can present the ray tracing method in a more formal way. Each ray i gets the initial intensity $I_i = \frac{P}{N}$ assigned with it, where N is the total number of rays sent. This intensity is attenuated during the propagation of the ray by the attenuation coefficient $A(l_i, f)$ with l_i the ray length and by reflection losses modelled by Γ_i. Thus, if we look at the acoustic pressure with 5.8 in mind, the final pressure amplitude becomes

$$\hat{p}_i = \frac{\sqrt{2ZI_i}\Gamma_i}{\sqrt{A(l_i, f)}} = \hat{p}_0 \frac{\Gamma_i}{\sqrt{A(l_i, f)}}. \tag{5.14}$$

with $\hat{p}_0 = \sqrt{2Z\frac{P}{N}}$ the initial amplitude assigned to all rays. To calculate the channel transfer function $H(f) = H(\frac{\omega}{2\pi})$ we take only the eigenrays into consideration and obtain

$$H(f) = \sum_{e=0}^{E-1} \frac{\Gamma_e}{\sqrt{A(l_e, f)}} e^{-j2\pi f \tau_e} \tag{5.15}$$

where E is the total number of eigenrays and $\tau_e = l_e/c$ is the time the eigenray e needs to find its way from the transmitter to the receiver. In the time domain this results in the channel impulse response of

$$h(t) = \sum_{e=0}^{E-1} \frac{\Gamma_e}{\sqrt{A(l_e, f)}} \delta(t - \tau_e) \tag{5.16}$$

with the dirac delta distribution $\delta(.)$ [7][15].

Energy Conversion

In order to be able to give a realistic channel model, the transmission power P must be determined in dependency of the voltage input $v_I(t)$. If Z_p is the complex impedance and G_p the gain of the sound projector, we get

$$P = G_p \frac{v_I^2(t)}{Z_p} \ . \tag{5.17}$$

Thus the transmission pressure function $p_0(t)$ calculates to

$$p_0(t) = \sqrt{2Z\frac{G_p v_I^2(t)}{NZ_p}} = v_I(t)\sqrt{\frac{2ZG_p}{NZ_p}} \ . \tag{5.18}$$

The output voltage function $v_O(t)$ on receiver side can be calculated in a similar manner:

$$v_O(t) = \sqrt{G_r Z_r P_r} \tag{5.19}$$

where G_r is the hydrophone sensitivity, Z_r the complex hydrophone impedance and P_r the received acoustic power. If we use $P_r = \sum_{e=0}^{E-1} I_e = \frac{p_r^2(t)}{2Z}$ with $p_r(t)$ the received pressure function, we get

$$v_O(t) = \sqrt{G_r Z_r \frac{p_r^2(t)}{2Z}} = p_r(t)\sqrt{\frac{G_r Z_r}{2Z}} \ . \tag{5.20}$$

Now we are able to give a more elaborate channel transfer function including the transfer effects of sound projector and hydrophone. We have

$$p_r(t) = p_0(t) * h(t) \tag{5.21}$$

where $*$ stands for convolution and $h(t)$ is the channel impulse response of equation 5.16. With 5.20 and 5.18 we obtain

$$\begin{aligned} v_O(t) &= \sqrt{\frac{G_r Z_r}{2Z}}[p_0(t) * h(t)] \\ &= \sqrt{\frac{2ZG_p}{NZ_p}}\sqrt{\frac{G_r Z_r}{2Z}}[v_I(t) * h(t)] \\ &= \sqrt{\frac{G_p G_r Z_r}{NZ_p}}[v_I(t) * h(t)] \ . \end{aligned} \tag{5.22}$$

If we calculate the fourier transform of v_O, we get

$$V_O(f) = \sqrt{\frac{G_p G_r Z_r}{NZ_p}}[V_I(f)H(f)] \ . \tag{5.23}$$

The elaborate channel transfer function $\tilde{H}(f)$ is given by $V_O(f)/V_I(f)$ and calculates to

$$\tilde{H}(f) = \sqrt{\frac{G_p G_r Z_r}{NZ_p}}H(f) = \frac{L_0}{\sqrt{N}}H(f) \tag{5.24}$$

with $H(f)$ given by equation 5.15 and L_0 a constant describing the amplitude diminishment by the projector and hydrophone losses.

Attenuation

The ray tracing approach requires a model of the amplitude attenuation a ray undergoes during its propagation in the channel. [12] gives such a model by assuming an exponential decay of the wave intensity at point r compared with the intensity at some starting point r_o. This gives

$$I(r) = I(r_0)e^{-\alpha(r-r_0)}. \tag{5.25}$$

α is the damping coefficient depending on the sound frequency and the acoustic absorption of the material. Thus, α needs to be measured to get a good match of the simulation to a specific environment. With 5.25 we can calculate the attenuation $A(l_e, f)$ of an eigenray in the previous section to

$$A(l_e, f) = e^{\alpha(f)l_e}. \tag{5.26}$$

Celerity Profile

The speed of sound in a given water environment is not constant but depends on the temperature, the salinity and the water depth. 5.13 shows that an acoustic wave that hits the interface to a medium with different sound speed is bent in a direction to or away from the interface. If there is no discrete interface but a celerity gradient, the direction of the wave changes continuously until it reaches a region with constant sound speed. As visible in figure 5.3, this can cause serious problems in a communication link since there can be large shadow zones in front of an emitter where no acoustic wave is transmitted to. For example, consider the case of a lake where the sun has heated the water surface such that the water temperature gets significantly higher with decreasing depth. Since this leads to a higher sound speed in lower depths, all acoustic waves emitted by a sender situated slightly under the water surface are bent away from the surface. As long as no reflections from the lake ground arrive at the receiver, he is not able to receive any signals. Essebar et al. [16] made measurements of the celerity profiles in the Mediterranean Sea and illustrate the problem of the formation of shadow zones especially during the summer months.

There exist several empirical formulas to calculate the speed of sound in underwater environments. A simplified equation is given by Coppens [17]:

$$\begin{aligned} c(D,S,t) = c(0,S,t) + (16.23 + 0.253t)D + (0.213 - 0.1t)D^2 \\ +[0.016 + 0.0002(S-35)](S-35)tD \end{aligned} \tag{5.27}$$

$$c(0,S,t) = 1449.05{+}45.7t{-}5.21t^2{+}0.23t^3{+}(1.333{-}0.126t{+}0.009t^2)(S{-}35). \tag{5.28}$$

Here, $t = T/10$ with T the temperature in °C, S is the water salinity in parts per thousand and D is the water depth in kilometres. The formula is valid for a temperature range of 0 to 35 °C, for a salinity level of 0 to 45 parts per thousand and for depths between 0 and 4000 metres.

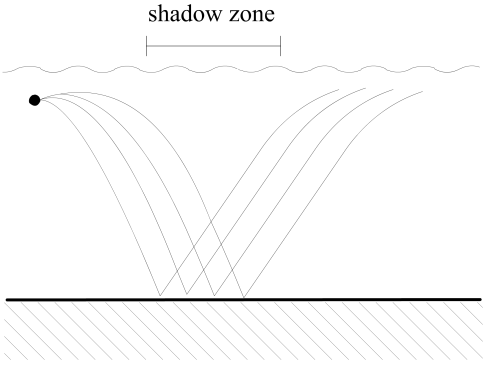

Figure 5.3: Shadow zone formed by celerity gradients.

Rayleigh Fading Channel

The described ray tracing method produces a channel model that is completely deterministic neglecting the fact that the identified eigenrays suffer from fluctuations at the receiver caused by temporal inhomogenities present along their paths. Sources of this inhomogenities are for example turbulences in the water and varying reflection coefficients and directions due to the movement of the water surface. It is not possible to modelise this fluctuations by determistic methods since they follow an unknown random distribution. Different approaches have been taken to account

for this problem and one of them is the Rayleigh fading channel model [14][18][19]. The Rayleigh fading channel model was originally introduced to model communications with electromagetic waves in environments with a very large amount of paths between the transmitter and the receiver like it can be found in urban regions. In fact, the Rayleigh fading channel model assumes that there is no dominant line-of-sight communication path but that a large number of the scattered paths influence the received signal in a similar way. The following formulae show the consequences of this assumptions [20]. First, a constant amplitude transmission signal is assumed:

$$s(t) = Re\left\{\hat{p}e^{j2\omega_c t}\right\} \tag{5.29}$$

with amplitude $\hat{p}$ and carrier angular frequency ω_c. Now the arriving signal composed by the large number of multipaths can be expressed as

$$r_s(t) = x(t)\cos(\omega_c t) + y(t)\sin(\omega_c t) = R(t)\cos[\omega_c t - \Phi(t)] \tag{5.30}$$

where

$$x(t) = \sum_n \hat{p}_n(t)\cos(\omega_c \tau_n(t)) \ , \tag{5.31}$$

$$y(t) = \sum_e \hat{p}_n(t)\sin(\omega_c \tau_n(t)) \ , \tag{5.32}$$

$$R(t) = \sqrt{x^2(t) + y^2(t)} \tag{5.33}$$

and

$$\Phi(t) = \arctan\left[\frac{y(t)}{x(t)}\right] \ . \tag{5.34}$$

$\hat{p}_n(t)$ is the amplitude and τ_n the time delay associated with path n and both are assumed to vary slowly compared with the carrier frequency ω_c. Under this circumstances it can be shown that $x(t)$ and $y(t)$ are independent random processes with a zero-mean Gaussian distribution. With that, the amplitude of the received signal $R(t)$ is Rayleigh distributed and the phase $\Phi(t)$ of this signal has a uniform distribution $p_\Phi(\phi) = 1/2\pi \ \forall \ \phi \in (-\pi, \pi]$.

If we look now at the original UWA channel with its set of distinct eigenpaths we could claim that each eigenpath produces a large number of subpaths on its way trough the channel. If this is true and the subpaths are generated in a way that there is no dominating path component, the channel could be viewed as a Multipath Rayleigh Fading channel with its eigenpaths that have all a rayleigh distributed amplitude and a normal distributed phase. With this, we have a model of the random fluctuations of the received signal.

Rice Fading Channel

Geng and Zielinski [20] state that the assumption of the existence of subpaths of each eigenpath holds, but - in contradiction with the multipath Rayleigh fading channel model - that these sub-eigenpaths are not always fully scattered. They believe that in the general case there is a dominant path component together with a number of smaller random subpath elements. Since the multipath Rayleigh fading channel model forms a special case of this assumption with the intensity of the dominant component equal to the intensities of the sub-eigenpaths, the Zielinski channel model is at least as good as the multipath Rayleigh fading channel model. Furthermore it seems reasonable to assume that in many cases the Zielinski model matches reality better than the multipath Rayleigh fading channel model. If we consider for example the case of short-range, point-to-point underwater communication

in an indoor pool, we could claim that the different eigenpaths undergo only small scattering due to the stable environment conditions and that therefore the dominant eigenpath components are large compared with the different sub-eigenpaths. Since the receiving statistics of a signal with a dominant component are significantly different from the statistics of a fully scattered signal, a formal description of the Zielinski model follows.

The transmitted signal is again 5.29, but with the dominant received component $S\cos(\omega_c t - \theta)$ the received signal can be expressed as

$$r_s(t) = S\cos(\omega_c t - \theta) + x(t)\cos(\omega_c t) + y(t)\sin(\omega_c t) \tag{5.35}$$

where x(t) and y(t) still have the form of 5.31 and 5.32, respectively. This could be rearranged to get

$$\begin{aligned} r_s(t) &= [S\cos(\theta) + x(t)]\cos(\omega_c t) + [S\sin(\theta) + y(t)]\sin(\omega_c t) \\ &= R(t)\cos[\omega_c t - \Phi(t)] \end{aligned} \tag{5.36}$$

with

$$R(t) = \sqrt{[S\cos(\theta) + x(t)]^2 + [S\sin(\theta) + y(t)]^2} \tag{5.37}$$

and

$$\Phi(t) = \arctan\left[\frac{S\sin(\theta) + y(t)}{S\cos(\theta) + x(t)}\right] . \tag{5.38}$$

Under these circumstances, the probability density of $R(t)$ calculates to

$$p_R(r) = \frac{r}{M^2}\exp\left[-\frac{r^2 + S^2}{2M^2}\right]\mathrm{I}_0\left(\frac{Sr}{M^2}\right) \tag{5.39}$$

where $\mathrm{I}_0(.)$ is the first kind, zero-order modified Bessel function and M^2 is the variance of the sub-eigenpath components, given by

$$M^2 = \mathrm{E}\left\{x^2(t)\right\} = \mathrm{E}\left\{y^2(t)\right\} . \tag{5.40}$$

The function $p_r(.)$ is called a Rice distribution, therefore we refer to a channel with this amplitude distribution as a Rice Fading Channel. At this point Geng and Zielinski introduce the signal-to-multipath ratio SMR, defined as the ratio between the amplitude of the dominant component and the standard deviation of the multipath components. Thus, the SMR is given by

$$SMR = S/M . \tag{5.41}$$

It can be shown that for small signal-to-multipath ratios $SMR \ll 1$, the amplitude has a Rayleigh distribution which confirms the earlier statement that the Rayleigh fading channel is an approximation of the Rice fading channel for very small dominant path components. For $SMR \gg 1$, the amplitude is Gauss-distributed with mean value S, showing that the subpath components act in this case like additive white gaussian noise. If we look at the phase $\Phi(t)$ of the received signal for a given value of θ, the distribution of $\Phi(t)$ can be given by

$$\begin{aligned} p_\Phi(\phi) =& \frac{1}{2\pi}\exp\left[-\frac{(SMR)^2}{2}\right] \\ &+ \frac{(SMR)}{\sqrt{2\pi}}\cos(\phi-\theta) \\ &\times \Psi[(SMR)\cos(\phi-\theta)] \\ &\times \exp\left[-\frac{(SMR)^2}{2}\sin^2(\phi-\theta)\right] \end{aligned} \tag{5.42}$$

where $\Psi(\xi) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\xi}\exp\left(-\frac{x^2}{2}\right)dx = \frac{1}{2}\left[1+\mathrm{erf}(\frac{\phi}{\sqrt{2}})\right]$. Again, a small $SMR \ll 1$ leads to a uniform distributed phase and a large $SMR \gg 1$ produces a gaussian phase distribution with mean value θ and variance $\left[\frac{1}{(SMR)}\right]^2$.

Estimation of the SMR parameter

The signal-to-multipath ratio of an eigenray depends on the length l of the ray and on the boundary reflections the ray experienced on its path through the channel. It is assumed that the SMR decays linearly with the number of involved ray reflections, thus

$$SMR_e = \frac{\widehat{SMR}(l)}{\sum_{q=0}^{Q_e} w_q} \tag{5.43}$$

where $\widehat{SMR}(l)$ represents the dependency of the SMR on the path length, Q_e is the total number of reflections of eigenray e and w_q, $q \in \{1,\ldots,Q_e\}$, is a boundary parameter called roughness associated with the reflection q and w_0 is the "roughness" of the wave emission by the sound projector. If we assume that $\widehat{SMR}$ is independent of the ray length, we get

$$\widehat{SMR}(l) = \widehat{SMR}_0 \tag{5.44}$$

Calculation of S and M

If I_e is the intensity of ray e, we can use 5.8 to get

$$I_e = \frac{S_e^2 + M_e^2}{2Z} \tag{5.45}$$

or equivalent

$$\hat{p_e}^2 = S_e^2 + M_e^2 \tag{5.46}$$

where $\hat{p_e}$ is the amplitude of eigenray e before the partition into S_e and M_e is done. Furthermore, if the $(SMR)_e$ is known, we have

$$S_e = M_e(SMR)_e\ . \tag{5.47}$$

This leads to

$$\hat{p_e}^2 = M_e^2[1+(SMR)_e^2] \tag{5.48}$$

and thus

$$M_e = \frac{\hat{p_e}}{\sqrt{1+(SMR)_e^2}} \tag{5.49}$$

$$S_e = (SMR)_e \frac{\hat{p_e}}{\sqrt{1 + (SMR)_e^2}} \tag{5.50}$$

where the positive solution of the quadratic equation 5.48 is taken since M_e and S_e are not allowed to be negative.

5.2 Channel Measurements

In order to be able to match the simulation to a real environment, measurements of the important channel parameters have to be done. In this section, the two basic parameters damping coefficient α and noise are measured. The intention of these measurements lies mainly in the goal to get a basic idea of the size of the involved parameters. These parameter estimations will then be used afterwards as starting values in a genetic algorithm that uses a large set of impulse response measurements of the channel to match the simulation to the reality as good as possible. With already predetermined values, the genetic algorithm will be much faster and the probability of finding the global minimum is increased drastically.

5.2.1 Measurement Channel

The channel in which the measurements were conducted equals one of the intended test environments for the communication system. It is a 30m × 1m × 1m indoor water channel with cement walls and floor. Since it is an indoor channel, surface movement is only small compared with outdoor channels. There are also nearly no sound celerity gradients due to the small channel dimensions. These factors lead to a very stable environment compared to other AUW channels like a lake or the open sea.

5.2.2 Measurement of the Damping Coefficient α and of the System Loss L_0

As 5.25 states, there is an exponential decay of the intensity of a sound wave with increasing distance r and decay constant α. Since α depends on the wave frequency, measurements of α have to be done by means of a sweep not only in distance but also in frequency. Afterwards, $\alpha(f)$ could be determined by interpolating the natural logarithm of the obtained intensity values both in the distance with a linear fit and in frequency using a custom fit matching the observed curvature. During the tests, not sound intensities were measured but amplitudes of sine waves. Since the signal amplitude is not equal to the sound intensity, this has to be considered in the intepretation of the results and in the calculation of α. It follows from 5.8 and 5.25 that the measured amplitude obeys

$$\hat{p} = 2Z\sqrt{I_0 e^{-\alpha d}} = 2Z\sqrt{I_0} e^{-\frac{\alpha d}{2}} \tag{5.51}$$

where I_0 includes both the projector and the hydrophone losses.
Thus, it follows that

$$\ln \hat{p} = \ln 2Z\sqrt{I_0} - \frac{\alpha d}{2} \,. \tag{5.52}$$

Using 5.52, we can obtain $\alpha/2$ out of the measured amplitudes by a linear fit through $\ln \hat{p}$.

Measurement Set-Up

The measurements were conducted using a function generator connected to a Reson TC4013 transducer as transmitter and a Aquarian H1a hydrophone connected to an oscilloscope as receiver. The transducer and the hydrophone were situated in the middle of the channel with distance d between them. Both the transducer and the hydrophone were positioned in the center of the channel's cross section. 60 measurements were conducted for distances $d \in \{0.1\text{m}, 0.2\text{m}, 0.4\text{m}, 0.8\text{m}, 1.6\text{m}, 3.2\text{m}\}$ and frequencies $f \in \{15\text{kHz}, 30\text{kHz}, 45\text{kHz}, 60\text{kHz}, 75\text{kHz}, 90\text{kHz}, 105\text{kHz}, 120\text{kHz}, 135\text{kHz}, 150\text{kHz}\}$. During the investigations always a sine wave with an amplitude of $\hat{v}_m = 10\text{V}$ and the specified frequency was generated in the function generator and the received amplitude was measured.

Results

After the completion of the measurements, for every frequency f out of the frequency sweep a set of 6 received amplitudes belonging to the different distances was generated. This amplitude set was then used to calculate the α-values for the different frequencies. Table 5.2.2 shows the results of this calculation.

Frequency	α	$2Z\sqrt{I_0}$
15kHz	5.6	0.11
30kHz	1.18	0.42
45kHz	1.06	0.39
60kHz	1.38	0.78
75kHz	0.94	0.86
90kHz	0.94	1.27
105kHz	0.90	1.28
120kHz	1.18	1.5
135kHz	0.56	0.83
150kHz	0.94	1.06

Table 5.1: Measured α and $2Z\sqrt{I_0}$ for different sine wave frequencies

A plot of α versus the different frequencies as in figure 5.4 shows that, except for the lowest frequency, α seems to be considerable constant. This is illustrated by the red linear fit in figure 5.4, for which the first measured point that belongs to the frequency of 15kHz was excluded.

The value of $2Z\sqrt{I_0}$ also presented in table 5.2.2 describes the measured pressure amplitude for the theoretical case of zero distance between transmitter and receiver. As the next section will show there are five distinct paths from transmitter to receiver in the used test setup. Three of them interfere constructively, one destructively and the last - the direct path - has a random phase compared with the other paths. Since the measurements were conducted on six different distances for every frequency step, the random contributions of the direct path tend to cancel each other out. Thus the ratio of $2Z\sqrt{I_0}/\hat{v}_m$ could be interpreted as a very rough estimation of $2L_0$ with L_0 presented in 5.24. It can be seen from the measurements that L_0 is highly frequency dependent.

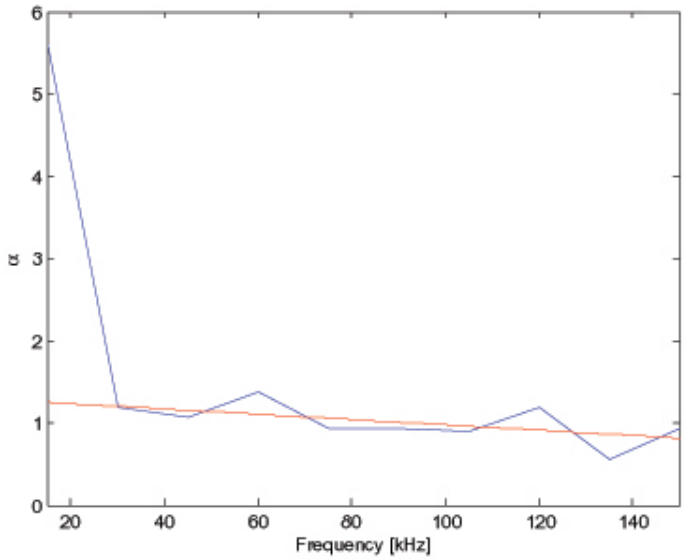

Figure 5.4: Plot of the damping coefficient α vs. frequency.

Discussion of the Results

The first thing to consider in the results is the large value of α obtained for the lowest frequency of 15kHz. This result has to be viewed with some reservation because in this low frequency region the resulting amplitudes are very low and sometimes even lie within the noise region. Thus, a reliable measurement of α in this frequency region was hardly possible.

Figure 5.5 shows the five different eigenpaths from the transmitter to the receiver in the test setup. Since the signal range of the transducer used in the tests was very low, no other reflected paths contributed to the received signal. Due to the high symmetry of the test setup, four of the five paths have the same length. Since three of these four paths are reflected by a cement boundary, they all have an additional phase shift of π and thus are in phase. The fourth path that is reflected by the water surface undergoes no phase shift and therefore has a phase difference of π with the other three paths of the same length. The last path is the direct or line-of-sight path and contributes to the overall received signal with an unknown phase. If we think of a sine wave of angular frequency ω_0 that is transmitted over the channel and we assume that there are no reflection losses, the received signal $v_O(t)$ calculates to

$$\begin{aligned} v_O(t) &= 3A\sin(\omega_0 t - \theta - \pi) + A\sin(\omega_0 t - \theta) + B\sin(\omega_0 t - \vartheta) \\ &= 2A\sin(\omega_0 t - \theta - \pi) + B\sin(\omega_0 t - \vartheta) \end{aligned} \tag{5.53}$$

where A and B are the amplitudes of the different paths that differ due to the different attenuation. We assume that the phase difference $\Delta = (\theta - \vartheta) \bmod 2\pi$ has a uniform distribution:

$$p_\Delta(\delta) = \frac{1}{2\pi} \;\forall\, \delta \in (-\pi, \pi]\,. \tag{5.54}$$

With that, the contribution of the direct path could be viewed as some sort of noise that cancels out with an increasing number of measurements. Therefore the measurements seem to be reasonable but at the same time it is obvious that they suffer

from large distortions. Thus in the following they are treated as rough estimations of the true parameters.

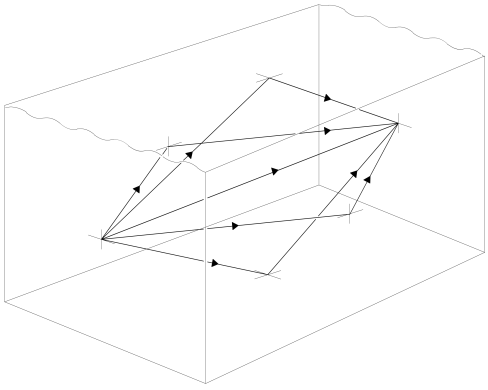

Figure 5.5: Different paths from emitter to receiver.

5.2.3 Measurement of a Set of Impulse Responses

In order to improve the matching of the simulation results to the real channel, the use of a genetic algorithm is proposed. This algorithm uses measured impulse responses and compares them with the simulation output for a given set of simulation parameters. After the comparison, a fitness value is assigned to the used simulation parameters. The algorithm then varies the parameters until a local fitness maximum is found. To be able to conduct the algorithm, a large set of measured channel impulse responses is needed. Thus it was one goal of the measurements to obtain such a set of impulse responses. Unfortunately, the measurements showed that the used transmitting device (the Reson TC4013) is not capable of transmitting enough power to create acoustic waves that travel far enough to provide a reliable impulse response of the channel with all its reflections. In fact, the transmitting range was around 5m, far away from the needed range of at least 200m. Therefore the measurement of the impulse response set is postponed to the future when a transmitter with larger range is available.

5.3 Matlab Implementation

In the following the implementation of the previously developed acoustic underwater channel model is discussed.

5.3.1 Concept

The focus of the present work lies in the design of an underwater communication system for short ranges in a pool-like environment. Since different underwater environments result in a completely different signal propagation behaviour there is no general simulation of the underwater acoustic channel that does best in all possible cases. In fact a specific simulation concept has to be derived from the general channel model in order to balance computing time and accuracy of the simulation.

For the present case of a pool-like environment the following assumptions were made:

1. The gradients in water density were assumed to be very small, thus the existence of water turbulences was negated.

2. The influence of vertical temperature inhomogenities was supposed to be neglectable.

3. Reflection losses were also neglected.

4. Encouraged by the data sheets the sound projector and hydrophone directivities were supposed to be isotropic.

5. Due to the small channel dimensions a very high number of distinct paths between transmitter and receiver was assumed.

6. The total noise as the sum of ambient and system noise was thought to be additive white gaussian with variance σ.

These assumptions had different consequences in the simulation development. As a first consequence, due to 1. and 2. the sound celerity was chosen to be homogeneous, thus the implementation of a celerity profile could be left away leading to a much easier and with that a much less computation intensive ray propagation algorithm. This gave space to a use a large number of rays, which - motivated by 5. - seemed to be necessary. The assumptions of 3. and 4. drastically reduced the channel parameter space and thus facilitate the future use of a genetic algorithm to enhance the simulation.

Boundary

Most of the underwater acoustic channel simulations developed in the past dealt with open sea applications and could therefore use a 2D channel boundary. In the present case this wasn't possible since the lengths of the paths that are reflected by the channel side boundary lie in the same region as the direct path length. In order to keep the simulation as simple as possible no complete 3D boundary model was implemented but a 2D channel cross section with an additional thickness was used to create a 3D boundary that always has flat side walls. Figure 5.6 illustrates this procedure.

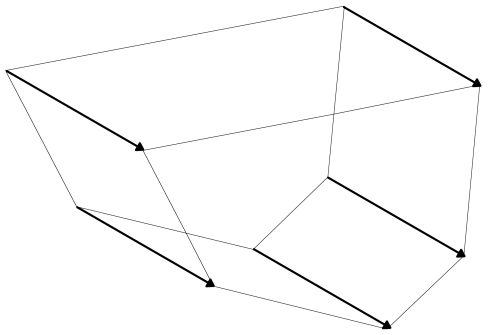

Figure 5.6: Generation of the boundary out of the single points.

Optimal Ray Depth

One of the critical simulation parameters is the optimal ray depth meaning the optimal maximum number of reflections a ray could experience. If the ray depth is chosen too small, some of the important paths between transmitter and receiver could not be identified. On the other hand, if the ray depth is too large valuable computation time is lost.

To determine the optimal ray depth, as a first step the minimum edge length $l_{e,min}$ of all faces was used as an estimation of the minimum path length between two reflections. Then it was calculated how many reflections a ray could experience with the minimum reflection path length until the ray is attenuated to 1% of the line-of-sight path. Using equation 5.26 with l the path length of the ray with reflections and l_0 the path length of the line of sight path we get

$$\frac{e^{\alpha l_0}}{e^{\alpha l}} \doteq 0.01 \tag{5.55}$$

leading to

$$l = -\frac{\ln(0.01)}{\alpha} + l_0 \; . \tag{5.56}$$

Thus the optimal ray depth n calculates to

$$n = \left\lceil \frac{l}{l_{e,min}} \right\rceil \tag{5.57}$$

where $\lceil . \rceil$ rounds towards the next larger integer.

Monte-Carlo Simulation

Rather theroretic importance has the used Monte-Carlo approach in the simulation. With this approach a ray has not a deterministic number of reflections but decides randomly after each reflection whether it continues its path through the channel or not. The probability that a ray continues its path is thereby chosen in a way that the expectation value of the number of reflections a ray experiences equals the optimal ray depth. We have for the expected number of reflections with K a random variable representing the actual number of reflections

$$\mathrm{E}\{K\} = \sum_{k=0}^{\infty} (k+1) p_c^k = \frac{1}{(1-p_c)^2} \tag{5.58}$$

where p_c is the ray continuation probability. Since $\mathrm{E}\{K\}$ should be equal to the optimal ray depth n, we get

$$p_c = 1 - \frac{1}{\sqrt{n}} \; . \tag{5.59}$$

By using this approach it can be guaranteed that an infinite number of rays will lead to the exact result. With a deterministic ray depth this wouldn't be the case.

5.3.2 Algorithm

Now in the following the algorithm to compute the ray model is presented. The computation of the channel transfer function is not included here since its implementation is straight-forward and the important theoretic aspects are discussed in the previous sections of this report.

The Algorithm in Pseudo-Code

Algorithm 1 illustrates the ray model computation by pseudo-code. As a first step a unit sphere is computed using a refinement of an icosahedron with N_s refinement steps. The points on the unit sphere are then used to determine the initial directions of the rays. Afterwards the environment is set up out of the boundary points and the sound projector and hydrophone positions. At this point the computation

intensive part begins. Each ray is pursued as long as a uniform on the interval $[0, 1]$ distributed random number is smaller than the continuation probability. If the ray hits the hydrophone it is stored and afterwards used in the calculation of the transfer function. Otherwise the next intersection of the ray with a boundary face is computed and set as the new ray position. At the same time following the law of reflection the new direction of the ray is determined. If all rays are terminated either by hitting the hydrophone or by exceeding p_c, the algorithm returns the paths and experienced reflections of the received rays.

Input: 2D boundary points, projector position $\vec{p}$, hydrophone position $\vec{h}$, ray continuation probability p_c, number of sphere refinements N_s
Output: receivedRays[ ]
Compute points *pts* on a unit sphere using N_s;
Set all ray positions $\vec{r_i}$ to $\vec{p}$;
Compute the initial ray directions $\vec{d_i}$ using *pts*;
Initialise environment;
foreach *ray* **do**
  while $randomNumber(0, 1) < p_c$ **do**
    if *hydrophone lies on ray direction* **then**
      Store ray with path and reflections in receivedRays[ ];
      break;
    end
    Compute nearest boundary interception $\vec{i}$ of ray;
    Set $\vec{r_i} = \vec{i}$;
    Compute new $\vec{d_i}$;
  end
end

Algorithm 1: Computation of the ray model.

Geometric Considerations

Some problems had to be solved concerning ray propagation. Since the sound projector was assumed to radiate in an isotropic fashion, the need after a sphere mesh with a homogeneous point distribution arose to be able to transmit an equal density of rays in all directions. This mesh was computed using an icosahedron, a platonic solid consisting of 20 equally sized triangles with vertices on the unit sphere. In order to get the desired density of mesh points, the edges of the triangles were halved and the resulting points again projected on the unit sphere. With this new set of points it was then possible to generate subtriangles of the original triangles. This procedure could be repeated as long as desired leading to an increasing number of points on the unit sphere. The number of these iterations is the number of sphere refinements N_s used in the previous section. Figure 5.7 shows the points generated out of an icosahedron together with the resulting subtriangles.

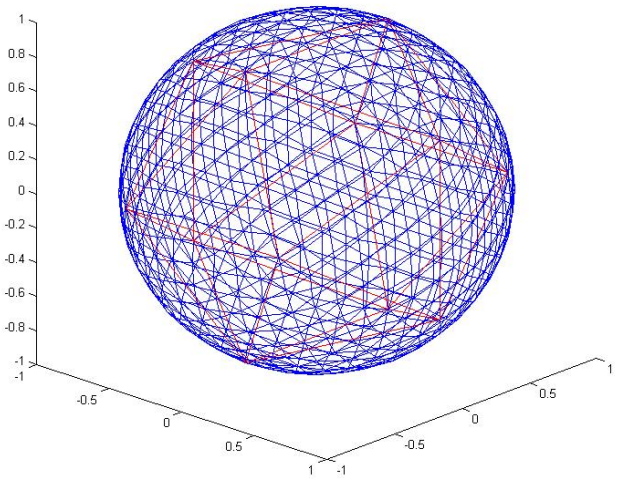

Figure 5.7: Generated sphere.

Another problem lay in the calculation of the reflection of a ray on a boundary face. This was solved by computing the projection of the negative of the incident ray vector $\vec{d}$ on the normal vector $\vec{n}$ of the boundary face and then determining the vector $\vec{a}$ visible in figure 5.8. With $\vec{a}$ added to the previously calculated projection we obtain the reflected ray direction vector $\vec{d'}$. If we formalize the problem we get

$$\vec{d'} = (-\vec{d} \cdot \vec{n})\vec{n} + \vec{a} \tag{5.60}$$

and with $\vec{a} = (-\vec{d} \cdot \vec{n})\vec{n} - (-\vec{d})$

$$\vec{d'} = 2(-\vec{d} \cdot \vec{n})\vec{n} + \vec{d}\,. \tag{5.61}$$

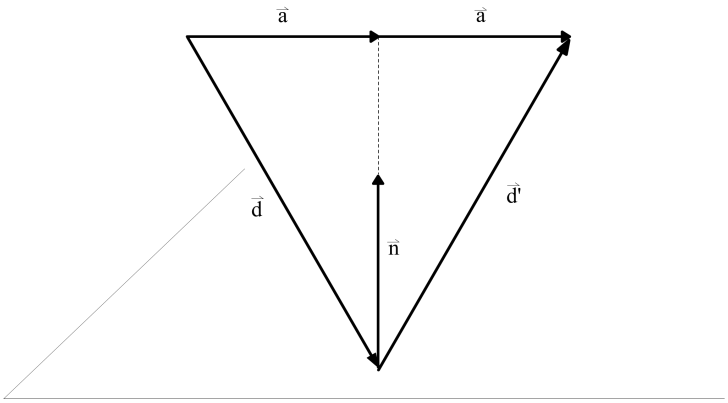

Figure 5.8: Calculation of the ray reflection.

5.3.3 Runtime Considerations

Originally the simulation was implemented by consequently employing object oriented programming. But investigations using the Matlab profiler showed that this proceeding leads to very slow running code. Therefore the ray model was reprogrammed using structs instead of objects which finally accelerated the simulation by a factor of about 100.

5.3.4 Sample Run

Figures 5.9 and 5.10 show the results of a sample run with attenuation $\alpha = 0.025$, system gain $L_0 = 0.05$ and sound speed $c = 1500\text{m/s}$ in a $30\text{m} \times 1\text{m} \times 1\text{m}$ channel with 10242 rays sent. In figure 5.9 the received rays are plotted together with the sound projector (red) and the hydrophone (green) position. It is visible from figure 5.10 that the absolute value of the transfer function is highly volatile due to the constructive and destructive interferences of the different eigenpaths. The fact that the upper and lower bounds of the fluctuations are also varying with frequency illustrates the impact of rice fading. As a last thing we can state that in reality the absolute value of the transfer function would decrease towards higher frequencies. In this simulation run this is not the case because a frequency-independent α was used.

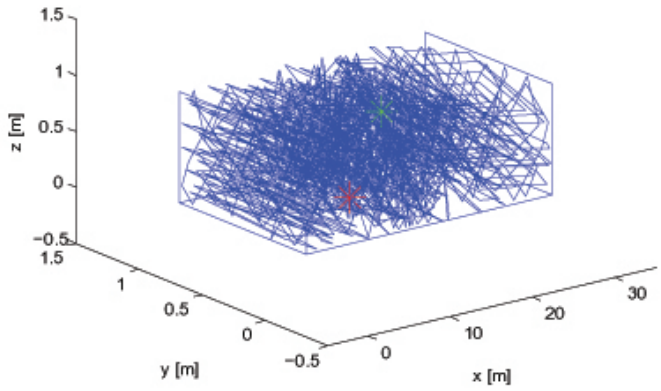

Figure 5.9: Received rays during one sample run of the simulation.

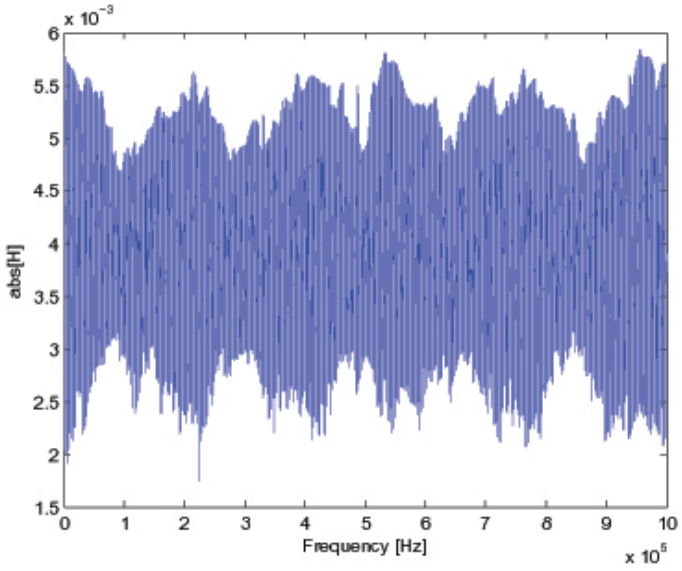

Figure 5.10: Absolute value of the calculated transfer function of one sample run of the simulation.

5.4 Improving the Channel Simulation

The developed simulation uses a set of parameters to describe the behaviour of the channel. These parameters are:

- Sound speed c
- Attenuation factor $\alpha(f)$
- System gain $L_0(f)$
- Boundary roughness vector $\vec{w}$
- Noise variance σ.

The exact measurement of these parameters is a hard task and has to be repeated for every new channel simulation with different environmental conditions. Thus at this point the use of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) together with a fitness function that determines the difference between a measured and a simulated impulse response is proposed to "learn" the parameter set. The CMA-ES algorithm is described in detail in [21]. Algorithm 2 provides a sketch of a fitness function usable with the CMA evolution strategy.

Input: Set of simulation parameters $\mathfrak{p}_{sim}$, Set of measured impulse responses $\mathfrak{h}$ with position parameters $\mathfrak{p}_{pos}$ and $|\mathfrak{h}| = M$
Output: Fitness f
Select a random element $h_{meas,i} \in \mathfrak{h}$;
Simulate the channel using $\mathfrak{p}_{sim}$ and $\mathfrak{p}_{pos,i}$;
Calculate the simulated channel impulse response h_{sim};
Calculate $f = \|h_{sim} - h_{meas,i}\|$;

Algorithm 2: Computation of the fitness of an impulse response.

5.5 Simulink Model

Figure 5.11 presents the Simulink model used to investigate the effects of the simulated channel transfer function $H(.)$. The model uses a source that generates random binary bernoulli distributed numbers and seeds them to a BPSK (binary phase shift keying) digital modulator. The modulator output forms a phase shift that is analogously modulated onto a 30kHz carrier represented by the sine and cosine blocks in the Simulink model. The digital and anolog modulated signal is then convoluted with the impulse response calculated by the channel simulation and additive white gaussian noise is added. The signal obtained at this point forms an estimation of the channel output. Now the demodulation begins with shifting the signal back to baseband, again by multiplying it with a 30kHz sine and cosine function. The resulting phase is presented to the demodulator that estimates the original transmitted bit. At the end, an oscilloscope visualises the channel output and another shows the difference between input and output, thus providing a tool to identify transmission errors.

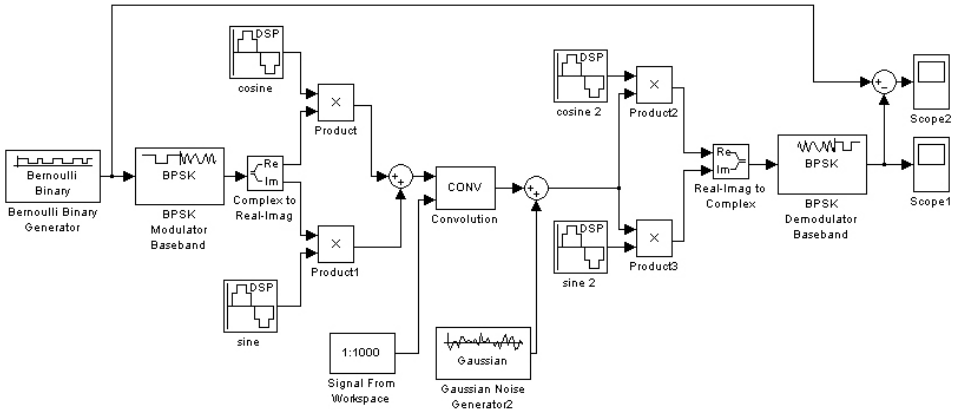

Figure 5.11: Simulink model to examine the channel simulation.

Chapter 6

Summary and Outlook

It is well known that the acoustic underwater channel poses great challenges on the design of a communication system, especially in the present case of high speed communication in a shallow-water channel. Thus during this first project part the focus of the work lay on one hand on the careful design of the receive and transmit circuitry and on the other hand on the development of a highly specialised simulation tool for stable shallow water channels with rather small dimensions.

With the present results, the second stage of the project will include the following tasks:

- Application of the CMA evolution strategy onto the simulation
- Implementation of the electronic design
- Refinement of the simulation using tests with the implemented electronics
- Development of powerful signal processing algorithms using the real-world environment and simulation results.

Chapter 7

Acknowledgement

The author would like to thank Jérôme Maye and Andreas Breitenmoser for their constructive and pleasant supervision. Furthermore a special thanks goes to Milian Seiler for helping out with some of the graphics and saving the author's neck.

Appendix A

Matlab Source Code of the Simulation

```
function H = UWAChannelSimulation()

    numberOfSphereRefinements = 2;
    points = [0 1; 0 0; 30 0; 30 1;];


    alpha = 1; %attenuation factor alpha

    bThickness = 1;
    pPos = [10 0.2 0.3]';
    hPos = [20 0.6 0.7]';
    hDia = 0.02;

    %determine optimal ray depth

    lmin = norm(points(end,:)-points(1,:)); %lmin = minimum edge length
    for i=2:size(points,1)
        if norm(points(i,:)-points(i-1,:))<lmin
            lmin = norm(points(i,:)-points(i-1,:));
        end
    end
    if bThickness<lmin
        lmin = bThickness;
    end

    l0=norm(hPos-pPos); %line-of-sight path length

    l = -log(0.01)/alpha + l0;

    n=ceil(l/lmin);

    rayContinuationProbability = 1-(1/sqrt(n)); %Probability, that ray
    %continues its way after hitting the boundary


    channel = UWAChannel(numberOfSphereRefinements);
    channel = channel.computeRayModel(points,bThickness,pPos,hPos,hDia, ...
                        rayContinuationProbability);


    %bParams(i) = [roughness of face i, phase shift at face i while
    %reflecting];
    %bParams(1) = [roughness of wave emission, phase shift of wave emission]
    bParams = [0.1 0;0.01 pi;0.01 pi; 0.01 pi; 0.2 0; 0.01 pi; 0.01 pi];
```

```
    %sound speed
    c = 1500;

    %system gain
    L0=0.05;

    channel = channel.computeTransferFunction(c,bParams,alpha,L0);

    H = channel.returnTF;
end


classdef UWAChannel

%Constants
properties(Constant = true)
    FREQUENCYBAND = 1000000; %1MHz
    NOTFVALUES = 1000000;
end
%Attributes
properties
    pPos = [0 0 0]';
    hPos = [0 0 0]';
    numberOfSphereRefinements = 0;
    sp; %points on unit sphere
    boundary;
    nOfRays = 0;
    receiptedRays = struct('path',[],'reflections',[]);
    H; %Channel transfer function
end


methods

    %Constructor Input:
    %nOfRays: Integer describing the number of eigenrays used
    function obj = UWAChannel(numberOfSphereRefinements)

        obj.numberOfSphereRefinements = numberOfSphereRefinements;
        obj = obj.computeSpherePoints();
        obj.nOfRays = size(obj.sp,1);
    end

    %Method Input:
    %bPoints: 2D vectors describing the boundary points, coordinates in meter
    %bThickness: Thickness of the boundary in meter
    %pPos: 3D Vector describing the position of the ultrasound
    %projector, coordinates in meter
    %hPos: 3D Vector describing the position of the hydrophone, coordinates in
    %meter
    %hDia: Diameter of hydrophone (approximated by a sphere)
    %rayContinuationProbability: Value describing the probability that a
    %ray continues after it hits a boundary face

    function self = computeRayModel(self,bPoints,bThickness,pPos,hPos,hDia, ...
    rayContinuationProbability)

        tic
        disp('Computing ray model ...');

        self.boundary = Boundary(bPoints,bThickness);
        self.pPos = pPos;
        self.hPos = hPos;

        recCounter = 0;
```

```
    for i=1:self.nOfRays
        %project rays on a sphere
        dir = self.sp(i,1:3)'; %initial ray direction

        %project rays on a circle
        %dirPhaseXY = ((i-1)/self.nOfRays)*2*pi;
        %dir = [cos(dirPhaseXY) sin(dirPhaseXY) 0]';

        pos = pPos; %initial ray position = projector position
        reflections = 0;
        numberOfReflections = 0;
        path = [];
        path(1,1:3) = pos';
        receipted = false;

        while rand(1)< rayContinuationProbability
            [newPos,newDir,planeID] = ...
            self.boundary.computeNearestIntersection(pos,dir,reflections(end));

            %Check whether ray collides with hydrophone
            h = hPos - pos; %vector from ray start point to hydrophone
            %angle between ray direction and h
            phi = acos(dot(h,dir)/(norm(h)*norm(dir)));
            %distance between hydrophone and ray
            d = sin(phi)*norm(h);
            %hydrophone should lie in the direction of the ray
            if (d ≤ hDia) && (abs(phi)<pi/2)
                %h is used instead of exact ray direction
                pos = hPos - (hDia/2) * (h/norm(h));
                path(end+1,1:3) = pos';
                receipted = true;
                break;
            end

            pos = newPos;
            path(end+1,1:3) = pos';
            dir = newDir;

            numberOfReflections = numberOfReflections + 1;
            reflections(numberOfReflections) = planeID;

        end

        if receipted == true
            recCounter = recCounter + 1;
            self.receiptedRays(recCounter).path = path;
            self.receiptedRays(recCounter).reflections = reflections;
        end

    end

    disp(['done: ' num2str(self.nOfRays) ' ray(s) sent, ...
    ' num2str(recCounter) ' ray(s) received']);
    toc
end



function self = computeSpherePoints(self)

    %ICOSAHEDRON (vertice density does not change on sphere surface)

    %Coordinates of icosahedron's vertices
    a = 0.850650808352040;
    b = 0.525731112119134;
    self.sp(1,1:3) = [ 0  b -a];
    self.sp(2,1:3) = [ b  a 0];
```

```
        self.sp(3,1:3) = [ -b  a 0];
        self.sp(4,1:3) = [ 0  b a];
        self.sp(5,1:3) = [ 0  -b a];
        self.sp(6,1:3) = [ -a  0 b];
        self.sp(7,1:3) = [ 0  -b -a];
        self.sp(8,1:3) = [ a  0 -b];
        self.sp(9,1:3) = [ a  0 b];
        self.sp(10,1:3) = [ -a  0 -b];
        self.sp(11,1:3) = [ b  -a 0];
        self.sp(12,1:3) = [ -b  -a 0];

        self = self.computeSubtriangles(0,self.sp(1,1:3), ...
        self.sp(2,1:3),self.sp(3,1:3));
        self = self.computeSubtriangles(0,self.sp(4,1:3), ...
        self.sp(3,1:3),self.sp(2,1:3));
        self = self.computeSubtriangles(0,self.sp(4,1:3), ...
        self.sp(5,1:3),self.sp(6,1:3));
        self = self.computeSubtriangles(0,self.sp(4,1:3), ...
        self.sp(9,1:3),self.sp(5,1:3));
        self = self.computeSubtriangles(0,self.sp(1,1:3), ...
        self.sp(7,1:3),self.sp(8,1:3));
        self = self.computeSubtriangles(0,self.sp(1,1:3), ...
        self.sp(10,1:3),self.sp(7,1:3));
        self = self.computeSubtriangles(0,self.sp(5,1:3), ...
        self.sp(11,1:3),self.sp(12,1:3));
        self = self.computeSubtriangles(0,self.sp(7,1:3), ...
        self.sp(12,1:3),self.sp(11,1:3));
        self = self.computeSubtriangles(0,self.sp(3,1:3), ...
        self.sp(6,1:3),self.sp(10,1:3));
        self = self.computeSubtriangles(0,self.sp(12,1:3), ...
        self.sp(10,1:3),self.sp(6,1:3));
        self = self.computeSubtriangles(0,self.sp(2,1:3), ...
        self.sp(8,1:3),self.sp(9,1:3));
        self = self.computeSubtriangles(0,self.sp(11,1:3), ...
        self.sp(9,1:3),self.sp(8,1:3));
        self = self.computeSubtriangles(0,self.sp(4,1:3), ...
        self.sp(6,1:3),self.sp(3,1:3));
        self = self.computeSubtriangles(0,self.sp(4,1:3), ...
        self.sp(2,1:3),self.sp(9,1:3));
        self = self.computeSubtriangles(0,self.sp(1,1:3), ...
        self.sp(3,1:3),self.sp(10,1:3));
        self = self.computeSubtriangles(0,self.sp(1,1:3), ...
        self.sp(8,1:3),self.sp(2,1:3));
        self = self.computeSubtriangles(0,self.sp(7,1:3), ...
        self.sp(10,1:3),self.sp(12,1:3));
        self = self.computeSubtriangles(0,self.sp(7,1:3), ...
        self.sp(11,1:3),self.sp(8,1:3));
        self = self.computeSubtriangles(0,self.sp(5,1:3), ...
        self.sp(12,1:3),self.sp(6,1:3));
        self = self.computeSubtriangles(0,self.sp(5,1:3), ...
        self.sp(9,1:3),self.sp(11,1:3));

        %Delete double points
        self.sp = unique(self.sp,'rows','first');
    end

    function self = computeSubtriangles(self,i,a,b,c)

        p1 = self.edgeMidpointOnSphere(a(1:3),b(1:3));
        self.sp(end + 1,1:3) = p1;
        p2 = self.edgeMidpointOnSphere(b(1:3),c(1:3));
        self.sp(end + 1,1:3) = p2;
        p3 = self.edgeMidpointOnSphere(c(1:3),a(1:3));
        self.sp(end + 1,1:3) = p3;

        if i< self.numberOfSphereRefinements
            self = self.computeSubtriangles(i+1,a(1:3),p1(1:3),p3(1:3));
```

```
            self = self.computeSubtriangles(i+1,p1(1:3),b(1:3),p2(1:3));
            self = self.computeSubtriangles(i+1,p2(1:3),c(1:3),p3(1:3));
            self = self.computeSubtriangles(i+1,p1(1:3),p2(1:3),p3(1:3));
        end
    end

    function p = edgeMidpointOnSphere(self,a,b)
        edgeMidpoint = (a' + b')/2;
        p = edgeMidpoint/norm(edgeMidpoint);

        p = p';
    end



    %Method Input:
    %c: Sound speed
    %bParams: boundary reflection parameters
    function self = computeTransferFunction(self,c,bParams,alpha,L0)

        tic
        disp('Computing transfer function ...');

        %Add all ray transfer functions of the eigenrays together and calculate Y
        f = 1:self.NOTFVALUES;
        f = f*self.FREQUENCYBAND/self.NOTFVALUES; %frequency in Hz

        self.H(1:self.NOTFVALUES) = 0;
        H_tot(1:self.NOTFVALUES) = 0;
        for i=1:size(self.receiptedRays,2)

            %calculate path length and path delay
            l = 0;
            for j=1:(size(self.receiptedRays(i).path,1)-1)
                l = l + norm(self.receiptedRays(i).path(j+1) - ...
                self.receiptedRays(i).path(j));
            end

            pathDelay = l/c;

            %compute transfer function of eigenray i

            %Estimate SMR
            roughness=bParams(1,1);
            for q=1:size(self.receiptedRays(i).reflections,2)
                planeId = self.receiptedRays(i).reflections(q);
                roughness = roughness + bParams(planeId+1,1);
            end
            SMR=1/roughness;

            %amplitude calculation

            p = (L0/sqrt(self.nOfRays)); %initial amplitude
                A = exp(alpha*l); %attenuation
            p = p/sqrt(A);
            %rice fading
            M = p/sqrt(1 + SMR^2);
            S = M * SMR;
            p_rice = ricernd(S,M); %Amplitude = sample from Rice distribution

            %phase calculation

            phi = -2*pi*f*pathDelay;
            %rice fading
            phi_rice = atan((S*sin(phi)+M*randn(1))./(S*cos(phi)+M*randn(1)));
            phaseDelay = exp(sqrt(-1)*phi_rice);
```

```
            %transfer function
            H_i(1:self.NOTFVALUES) = 1;
            H_i = p_rice*H_i.*phaseDelay;

            H_tot = H_tot + H_i;

        end

        self.H = H_tot;

        disp('done');
        toc
    end

    function Y = channelResponse(self,X)
        Y = X.*self.H;
    end

    function H = returnTF(self)
        H = self.H;
    end

end
end


classdef Boundary
    %Constants
    properties(Constant = true)
        MIN_X = 0;
        MIN_Y = 0;
        MIN_Z = 0; %Max z = thickness
        MAX_X = 10000;
        MAX_Y = 10000;
        ACCURACY = 8; %accurate digits after comma
    end
    %Attributes
    properties
        points = []; %Should be [x1, y1; x2 y2; ...]
        n = []; %Set of normal vectors
        thickness=0;
        numberOfPoints = 0;
    end

    methods

        %Constructor
        function obj = Boundary(points, thickness)
            for i=1:size(points,1)
                if points(i,1) > obj.MAX_X || points(i,1) < obj.MIN_X || ...
                points(i,2) < obj.MIN_Y || points(i,2) > obj.MAX_Y
                    error('Boundary point exceeds allowed values');
                end
            end
            obj.points = points;
            obj.thickness = thickness;
            obj.numberOfPoints = size(points,1);
            obj = obj.computeNormalVectors();
        end



        function points = get.points(self)
            points = self.points;
        end
```

```
function self = computeNormalVectors(self)

    for i=1:self.numberOfPoints
        if i==1
            p0_x = self.points(self.numberOfPoints,1);
            p0_y = self.points(self.numberOfPoints,2);
            p1_x = self.points(i,1);
            p1_y = self.points(i,2);
        else
            p0_x = self.points(i-1,1);
            p0_y = self.points(i-1,2);
            p1_x = self.points(i,1);
            p1_y = self.points(i,2);
        end

        %Compute plane normal vector
        %(point(i) - point(i-1)) x [0 0 1]' / norm( ... )
        self.n(i,1:3) = [p1_y - p0_y; -(p1_x - p0_x); 0]';
        self.n(i,1:3) = self.n(i,1:3)/norm(self.n(i,1:3)');
    end

    self.n(end+1,1:3) = [0 0 1]; %Side wall normal vector

end



function [intersection, reflectedDirection, planeIdentifier] = ...
computeNearestIntersection(self,rayStartingPoint,rayDirection,previousPlaneId)

    %initialise min_mu with maximal possible value
    max_mu = norm([self.MAX_X-self.MIN_X; self.MAX_Y-self.MIN_Y; ...
    self.thickness - self.MIN_Z])/norm(rayDirection);
    min_mu = max_mu;
    min_e = [0 0 0]';
    stuck = false;
    planeIdentifier = 0; %Integer describing the reflecting plane
    %(index of the second point or numberOfPoints + {1,2} in case of
    %reflecting side walls)

    pts = self.points; %use local array to make computation faster

    for i=1:self.numberOfPoints
        if i==1
            p0_x = pts(self.numberOfPoints,1);
            p0_y = pts(self.numberOfPoints,2);
            p1_x = pts(i,1);
            p1_y = pts(i,2);
        else
            p0_x = pts(i-1,1);
            p0_y = pts(i-1,2);
            p1_x = pts(i,1);
            p1_y = pts(i,2);
        end

        %Plane normal vector
        e = self.n(i,1:3)';

        %compute mu
        a = -dot((rayStartingPoint-[p0_x;p0_y;0]),e);
        b = dot(rayDirection,e);

        if abs(a)>10^-self.ACCURACY % a¬0 => rayStartingPoint lies on plane
            if b≠0 %b=0 => ray is parallel to plane
                mu = a/b;
```

```
                if (mu<min_mu) && (mu > 0) %no negative mu allowed since
                %only positive ray direction is of interest
                    %Check whether intersection lies within plane
                    %boundaries
                    inter = rayStartingPoint+mu*rayDirection;
                    %Check x boundaries
                    if(inter(1)≥(min(p0_x,p1_x)−10^−self.ACCURACY)) ...
                    && ( inter(1) ≤ (max(p0_x,p1_x)+10^−self.ACCURACY) )
                            %Check y boundaries
                        if(inter(2)≥(min(p0_y,p1_y)−10^−self.ACCURACY)) ...
                        &&(inter(2)≤(max(p0_y,p1_y)+10^−self.ACCURACY))
                                % Check z boundaries
                            if(inter(3)≥(self.MIN_Z−10^−self.ACCURACY)) ...
                            &&(inter(3)≤(self.thickness+10^−self.ACCURACY))
                                min_mu = mu;
                                min_e = e;
                                planeIdentifier = i;
                            end
                        end
                    end
                end
            else %if startingPoint lies on plane, check whether it is
            %the starting plane
                if i ≠ previousPlaneId %it is not the starting plane
                    min_mu = 0;
                    min_e = e;
                    planeIdentifier = i;
                    stuck = true;
                end
            end
        end


        %Check side walls

        %side wall normal vector
        e = self.n(end,1:3)';

        for i=1:2
            %compute mu
            a = −dot((rayStartingPoint−(i−1)*[0;0;self.thickness]),e);
            b = dot(rayDirection,e);
            if (abs(a)>10^−self.ACCURACY)   %a¬0 => rayStartingPoint lies on plane;
                if b ≠ 0   %b=0 => ray is parallel to plane
                    mu = a/b;
                    if (mu<min_mu) && (mu > 0) %no negative mu allowed since
                    %only positive ray direction is of interest
                        min_mu = mu;
                        min_e = e;
                        planeIdentifier = self.numberOfPoints + i;
                    end
                end
            else %if startingPoint lies on plane, check whether it is
            %the starting plane
                %it is not the starting plane
                if (self.numberOfPoints + i) ≠ previousPlaneId
                    min_mu = 0;
                    min_e = e;
                    planeIdentifier = self.numberOfPoints + i;
                    stuck = true;
                end
            end
        end


        if norm(min_e) == 0
```

```
                error('No intersection found');
            end

            if stuck == false
                reflectedDirection = 2*dot(-rayDirection, min_e)*min_e + rayDirection;
            else
                reflectedDirection = -rayDirection;
            end
            reflectedDirection = reflectedDirection/norm(reflectedDirection);

            intersection = rayStartingPoint+min_mu*rayDirection;
        end

    end
end
```

Bibliography

[1] R. Vahldieck and P. Leuchtmann: *Felder und Komponenten II - Skript zur Vorlesung im 4. Semester*. 5. Auflage, Institut für Feldtheorie und Höchstfrequenzelektronik (IFH), ETH Zürich, 2005.

[2] M. Siegel and R. W. P. King: *Electromagnetic Propagation Between Antennas Submerged in the Ocean*. IEEE Transactions on Antennas and Propagation, vol. 21, pp. 507-513, July 1973.

[3] Y. Liu and X. Ge: *Underwater Laser sensor network: A New Approach for Broadband Communication in the Underwater*. 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, pp. 421-425, May 2006.

[4] I. Vasilescu et al.: *Data Collection, Storage, and Retrieval with an Underwater Sensor Network*. Proceedings of ACM Sensys 2005, pp. 154-165, November 2005.

[5] H. Momma and T. Tsuchiya: *Underwater Communication by Electric Current*. OCEANS '76, pp. 24C-1-24C-6, 1976.

[6] J. Joe and S. H. Toh: *Digital Underwater Communication Using Electric Current Method*. OCEANS 2007, pp. 1-4, June 2007.

[7] M. Stojanovic: *On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel*. In WUWNet'06, Los Angeles, California, USA, 2006.

[8] A. I. Al-Shamma'a et al.: *Propagation of Electromagnetic Waves at MHz Frequencies Through Seawater*. IEEE Transactions on Antennas and Propagation, vol. 52, pp. 2843-2849, November 2004.

[9] A. Shaw et al.: *Experimental Investigations of Electromagnetic Wave Propagation in Seawater*. 36th European Microwave Conference, pp. 572-575, Manchester, UK, September 2006.

[10] Texas Instruments Inc.: *TMS320C6000 DSP Multichannel Buffered Serial Port (McBSP) Reference Guide*. http://focus.ti.com/lit/ug/spru580g/spru580g.pdf, December 2006.

[11] Texas Instruments Inc.: *Active Filter Design Techniques*. http://focus.ti.com/lit/ml/sloa088/sloa088.pdf, 2008.

[12] E. Hering, R. Martin and M. Stohrer: *Physik für Ingenieure*. 9. Auflage, Springer Verlag Berlin-Heidelberg, 2004.

[13] M. A. Munoz Gutiérrez et al.: *An Eigenpath Underwater Acoustic Communication Channel Simulation*. OCEANS, vol. 1, pp. 355-362, 2005.

[14] A. Bouzoualegh et al.: *Modelling and Simulation of Underwater Acoustics Communication based on Stateflow and Simulink Models.* SETIT 2005, Tunisia, March 2005.

[15] C. Bjerrum-Niese and R. Lützen: *Stochastic Simulation of Acoustic Communication in Turbulent Shallow Water.* IEEE Journal of Oceanic Engineering, vol. 25, pp. 523-532, October 2000.

[16] A. Essebar et al.: *Underwater Acoustic Channel Simulation for Communication.* OCEANS '94, vol. 3, pp. 495-500, September 1994.

[17] A. B. Coppens: *Simple equations for the speed of sound in Neptunian waters.* Journal of the Acoustic Society of America, vol. 69, pp. 862-863, 1981.

[18] R. Galvin and R. F. W. Coates: *Analysis of the Performance of an Underwater Acoustic Communications System and Comparison with a Stochastic Model.* OCEANS '94, vol. 3, pp. 478-482, September 1994.

[19] H. S. Chatha et al.: *Simulation Studies of Underwater Communication System in Shallow Oceanic Channel.* OCEANS '02, vol. 4, pp. 2401-2408, October 2002.

[20] X. Geng and A. Zielinski: *An Eigenpath Underwater Acoustic Communication Channel Model.* OCEANS '95, vol. 2, pp. 1189-1196, October 1995.

[21] N. Hansen: *The CMA Evolution Strategy: A Comparing Review.* Studies in Fuzziness and Soft Computing, Springer Berlin/Heidelberg, vol. 192, pp. 75-102, 2006.