ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

WS 2018-2019, MT-MA1, No. DISAL-SP124

Start: 18.09.2018
Finish: 11.01.2019

Underwater Acoustic Communication

Darko Lukic

Professor: Alcherio Martinoli
Assistant: Quraishi Anwar Ahmad

This page intentionally left blank.

Contents

Chapter 1 Introduction

Autonomous Underwater Vehicles (AUVs) are developed by DISAL[1] in the purpose of gathering environmental data in water bodies. The AUVs are equipped with a suite of sensors that measure various parameters eg. temperature, a concentration of various substances and turbidity. Multiple AUVs have to collaborate in order to localise and gather data more efficiently, therefore a proper communication between the AUVs is required. Since electromagnetic waves do not propagate in water, communication and localisation is a challenge. The AUVs have to periodically surface to receive a Global Navigation Satellite System (GNSS) update or to communicate with the base station.

Acoustic waves are a suitable alternative to electromagnetic waves for wireless underwater communication as they can achieve long ranges [Coa90]. However, communication-based on acoustic waves brings many challenges, such as frequency-dependent attenuation, multipath propagation and low speed of sound. As the result, underwater wireless communication based on acoustic waves has limited bandwidth and cause a signal dispersion in time and frequency [SP09]. Despite all those limitations, acoustic waves are the most promising solution for underwater communication. [JGWu15]

Taking it into consideration, DISAL developed hardware support for acoustic underwater communication. It consists of a custom built transducer, receiver and PCB with AVR 32bit microcontroller AT32 UC3C2512C[2] dedicated to signal processing. Therefore, the goal of this project was to study and explore possible solutions for underwater acoustic signal modulation and demodulation, simulate it, implement it to the microcontroller and measure the potential of the given hardware.

In Chapter 2 a basic background about communication will be described. Modulation and demodulation methods will be examined by putting the latest and the most common methods for underwater communication into the focus. The goal is to describe the advantages and disadvantages of each approach and to discover the best method for the required use-case.

The project consists of a few stages. In the first, a simple simulator is

---

[1] Distributed Intelligent Systems and Algorithms Laboratory at EPFL

[2] Datasheet of AVR AT32 UC3C2512C is available at http://ww1.microchip.com/downloads/en/DeviceDoc/doc32117.pdf

built in order to simulate a signal propagation through the water and in the simulator different modulation and demodulation strategies are evaluated. Therefore, in Chapter 3 a simulator will be described and how it helped to evaluate different modulation methods.

In the second stage, modulator, as a less complex component, is implemented in the microcontroller. And in the receiving side, the demodulator, from the simulator, is used to decode signal to corresponding data. This provides us with closely real data (transducer and receiver are located in a bucket of water) from hardware. In the final stage, the demodulator is implemented in the microcontroller and measurements are performed. By decoupling the project into multiple stages it enables us faster testing and it gave us a better understanding of the system. Both, modulator and demodulator, will be described in Chapter 4.

Performance of the implementation will be evaluated in Chapter 5. Based on the results and experience further research proposal will be given in Chapter 6.

Chapter 2 Background

Underwater acoustic communication has a growing trend in the past decades in research, as well as in engineering. It is lead by demand in commercial and military sector for real-time communication for submarines and autonomous underwater vehicles. All those applications require the development of a robust, fast and reliable underwater communication.

2.1 Communication Mediums

The acoustic waves are not the only medium for underwater communication. Before proceeding to the next section, it is important to compare acoustic waves to alternative mediums of communication.

2.1.1 Radio Waves

Radio waves are a widely used medium for real-time communication. Communication devices based on radio waves offer a robust, reliable and high-speed exchange of data. Unfortunately, radio waves don't propagate over long distance underwater.

Radio waves of lower frequencies, around 40 Hz - 300 Hz, can travel long distances through the water but it requires huge antennae and high transmitter powers. [Coa90] On the other hand, radio waves of 10 kHz can propagate the water only a few meters. [CWD$^+$10] This makes a wide variety of communication modules, based on radio waves (such as Wi-Fi which uses 2.4 GHz or 5.8 GHz radio bands), unusable for the most applications that require underwater wireless communication.

2.1.2 Optical Waves

One more medium for underwater communication, that is experimented with, are optical waves. Optical waves can propagate through the blue-green region quite good, but they are limited to a few hundred meters and they are affected by scattering. Depending on the type of source it can power be inefficient. Laser beams are power efficient but they require a high precision whereas diodes are high energy consumers. [FGS$^+$05a]

2.1.3 Acoustic Waves

Acoustic waves can propagate through the water many kilometres, up to hundreds of kilometres by using low frequencies. A power consumption of modules for underwater communication that use acoustic waves is also quite low. [FGS$^+$05b]

Frequencies and Distances

A distance that acoustic waves can carry a signal depends on a frequency and power of transducer. Therefore, it very important to provide an accurate dependency between of signal loss, distance and frequency.

In Milica's paper "Underwater acoustic communication channels: Propagation models and statistical characterization" this dependency is well described (Figure 2.1).

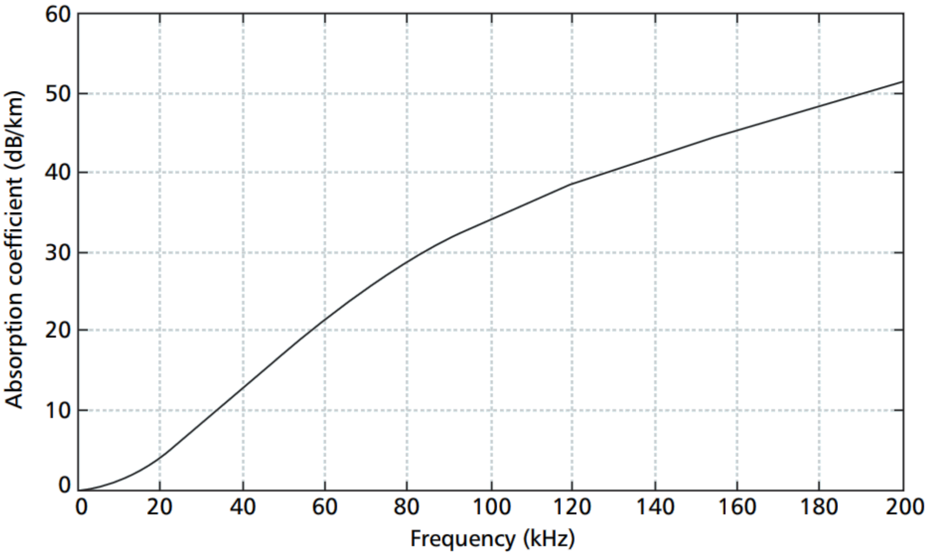


Figure 2.1: Absorption coefficient [SP09]

The results on ordinate (from Figure 2.1) are obtained according to:

$$A(l,f) = (l/l_r)^k * a(f)^{l-l_r} \tag{2.1}$$

where f is the signal frequency, l transmission distance, l_r reference point and k modes a spreading loss.

The signal loss at lower frequencies, around 30 kHz that many conventional underwater modems use, is significantly lower than at higher frequencies (eg. 200 kHz).

Speed of Acoustic Waves

A speed of acoustic waves in the water is about 1500m/s, depending on temperature, salt and other parameters. [CM77] It is very slow comparing to the speed of radio waves which travel at speed of light ($c_{light} = 299,792,458 m/s$), about 200,000 times slower. It makes acoustic waves sensitive to Doppler effect and scattering, which challenges engineers to use it as a medium for underwater communications.

To put this into the perspective, let's say we have the underwater vehicle moving 54km/h and transmitting acoustic waves. Frequency shift would be around 1% relative to a stationary receiver. It also means that a signal will be shifted about 10ms after the vehicle has moved towards referent receiver for 1s.

As the signal is slow, echoes can put an additional challenge to engineers. If the obstacles are only 100m away the echo can appear after about 130ms. That makes hard to differentiate old and new signal.

2.2 Modulation

Modulation is a process of changing one or more parameters of a signal in order to encode information. A device used for modulation is called modulator. It transmits a signal through some medium (eg. water) and demodulator extracts information from the observed signal in a process called demodulation.

Typical properties of signal that are used for modulation are amplitude, frequency and phase. For underwater applications, those are most often frequency and phase. As we will be talking about digital systems, the most often used modulation methods for acoustic underwater communications are PSK, FSK and QAM. [APM05] Variation and the combination of these methods are also possible and it potentially can increase the quality of communication.

Type	Year	Rate [kbps]	Band [kHz]	Range [km]
FSK	1984	1.2	5	3 (shallow water)
PSK	1989	500	125	0.06 (deep water)
FSK	1991	1.25	10	2 (deep water)
PSK	1993	0.3–0.5	0.3–1	200 (deep water) – 90 (shallow water)
PSK	1994	0.02	20	0.9s
FSK	1997	0.6–2.4	5	10 (deep water) – 5 (shallow water)
DPSK	1997	20	10	1d
PSK	1998	1.67–6.7	2–10	4 (deep water) – 2 (shallow water)
16-QAM	2001	40	10	0.3 (shallow water)

Table 2.1: Overview of underwater modulation methods during period of 1984–2001 [APM05]

Type	Rate [kbps]	Band [kHz]	Range [km]
FSK	1.2	5	3 (shallow water)
PSK	500	125	0.06 (deep water)
FSK	1.25	10	2 (deep water)
PSK	0.3–0.5	0.3–1	200 (deep water) – 90 (shallow water)
PSK	0.02	20	0.9s
FSK	0.6–2.4	5	10 (deep water) – 5 (shallow water)
DPSK	20	10	1d
PSK	1.67–6.7	2–10	4 (deep water) – 2 (shallow water)
16-QAM	40	10	0.3 (shallow water)

Table 2.2: Overview of underwater modulation methods [APM05]

In the table 2.2 there is a trend acquiring PSK over FSK as of the late 90s.

2.2.1 Phase-shift Keying

Phase-shift keying (PSK) modules a signal by shifting its phase.

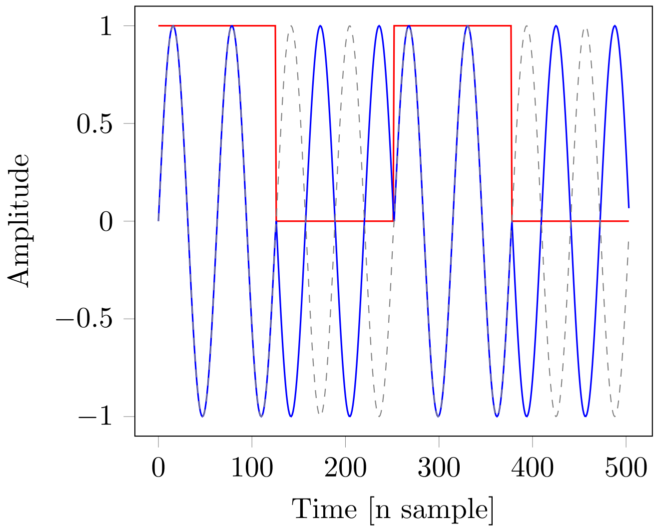


Figure 2.2: Example of PSK modulation. Blue line is a modulated signal, red line is a digital signal that has to be modulated and dashed grey line would be a signal if there is no PSK applied.

In the Figure 2.2 an example of PSK is presented. It shifts a phase of the signal for π radians if "1" has to be modulated and for 0 radians otherwise. PSK can use smaller phase shift in order to increase the speed of data transfer.

The signal from Figure 2.2 can be described by a simplified equation 2.2

$$s_n(t) = cos(2\pi f_c t + \pi(1 - n)) \tag{2.2}$$

Where, f_c is carrier frequency and n is a digital value ("0" or "1") that we want to modulate.

2.2.2 Quadrature Amplitude Modulation

Quadrature amplitude modulation (QAM) modulates a signal by the alternating amplitude of two carrier waves. The two carrier waves are the same frequencies with a phase shift of $\pi/2$ radians.

QAM modulated signal can be formulated as of equation 5.1.

$$s(t) = cos(2\pi f_c t)I(t) + cos(2\pi f_c t + \pi/2)Q(t) \tag{2.3}$$

Where $I(t)$ and $Q(t)$ are values we want to modulate.

Demodulation of QAM generated signal is by multiplying the signal by cosine (equation 2.4) and then applying a low-pass filter. All except $I(t)$ component will cancelled and modulate $I(t)$ can be extracted.

$$r(t) = s(t)cos(2\pi f_c t) \tag{2.4}$$

A similar approach can be applied for extracting $Q(t)$ component by multiplying signal by sinus (equation 2.5).

$$r(t) = s(t)sin(2\pi f_c t) \tag{2.5}$$

By plotting I and Q components the following graph can be obtained (Figure 2.3).

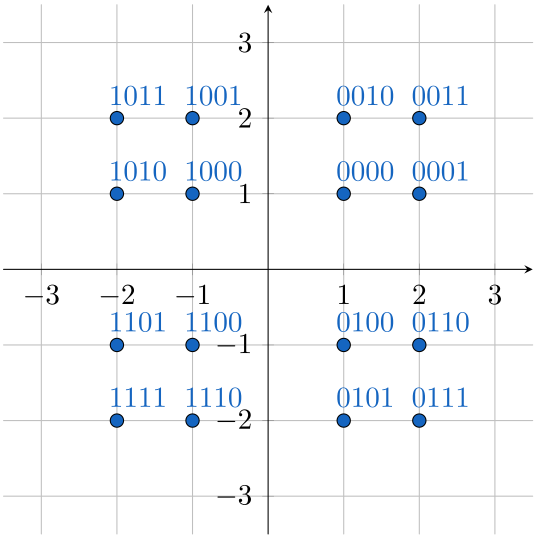


Figure 2.3: 16-QAM (4 bits per symbol)

2.2.3 Frequency-shift Keying

Frequency-shift keying (FSK) acquires a frequency of the signal in order to modulate a digital signal. It simply alternates a frequency in defined time frames depending on digital value.

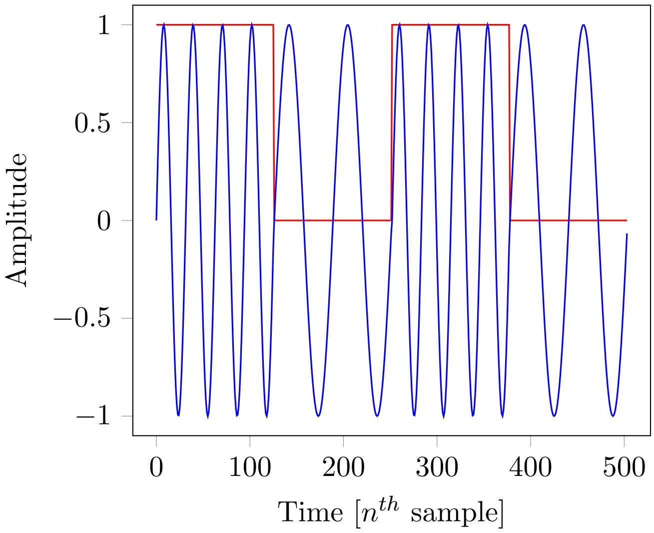


Figure 2.4: Like in the plot 2.2, blue line represents a signal in time domain and red line corresponding digital values

In the plot above 2.4, an example of FSK is presented. It uses two frequencies (BFSK) to modulate a signal. Demodulation is done by performing a Fourier transformation on predefined time windows.

Chapter 3 Simulation

In order to evaluate different modulation methods with, different parameters, a simple simulator is created. It aims to model a constrains of the given physical device as well as a noise in the water.

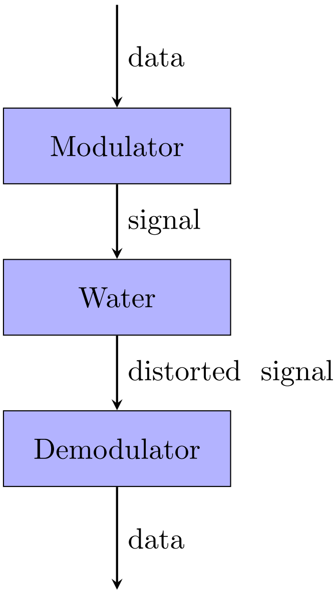


Figure 3.1: Available modules in the simulator and it's interconnections

As shown in the Figure 3.1, the simulator provides lightweight building blocks for modulation, demodulation and a model of the water.

3.1 Modulator

Modulator supports FSK (BFSK, QFSK and 8-FSK) and PSK modulation. Configurable parameters of FSK modulator are:

- **fs** – sample frequency (f_s),
- **symbol_length** – length of a single symbol (T_s),

- **constilation_schema** – number of bits per symbol (eg. BFSK, QFSK and 8-FSK),
- **pause_after_symbol** – pause after symbol and
- **frequency_range** – frequency range that can be used.

Based on a given frequency range and constellation schema the modulator will automatically pick the most suitable frequency carriers by spreading evenly across the given spectre. Pause after a symbol is implement in order to avoid interference of a signal with echoes at a receiving side.

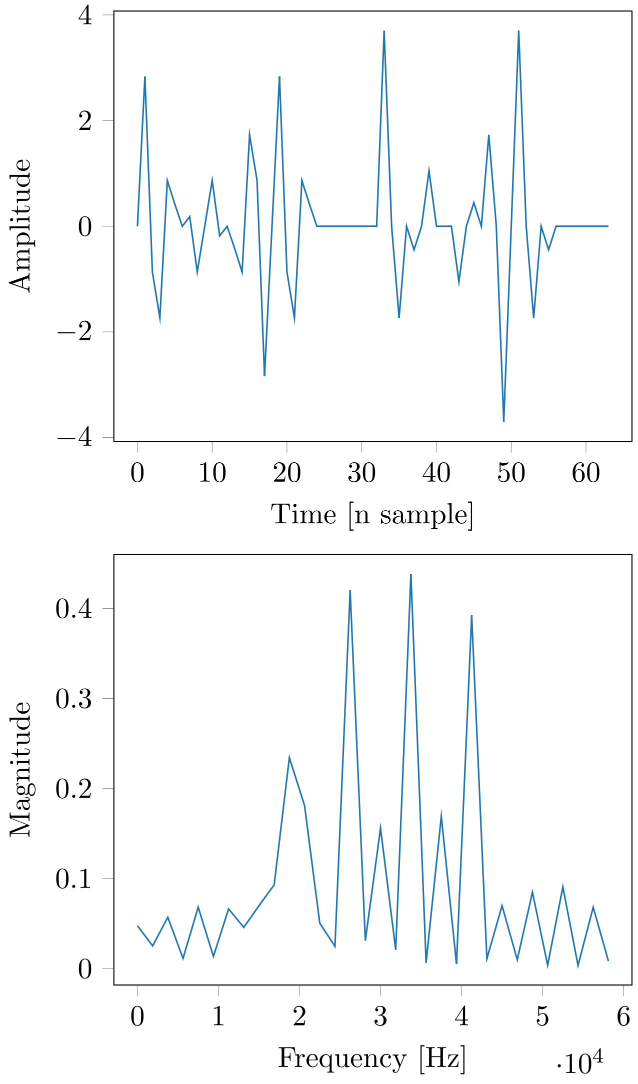


Figure 3.2: Sample output from modulator

In the Figure 3.2 a low sample rate is used, close to the Nyquist criterion, which gives sharp waves in the time domain.

3.2 Water

In order to simulate disturbances of the signal, water is modelled. It introduces a white noise, as well as echoes, to the signal which is passed through the model.

The following parameters of the water model are available:

- **noise_mean** – mean value of added white noise,
- **noise_standard_deviation** – standard deviation of the white noise,
- **echo_multiplication_factor** – weakening of echoes are described by this parameter which suppose to be less than 1,
- **echo_delay** – time needed of the signal to debounce of an obstacle is described by this value and
- **echo_max_n** – maximal number of repetitions of the same echo is limited by this parameter.

An example of output (in time and frequency domain) from a water module is given by Figure 3.3.

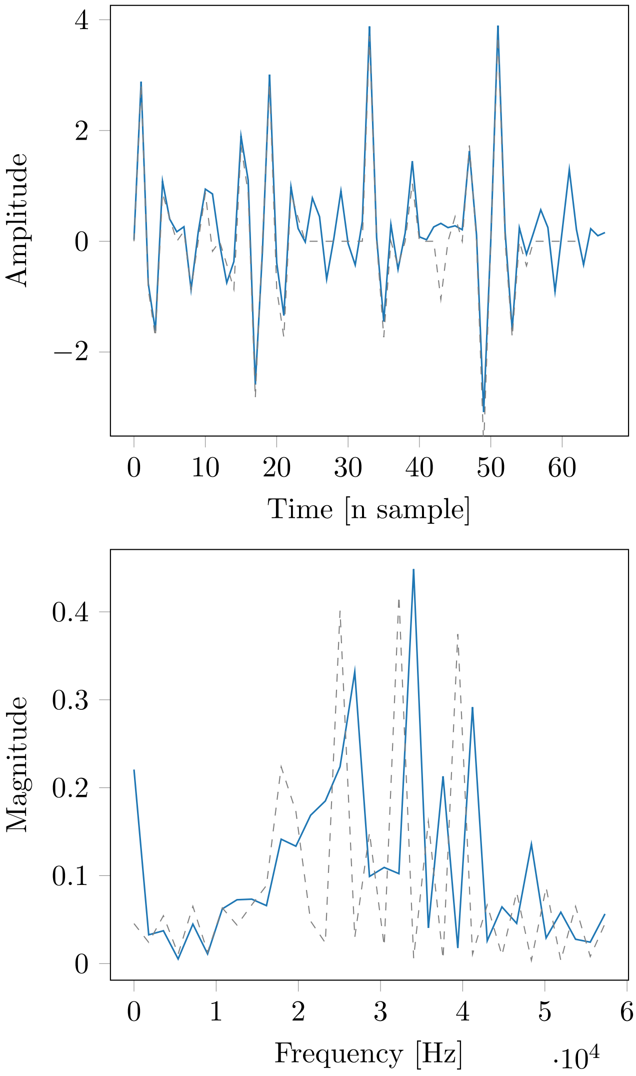


Figure 3.3: Signal from Figure 3.2 after white noise and echoes are added. Dashed grey line shows an expected signal if it wasn't disturbed by water

3.3 Demodulator

The demodulator is the most complex and the most important component in the simulator. It uses FSK in order to demodulate a signal. The following parameters are configurable:

- **fs** – sample frequency (f_s),
- **symbol_length** – length of a single symbol (T_s),

- **constilation_schema** – number of bits per symbol (eg. BFSK, QFSK and 8-FSK),
- **pause_after_symbol** – pause after symbol,
- **frequency_range** – frequency range that can be used and
- **frequency_deviation** – maximal allowed deviation from the carrier frequencies.

In order to demodulate the signal the following algorithm is applied:

Algorithm 1 Demodulator

```
procedure DEMODULATE
    repeat
        $signal \leftarrow ReadSignal()$
        $frequencies \leftarrow FFT(signal)$
        if $HasCarrierFrequency(frequencies)$ then
            $frequencies \leftarrow FilterByThreshold(frequencies)$
            $carrierFrequencies \leftarrow FindSimilarToCarriers(frequencies)$
            $symbol \leftarrow DecodeFrequencies(carrierFrequencies)$
        end if
    until end of simulation
end procedure
```

Chapter 4 Implementation

After the simulator is built and different modulation methods and configuration are evaluated, the most suitable approach is implemented to microcontroller. Implementation of the algorithms is performed in stages as there were a few challenges that had to be solved, the biggest of which are:

- debugger was not available,
- performance of the microcontroller are limited →limited number of FFT can be performed,
- communication speed with the microcontroller is limited →visualisation of all samples is not possible,
- memory management is not handled by programming language and
- integration with existing code base was required.

4.1 Development Environment

Development environment is adapted to fast prototyping (Figure 4.1). Both microcontrollers, one connected to receiving side and other one connected to transducer, communicates to PC over mavlink protocol [1]. [MCG+13] Mavlink is primarily used to change value of parameters, send sampled signal to PC and send debug information.

---

[1] "MAVLink is a very lightweight messaging protocol for communicating with drones" - https://mavlink.io/en/

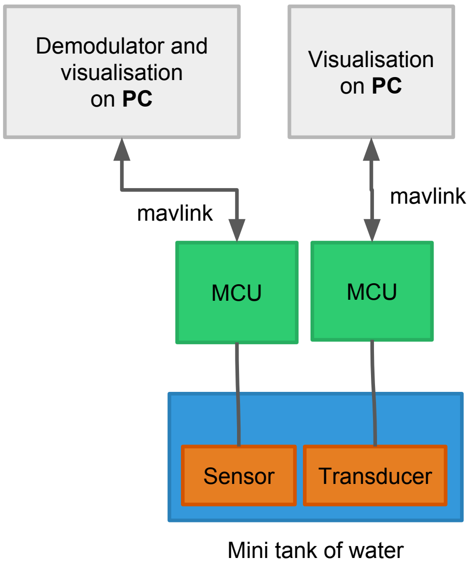


Figure 4.1: Environment for development and testing

In the first stage, demodulation was done in PC in which demodulator from simulator was used. After the algorithm is evaluated, demodulator is implemented on microcontroller.

4.2 Modulator

As the given system has highly precise epoch time measurement[2] and AUVs have to broadcast a message (unicast is not required) data is modulated and sent in the beginning of a second. This simplifies a communication as time synchronisation, on receiving and sending side, is done by the given system.

Initially, data is decoupled into stream of bits and multiple bits are assigned to symbols. The number of bits assigned to symbol corresponds to modulation schema, eg. 2 bits for BFSK, 4 bits for QFSK, 8 bits for 8-FSK... A modulated signal is generated by summing all corresponding carrier frequencies. Check equation 4.1 and algorithm 2.

$$s_{modulated} = \sum_{f_i \subset f_i, bit_i = 1} s(f_i) \tag{4.1}$$

Generated signal ($s_{modulated}$) is then loaded to a buffer in order to make it accessible to microcontroller's DAC module. Loading the signal to the buffer

---

[2] A precise time measurement is achieved by using PPS (Pulse-Per-Second) interrupt generated by GPS (Global Positioning System) receiver. Internal clock is acquired for time measurement between PPS' or in case of GPS signal loss.

takes about 40 µs for the given microcontroller, therefore the generated signal is shifted accordingly.

Algorithm 2 Modulator

```
procedure ModulateSymbol($symbol$)
    $s_{modulated} \leftarrow ZerosArray()$
    $bits \leftarrow DataToBits(symbol)$
    for $i$ in $1, 2...length(symbol)$ do
        if $bits_i = 1$ then
            $s_{modulated} \leftarrow s_{modulated} + GenerateSamples(f_i)$
        end if
    end for
end procedure
```

4.3 Demodulator

Demodulator is described by algorithm 1. However, the algorithm had to be extended and improved in order to work in the microcontroller. The algorithm is alternated because we had to consider:

- Trigger takes about 150 µs to detect beginning of a symbol (checking first 512 samples in array).
- FFT of 256 samples takes about 380 µs to calculate a signal in frequency domain.
- Required array shifting and copy new samples to buffer takes around 120 µs.

All those calculations take around 0.65 ms and they are applied on every 256 samples, therefore they have to be considered. On the other hand, as the microcontroller is configured to sample signal at 160 kHz, 256 samples are generated in 1.6 ms ($\frac{1}{160000} \times 256 \times 1000$).

Algorithm 3 Extended Demodulator

```
$symbolIndex \leftarrow 0$
$channelTriggered \leftarrow False$
$nextSymbolTime \leftarrow None$
procedure DemodulateTask
    if $SinceLastCheck() > 800\,\mathrm{ms}$ then
        $nextSymbolTime \leftarrow FindStartTime(buffer)$    ▷ Trigger demodulation if a sample is above a threshold
        if $IsFound(startIndex)$ then
            $symbolIndex \leftarrow 0$
            $channelTriggered \leftarrow True$
        end if
    end if
    if $channelTriggered = True$ then
        $startIndex = FindIndexFromTime(nextSymbolTime)$
        $Demodulate(signalfromstartIndex)$    ▷ See Algorithm 1
        $symbolIndex \leftarrow symbolIndex + 1$
        $nextSymbolTime \leftarrow nextSymbolTime + SYMBOL\_LENGTH + SYMBOL\_GUARD\_TIME$
    end if
    if $symbolIndex = N$ then    ▷ N depends on demodulation schema
        $channelTriggered \leftarrow False$
    end if
end procedure
```

Algorithm 3 provides optimised demodulation technique for the given microcontroller. Procedure “DemodulateTask” is executed by scheduler.

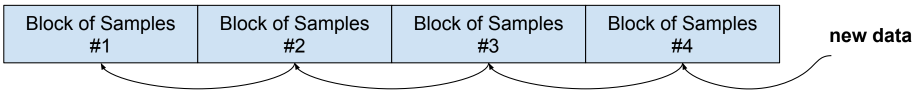


Figure 4.2: Adding data to buffer

Buffer of signal samples is given by two figures 4.2 and 4.3. The first figure 4.2 describes how samples are appended to array. The whole block of memory is shifted in a left by coping blocks.

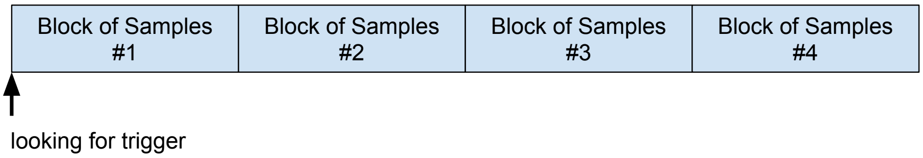


Figure 4.3: Searching for the first symbol

The second figure 4.3 shows that trigger sample, the beginning of a symbol, is searched from the beginning of buffer. Symbol is consider to be found if a value of a sample is above given threshold.

Chapter 5 Results

5.1 Modulator in MCU and Demodulator in Python

The acquired results are from the first stage of development in which demodulation is done on PC.

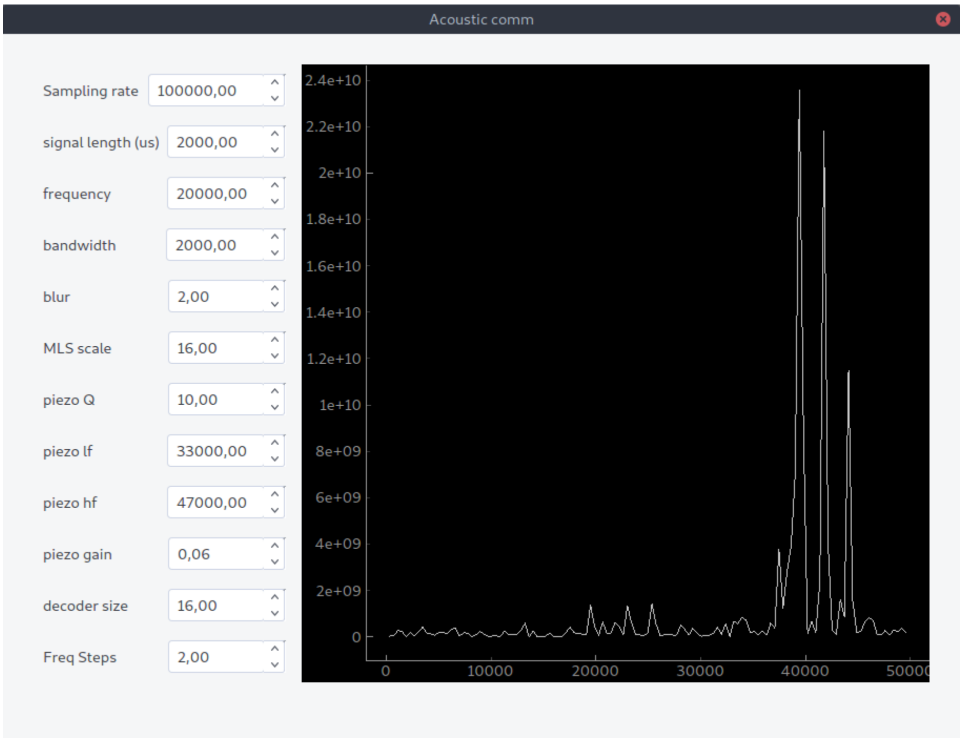
5.1.1 Snapshot of QFSK in Frequency Domain

The goal of the experiment was to measure a error rate of communication by using a demodulator on PC. The following two figures 5.1 and 5.2 show a response of a modulated signal in frequency domain.

Parameter	Value
Frequency carriers with amplification $f_{carriers}(amplification)$	42 kHz (1), 44 kHz (1), 46 kHz (1), 48 kHz (1)
Sample rate (f_s)	100 kHz
Samples per symbol	256
Symbol Guard Samples	2048
Allowed Frequency Deviation	1%
Threshold	300
Symbol Frequency Carrier	39 kHz

Table 5.1: Parameters used for the experiment

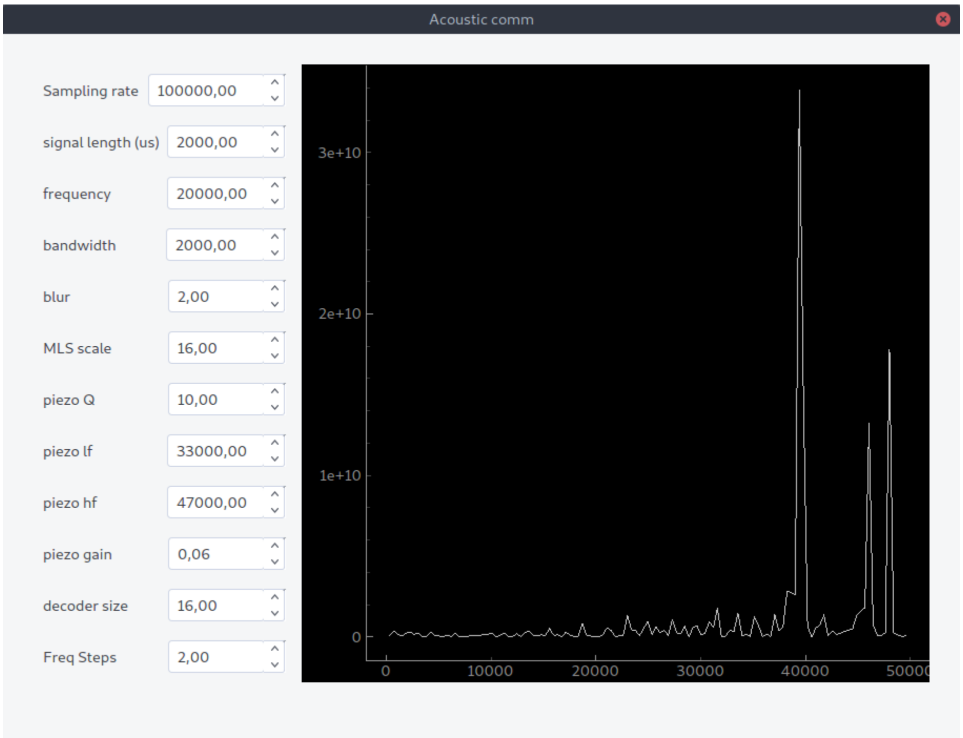
Parameters from table 5.1 are used for the error rate measurement. In the experiment 195 (binary 1100 0011) is used as data. The measured success rate was 149/150 (symbols successfully demodulated / symbols sent).



Sampling rate 100000,00
signal length (us) 2000,00
frequency 20000,00
bandwidth 2000,00
blur 2,00
MLS scale 16,00
piezo Q 10,00
piezo lf 33000,00
piezo hf 47000,00
piezo gain 0,06
decoder size 16,00
Freq Steps 2,00

Figure 5.1: Screenshot from Mavue software shows modulate symbol 1100

In the Figure 5.1 a first symbol (1100) in frequency domain is shown. Beside two carrier frequencies, 42 kHz and 44 kHz, symbol carrier frequency (39 kHz) is also notable.



Sampling rate 100000,00
signal length (us) 2000,00
frequency 20000,00
bandwidth 2000,00
blur 2,00
MLS scale 16,00
piezo Q 10,00
piezo lf 33000,00
piezo hf 47000,00
piezo gain 0,06
decoder size 16,00
Freq Steps 2,00

Figure 5.2: Screenshot from Mavue software shows modulate symbol 0011

In next Figure 5.1 a second symbol is also shown (0011) with symbol carrier frequency (39 kHz).

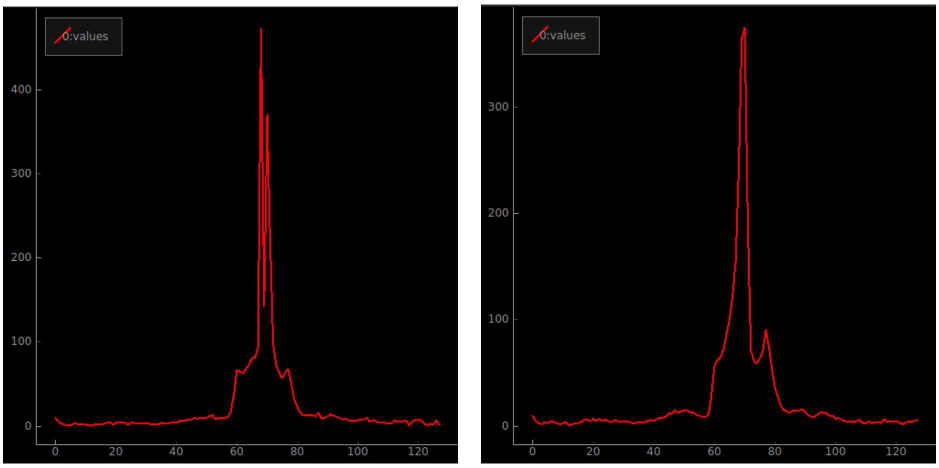
In both figures frequencies closer to (38 kHz) have greater magnitude because a resonant frequency of transducer is around (38 kHz).

5.2 Modulator and Demodulator in MCU

The following results are acquired by using demodulator implemented on microcontroller.

5.2.1 Minimal Frequency Difference

In order to maximise usage of the given spectrum it is important to put as many carrier frequencies as possible in the given spectrum. Therefore, minimal distance between two frequencies is investigated. The minimal difference is determined by size of window used by FFT (256 samples in our use-case), noise and missed samples.



(a) Captured signal of 42.5 kHz and 44 kHz on microcontroller

(b) Captured signal of 43 kHz and 44 kHz on microcontroller

Figure 5.3: Values on abscissa are not scaled, but represent real output from FFT implemented on microcontroller. There are 128 values as the FFT uses window of 256 samples.

Minimal distance between two frequencies in perfect conditions:

$$d_{minimal} = \frac{f_s}{N_{window\_size}} \tag{5.1}$$

if frequencies are chosen by $d_{minimal} \times N$ where $N \in \{1, 2, 3...\}$

As in real conditions there is a noise and frequencies can deviate (because of eg. Doppler effect) a safe distance for our case is about 1.5 kHz (see Figure 5.3). It means that in the available range, 39 kHz – 50 kHz, 8 carrier frequencies can be used, eg. 39 kHz, 40.5 kHz, 42 kHz, 43.5 kHz, 45 kHz, 46.5 kHz, 48 kHz and 49.5 kHz. By using 8 carrier frequencies per symbol, 1 byte would be able to be transmitted through 1 symbol.

Possible way to improve density of frequency carriers is given in chapter 6.

5.2.2 Demodulation Performance

The goal of the following experiment is to measure an error rate of symbol demodulation for QFSK.

Parameter	Value
Frequency carriers with amplification $f_{carriers}(amplification)$	40 kHz (1), 42 kHz (1), 44 kHz (1), 46 kHz (1)
Sample rate (f_s)	160 kHz
Symbol length	1.6 ms
Symbol Guard length	1.6 ms
Allowed Frequency Deviation	1%
Threshold	300

Table 5.2: Parameters used for testing demodulation performance

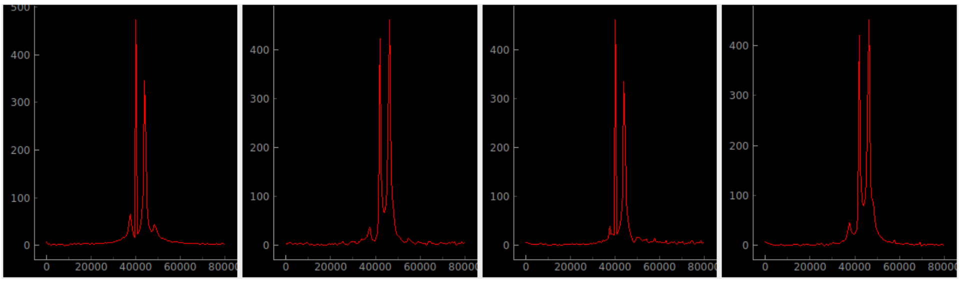


Figure 5.4: Signal in frequency domain modulated using the following binary data **1010 0101 1010 0101**

By figure 5.4 are shown 4 successive symbols in frequency domain. As described in chapter 4 symbols are sent after each PPS interrupt with equal pauses between each symbol.

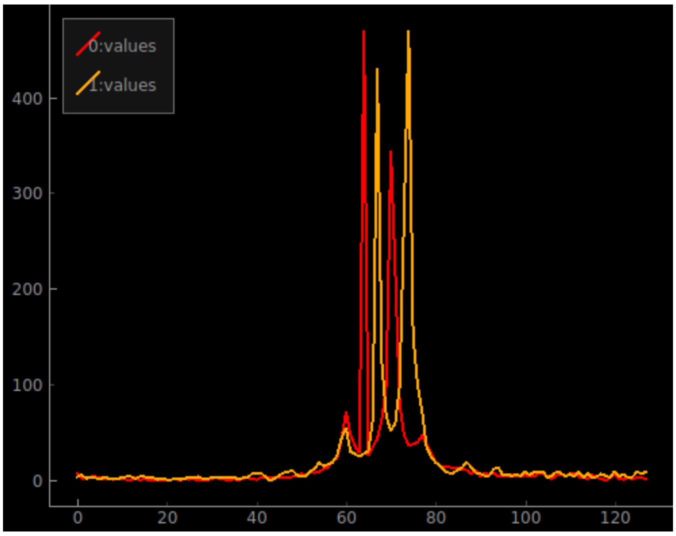


Figure 5.5: Two symbols presented in the same plot. By red colour modulated symbol **1010** (40 kHz and 44 kHz) is shown, and symbol **0101** by yellow colour (42 kHz and 46 kHz

During the experiment no error occurred for 152 transferred samples.

5.2.3 Distortion Caused by Water

In the following two figures (Figure 5.6 and 5.7) modulated signal by water in a bucket is shown.

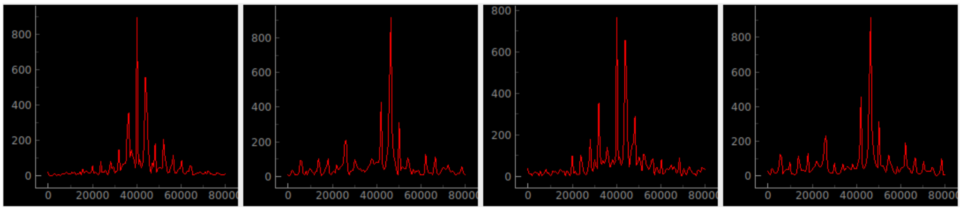


Figure 5.6: The same signal as in figure 5.4 distorted by water in a bucket

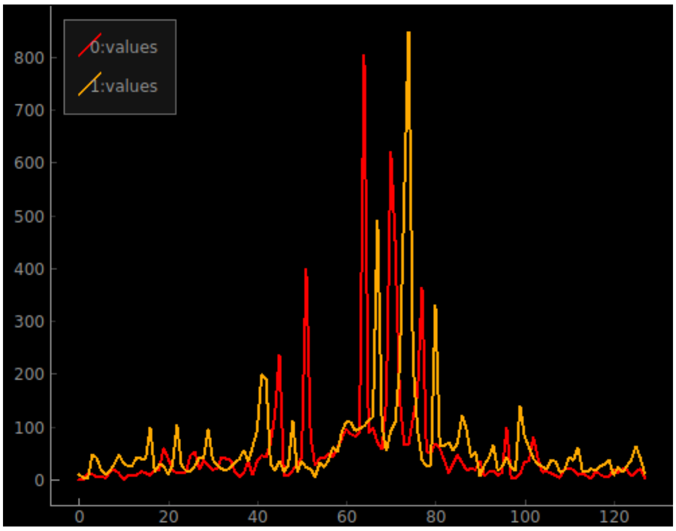


Figure 5.7: The same signal as in figure 5.5 distorted by water in a bucket

The bucket of water is small, therefore there are many echoes bouncing the water. The figures show a lot of noise in comparison to signal in figures 5.5 and 5.4. However, the noise is not strong enough to significantly disturb carrier frequencies.

5.2.4 Bitrate

According to parameters used for error rate measurement (Table 5.2) achieved bitrate is given by Equation 5.2.

$$\frac{N_{bits\_per\_symbol}}{t_{symbol\_length} + t_{guard\_delay}} = \frac{4}{3.2\,\mathrm{ms}} = 1.25kb/s \tag{5.2}$$

However, it has to be considered that the achieved sequence length is 8 symbols. Therefore, the bitrate can be calculated as of Equation 5.3.

$$\frac{N_{symbols\_per\_sequence} \times N_{bits\_per\_symbol}}{1\,\mathrm{s}} = \frac{32}{1\,\mathrm{s}} = 32b/s \tag{5.3}$$

In experiments (Subsection 5.2.1) it is shown that more frequency carries can be packed into the same frequency range. Therefore, the communication speed can be further increased.

Chapter 6 Further development

During development I discovered a lot of ways to improve current implementation. Unfortunately, because of time limitation I was not able to evaluate all approaches.

6.1 Downsampling to Use Wider Spectre

Currently, spectre between around 38 kHz and 50 kHz is used. Usage of frequencies out of the range is limited by various factors. It means that only around 25% of spectre is effetely used for demodulation.

By using downsampling technique more frequencies in the targeted range could be used for demodulation. [Mil09]

6.2 Improvement of Simulator

The simulator was a crucial part during development. Further development of the simulator would be very useful for more accurate testing of various configurations and demodulation methods.

The downsides of the current simulator are:

- It does not consider a resonance frequency of the transducer.
- Even if echos and noise are implemented it still does not faithfully simulates it.

6.3 Evaluation of QAM

QAM (Quadrature Amplitude Modulation) gives promising results according to other research (check table 2.2) Therefore, I believe more focus should be put into this method.

6.4 Implementation of Data Link

Currently, only a stream of symbols is exchanged between sender and receiver. There is no validation if data is transported correctly and there are no packets.

In a further development a new layer of communication should be introduced with the following features:

- Packetisation. Packing data as well as meta-data to packets.
- Different checksum sizes and different error-correcting codes should be evaluated.
- In addition, for more general purpose usage, addresses should be assigned to nodes.

Chapter 7 Conclusion

During this project, I have broadened my knowledge in digital signal processing. I have learned the advantages and disadvantages of different modulation and demodulation methods used for underwater communication. Also, I discovered a rising interest in this area of research as well as many challenges that the researches are facing with.

The implementation is done through a few stages which helped me to better understand the system as well as to easier recognise side-effects of acoustic communication (eg. echo or resonance). Simulator enabled me a much faster evaluation of algorithms and demodulation methods. It significantly reduced time, as different algorithms didn't need to be tested on a microcontroller.

Microcontroller brought various challenges, mostly related to memory management and processing time. There I learned some techniques on how to reduce memory usage and processing time, or at the least to postpone processing when the time is not critical.

The results showed successful communication between devices. However, more tests have to be performed, in which AUVs (with the acoustic modules) are moving at different speeds and in different environments. In order to increase robustness, more modulation and demodulation techniques should be investigated as suggested in Chapter 6.

Bibliography

[APM05] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater acoustic sensor networks: research challenges. *Ad hoc networks*, 3(3):257–279, 2005.

[CM77] Clarence Samuel Clay and Herman Medwin. Acoustical oceanography: principles and applications. 1977.

[Coa90] Rodney FW Coates. *Underwater acoustic systems.* Macmillan International Higher Education, 1990.

[CWD+10] Xianhui Che, Ian Wells, Gordon Dickers, Paul Kear, and Xiaochun Gong. Re-evaluation of rf electromagnetic communication in underwater sensor networks. *IEEE Communications Magazine*, 48(12):143–151, 2010.

[FGS+05a] Lee Freitag, Matthew Grund, Sandipa Singh, James Partan, Peter Koski, and Keenan Ball. The whoi micro-modem: An acoustic communications and navigation system for multiple platforms. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1086–1092. IEEE, 2005.

[FGS+05b] Lee Freitag, Matthew Grund, Sandipa Singh, James Partan, Peter Koski, and Keenan Ball. The whoi micro-modem: An acoustic communications and navigation system for multiple platforms. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1086–1092. IEEE, 2005.

[JGWu15] John Wiley & Sons 1999; revised John G. Webster, Ed. and updated 2015. *Underwater Acoustic Communications.* Encyclopedia of Electrical and Electronics Engineering, 2015.

[MCG+13] Lorenz Meier, J Camacho, B Godbolt, J Goppert, L Heng, M Lizarraga, et al. Mavlink: Micro air vehicle communication protocol. *Online]. Tillgänglig: http://qgroundcontrol.org/mavlink/start.[Hämtad 2014-05-22]*, 2013.

[Mil09] Ljiljana Milic. *Multirate Filtering for Digital Signal Processing: MATLAB Applications: MATLAB Applications.* IGI Global, 2009.

[SP09] Milica Stojanovic and James Preisig. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE communications magazine*, 47(1):84–89, 2009.