UNIVERSITÀ DEGLI STUDI DI PALERMO

Information And Communication Technologies
ING-INF/03-Telecomunicazioni
Dipartimento di Ingegneria

A Channel-Aware Adaptive System for Underwater Acoustic Communications

Ph.D. Candidate
Ing. GIOVANNI ETTORE GALIOTO

Coordinator
Prof. ILENIA TINNIRELLO

Tutor
Prof. ILENIA TINNIRELLO

Co-Tutors
Prof. CHIARA PETRIOLI
Ing. DANIELE CROCE

XXXIII CYCLE - ACADEMIC YEAR 2019-2020

Abstract

In recent years, underwater communications have seen a growing interest pushed by marine research, oceanography, marine commercial operations, offshore oil industry and defense applications. Generally, underwater communications employ audio signals which can propagate relatively far. However, acoustic underwater channels are very challenging, because of limited bandwidth, long propagation delays, extended multipath, severe attenuation, rapid time variation and large Doppler shifts. A plethora of underwater communication techniques have been developed for dealing with such a complexity, mostly tailoring specific applications scenarios which can not be considered as one-size-fits-all solutions. Indeed, the design of environment-specific solutions is especially critical for modulations with high spectral efficiency, which are very sensitive to channel characteristics. In this thesis, we design and implement a software-defined modem able to dynamically estimate the acoustic channel conditions, tune the parameters of a OFDM modulator as a function of the environment, or switch to a more robust JANUS/FSK modulator in case of harsh propagation conditions. To achieve this ambitious goal, various topics have been addressed and will be presented in this thesis.

We study how to compensate for the Doppler effect in transmission employing the JANUS standard, that is a popular modulation scheme for underwater communication developed by the NATO Center for Maritime Research. Although JANUS is resistant to multipath thanks to frequency hopping, a specific correction must be done to cope with the distortions caused by Doppler effect and make it a more reliable solution for harsh environments.

With the aim of developing an adaptive system, we implemented FLUMO, a flexible Software-Defined Acoustic (SDA) underwater modem. The system architecture follows two key ideas. First, the modulation/demodulation part runs fully in software and is completely decoupled from the rest of the system. The system can be used to send/receive modulated signals from different systems and, on the other side, it can be used to modulate signals that are sent from different systems. Second, we provide to the users a flexible and tunable system where the different communication parameters can be adjusted in order to reach the best performance in every scenario.

In order to develop a system supporting high data rate communication, we also implemented an OFDM transceiver which use a feedback chain to follow channel variations and correct the transmission recursively. We show the results obtained using the simulated channels in Watermark and how in these channels our technique achieves good performance compared to a traditional receiver. We also show realtime test performed in indoor (aquarium) and outdoor environments (at sea).

Finally, to allow our platform to decide between different solutions we address the problem of the channel characterization. Since the temporal variability of the channel behavior can be summarized in terms of maximum delay spread and Doppler spread, we present a solution for deriving these parameters and discuss the limit conditions under which the OFDM modulator can work. In such scenarios, we also calibrate the prefix length and the number of subcarriers for limiting the inter-symbol interference and signal distortions due to the Doppler effect. We validate our estimation and adaptation techniques by using both a custom-made simulator for time-varying underwater channels and the Watermark simulator, as well as real in field experiments. Our results show that, for many practical cases, a dynamic adjustment of the prefix length and number of subcarriers may enable the utilization of OFDM modulations in underwater communications, while in harsher environments JANUS can be used as a fall-back modulation.

Acknowledgement

I am deeply grateful to my tutors, Prof. Ilenia Tinnirello, Prof. Chiara Petrioli and Eng. Daniele Croce for their support during the development of this research project.

I would like to offer my special thanks to all members of the Telecommunications Lab of the Engineering Department of the University of Palermo for sharing their knowledge and experience with me and for helping me in many other ways.

I would like to express my sincere gratitude to WSENSE S.R.L. and all its members for their contribution in the realization of this research work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Humans have always been guided by the instinct to research their origins and overcome their limits. We have explored every far corner of the Earth and even the most distant planets. Despite this, we still do not know enough about our oceans and what lies beneath them. This is a particular knowledge gap, since 71% of Earth surface is water-covered and oceans hold about 96.5% of it. In the last decade underwater monitoring and exploration has been considered a crucial aspect for the economy and interests of nations.

Hence, there has been a rapidly growing research area on Underwater Acoustic Sensor Networks (UASNs), due to their wide applications in many underwater scenarios such as remote control in the offshore oil industry, pollution monitoring in environmental systems, collection of scientific data from ocean-bottom stations, disaster detection and early warning, assisted navigation, national security and defense, e.g. intrusion detection and underwater surveillance [1]. A fundamental feature for all of these applications is the increasing need of underwater communications technologies that enable collaborative monitoring tasks. This tasks are often performed by teams of heterogeneous mobile and static underwater devices.

Recently, several solutions, in the literature, have been proposed, at all layers of the protocols stack, to realize these technologies. In particular, a novel paradigm called the Internet of Underwater Things presented in [2], is emerging in the Blue Economy sector. Through low-cost technologies, typical of terrestrial IoT and personal devices, IoUT brings to marine and ocean environments the ability to sense, actuate and exchange information.

However, terrestrial wireless communication are based on Radio Frequency (RF) technologies. In underwater environments, common RF technologies show very limited performances because of the extreme attenuation of the medium. Thus, emerging industrial applications (such as submarine oil & gas and fishing industries) have pushed the development of new communication systems, to provide connectivity between underwater sensors, actuators and smart objects interconnected with each other and with the outside world. The research of new underwater wireless communication techniques has played an important role in the exploration of oceans and other aquatic environments.

Although transmissions of information can be carried out by means of acoustic, electromagnetic (EM) or optical waves, most of today's underwater communication systems make use of sound waves which have the advantage of significantly less absorption at low frequencies (around a dozen kHz) compared to, for example, optical communications. This allows to cover longer distances, though the bit rate is lower (dozens of kbps).

Indeed, underwater acoustic communications are very challenging, because the acoustic channel is characterized by limited bandwidth, severe attenuation (increasing with the signal frequency), long propagation delays due to the sound speed (about 1500 m/s), frequency dispersion, and time-varying multipath [3]. Sound speed profiles due to the heterogeneous water salinity increases the number of potential reflections, while surface waves, internal turbulence, fluctuations in the sound speed at different depths, and other small-scale phenomena contribute to random signal variations, which may strongly vary from one environment to another (e.g. from shallow waters, to pools or port environments).

1.1 Motivation and contribution

The large diversity of underwater wireless scenarios and environments is reflected in different varieties of acoustic modems and architectures which are present in literature and on the market. Most of the approaches that have been proposed can be roughly divided into two categories: i) very robust, low rate, modems reaching a rate in the order of hundreds of bits/s, exploiting long-lasting symbols and various types of spread spectrum modulations; ii) high speed modems for short/medium distance connections (tens to hundreds of meters), employing for example OFDM modulations and high frequency acoustic transducers operating in ultra-sonic bands (around 100 kHz or above), reaching nominal data rates of hundreds of kbps.

The first category of solutions work by using symbols whose duration is higher than the propagation delays of the most powerful reflected path. In some cases, gap intervals are added between consecutive symbols, in order to allow the extinction of the symbol reflection before transmitting a new symbol. Moreover, consecutive symbols can be transmitted by hopping from one carrier frequency to another, as done in JANUS [4], for leaving signal reflections in a sub-band different from the current used one and avoiding inter-symbol interference (ISI). Other spread-spectrum solutions devised to work in challenging environments exploit chirp-based modulations, such as the Sweep Spread Carrier (S2C) modulation used by Evologics modems [5].

The second category of solutions address the problem of inter-symbol interference and scarce bandwidth availability by using multi-carrier modulations, trying to achieve higher spectral efficiency. However, applying multi-carrier modulations such as OFDM to acoustic channels is a challenging task, because of its sensitivity to frequency offsets due to motion, currents, and clock stability of the nodes. In particular, because of the low speed of sound, motion-induced Doppler effects due to waves and node's drifting may result in critical problems, such as non-uniform frequency shifts across the signal bandwidth and inter-carrier interference (ICI).

In this work we evaluate to conjugate the two categories of solutions using a single SDA modem that is able to choice the better one as a function of environmental conditions. In other words, at the end of the work, the modem should be able to evaluate the channel condition and eventually use a reliable low rate solution only when the channel conditions do not allow to reach high speed.

To achieve this ambitious goal, various topics have been addressed and will be presented in this work. More specifically, the work contributions are summarized as follow:

- A method of estimation and compensation of Doppler effect in JANUS signals. Through the use of Cross Ambiguity Function (CAF) applied to the m-sequence of the JANUS preamble, the Doppler effect can be estimated without any modification to the standard. The model was tested both in Watermark simulated channel and in real at-sea experiments achieving performance until 90% of success in the former and 75% in the latter. This make JANUS a valid and reliable candidate to be used for low rate communication.

- An architecture for SDA modem that can easily change its configuration without the need to modify the underlying hardware. For example, the proposed modem can switch between modulation schemes or other characteristics like Forward Error Correction (FEC) scheme, interleaving, etc as a function of the environment. On the other hand, it can also drive third party hardware if this has some Software Defined Modem (SDM) capability.
- An OFDM transceiver with an enhanced a receiver based on a feedback chain to improve channel equalization. Through a comparison between received symbol and its regeneration, obtained after FEC decoding and remodulation, the channel estimate is updated. This new estimate is then used for correcting the next symbol in a recursively manner. This improved OFDM receiver, that was tested in Watermark, in aquarium environment and at sea, results in better performance compared to traditional receiver, generally.
- A method to obtain the delay spread and Doppler spread in order to characterize a channel. Through a periodic signal sent in the channel such as chirp train or pn sequences is possible to estimate the channel spreading characteristics. This allows to an intelligent module to understand if in this channel high data rate OFDM is feasible and size it eventually. Otherwise a fallback robust modulation like JANUS can be chosen to transmit. The method was validated trough a custom-made time-variant channel simulator and used both in simulation and at sea.

1.2 Structure of the thesis

The thesis is organized in chapters. Each chapter deal with different problems and aspects belonging to underwater communication systems, both low rate and high rate.

In the chapter 2 some background on underwater channels and the related work are presented. After a description of the Doppler effects concept, we introduce the general approaches present in literature to estimate and correct it. Then, an overview of existing underwater platforms are presented both for low rate and high rate OFDM communication. Finally we provide a brief description of Watermark, a realistic underwater channel simulator used in several parts of this thesis.

In the chapter 3 we present a method for estimating and correcting the Doppler effect in underwater communications using the JANUS standard. To demonstrate the effectiveness of the proposed method, we show the experimental results obtained through simulation tests using Watermark and real in-field experiments.

In the chapter 4 a software-defined approach to underwater acoustic modem is proposed. This approach allows to adapt the transmission in different channel conditions. This is possible because all the physical parameters of the transmission can be changed via software and do not require hardware changes. Although custom hardware is initially used as example, the software-defined nature of the proposed architecture allows to use generic third-party hardware.

In the chapter 5 the topic of the high rate modulation is dealt with particular reference to OFDM. First we provide a general description of the OFDM modulation. Then we describe the fundamental blocks composing the transmitter and the receiver, as well as the framing used in our OFDM implementation. Finally we present the experimental results obtained in the realistic channels of the Watermark simulator, as well as experimental results obtained in real indoor and outdoor environments, varying configuration parameters.

In the chapter 6 we propose an intelligent module for the estimation of the channel parameters and the optimization of the modem performance. Hence, we validate the propose method using a custom-made time-varying channel simulator and we apply the module on the Watermark channels and real at-sea channels. We show the experiments results obtained sizing the OFDM on the base of estimated channel parameters.

Finally, conclusions will be drawn.

Chapter 2

Background and related work

2.1 Doppler effect

Consider the case of a source transmitting a sine wave at frequency f_T and moving at constant speed v towards a fixed receiver as showed in figure 2.1. The Doppler effect will alter the wavelength perceived by the receiver: the sine wavelength as seen by the receiver will be decreased by the space covered by the source in a period or, in terms of frequency, the receiver will observe a greater number of wave fronts in the unit of time, i.e. the received frequency f_R will be greater than the transmitted frequency. If by convention we use a positive velocity v to indicate the direction of an approaching source and a negative velocity $-v$ to indicate the direction of a departing source, then we can express the frequency perceived by the receiver as $f_R = \gamma f_T$, where:

$$\gamma = \frac{c}{c - v} \tag{2.1}$$

and c is the speed of sound in water, approximately equal to 1540 m/s.

Now, depending on the fact that the transmitted signal can be considered narrowband or wideband, the Doppler effect must be differently modeled. For narrowband signals, the Doppler effect can be treated as a simple frequency shift of the whole signal spectrum (i.e. each component is shifted by the same amount of the carrier frequency shift). Instead, for wideband signals, Doppler effect influences differently each frequency component, so the effect is more accurately modeled as expansion/-compression of the signal waveform, characterized by a time scaling factor.

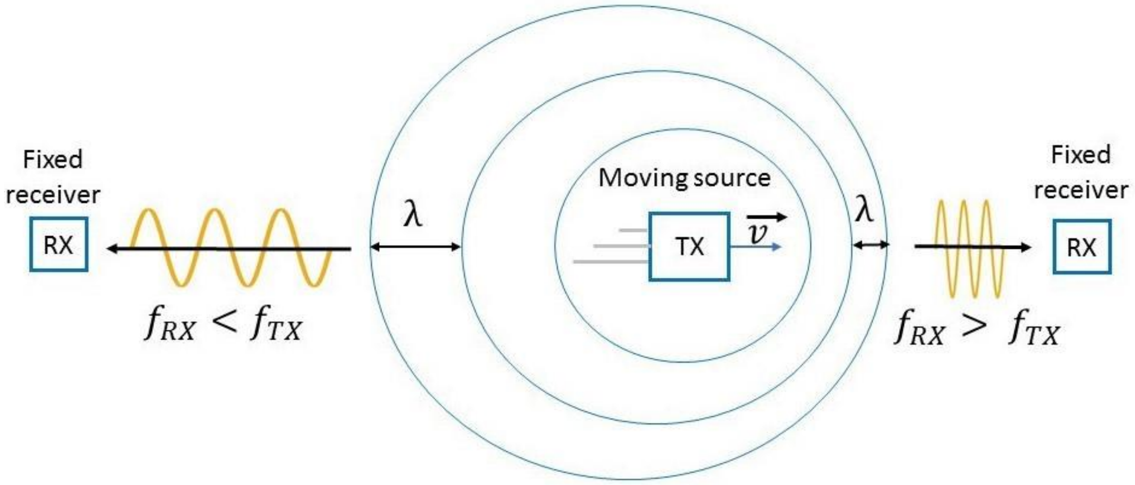


Figure 2.1: Representation of the Doppler effect.

2.1.1 General approach to Doppler correction

In the literature, several methods have been designed to estimate and compensate Doppler effects in underwater communications. Many of the proposed algorithms perform single or multiple correlation processes on known signals attached to the transmitted sequence to extracts the doppler estimation in term of compression/dilatation factor as well as a frequency shift. For example, in [6] the Doppler-induced time scaling distortion is estimated by applying a known waveform before and after the data packet so that, correlating the received packet with a copy of the known waveform, data packet duration can be obtained by measuring the time interval between the two correlation peeks. Furthermore, to minimize possible errors due to multipath copies of the transmission, an highly Doppler-tolerant waveform such as linear or log frequency-modulated signal is used. In [7] Doppler shift estimation is accomplished using the CAF method on a preamble (know training sequence) prefixed to the data. The obtained Doppler shift estimation is first used to shift the center frequency and a linear interpolation is then applied to correct the time scaling of the signal. Finally, a fine compensation is applied to remove residual Doppler effects and to improve the performance of the coherent receiver considered under test. A different approach is used in [8], where the estimate of the Doppler dilatation factor is accomplished by applying a Fourier analysis to three wake-up tones (preceding a JANUS packet).

2.1.2 OFDM-focused approach to Doppler correction

In the last decade, multi-carrier modulations, such as OFDM, have emerged as a promising solution for underwater communications [9], because of their robustness to channels which exhibit long delay spreads and frequency selectivity. However, these modulation solutions also have to specifically address the significant Doppler effect which affects underwater acoustic channels. Indeed, even for limited node motion and good clock stability, the low speed of sound and the fact that the acoustic bandwidth is not negligible with respect to the center frequency, make the Doppler effects critical. Different solutions have been proposed for addressing this problem, by designing receivers with some adaptation capabilities [10]. Adaptions can involve traditional approaches based on the choice of per-carrier constellations, such as the first scheme demonstrated in [11] or the On-Off keying solution for non-coherent receivers presented in [12], as well as adaptive coding techniques [13].

However, to achieve high reliability under general propagation conditions, Doppler estimation and compensation techniques are often required [14].

Different assumptions on the channel conditions lead to different complexity of the receivers. When the channel can be assumed quasi-static or with uniform Doppler shifts, an average Doppler scaling factor can be estimated with a given resolution by correlating special preamble signals with different distorted versions of the same known preamble [15]. The same assumptions were made in [16], where a single correlation is applied to a zero-padded (ZP) OFDM modulation. In this case, a preamble and a postamble surrounds the whole ZP-OFDM packet consisting of multiple OFDM symbols. In a first step, the scaling factor is again estimated by measuring the time duration between the preamble and postamble signals around the packet and a resampling is applied to compensate the coarse Doppler shift. Then, a fine compensation is carried out on each OFDM symbol to eliminate the residual Doppler shift.

Recent proposals have also addressed the case when the Doppler scaling cannot be assumed constant for the whole data packet. Indeed, with fast-varying movements, the Doppler estimation needs to be updated more frequently, even for each data symbol [17]. Per-symbol estimates are usually based on the evaluation of the cross-ambiguity function [18]. In [19] authors exploit the pilot signals, periodically inserted in the OFDM symbol, to compute the CAF. Since the CAF method is a computationally intense task, more recently multi-branch auto-correlation methods (MBA) have been devised in order to lower its complexity. In [20] the authors also exploiting pilot signal transmissions, and the results are compared with the ones in [19] and with single branch auto-correlation method (SBA). In the case of MBA, the Doppler effect is estimated using a single correlation on each sub-carrier. From the presented results, it is evident that the MBA and CAF algorithms provide comparable results, while SBA performs poorly.

2.2 SDA platform overview for underwater

Different types of underwater modems are present both in the literature and in the market. In this section we discuss the most relevant ones having similar characteristics to the one designed in this work.

2.2.1 Low data rate platforms

In [21], authors examine the performance of an underwater communication system that uses a common PC microphone as a receiver and the speaker of a Tmote Invent sensor as a transmitter. Using an 8-frequency FSK modulation, the system reaches a data rate of $48bps$ and a covered distance of $17m$.

An underwater sensor network hardware is proposed in [22]. It consists of static sensor nodes and mobile robots that are networked in a double configuration: optically, for point-to-point transmission at $330kbps$, and acoustically, for broadcast communications at $300bps$ up to $400m$.

In [23], authors propose an underwater acoustic modem for short-range and low data-rate sensor network applications. The modem reaches data rates up to $200bps$ with a 10^{-2} bit error rate over a signal to noise ratio (SNR) of $10dB$. Underwater tests show the modem can detect signals at distances greater than $350m$.

Micro-Modem [24] is another compact underwater acoustic communications system. It uses two modulation types: frequency-hopping frequency-shift keying (FH-

FSK), reaching low data rates of 80bps, and phase shift keying (PSK) obtaining variable data rates from 300 to 5000bps. The system can transmit in four different bands in the range from 3 to 30kHz, requiring a larger board for the lowest frequency.

In [25] is presented an underwater communication platform consisting in custom hydrophones interfaced with laptops that acts as SDA-based physical layer and allows higher layer protocols to interact via a plug-and-play interface. The platform uses an FSK modulation with center frequency of 16.5kHz and bandwidth 2kHz. The distances reached are between 5 to 10m with data-rates ranging 100 to 600bps.

The SeaModem described in [26] is an underwater acoustic modem that features a M-FSK (2, 4 and 8 tones) modulation with frequency band from 25 to 35kHz and data-rates of 750, 1500 and 2250bps depending on the number of tones used. A more advanced configuration of the modem is presented in [27], where the JANUS FH-BFSK scheme was added to the native M-FSK modulation. In the experiments, JANUS was observed to be quite robust against communication errors and was used to exchange first-contact information, later switching to the proprietary M-FSK physical layer. The SeaModem is reported to achieve a packet error rate below 0.1 at a distance below 400m.

2.2.2 Experimental platforms for high rate OFDM schemes

Nowadays, most commercial acoustic modems rely on fixed hardware designs and proprietary protocol solutions, designed for providing low-rate robust links and exposing a few configurable parameters. Only some recent modems support high-rate communications (up to hundreds kbits/s), and the possibility to switch among multiple pre-defined modulation schemes [28, 29]. Some models also permit the transmission of programmable waveforms, thanks to an hardware interface devised to load (and to read) the digital samples of a base-band modulated signal [30]. However, in this configuration, the base-band processing is performed in external devices e.g. using external libraries or software stacks (e.g. [31]).

Because of the limited flexibility of commercial solutions, most of the OFDM schemes designed for underwater environments have been experimentally validated by means of expensive research prototypes [32]. As an example, in [33], an OFDM base-band implementation based on a DSP board is presented, with a single-core CPU running at 1GHz, able to support modulations from QPSK up to 64-QAM and working with multiple transmitters. The aforementioned implementation achieves a data rate of 125kbps over a bandwidth of 62.5kHz, using a 16-QAM modulation and a coding rate of 1/2.

A complete implementation, including an embedded computer for high-layer processing, a DSP board for base-band processing and an FPGA board for pass-band processing is described in [34]. Different OFDM variants, which include incoherent and coherent OFDM schemes, are available for this implementation as DSP pre-configured modules.

A complete software-defined-acoustic-modem (SDAM) able to work at high-data rates is described in [35]. The model can seamlessly switch between different communication technologies such as OFDM and spread-spectrum modulations. At high rates, it can support up to 260kbps in real-time over a link of 200m, by adapting the modulation order and coding rate as a function of the channel conditions. For supporting this rate, the platform makes use of expensive hardware, including the USRP N210 (FPGA-based software defined radios), and a powerful PC for running the GNU Radio software.

The SEANet Project [36] aims to develop a SDM prototype with architecture both open hardware and open software. The hardware is based on a dual-core ARM Cortex-A9 platform and a Xilinx FPGA board. A Zero-Padded OFDM (ZP-OFDM) is implemented in FPGA and is able to reach a data rate of 596 kbit/s (with QPSK subcarrier modulation) using a transducer array over a bandwidth of 500kHz. Moreover, In FPGA the prototype also incorporates a FIFO and a mixer for both transmitter and receiver chains that allows to transmit/receive baseband samples of custom waveforms. The physical layer parameters can be also reconfigured in realtime.

2.3 Watermark

The underWater AcousTic channEl Replay benchMARK (Watermark) [37] is a realistic simulation tool that convolves user signals with at-sea measurements of the time-varying impulse response, thus providing a benchmark for physical-layer algorithms under realistic and reproducible conditions.

The simulator work on the base of the channel replay:

$$y(t) = \int_{-\infty}^{+\infty} \hat{h}(t,\tau)\, x(t-\tau)\, \mathrm{d}\tau + n(t) \tag{2.2}$$

where x(t) is the input signal, $\hat{h}(t,\tau)$ is the measured channel time-varying impulse response (TVIR), n(t) is a noise term, and y(t) is the distorted output signal. The maximum simulation time is bounded to the channel measurement time.

Receivers experience almost the same BER or SNR as a real in at-sea measurement. Indeed, channel replay includes generally all the typical channel effects such as non-stationarity or time-varying delays, except for the overspread channel because of difficult to measure it.

Watermark is issued with a library of five channels measured in different sites of the world. Figure 2.2 represents a picture of the measurements setup for the five available channels. More in detail, in Norway-Oslofjord (NOF1) and Norway-Continental Shelf (NCS1) (Norwegians channels) both the transmitter and the receiver are bottom-mounted. These two channels use a single hydrophone receiver and are then single-input-single-output (SISO). In the Brest Commercial Harbor (BCH1) (French channel) both the transmitter and the receiver are suspended in water close to the dock of the harbour. Since the receiver is composed by a hydrophone array this channel is single-input-multiple-output (SIMO). In Kauai site (KAU1 and KAU2) (Hawaiian channels) the transmitter is suspended in water and towed by a ship while the receiver is bottom-anchored and vertical suspended trough a buoy. Also in this setup the receiver is composed by a hydrophone array and the channel is SIMO type.

Table 2.3 summarizes all the characteristics of the channels. For each channel, the probe signal used fix the band and the maximum signal duration that can be used for the simulations. Two types of probing signal have been used: pseudonoise (m-sequence) and linear frequency modulated (LFM) pulses train. Denoting with fc the center frequency, B the bandwidth, T the duration, and u(t) the pulse shape function, a single pulse can be expressed as:

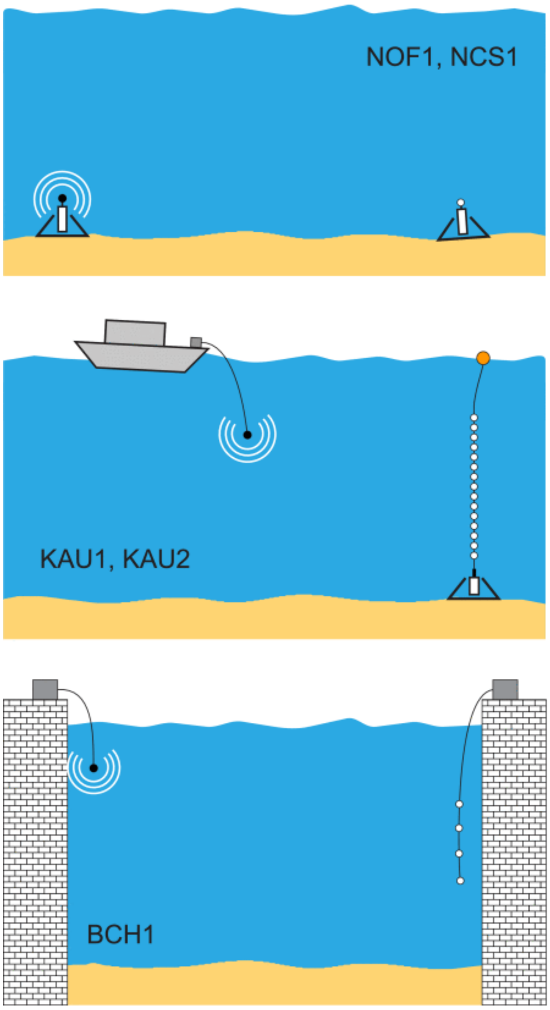


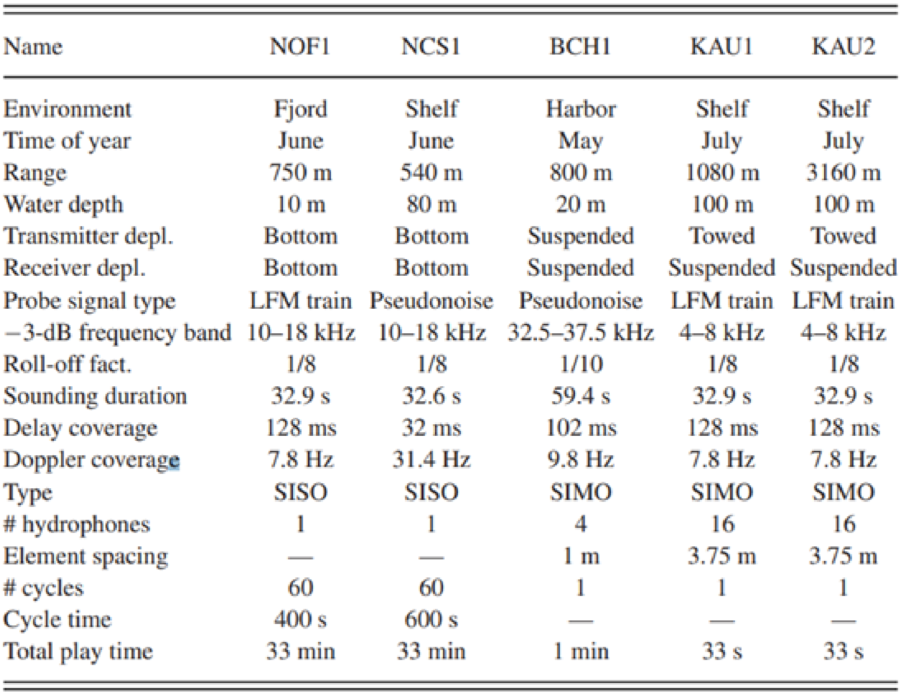
Figure 2.2: Deployment setup for Watermark channels sounding.

$$p(t) = sin(2\pi f_c t) \sum_{m=0}^{M-1} c_m u(t - \frac{m}{M}T) \tag{2.3}$$

and

$$p(t) = sin(2\pi[(f_c - \frac{B}{2})t + \frac{B}{2T}t^2]) \tag{2.4}$$

for the pseudonoise and for the LFM respectively. In both trains the pulses were transmitted without any pause between them. Naming with N the number of pulses in the train, in both cases the probing signal can be expressed as:



Name	NOF1	NCS1	BCH1	KAU1	KAU2
Environment	Fjord	Shelf	Harbor	Shelf	Shelf
Time of year	June	June	May	July	July
Range	750 m	540 m	800 m	1080 m	3160 m
Water depth	10 m	80 m	20 m	100 m	100 m
Transmitter depl.	Bottom	Bottom	Suspended	Towed	Towed
Receiver depl.	Bottom	Bottom	Suspended	Suspended	Suspended
Probe signal type	LFM train	Pseudonoise	Pseudonoise	LFM train	LFM train
−3-dB frequency band	10–18 kHz	10–18 kHz	32.5–37.5 kHz	4–8 kHz	4–8 kHz
Roll-off fact.	1/8	1/8	1/10	1/8	1/8
Sounding duration	32.9 s	32.6 s	59.4 s	32.9 s	32.9 s
Delay coverage	128 ms	32 ms	102 ms	128 ms	128 ms
Doppler coverage	7.8 Hz	31.4 Hz	9.8 Hz	7.8 Hz	7.8 Hz
Type	SISO	SISO	SIMO	SIMO	SIMO
# hydrophones	1	1	4	16	16
Element spacing	—	—	1 m	3.75 m	3.75 m
# cycles	60	60	1	1	1
Cycle time	400 s	600 s	—	—	—
Total play time	33 min	33 min	1 min	33 s	33 s

Figure 2.3: Characteristics of the Watermark channels.

$$s(t) = \sum_{n=0}^{N-1} p(t - nT) \tag{2.5}$$

Moreover probing signals were shaped through a root raised-cosine filter using a small roll-off factor (0.125), which lead to an almost rectangular spectrum.

The maximum delay spread and Doppler spread in the replay channels are determined from the duration of the probing signal pulse. The former is exactly the duration of the pulse while the latter is its reciprocal. These amounts represents the maximum distortions that can be affect the simulated output signals.

Chapter 3

Low rate solution

3.1 Overview and motivations

Acoustic communication is affected by frequency-selectivity, multipath and undesidered movements [38]. The latter phenomenon is typical of subsea channels since, in addition to deliberate node's movements, spatial and physical changes in the channel itself can cause severe Doppler effects. In fact, tides, currents and waves can cause the transceivers to move significantly. Thus, measuring and compensating Doppler effects is essential to create a reliable underwater communication system. In the literature, two approaches are often used: the cross-ambiguity function (CAF) and the single branch auto-correlation (SBA) [20]. The CAF method results in accurate estimations but with a high complexity, whereas SBA is less complicated but also less accurate.

In this work, we study how to estimate and correct Doppler effects in underwater communications. In particular, we propose a simplified CAF method that can be used in real-time applications, and we apply it to the JANUS [4] standard.

JANUS is a first standard for underwater communications developed by the NATO Centre for Maritime Research and Experimentation (CMRE) and designed to enable communication interoperability between maritime heterogeneous assets: NATO and NON-NATO, military and civilian. It has been designed to be simple, so that it can be easily adopted by existing or newly manufactured equipment. In [27] JANUS is proposed as a second "language" to be implemented in manufacturer-specific devices in parallel with their proprietary digital coding schemes: JANUS could be used to establish the first contact, notifying the presence of an asset in the area and its capabilities to negotiate communication parameters. After, the devices can switch to a suitable modulation scheme supported by all, or a subset of, the devices in the area. In addition, it also offers the freedom to use sophisticated receivers and decoders that significantly improve performance. One of the improvements that can be made concerns the performances of this communication standard in the presence of the Doppler effect. Indeed, although the modulation scheme adopted in this standard shows a good robustness for multipath, explicit Doppler compensation is required in all those scenarios where the Doppler effect introduces significant distortions.

All the features described above make JANUS an attractive candidate to be taken into account in the development of underwater software-defined platform equipped with different capabilities and able to communicate with other manufacturer-specific devices.

We exploit the JANUS preamble, composed of an m-sequence of 32 pseudo-random symbols, to estimate and compensate the Doppler effect without requiring any modification to the standard. We validate the proposed method using Watermark, a realistic underwater channel simulator, and with in-field experiments. As we will show, our estimation technique is able to correct and receive over 90% of the packets even with severe Doppler, compensating relative speeds up to 5 m/s.

In the rest of the chapter, we present the JANUS standard in section 3.2 and our estimation technique for JANUS in section 3.3. We demonstrate the effectiveness of the proposed method in section 3.4 using both simulations and real experiments.

3.2 JANUS Standard

JANUS adopts a Frequency-Hopped (FH) Binary Frequency Shift Keying (BFSK) modulation scheme because of its robustness in underwater propagation and its implementation simplicity. In this scheme, each bit is mapped into a pair of tones, selected among 13 possible equidistant pairs covering the entire frequency band whose initial allocation ranges from 9440 to 13600Hz. The order in which the 13

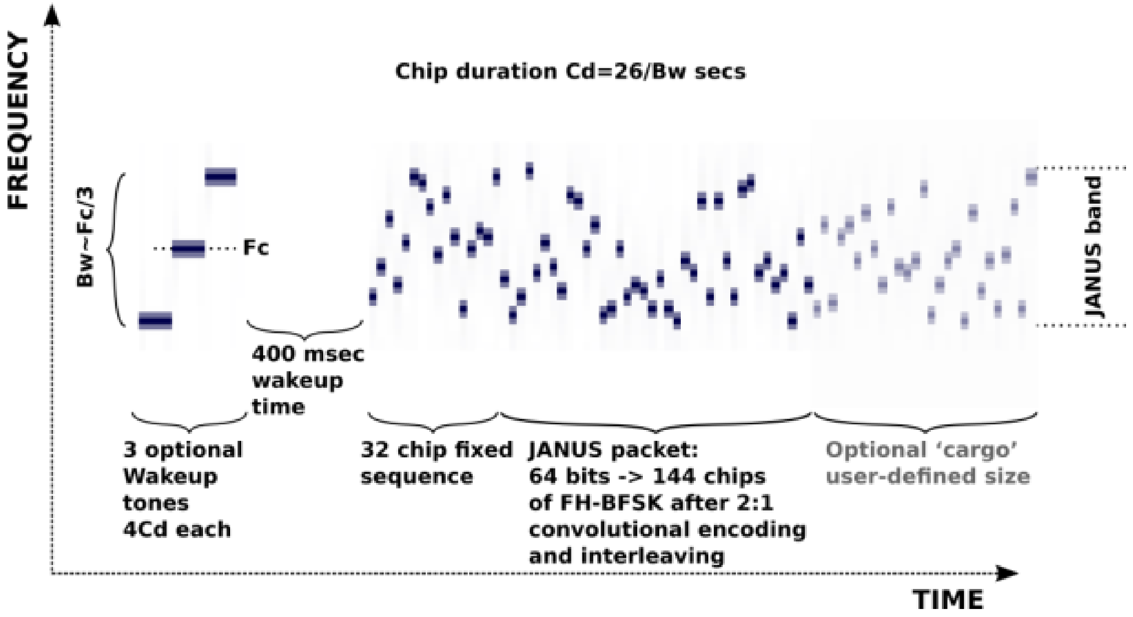

FREQUENCY
Chip duration Cd=26/Bw secs
Bw~Fc/3
Fc
JANUS band
400 msec wakeup time
3 optional Wakeup tones 4Cd each
32 chip fixed sequence
JANUS packet: 64 bits -> 144 chips of FH-BFSK after 2:1 convolutional encoding and interleaving
Optional 'cargo' user-defined size
TIME

Figure 3.1: Time-Frequency structure of a JANUS packet.

tone pairs are used is chosen to minimize Inter-Symbol Interference (ISI).

A characteristic of JANUS is that some parameters can not be freely chosen but are a function of the selected frequency band. For example, the Chip duration (Cd) and the Frequency Slot width (FSw) are computed directly from the upper and lower band values. Other parameters, such as the Frequency Hopping, remain constant for any chosen band.

The standard provides a baseline JANUS packet consisting of an acoustic waveform that encodes 64 bits of information. Furthermore, at the end of the baseline packet, an additional Data Cargo payload can be added. The alignment mechanism is obtained by preceding the baseline JANUS packet with a fixed synchronization sequence of 32 symbols. Optionally, three tones can precede the whole packet and are used to wake up a modem from a power-saving sleep state.

Figure 3.1 shows the Time-Frequency structure of a JANUS packet. Notice the three wake-up tones at the start of the packet.

The robustness to temporal and frequency fading is provided by a 2:1 convolutional coding redundancy with a constraint length of 9, using the following polynomial generators:

$$\begin{aligned} g_1(x) &= x^8 + x^7 + x^5 + x^3 + x^2 + x^1 + 1 \\ g_2(x) &= x^8 + x^4 + x^3 + x^2 + 1 \end{aligned} \tag{3.1}$$

The coded signal is then followed by interleaving. The baseline packet and the cargo are always separately encoded and interleaved as the baseline packet needs to be decoded to know how many bytes are in the cargo.

Error checking is performed using 8-bit CRC (Cyclic Redundancy Check) which uses the polynomial $p(x) = x^8 + x^2 + x^1 + 1$, initialized to 0.

3.3 Doppler estimation

Doppler measurements are generally performed by adding particular waveforms to signal to be transmitted. For example, m-sequences, linear-frequency-modulated (LFM) or hyperbolic-frequency modulated (HFM) waveforms could be added before the data symbols for evaluating the frequency distortion [7], both before and after the data frame to measure the time differences between arrivals (then converted to time compression factors [6]).

One of the most used technique is the cross-ambiguity function (CAF). The CAF represents the output of a matched filter to an input signal that is shifted in terms of delay and frequency (Doppler effect) [6], [39]. The mathematical expression that defines the CAF is dependent on the way the Doppler effect is modeled. Following the above references, for the narrowband model the cross-ambiguity function can be expressed as:

$$A_{narrowband}(\tau, f_D) = \int_{-\infty}^{+\infty} s(t)s^*(t+\tau)e^{j2\pi f_D t}dt \tag{3.2}$$

where $f_D = f_R - f_T = (\gamma - 1)f_T$ is the (fixed) frequency distortion and γ is the Doppler factor defined in equation (2.1). Instead, using the wideband model, we have:

$$A_{wideband}(\tau, \gamma) = \int_{-\infty}^{+\infty} s(\gamma t)s^*(t+\tau)dt \tag{3.3}$$

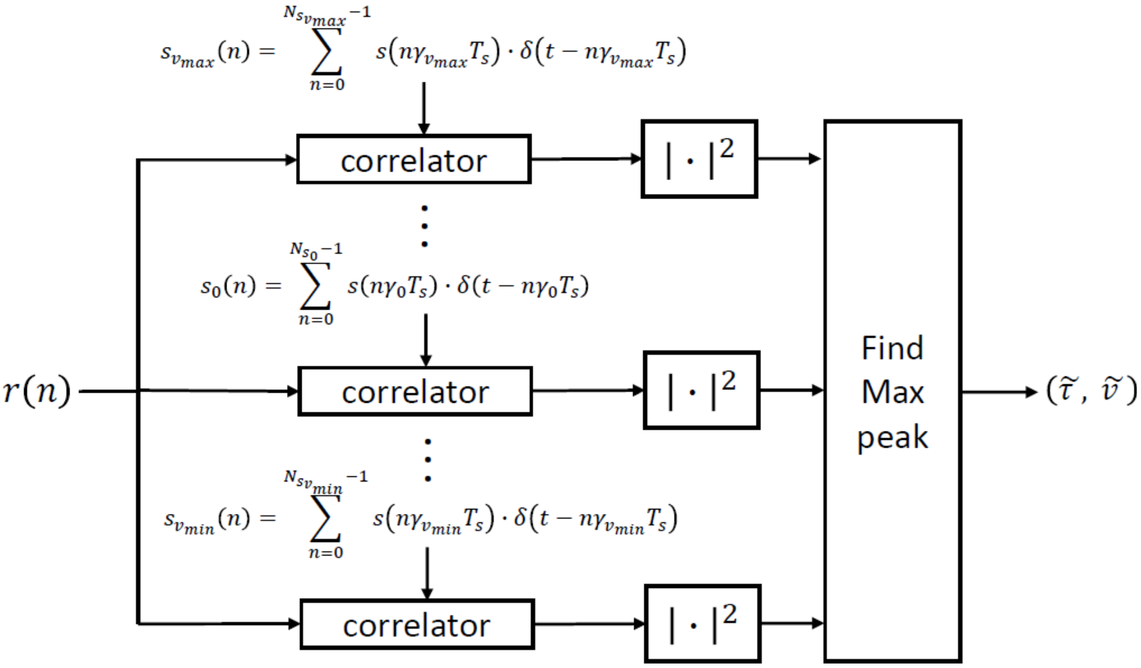

$s_{v_{max}}(n) = \sum_{n=0}^{N_{s_{v_{max}}}-1} s(n\gamma_{v_{max}}T_s) \cdot \delta(t - n\gamma_{v_{max}}T_s)$
correlator
$|\cdot|^2$
$s_0(n) = \sum_{n=0}^{N_{s_0}-1} s(n\gamma_0 T_s) \cdot \delta(t - n\gamma_0 T_s)$
$r(n)$
correlator
$|\cdot|^2$
Find Max peak
$(\tilde{\tau}, \tilde{v})$
$s_{v_{min}}(n) = \sum_{n=0}^{N_{s_{v_{min}}}-1} s(n\gamma_{v_{min}}T_s) \cdot \delta(t - n\gamma_{v_{min}}T_s)$
correlator
$|\cdot|^2$

Figure 3.2: Block scheme for computing the CAF using a bank of correlators.

In both models, it can be easily seen that for $\gamma = 1$ the CAF is nothing but a simple auto-correlation of the signal. Instead, for $\gamma \neq 1$, the CAF consists in correlating the signal with a modified version of itself through the Doppler factor γ which, in the narrowband model, appears as a fixed frequency shift, while in the wideband model appears as a scale factor of time. In practice, the CAF can be computed employing a bank of correlators, each of which is used to correlate the input signal with a Doppler-scaled replica of itself. This operation leads to a delay-Doppler scale grid in which the maximum of the CAF magnitude is searched. This measurement is used for both Doppler and time delay estimation (used for synchronization).

Clearly, the resolution depends on the number of correlators and on the expected range of Doppler distortions. Let M be the number of correlators and $s_0(t)$ the original transmitted signal. By sampling the signal at frequency f_s, we obtain:

$$s_0(n) = s_0(t) \cdot \sum_{n=0}^{N_{s_0}-1} \delta(t - nT_s) \tag{3.4}$$

where $T_s = 1/f_s$ is the sampling period and N_{s_0} is the number of samples required to cover the entire signal. Then, we can express the generic Doppler-scaled version $s_{v_i}(n)$ of the reference preamble as:

$$\begin{aligned} s_{v_i}(n) &= s_0(t) \cdot \sum_{n=0}^{Ns_{v_i}-1} \delta(t - n\gamma_{v_i} T_s) \\ &= \sum_{n=0}^{Ns_{v_i}-1} s_0(n\gamma_{v_i} T_s) \cdot \delta(t - n\gamma_{v_i} T_s) \end{aligned} \tag{3.5}$$

where the subscript v_i represents the relative speed of the signal affected by Doppler and Ns_{v_i} the number of samples of this same signal, which is equal to:

$$Ns_{v_i} = \frac{N_{s_0}}{\gamma_{v_i}} \qquad i = 0, \pm 1, ..., \pm \frac{M-1}{2} \tag{3.6}$$

To compute the CAF, the received signal $r(t)$ is sampled, converted to baseband and filtered to obtain $r(n)$. Then, $r(n)$ is correlated with all the Doppler-scaled versions of the reference signal $s_0(n)$ as shown in figure 3.2. Finally, considering the squared module of each correlation, the CAF is obtained from the output of all the correlators.

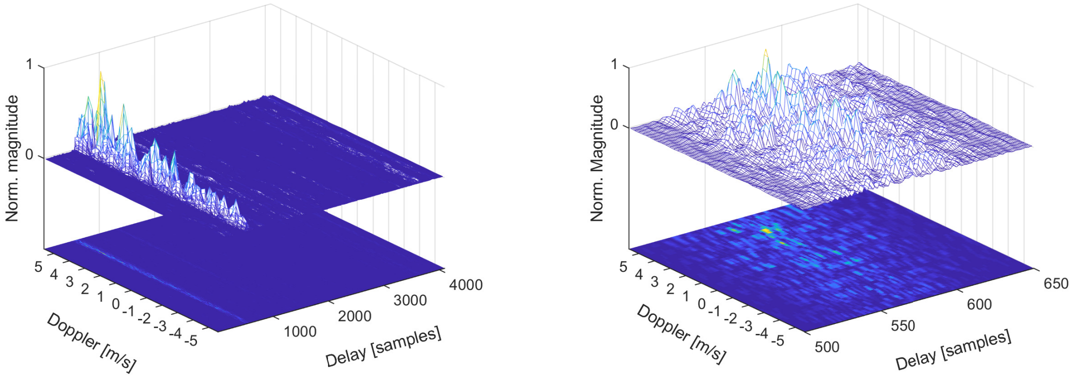
Since in this work we are interested in the impact of relative motion between transmitter and receiver, we will express the Doppler range in terms of speed (measured in m/s) and we will estimate the Doppler scale factor via equation (2.1).

For this purpose, M doppler-scaled versions of the JANUS preamble must be generated, each of these corresponding to a different relative speed v_i between transmitter and receiver. Without loss of generality, in this work the CAF is designed to estimate and correct distortions caused by relative movement up to $\pm 5m/s$ and with resolution step of $\Delta v = 0.25m/s$. Thus, the total number of correlators M required is computed as the ratio between the speed range $[v_{min}, v_{max}]$ and the resolution plus 1:

$$M = \frac{v_{max} - v_{min}}{\Delta v} + 1 \tag{3.7}$$

while the relative speed v_i, corresponding to each correlator i is:

$$v_i = i \cdot \Delta v \qquad i = 0, \pm 1, ..., \pm \frac{M-1}{2} \tag{3.8}$$



(a) CAF

(b) Details of CAF

Figure 3.3: Example of CAF using the JANUS preamble. Watermark NOF1 channel, simulated motion 4 m/s.

The estimated Doppler will correspond to the highest value shown by the CAF, obtaining the relative speed $\hat{v}$. Although out of the scope of this work, the CAF peak may be also exploited to extract the time delay $\hat{\tau}$, used for time synchronization:

$$[\hat{v}, \hat{\tau}] = \underset{v,\tau}{\arg\max} |A(v, \tau)| \tag{3.9}$$

Since the CAF time and Doppler scales are quantized, searching for the CAF maximum value means finding the index of a matrix representing respectively the approximated values of the relative motion $\tilde{v}$ and the time delay $\tilde{\tau}$. Quadratic interpolation is then performed to improve the Doppler estimate.

For example, figure 3.3 shows the CAF obtained when the JANUS preamble is received through Watermark NOF1 channel (more details in section 3.4), with a simulated motion of 4 m/s. Note that the effect of node's movement sums up with the Watermark channel characteristics (i.e. in terms of Doppler spread). From the figure, the maximum of the CAF is clearly distinguishable from other minor peaks, despite the fact that the JANUS preamble is generally not used for this purpose.

3.4 Experimental results

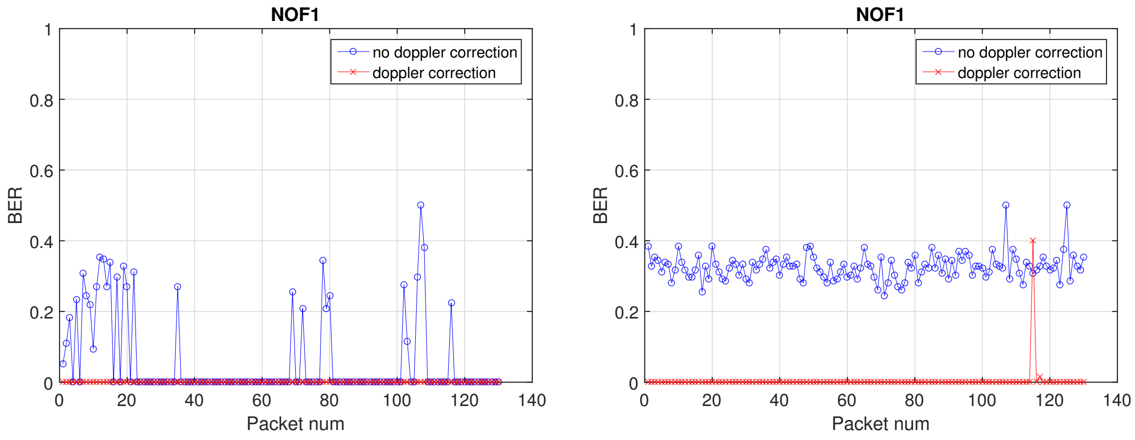
We implemented the proposed technique in the WSense srl and Rome La Sapienza implementation of the JANUS STANAG standard modulation. First, we tested the proposed mechanism using the realistic underwater simulator Watermark, in order to combine real channels and reproducible conditions. Then, we also validate the proposed Doppler estimation technique in field, with an unmanned surface vehicle (USV) towing one of the two modems.

3.4.1 Watermark simulation

In our experiments, we used the NOF1 and NCS1 Watermark channels having center frequency of 14 kHz (10-18 kHz band), the closest to JANUS standard (center frequency $fc = 11520$ Hz and band between $9440 - 13600$ Hz). Additionally, we shifted the JANUS signal to a center frequency of 14 kHz, sampled at 48 kHz, and we tested different relative speeds, from -5 to 5 m/s with steps of 1 m/s. We performed the tests setting the size of the JANUS Cargo to 8, 16, 32 and 64 Bytes. Depending on the size of the Cargo, the packet transmission is repeated many times to fill the Watermark trace (almost 300 packets with 8 Bytes Cargo). In the follow, we report the results obtained using a JANUS Cargo of 16 Bytes leading to about 130 packets transmission in NOF1 channel and about 140 in NCS1 channel.

Figure 3.4 shows the results obtained on the Watermark NOF1 channel, with node's relative speed of 2 and 4 m/s respectively. From the figure, it is clear that with our Doppler correction almost all the packets are received without errors. Instead, without Doppler correction, the BER becomes quickly unacceptable, over 30% on average of bit errors per packet for speeds of 4 m/s. We repeated the same experiments using the Watermark NCS1 channel, which is the most challenging trace available. Figure 3.5 summarizes the results obtained in this scenario, with node motion of 2 and 4 m/s respectively. With the NCS1 trace, errors appear also when employing the proposed Doppler correction. Note, however, that packets received with a 50% bit errors indicate a possible detection or synchronization problem, more than a Doppler distortion[1]. In any case, using our method, packets are correctly

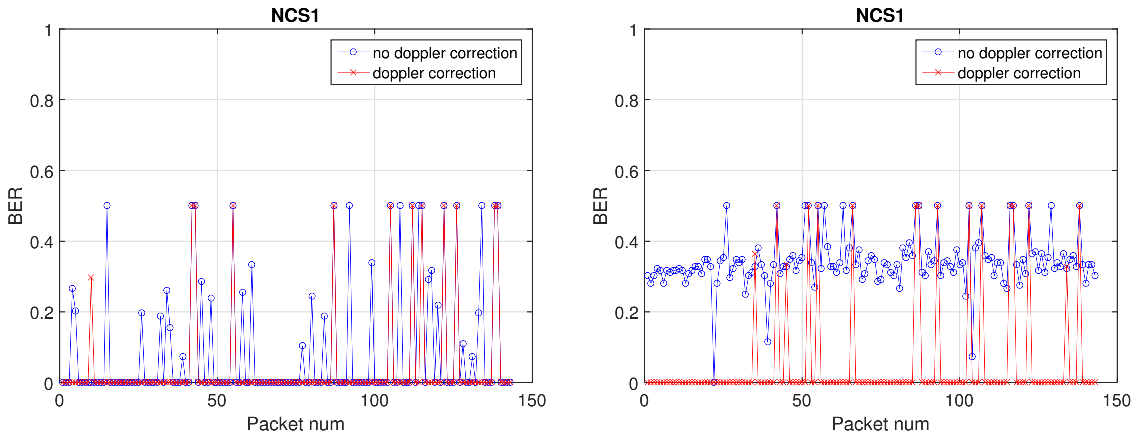
[1] In these experiments, for simplicity we employ a separated algorithm for packets detection



(a) 2 m/s

(b) 4 m/s

Figure 3.4: Results obtained on Watermark NOF1 channel.



(a) 2 m/s

(b) 4 m/s

Figure 3.5: Results obtained on Watermark NCS1 channel.

received in over 90% of the cases, even with speeds of 4 m/s.

Figure 3.6 compares the results obtained on both channels in terms of BER when employing the proposed Doppler correction (solid line) or without it (dashed line). In the figure, each point represents the average bit errors per packet computed on the overall packets (about 130 for NOF1 and 140 for NCS1) obtained for each speed. As an example, at 2 m/s we can see the BER obtained from the bit errors

before applying the CAF.

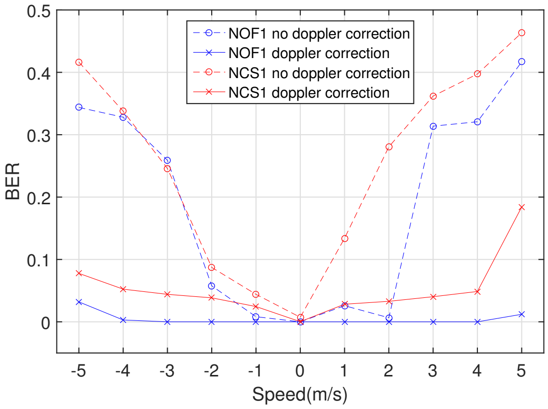


Figure 3.6: Average BER obtained with and without Doppler correction on NOF1 and NCS1 channels.

of the packets shown in the figures 3.4(a) and 3.5(a). As already demonstrated in [8], JANUS is robust to Doppler speeds up to 2 m/s but above these relative speeds the BER sharply increases on both NOF1 and NCS1 channels. In these scenarios, a Doppler estimation and correction mechanism is required and the proposed method succeeds to this purpose, keeping the BER generally lower than 5%.

We also tested the performance obtained at different values of normalized signal-to-noise ratio (SNR), by adding AWGN noise to the Watermark traces (which are already affected by noise). We vary the E_b/N_0 SNR from 0 dB to 20dB, with steps of 2 dB, and tested the proposed method in the worst case scenario of 5 m/s speed. Figure 3.7 shows the results obtained with a 16 Bytes cargo on the NOF1 trace, from which it is clear that the proposed method successfully estimates the Doppler already with a SNR of 2 dB on top of the Watermark trace. Similar results were obtained on the NCS1 trace.

3.4.2 At sea results

Finally, we tested our implementation with in field experiments, in the waters of the Italian harbor of La Spezia. The experiments were performed during the JANUS Interoperability Fest, an event organized by the Centre for Maritime Research and Experimentation (CMRE) in collaboration with the Italian Navy Naval Support and Experimentation Centre (CSSN), for allowing to the guest companies (as WSENSE

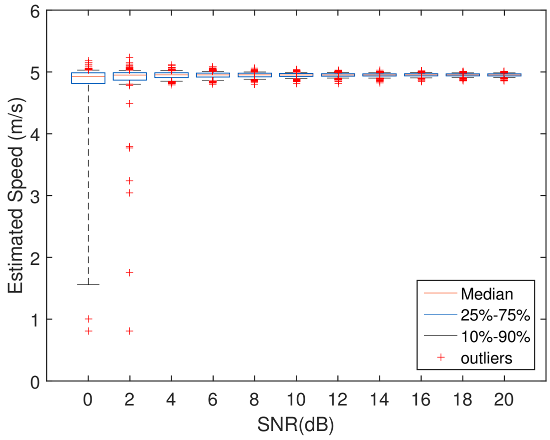


Figure 3.7: Estimated Doppler speed varying the SNR level.

S.R.L.) to test the own JANUS implementation[2]. The setup was composed of a moving transmitter placed on a USV provided by CMRE and located at about 200m away from the receiver that was instead stationary and dropped from the pier where water depth is about 10m. During these experiments several JANUS packets with a size of 64 Bytes were transmitted from the USV moving at variable speeds spanning a range from -4 m/s to 4 m/s. Figure 3.8 shows the results obtained in this challenging scenario. Note that our implementation is designed to compensate Doppler effects due to constant relative speeds, so large accelerations of the USV during the packet transmission might influence the results. Nevertheless, the proposed Doppler correction allows error-free reception in about 75% of the cases, with a BER of about 10%, while without Doppler correction the BER is 37% and only 16% of packets received without errors. In this last case, most of the error-free receptions are around the center of the experiment, when the USV spanning the -4m/s to 4m/s range crosses speeds around 0 m/s, i.e. no or slight Doppler effects that the robust JANUS modulation is able to tolerate.

[2] This work acknowledges the use of data that was made possible thanks to the Second JANUS Interoperability Fest, organised by CMRE with the support of CSSN.

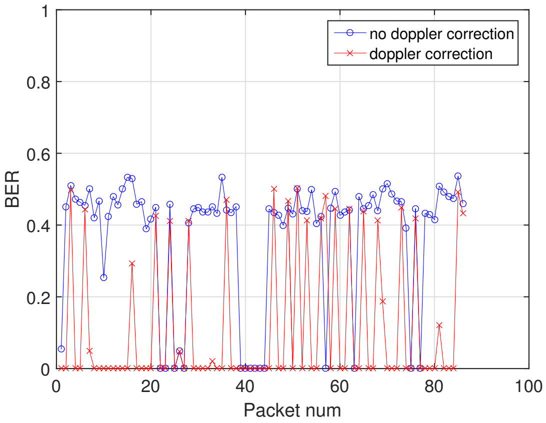


Figure 3.8: Results obtained in field using a USV.

Chapter 4

Experimental SDA platform

4.1 Introduction

The large varieties of underwater wireless scenarios and environments is reflected in different varieties of modems and architectures which are present in literature and on the market. All these modems generally have limited input parameters to configure the link and, most importantly, the HW and SW of these modems are not decoupled in separate modules, i.e. the modulation/demodulation SDA and the HW front-end of the modem.

More recently, the new *Software-defined Acoustic* (SDA) modem paradigm has been introduced, where part of the modem operations are done in software and can thus be changed or tuned at will.

In this work we present FLUMO (FLexible Underwater MOdem), a new, flexible SDA underwater modem architecture, composed of three different parts, i) a digital platform; ii) an analog front-end board; and iii) a piezoelectric transducer. The design and implementation of the modem architecture follows two key ideas: first, the signal modulation/demodulation runs fully in software and is completely decoupled from the rest of the system. This separation is very helpful for the validation and testing of both of the SDA and HW of the modem, e.g. employing externally modulated signals contained in a wav file. Second, we provide a flexible and tunable system where the different communication parameters can be adjusted in order to reach the best performance in every scenario. In other words, FLUMO offers to the users the possibility to better fit the different application scenarios, easily adapting to changing underwater channel conditions and variable transmitter-receiver distances.

This complete separation between the SDA modem and the transmission hardware (HW), allows FLUMO to send/receive signals modulated with other modems (recorded through a wav file) or, alternatively, modulate/demodulate signals to be sent/received from other transceivers. For example, in this work we use the same HW to validate the performance of a Continuous Phase Frequency Shift Keying (CPFSK) modulation scheme. and compare it with the WSENSE srl and Rome La Sapienza implementation of the JANUS STANAG standard modulation.

The implementation of FLUMO modem is fully based on software – using the SDA modem paradigm – and runs on the CPU of the digital platform, while the CPFSK modulation is based on the open source *liquid-dsp* library [40]. To validate the implemented SDA system, we performed several experiment campaigns in controlled environments, as well as co-simulation tests to validate the modulation only (decoupled from the hardware) using Watermark. Depending on the channel conditions, the CPFSK modulation tested with our SDA modem can transmit over 500 symbols per second at $25kHz$ carrier frequency, which translates to a gross data rate of 500bps if each symbol codes one bit. However, the focus of this work is not on the modulation performance itself, but on the flexibility of the SDA architecture. On the other hand, our solution can be easy extended with other types of innovative modulations, like OFDM, S2C or DSSS.

In the rest of the chapter we present the characteristics of the proposed system architecture in section 4.2. Then we analyze and tune the parameters of our flexible modem in section 4.3, where experimental results are performed both indoors, in a small water tank (a plastic barrel), and outdoors, in a large irrigation cistern. Finally, in section 4.4, we extend the experimental validation with the Watermark simulator and perform a co-simulation of our modem implementation.

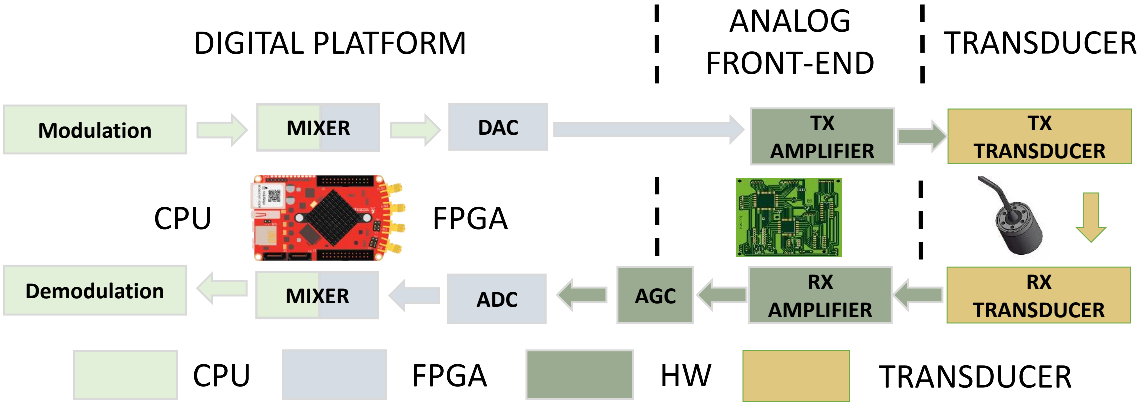


Figure 4.1: Block diagram of the system.

4.2 HW/SW architecture for FLUMO

In this section we present the characteristics of the proposed system. Figure 4.1 describes the overall system architecture, mainly composed of three parts (the digital platform, the analog front-end and the transducer), showed with different colors in a block diagram. From left to right we have:

- *The digital platform*, provided by a RedPitaya board [41], an open-source hardware project that provides both high and low speed ADCs and DACs, with SMA standard connectors. It includes a dual-core ARM CPU, which allows to run a Linux operating system, and a high performance FPGA where the more demanding digital signal processing algorithms are performed.

- *The analog front-end board*, a custom made analog board to interface the digital platform to the transducer.

- *The underwater Transducer*, a $25kHz$ cylindrical transducer produced by Neptune Sonar, with a bandwidth comprised between $14kHz$ to $30kHz$.

We will now describe in details the implementation of these three parts of FLUMO.

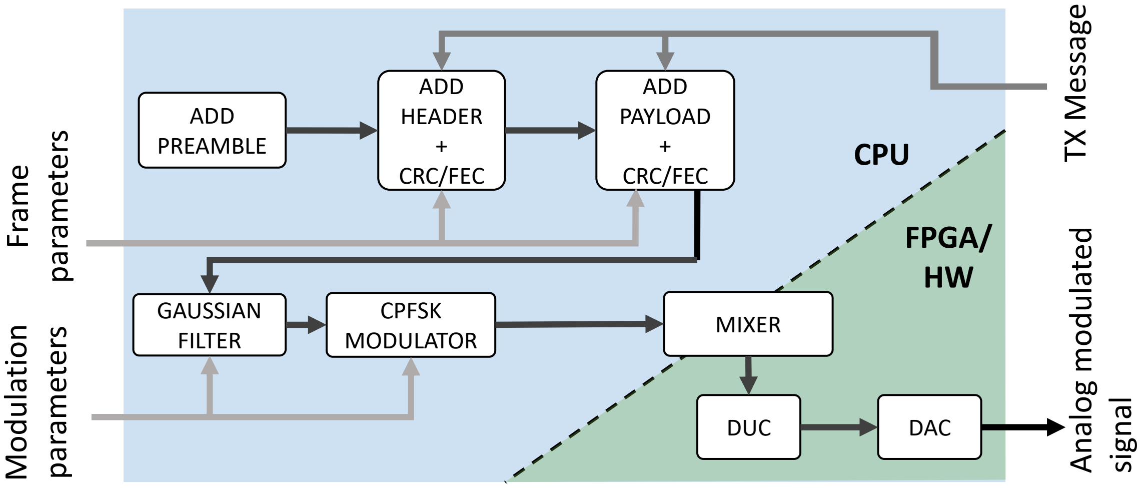


Figure 4.2: Block diagram of the transmission chain.

4.2.1 Digital platform and modulation

The RedPitaya board represents the core of the modem, providing both the dual-core ARM CPU and the high performance FPGA. The RedPitaya provides 50MHz (3dB) of bandwidth, a sampling rate of 125 Msps and a DAC resolution of 14 bits. The board can be accessed through ethernet, serial port and WiFi. Thanks to the RedPitaya board, it is possible to reach high frequencies bands, with a cost that is contained to few hundred euros. The designed and developed SDA modulation scheme runs on the RedPitaya's CPU. We also generate a bitstream, placed in the FPGA, with Cascaded Integrated Comb (CIC) filters. The board is clearly oversized (we exploit an audio bandwidth of the order of hundreds of kHz) and was chosen because of the presence of FPGA for future extensions. Indeed, we can demand the most computationally expensive tasks (like filtering, synchronization or for example entire modulation/demodulation blocks) to FPGA.

The Digital Up Conversion (DUC) and the Digital Down Conversion (DDC) modules respectively increase the sampling rate (interpolation) in transmission before the DAC, and reduce the sampling rate (decimation) in reception after the ADC. The DAC and ADC are already part of the RedPitaya board. The result is a multirate transceiver system, depicted in figure 4.2 for transmission and figure 4.3 for reception (FPGA area). The developed bitstream provides a complete

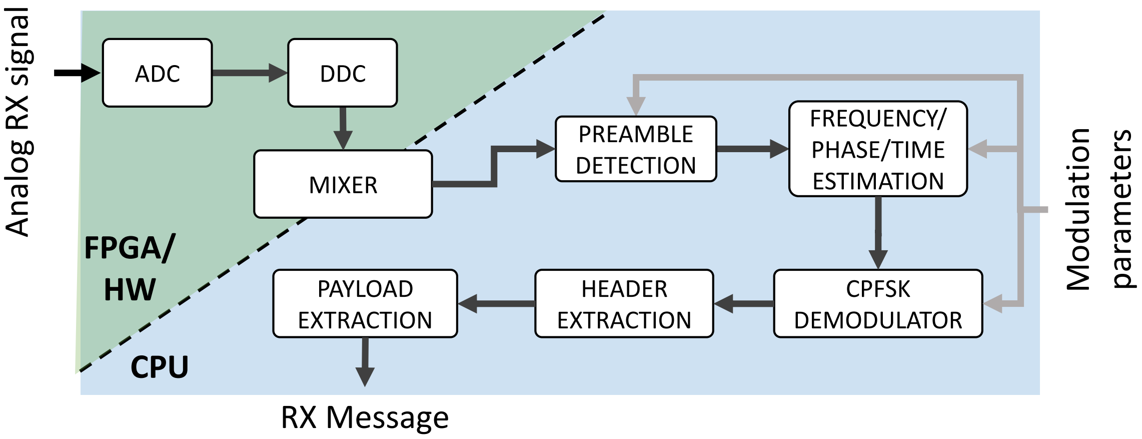


Figure 4.3: Block diagram of the reception chain.

Direct Digital Synthesizer (DDS) module that exploits the signals of a numerically controlled oscillator (NCO). NCO sine and cosine signals are multiplied with the input samples in the mixer block to obtain the baseband I/Q complex signal (i.e. the in-phase and in-quadrature signal components) [42]. All the FPGA composed blocks are Xilinx core IP.

The bitstream presents a register to enable or disable the DDS and the mixer modules. This implemented register is fundamental for the decoupling between modulation/demodulation blocks and the transmission/reception module. Indeed, when the DSS and the mixer modules are disabled, the SDA is able to directly manage the passband signal. For example, it is possible to directly read or write the raw signal samples using wav file. When the signal modulation process is performed by a different modem (e.g. through a saved wav file), the signal is already in passband and up-conversion is disabled in the FPGA. In other words, the operation of bringing the signal from baseband to passband and viceversa can be performed in FPGA or in software according to the use case and the needs of the application (indeed, in figure 4.1, we show the mixer block with both colors). This characteristic allows the HW transceiver to send/receive a signal modulated from another system or, alternatively, the SDA modulator can modulate/demodulate signals that can be sent/received from a different transceiver. This flexibility is very important for the correct validation and testing of both SDA and HW transceiver blocks, as we will show later in section 4.4.

The high speed sampling and direct ADC/DAC conversion of the RedPitaya board allow to receive and decode in software the acoustic signals through the SDA modem, which runs on the CPU of the RedPitaya. The receiver is implemented in the following way: the analog front-end board is connected to one of the high-impedance RedPitaya inputs. The on-board ADC digitizes the received signal which is then processed by the DDC running on the FPGA. The data is transmitted via Linux socket to the modulation/demodulation block, a software module developed in C language. The complete list of tunable parameters in FLUMO is reported in Table 4.1, with the relative descriptions.

Modulation/Demodulation We use a CPFSK modulation scheme in which the digital data is transmitted through discrete frequency changes of a carrier signal. CPFSK is a continuous phase modulation method, with the advantage of high band utilization and low out-of-band power. CPFSK signals have constant envelope and are easy to amplify and robust to nonlinear channels.

As in FSK modulation, binary digits are mapped to a symbols alphabet a. With reference to binary FSK, binary digit 0 is mapped to the symbol $a = -1$, while binary digit 1 is mapped to the symbol $a = 1$. The carrier frequency is then changed according to symbol value:

$$f_n = f_0 + a_n \Delta f \tag{4.1}$$

where f_n is the frequency associated with symbol $a_n \in \{-1, +1\}$, f_0 is the carrier frequency and Δf is the frequency deviation.

Differently from FSK, CPFSK keeps the phase continuity between two consecutive symbols. Indeed, in this type of modulation the instantaneous phase of the signal, in the symbol interval nT $\leq$ t $\leq$ (n+1)T, changes according to:

$$\theta(t) = 2\pi\Delta f[a_n(t - nT) + T \sum_{k=-\infty}^{n-1} a_k] \tag{4.2}$$

where T is the symbol period. The instantaneous phase can be expressed as:

Table 4.1: Parameters of the FLUMO SDA modem.

Modem PHY parameters	**Description**
Sampling Rate [sps]	The number of samples per second
Payload	Message to send
AMP	The output signal level (Amplitude) from the digital platform (range between -1/+1 Volt)
Power	Power amplification level of the analog front-end (4 different levels)
Center Freq	Center frequency used by modem
Samples Per Symbol (SPS)	Specifies the number of samples dedicated to each symbol
Filter-Bandwidth (BT)	Bandwidth-Time product, where the symbol bandwidth B is the -3 dB (half-power) band of the pulse/filter and T is the symbol duration
Modulation index	FSK modulation index
CRC	Cyclic redundancy check used for the modulation/demodulation phase
FEC	Forward Error Correction used for the modulation/demodulation phase
PassBand Filter	Passband filter applied before the demodulation phase (real-time mode only)

$$\theta(t) = \pi h a_n \frac{t - nT}{T} + \alpha_n \qquad nT \leq t \leq (n+1)T \tag{4.3}$$

where the term:

$$h = 2T\Delta f \tag{4.4}$$

is named modulation index and the term:

$$\alpha_n = 2\pi T\Delta f \sum_{k=-\infty}^{n-1} a_k = \pi h \sum_{k=-\infty}^{n-1} a_k \tag{4.5}$$

is the phase accumulation due to memory of previous symbols. The CPFSK

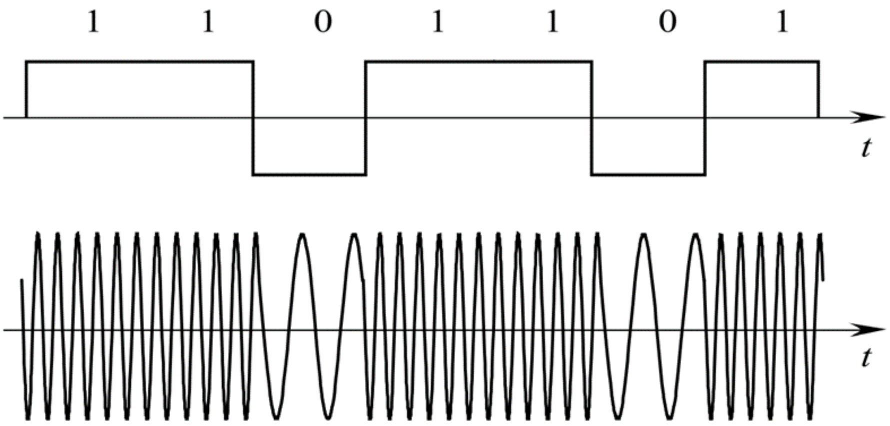


Figure 4.4: Example of CPFSK signal.

signal then can be expressed as:

$$s(t) = A_0 \sum_{n=-\infty}^{+\infty} cos(2\pi f_0 t + \pi h a_n \frac{t-nT}{T} + \alpha_n + \phi)g(t-nT) \tag{4.6}$$

where the term A_0 is the amplitude factor, ϕ an unspecified initial phase term and g(t) is a pulse shaping function which is non-zero in the interval t $\in$ [0,T] and 0 otherwise. In figure 4.4 we can see an example of CPFSK modulated signal where an ideal rectangular pulse shaping is used.

Because of the continuous phase between consecutive symbols, the spectrum of this type of modulated signal is less spreaded and then more concentrated around the carrier. Conversely, the accumulation phase mechanism lead to a more complex receiver structure.

In our implemented CPFSK modulator, we preliminary apply a Gaussian filter to the bit rectangular pulses, thus smoothing it and leading to a signal pulses g(t) which confines the emissions to a lower spectral band. The filter is defined by a zero-mean impulse response:

$$h(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{t^2}{2\sigma^2}}, \quad \sigma^2 = \frac{ln2}{(2\pi B)^2} \tag{4.7}$$

where B is the 3dB filter bandwidth, and can be specified through the Bandwidth-Time product value BT. Figure 4.5 shows the pulse shape for different BT values.

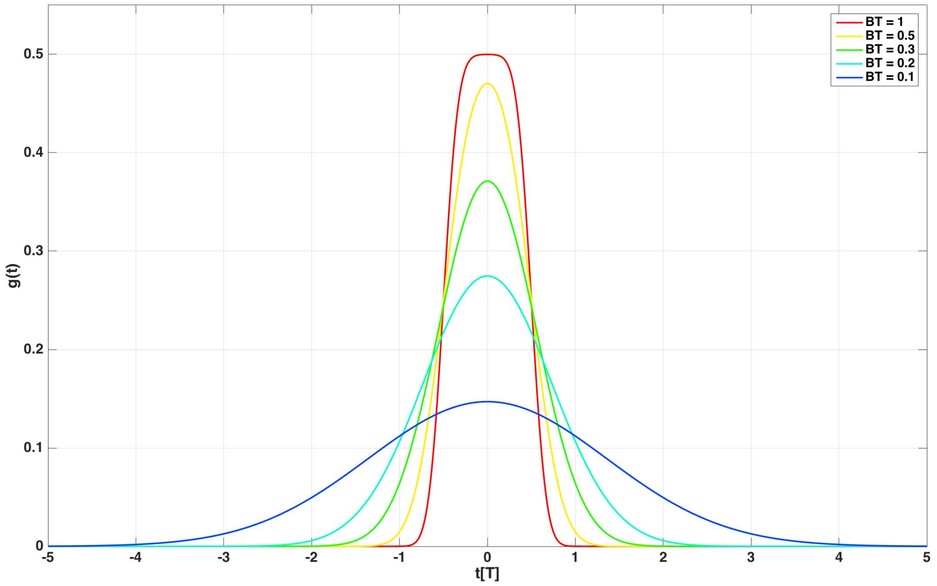


Figure 4.5: Guassian pulse shape with different BT values.

As we can see from the figure, when BT $\rightarrow +\infty$ then h(t) $\rightarrow \delta(t)$ and g(t) become a rectangular pulse. On the other hand, lower is its value, lower is the spreaded band but higher is the ISI interference because of the extension of the pulses to the adjacent symbols.

The modem has a tunable frequency range from 0 to $50MHz$. The development of the modem is based on the *liquid-dsp* library, a free and open-source signal processing library for software-defined radios written in C. Furthermore liquid-dsp provides a framing structure for sending and receiving data, which we describe later.

Figure 4.2 and figure 4.3 present the software architecture of the implemented modulation and demodulation schemes respectively. In the figures, the black arrows indicates the data flow, while the grays arrows represents the system inputs, i.e. the message to transmit and the tunable modulation parameters. The two blocks “CPFSK MODULATOR” and “CPFSK DEMODULATOR” implement the CPFSK modem. The output of the modulator is a signal with k Samples Per Symbol (SPS), and a fixed Bandwidth-Time product (excess bandwidth factor) BT. The demodulation is performed by two blocks, showed in figure 4.3: the “PREAMBLE DETECTION”, where a correlation procedure tries to find the presence of a preamble in the received signal, and the “FREQUENCY/PHASE/TIME ESTIMATION” block which computes the relative offset and implements a Least Means Squares (LMS) equalizer. The demodulator then identifies the instantaneous received frequency, by applying a matched filter to the resulting time-varying phase.

Framing The transmitted frames are constituted of six parts, handled by three blocks in the transmission module and two in the reception module:

- The Ramp-Up, implemented in the “ADD PREAMBLE” block, gracefully increases the output signal level to avoid key “clicking” and reduce spectral side-lobes in the transmitted signal. Furthermore, it allows the receiver’s Automatic Gain Control (AGC) to lock on to the incoming signal, preventing sharp transitions in its output.

- The Preamble phasing is a BPSK pattern which flips phase for each transmitted symbol (+1,-1,+1,-1,...). This sequence helps the receiver’s symbol synchronization circuit to lock onto the proper timing phase (“ADD PREAMBLE” block).

- The P/N sequence, also implemented in the “ADD PREAMBLE” block, is an m-sequence exhibiting good auto- and cross-correlation properties. This sequence aligns the frame synchronizer to the remainder of the frame, telling when to start receiving and decoding the frame header, as well as if the phase of the received signal needs to be reversed.

- The Header is a fixed-length data sequence which contains information about the rest of the frame. It is generated by the “ADD HEADER + CRC/FEC” block and decoded by the “HEADER EXTRACTION” block.

- Payload contains the raw data to be transferred across the link. The frame structure has a variable length payload and can be modulated using different schemes. The “ADD PAYLOAD + CRC/FEC” block optionally adds forward error-correction (FEC) and the “PAYLOAD EXTRACTION” block is then

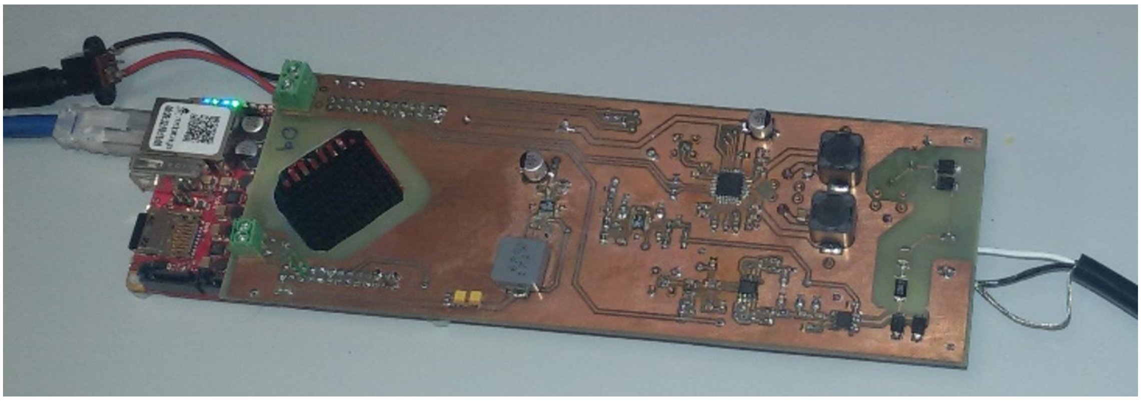


Figure 4.6: Analog front-end board, connected to the RedPitaya (left side) and the wet-end through the twisted cable (right side).

used at the reception of the payload. Different types of FEC and CRC are enabled in the modem, based on the liquid-dsp library.

The frames are modulated in complex baseband signals and can be either immediately processed by the FPGA or saved in a wav file for later transmission. After up-conversion (mixing up to a carrier frequency) the frame is transmitted over the channel where the receiver mixes the signal back down to complex baseband.

On the receiving side, raw samples at complex baseband are streamed to an instance of the frame synchronizer which invokes user-defined callback functions. The synchronizer corrects for gain, carrier, and sample timing offsets (channel impairments) in the complex baseband samples with a minimal amount of pre-processing. When seeking for a frame, the synchronizer initially sets high its internal loop bandwidths for acquisition, as well as for automatic gain control, symbol timing recovery, and carrier frequency/phase recovery.

4.2.2 Analog front-end

The analog front-end board is composed by two different sections, for transmission and reception. The transmission section performs the adaptation between the output signal of the digital platform and the transducer. The RedPitaya SMA connector produces an output between -1 and +1 Volt, with load impedance of 50Ω.

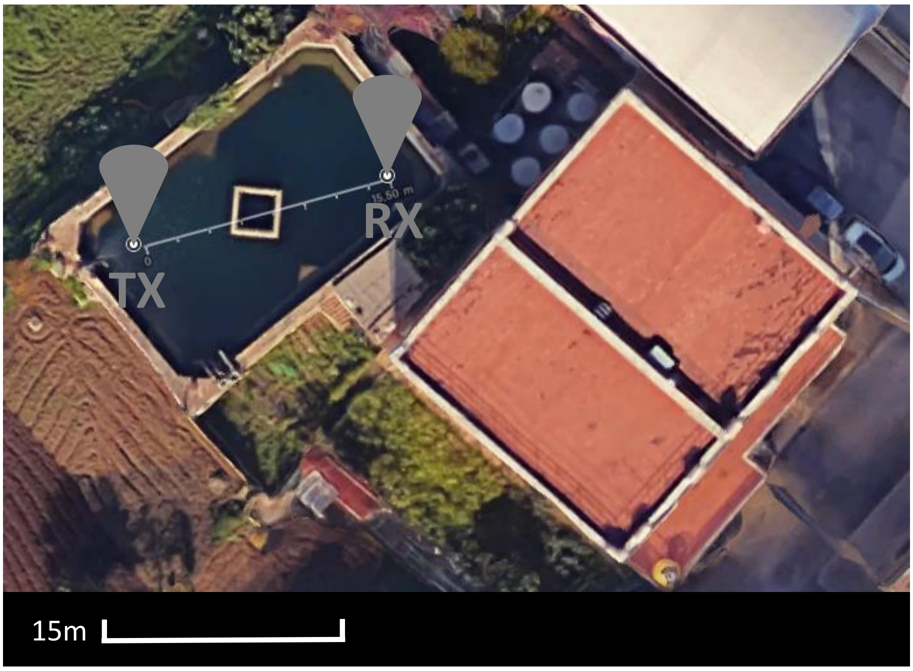


Figure 4.7: Outdoor experiments setup.

The transmission section implements the impedance matching and provides the correct power to the transducer through the custom amplifier, with 4 different power levels and a maximum power of 40 Watt. The gain of the amplifier ranges from 20 to 36 dB. The reception section of the front-end is composed by a second order filter (acting also as anti-aliasing filter) and the AGC amplifier. The received signal is then acquired by the Red Pitaya's ADC. Finally, the underwater transducer is connected to the Red Pitaya through the analog front-end board. We use a cylindrical transducer produced by Neptune Sonar, with bandwidth comprised between 14kHz to 30kHz, connected with the front-end via a twisted pair cable. We also tested an EvoLogics transducer working in the $18-34$kHz band, controlled through the SDM Protocol [30]. Figure 4.6 shows the analog front-end (outside from the enclosure), which is connected to the Red Pitaya on the left side and with the twisted pair on the right side, towards the transducer.

4.3 Experimental results

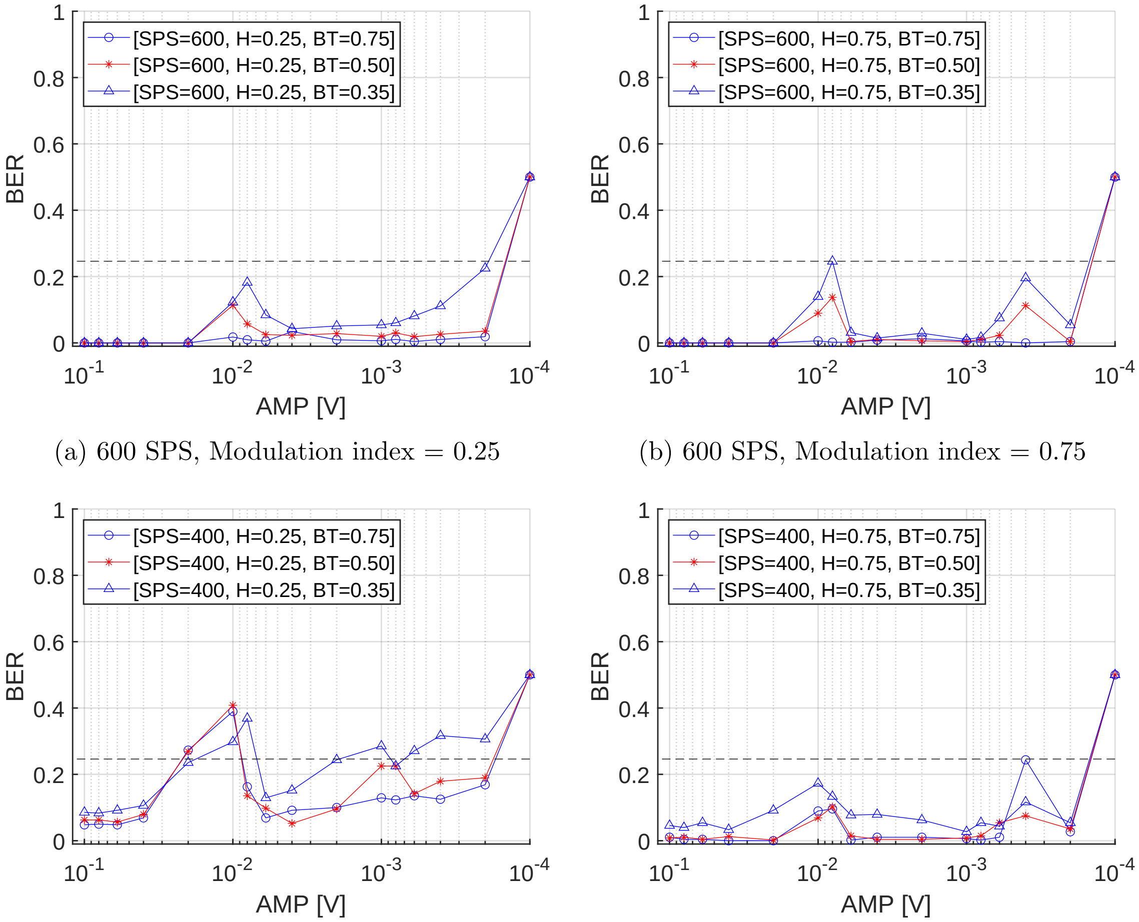
To validate the implemented SDA system, we performed several experiment campaigns in two controlled environments (indoor and outdoor), as well as co-simulation tests to validate the modulation only (decoupled from the hardware) using Watermark. For the indoor experiments, we use a plastic barrel container with a diameter of 60cm and 1m height, filled with about 30l of water. However, to avoid the strong multipath present in the barrel, we also performed outdoor experiments exploiting a rectangular irrigation cistern present in the campus of the University of Palermo, sized 18x12 meters. In this case, the transducers are placed in the opposite angles of the irrigation cistern at a distance of about 15.5 meters as shown in figure 4.7 and about 1.5 meters depth.

We first report the results obtained in the indoor environment, analyzing the impact of the different parameters, e.g. modulation index and filter bandwidth. Then, we reduce the number of SPS to find the minimum SPS required to maintain the communication performance acceptable. Finally, we confirm these settings in the outdoor cistern and we compare the performance of the proposed modulation with a JANUS standard implementation. This process was repeated also using Watermark, thus removing all possible hardware effects and simulating the transmission in a realistic underwater channel.

For both experimental and simulation results, we work in offline mode, i.e. we save the output signal of the modulator in a wav file and use the digital platform to transmit or receive the wav samples. Although FLUMO may also work in real time, this allows us to make separate validation of the implemented SDA and HW of the modem.

4.3.1 Barrel experiments with different configurations

For the indoor experiments in the barrel, we maintain the hardware output power at the lowest level. In the following figures, we plot the Bit Error Rate (BER) in the vertical axis and the output range of signal dynamics (in Volt) in the horizontal axis. Additionally, the horizontal dashed line in the plots represents the upper bound for receiving correctly a frame, i.e. all the packets received with a BER under



(a) 600 SPS, Modulation index = 0.25

(b) 600 SPS, Modulation index = 0.75

(c) 400 SPS, Modulation index = 0.25

(d) 400 SPS, Modulation index = 0.75

Figure 4.8: FLUMO experimental result: comparison between different Modulation index, Filter Bandwidth and SPS.

this threshold are successfully received thanks to the convolutional coding (we used Unpunctured FEC LIQUID_FEC_CONV_V39 rate=1/3 hard=5.41 soft=3.59 K=9, d=18). For each experiment, we send a message of 40 bits, which translates to frames of 264 bits (including the preamble). Finally, we use a signal sampling rate of 100Kbps and average 10 experiments for each point in the plots.

Figure 4.8 shows the BER obtained when we set the modulation index at 0.25 and 0.75 (figures from left to right), different values of SPS, respectively at 600, 500 and 400 SPS (from top to bottom) and filter bandwidth BT 0.35, 0.5 and 0.75 (in each plot). In these experiments the frames are not recognized when the signal dynamics

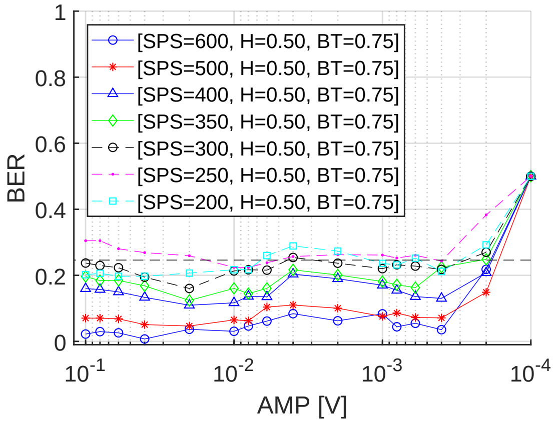


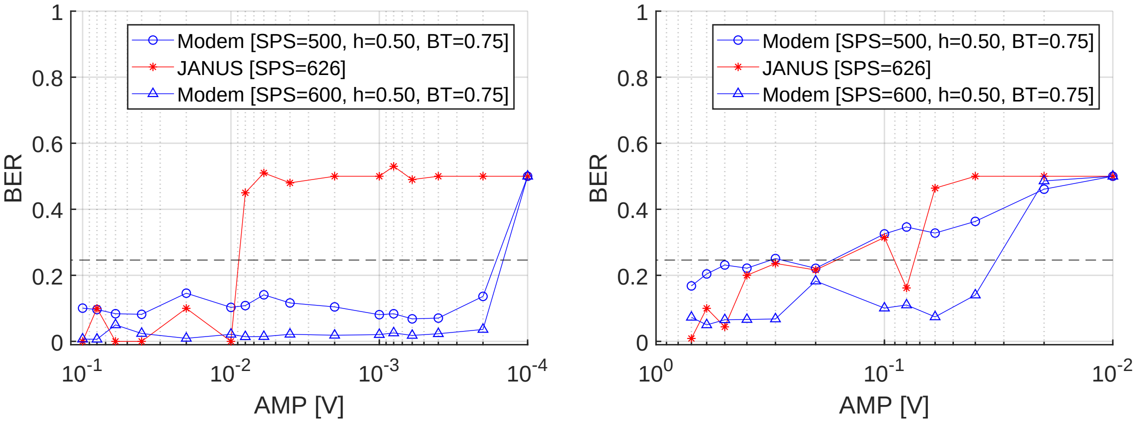
Figure 4.9: FLUMO results with different SPS.

are below 100μV. Indeed, at this range the DAC/ADC resolution is insufficient since the distance between two consecutive quantization levels is $2/2^{14} > 100\mu$. For other specific values where the performance was insufficient with all parameter settings (e.g. around 10^{-2}V), the behavior is probably due to the small barrel, where severe multipath can impact the reception of the frame. Figure 4.8 shows that the best performance settings are when the modulation index is 0.75 and filter bandwidth is 0.75. Moreover, the impact of the modulation index and filter bandwidth is more important at lower SPS.

We then select a fixed modulation index and filter bandwidth, and performed a set of experiments in order to find the lowest SPS possible for correct communication. Figure 4.9 shows the BER when we reduce the SPS from 600 to 200. From the figure, it is clear that FLUMO produces good performance results down to 300 SPS and in some cases even with 200 SPS, which translates to a gross data rate of 500bps if each symbol codes one bit.

4.3.2 Cistern experiments and comparison with JANUS

We repeated the same experiments in the outdoor cistern and compared our proposed modulation scheme with the JANUS standard implementation. The experiments confirmed the indoor results presented previously, while figure 4.10-a and figure 4.10-b compare the performance of JANUS against the proposed modulation (with 2 different SPS settings) both in the indoor and outdoor testbeds. In



(a) Small barrel

(b) Irrigation cistern

Figure 4.10: FLUMO modulation vs JANUS.

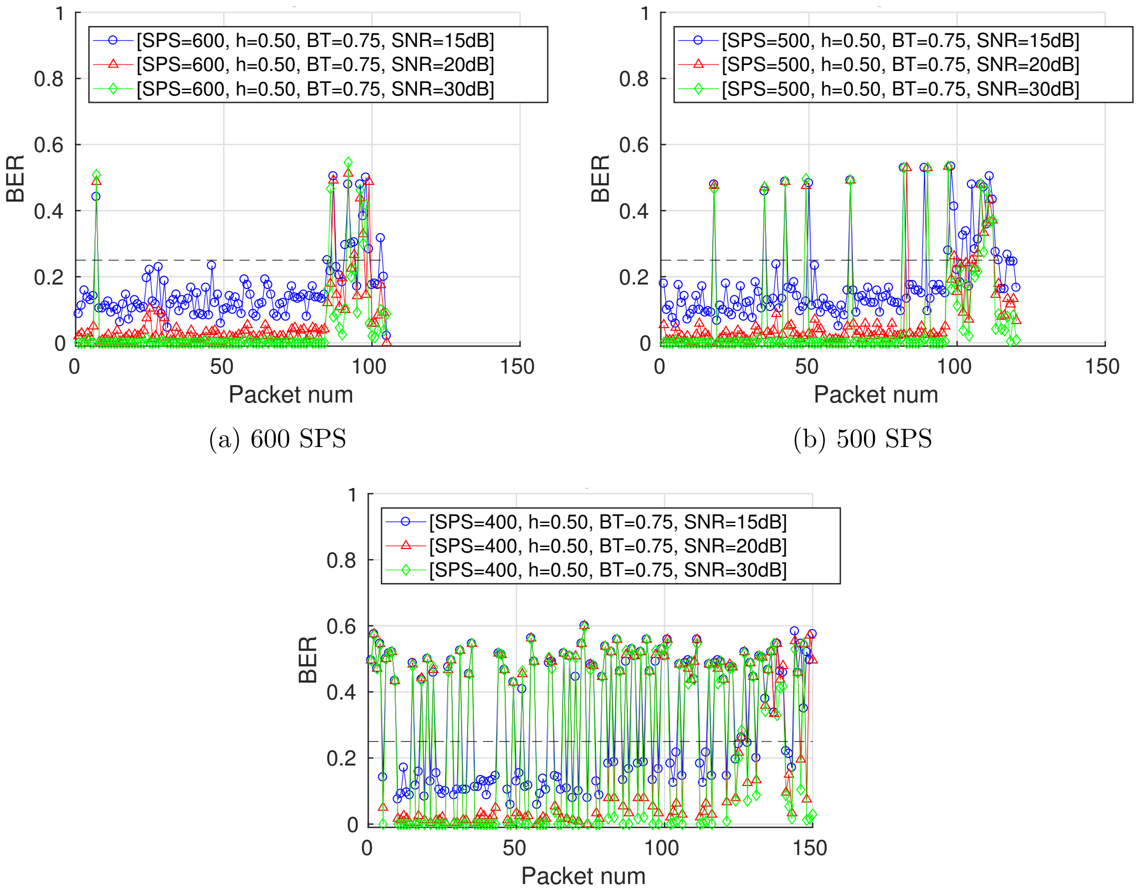
both environments, it seems that our proposed modulation scheme achieves better performance than JANUS implementation. However, as an example, the 50% bit errors of JANUS implementation shown in figure 4.10-a, may hide some detection (or synchronization) problem caused by very low amplitude of the signal leading to low SNR. Note that JANUS uses a preamble of only 32 bit while our modulation scheme uses until 120 bit.

4.4 Co-Simulation results

In this section we extend the experimental evaluation with Watermark, performing a co-simulation test of our modem implementation. In our experiments, we used the available SISO channels since most acoustic modems use a single receiver.

We used Watermark to validate in a realistic underwater environment the performance of our modulation scheme and to compare the results against JANUS. We selected the NOF SISO channel ($10-18kHz$ band) on top of which we added an increasing level of AWGN noise, with a residual Signal-to-Noise Ratio (SNR) from 30 dB downto 15 dB. We emphasize that this SNR level is not an absolute limit of the modulation, since the AWGN noise is added to the signal already affected by the Watermark channel.

Although simulated, such physical scenarios are much more challenging than the



(a) 600 SPS

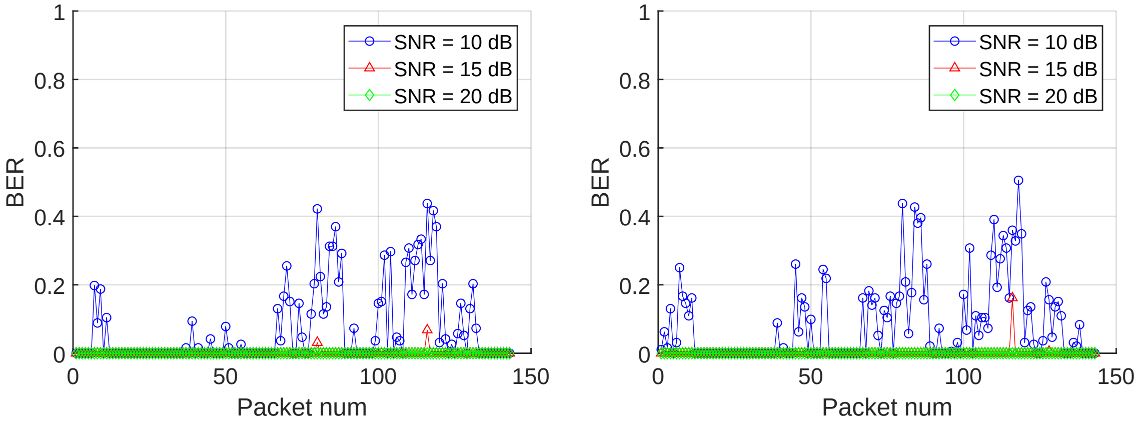
(b) 500 SPS

(c) 400 SPS

Figure 4.11: FLUMO results in Watermark with different SPS and SNR levels.

experiments in our testbeds, both because the NOF channel is a real at-sea measurement and because of the added noise level. Similarly to the testbed experiments, we varied different parameters settings (SPS, H and BT), transmitting at least 100 packets for each experiment and collecting hundreds of channel traces. We show here a small subset for the sake of simplicity.

Figure 4.11 shows the performance of FLUMO using 600, 500 and 400 SPS respectively. From the figure, it is clear that most of the time 500 SPS are usually sufficient even when the SNR is as low as 15dB, while 400 SPS are generally not enough. However, towards the end of the trace (about the last 20 packets) the underwater channel degrades and communications become difficult even with 600



(a) JANUS

(b) FH-FSK

Figure 4.12: JANUS results with Watermark and comparison with our FH-FSK modulation in different SNR conditions

SPS and 30dB of SNR. Similar results were obtained with different modulation index and filter bandwidth values. Differently from the experimental testbed results, with Watermark NOF the best performance settings are with modulation index 0.5 and filter bandwidth 0.75.

In order to assess the impact of carrier frequency hopping, we also tested the WSense/Sapienza implementation of JANUS. As reported in figure 4.12(a), the performance of JANUS on the same Watermark trace seems more stable and tolerates lower SNR levels. Indeed, JANUS enables a Doppler correction and uses a convolutional code for robust demodulation. Nevertheless, thanks to the flexibility offered by FLUMO, similar results can be obtained with a custom FH-FSK emulating JANUS, as shown in figure 4.12(b). In this latter case, the advantage stands in the fact that the modulation parameters can be easily customized to tailor the different underwater environments, adapting bandwidth occupation, symbol length, modulation index, etc., while in JANUS most of the parameters are defined by standard and can not be tuned.

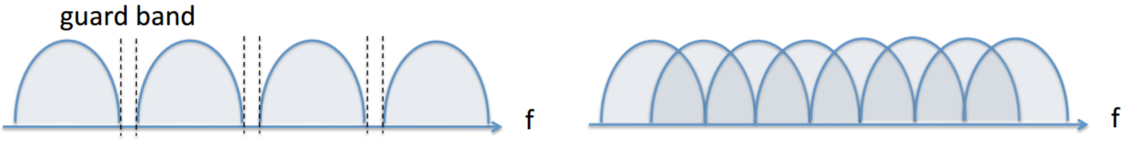
Chapter 5

High rate solution

5.1 Single-carrier vs Multi-carrier modulations

Single carrier modulation is the simplest form of coding data symbol onto a carrier signal. Generally, this process is carried out by varying one or more of the characteristic parameters of the carrier signal such as amplitude, phase and frequency. It is well known that, to obtain a symbol rate of R symbols per second, the minimum required bandwidth is the Nyquist bandwidth, which is given by $\frac{R}{2}$, so the only way to increase such as symbol rate is increase the bandwidth. In channels characterized by multipath fading, a bandwidth greater than the coherence bandwidth of the channel causes inter-symbol interference (ISI). In general, adaptive equalizers are employed to deal with the ISI incurred by the time-varying multipath fading channel. However, the complexity of an equalizer increases with the data rate, so beyond a certain limit high data rate single-carrier transmission may not be feasible due to too much complexity of the equalizer in the receiver.

Multi-carrier modulation represents the solution to achieve high data rate by overcoming the frequency selectivity of the wideband channel experienced by single-carrier transmission. This type of modulation is based on the idea of dividing the channel band into several smaller sub-bands, each of which is associated with a sub-carrier. Indeed, the flow of information is divided into several parallel sub-flows at lower speed and transmitted at the same time. In this way, the wideband channel can be approximated by multiple frequency-flat narrowband channels, and therefore



(a) FDM

(b) OFDM

Figure 5.1: FDM vs OFDM spectrum.

a low complexity equalizer can be applied because of in each of these sub-channels the attenuation is almost constant. This subdivision strategy can be also useful to perform optimization process, as an examples, it is possible to transmit larger amounts of information on the sub-bands that have a better Signal-to-Noise Ratio (SNR). In this sense, the width of each sub-band can be sized according to its channel attenuation characteristics and according to the noise present.

A widely multi-carrier modulation technique used in the past is Frequency Division Multiplexing (FDM). As can be view in the figure 5.1(a), in this approach the sub-bands don't overlap and are separated by a guard-band to ensure no inter-carrier interference with adjacent bands. It is clear that the introduction of guard bands, which are unused portion of the spectrum, lead to spectral inefficiency.

To improve spectral efficiency, Orthogonal Frequency Division Multiplexing (OFDM) is used.

5.2 OFDM overview

OFDM is a multi-carrier modulation technique in which adjacent bands are partially overlapped as in figure 5.1(b). Under suitable conditions, called orthogonality, the overlap among the sub-bands does not generate interference phenomena and in any case allows a right signal reception. The orthogonality condition can be obtained by sizing the system in such a way that the distance among the various subcarriers equals the reciprocal of the duration of an OFDM symbol. Denoting with Δf the subcarriers spacing and with T_s the symbol duration, orthogonality can be expressed as:

$$\Delta f = \frac{1}{T_s} \tag{5.1}$$

and the carriers frequencies follow the relation:

$$f_k = f_c + k\Delta f \qquad\qquad k = -\frac{N}{2}, ..., \frac{N}{2} - 1 \tag{5.2}$$

where f_c is the reference carrier frequency and N is the number of subcarriers.

The bit sequence to be transmitted with rate $R = \frac{1}{T}$ is distributed into the N different subcarriers. In each flow, the bit are mapped with a conventional modulation at a rate $R_s = \frac{R}{N} = \frac{1}{NT} = \frac{1}{T_s}$ and the resulting waveforms are summed together to compose one OFDM symbol.

The band-pass OFDM signal can be expressed as:

$$x(t) = \Re\{\hat{x}(t)e^{j2\pi f_c t}\} \tag{5.3}$$

where $\hat{x}(t)$ is the equivalent base-band OFDM signal:

$$\hat{x}(t) = \sum_{n=-\infty}^{+\infty} \sum_{k=-\frac{N}{2}}^{+\frac{N}{2}-1} a_{n,k} g(t - nT_s) e^{j2\pi k\Delta f t} \tag{5.4}$$

and $a_{n,k}$ is the mapped constellation symbol in the k-th sub-carrier at time n.

The signaling pulse g(t) is defined in the interval $t \in [0, T_s]$ and must be chosen to satisfy the Nyquist ISI-free criterion at time instants T_s:

$$g(mT_s) = \begin{cases} 1 & \text{for } m = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

Under these conditions the OFDM modulation can be implemented efficiently through the use of IFFT/FFT algorithms. Indeed, if we sample the OFDM signal at time instants $T = \frac{T_s}{N}$, we obtain:

$$\begin{aligned} \hat{x}(m\frac{T_s}{N}) &= \sum_{n=-\infty}^{+\infty} \sum_{k=-\frac{N}{2}}^{+\frac{N}{2}-1} a_{n,k} g(m\frac{T_s}{N} - nT_s) e^{j2\pi k\Delta f m \frac{T_s}{N}} \\ &= \sum_{n=-\infty}^{+\infty} \left[\sum_{k=-\frac{N}{2}}^{+\frac{N}{2}-1} a_{n,k} e^{j\frac{2\pi}{N}km} \right] g(m\frac{T_s}{N} - nT_s) \end{aligned} \tag{5.6}$$

The amount in the square brackets is N times the IDFT of the constellation symbols $a_{n,k}$:

$$b_{n,m} = \sum_{k=-\frac{N}{2}}^{+\frac{N}{2}-1} a_{n,k} e^{j\frac{2\pi}{N}km} \tag{5.7}$$

and the sampled OFDM signal become:

$$\hat{x}(m\frac{T_s}{N}) = \sum_{n=-\infty}^{+\infty} b_{n,m} g(m\frac{T_s}{N} - nT_s) \tag{5.8}$$

with $m = lN + i$, $l = 0, 1, ...$ and $i = 0, ..., N-1$ Consequently, at transmitter side, each baseband OFDM symbol can be generated performing an IFFT operation and then upconverted and shifted to pass-band carrier frequency through the use of the transmitter pulse shaping filter. At receiver side, the inverse operation will be executed, i.e. after a base-band shifting, the signal is then downconverted using the pulse shaping filter and through an FFT operation the constellation symbols can be recovered.

5.2.1 Cyclic prefix

The impact of the multipath can be mitigated by introducing an appropriate guard time between symbols. The guard time must be chosen long enough to cover the entire delay spread of the channel. In this way, the interference due to reflections impact only in the guard-time part of the symbol, thus leaving unchanged the rest. Naturally, the introduction of this guard time leads to an overhead because it does not carry any useful information.

Mainly, there are two methods to add a guard time: zero-padding and cyclic prefix insertion. The former is obtained just adding a simple silence, i.e. no-transmission time, between symbols. Its main advantage stands on the lower overall power consumption of the system.

With the cyclic prefix method, the last part of the OFDM symbol, corresponding to an amount equal to the guard time, is copied to the head of it as showed in figure 5.2. This method accomplish the two main target: first implement the aforementioned guard time dealing with ISI and second it allows to exploit an important property of convolution to simplify the channel equalization. Indeed we know that

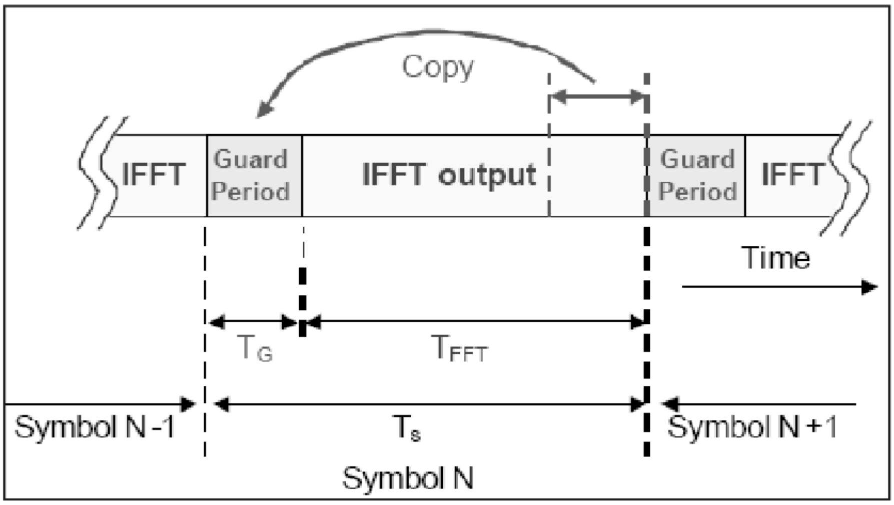


Figure 5.2: Cyclic prefix in OFDM symbol.

the transmission of the OFDM signal through the channel can be represented as a linear convolution of the signal itself with the impulse response of the channel:

$$y(n) = \sum_{l=0}^{L-1} h(l)x(n-l) \tag{5.9}$$

where h(n) is the discrete impulse response of the channel and x(n) is the OFDM signal. Because of the latter is cyclic-prefixed, the convolution with the cyclic prefix part can be viewed as the convolution with the last part of the OFDM symbol thus wrapping the impulse response of the channel around the FFT window of the OFDM symbol. In this sense the cyclic prefix turn the linear convolution to a circular convolution.

Making use then of the discrete convolution theorem:

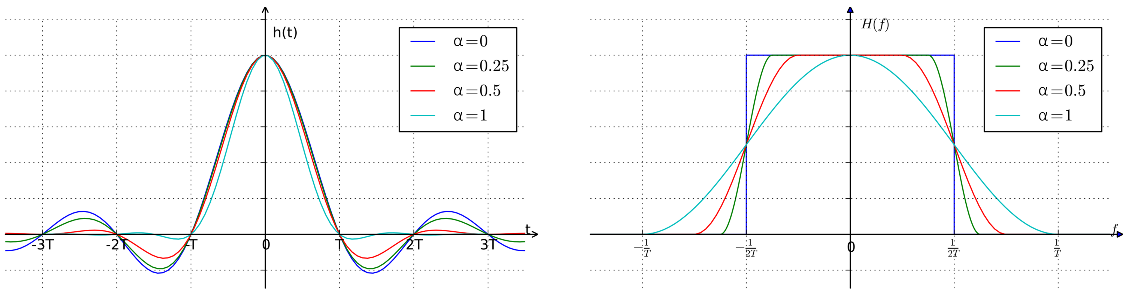
$$Y(k) = DFT_N\{h(n) * x(n)\} = H(k)X(k) \tag{5.10}$$

we can make easily channel equalization, estimating the transmitted symbol as:

$$\hat{X}(k) = \frac{Y(k)}{H(k)} \tag{5.11}$$

5.2.2 Pulse shaping

Although the pulse g(t) can be theoretically an ideal rectangular pulse with unlimited bandwidth, the real channel is band-limited. Therefore, when the signal pass



(a) Impulse response

(b) Spectrum

Figure 5.3: Raised Cosine filter at different values of roll-off factor α.

through the channel is deformed by it leading to unacceptable ISI.

To fit the signal to real channel bandwidth and avoid these phenomena a pulse shaping operation is executed. More in detail, this is performed by filtering the signal to be transmitted with a pulse shaping filter satisfying the Nyquist ISI-free criterion.

Usually the Raised Cosine filter is used as pulse shaping filter. Its spectrum is described by:

$$H(f) = \begin{cases} 1, & |f| \leq \frac{1-\alpha}{2T} \\ \frac{1}{2}\left[1 + cos\left(\frac{\pi T}{\alpha}\left[|f| - \frac{1-\alpha}{2T}\right]\right)\right], & \frac{1-\alpha}{2T} < |f| \leq \frac{1+\alpha}{2T} \\ 0, & \text{otherwise} \end{cases} \tag{5.12}$$

The filter is completely defined by means of the two parameters α and T, where α is the roll-off factor and T represents the symbol period of the transmission system. Figure 5.3 shows the impulse response of the raised cosine filter and its spectrum at different values of α. As can be viewed in the figure, the spectrum of the filter can be splitted in two parts: a flat central part and the two side parts composed by a piece of cosine function. The roll-off factor defines the excess bandwidth used compared to a pulse with a rectangular spectrum of equal bandwidth. It can vary from 0 to 1. When $\alpha = 0$ the impulse response become a sinc function and the frequency spectrum become an ideal rectangular pulse. Instead, with $\alpha = 1$ the whole spectrum is a cosine period shifted above the frequency axis.

Usually the filter is splitted between the transmitter and the receiver. Both of them use a Root Raised Cosine filter defined as $|H_R(f)| = |H_T(f)| = \sqrt{|H(f)|}$ with

$H_R(f) \cdot H_T(f) = H(f)$. In this way, at receiver side the filter act as matched filter maximizing then the Signal-to-Noise Ratio (SNR).

5.2.3 Advantages and disadvantages

From previous discussions can be deduced some considerations about benefits or drawbacks on the OFDM modulation. More in detail, among the advantages of OFDM modulation there are:

- Simplified equalization and high immunity to frequency selective fading. Splitting the entire band in smaller sub-bands ensures that in each sub-band the channel response can be considered almost flat.
- Robustness to interference phenomena. Indeed, interference generally affects the various sub-bands in a scattered mode and does not involve the entire signal but portions of it.
- Good spectral efficiency. Because the overlap among the sub-bands allows to safeguard bandwidth.
- Robustness to ISI. Indeed, lower speed transmissions in the different sub-bands are less prone to this phenomenon.

Among the disadvantages instead we can find:

- High Peak-to-Average Power Ratio (PAPR). It represents the ratio between the peak power value of the signal and its average power. In OFDM modulation this ratio can be very high and cause signal distortion due to the non-linearity of hardware power amplifiers.
- Sensitivity to Carrier Frequency Offset (CFO). This can be due to frequency difference between transmitter and receiver oscillators. It lead to inter-carrier interference (ICI).
- Guard time. Its presence, is needed to deal with ISI, against multipath phenomenon. The guard time lead to overhead decreasing speed performance.

5.3 OFDM Implementation

In this section we present our OFDM implementation. We made use of the C library liquid-dsp so that the implementation can be easily integrated into in various embedded systems, and more in detail, in our SDA platform. Moreover, the library is released with several already implemented functionalities and examples. Therefore, this allowed us to use different functions of the library to manipulate bit stream or signals, like, for examples, scrambler, various FEC schemes, interleaver, modulation schemes, CRC schemes, filters, oscillators and so on. These functions were in some case adapted or rewritten for our needs.

As we will see in the next sections we developed a complete transmitter and an improved receiver working on a symbol-by-symbol basis, i.e. where each symbol can be recovered independently from each others. This allow us to apply a recursive symbol-based channel estimation improving performances.

The resulting implementation is fully parameterizable and can be employed in different environments. Its flexibility also allowed to execute various experiments with different configurations. More in detail, the system allows to change the most important parameters of an OFDM system such as the number of subcarriers, the length of the cyclic prefix and the number of pilot subcarriers. Also the modulation scheme and FEC scheme can be selected between a variety contained in the liquid-dsp library. Moreover the pulse shaping parameters can be changed. Finally custom add-on can be implemented and easily added to the system.

5.3.1 Framing

The digital platform takes care of assembling the frame at transmitter side and extracting the message at receiver side, performing all the low-level modulation/demodulation functions.

Figure 5.4 shows the OFDM frame structure implemented in our system. More in detail, it consists of three parts:

- Preamble. It consists of one or more symbols composed by pseudo-random sequences that are not scrambled and no channel coding and interleaving is applied to them. These sequences are always modulated QPSK and the mapped

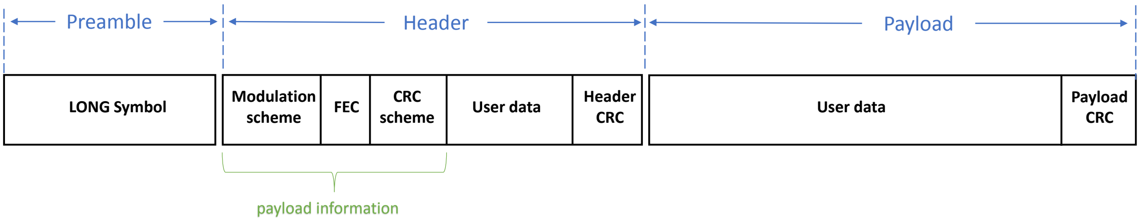


Figure 5.4: OFDM frame structure.

symbols are distributed over all the subcarriers except the null ones. The preamble is used for detection and synchronization and for the first estimate of the channel response.

- Header. It consists of a variable number of symbols carrying a fixated number of bytes that is known to both the transmitter and the receiver. The number of symbols depends of the number of data subcarriers that has been allocated. The header bytes are composed by a fixed part and a user-defined part. In the fixed part, are always carried the payload framing information such as the modulation scheme, the FEC scheme, the CRC scheme and its length. A CRC is appended in order to allow the receiver to detect a possible reception error. Once the header is correctly extracted, the receiver continue for extracting the payload.
- Payload. It consists of an arbitrary number of symbols that carries the message bytes. The number of symbols depends on the length of the message bytes length, and, as for the header, varies with the number of data subcarriers allocated. As for the header, for similar reasons, a CRC is appended to it.

5.3.2 Transmitter

Figure 5.5 shows the block diagram of the implemented OFDM transmitter considered in this work. We can ideally consider the system as composed of two main parts: the high-level bit stream part that deals with the manipulation of the bit stream and the low-level OFDM modulator part that deals with conversion of the bit stream in a OFDM signal to be sent to the hardware.

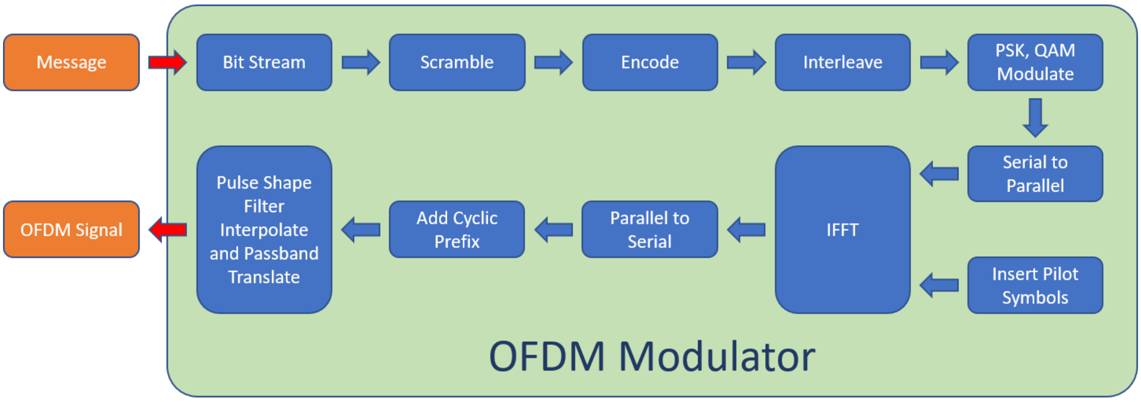


Figure 5.5: Block scheme of the OFDM transmitter.

More in detail, in the high-level part, the transmitter execute sequentially the following functions:

- Bit stream. The user message (for example an ASCII encoded string) is converted into a bit stream. The obtained bit stream is then subdivided into blocks of equal length, where the number of bits contained in each block is pre-computed in such a way that, after the PSK/QAM mapping, the number of constellation symbols fit exactly in one OFDM symbol. If the number of bits is not enough to cover the last block, a pad of 0s is added to the bit stream.

- Scrambling. Each blocks of bit pass through an independently scrambling operation. The scrambling purpose is to remove any periodic sequences or long sequences of 0s or 1s from the input bit sequence. These periodic sequences can bring to unwanted frequency components in the generated signal. This help also to reduce the PAPR of the final OFDM signal. The scrambling operation is performed by multiplying or adding to the original binary sequence a pseudo-noise binary sequence (XOR and OR), named also whitening sequence.

- Encoding and interleaving. The scrambled bits are individually encoded with a convolutional code and then interleaved. The convolutional code belongs to the channel coding also named Forward Error Correction (FEC) used to preserve the integrity of the data sent through the channel. Channel coding introduce redundancy in the information flow that is reflected in a lower effective communication speed, but it is essential to recover and correct errors in the received information. Interleaving process is performed to give robustness to the used channel coding by shuffling the coded bit stream and thus breaking its continuity. If contiguous errors (i.e. burst errors), occurs during transmission, they appear as scattered errors after the deinterleaving operation executed by the receiver. This improves the decoding procedure.

Once the logic bit stream is ready, the transmitter can use it to create the OFDM signal in the low-level part executing the following functions:

- PSK/QAM mapper. The information bits are mapped to symbols according to the chosen modulation scheme. The symbols mapping is performed using grey code to facilitate the decoding process.

- Baseband OFDM modulation. This action is performed by five elements. More in detail, the constellation symbols are parallelized (each corresponding to one sub-flow) and the pilot symbols are inserted before the IFFT execution. Then, after IFFT computation, the symbols are re-serialized, and the cyclic prefix is added obtaining the OFDM baseband signal.

- Upconversion. The baseband OFDM signal is filtered by the pulse shaping filter. Through this operation the baseband OFDM signal is interpolated and therefore upsampled to the sampling frequency required by the hardware. Moreover, the signal is multiplied by carrier frequency and therefore shifted to the channel band, obtaining the final passband OFDM signal.

5.3.3 Receiver

Figure 5.6 shows the block scheme of the OFDM receiver, which performs channel frequency response estimation on a symbol-by-symbol basis, by using decision-feedback detection (rather than on the preamble only, at the beginning of the packet). The decision feedback receiver structure is needed in order to be able to track channel frequency response variations during the frame. The process can be summarized as follow:

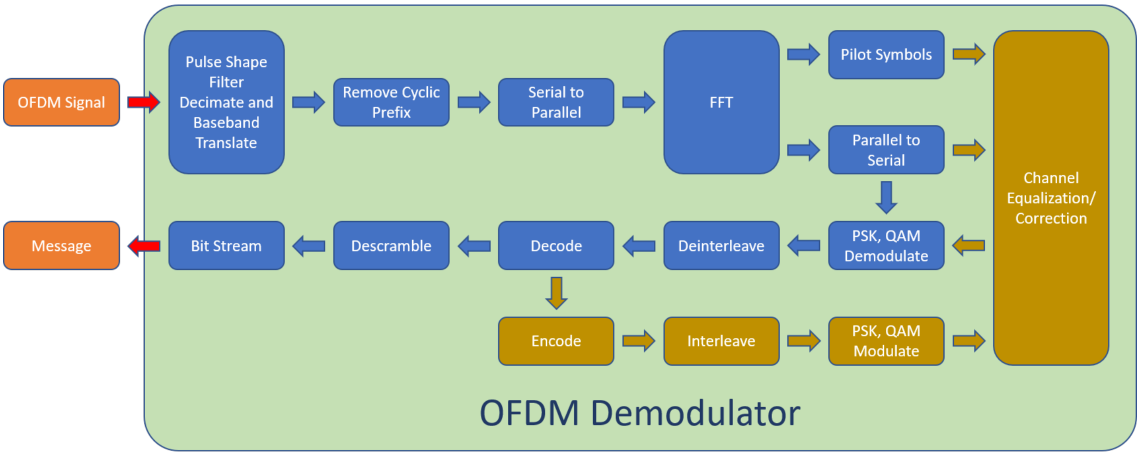


Figure 5.6: Block scheme of the OFDM receiver. The traditional receiver scheme is enhanced with re-encoding and channel correction blocks in yellow.

- First, the receiver performs a correlation operation by which it identifies the beginning of the packet. Once the signal is detected, the receiver achieves an initial channel estimation by comparing the received preamble symbol with the expected one. Denoting with Y_0 the received preamble symbol and with X_0 the transmitted preamble symbol, we can obtain the initial channel estimate $\hat{H}_0$ as
$$\hat{H}_0(k) = \frac{Y_0(k)}{X_0(k)} \tag{5.13}$$
- The channel estimate $\hat{H}_0$ is used to correct the impact of the channel on the first data symbol. Therefore, this one is equalized using $\hat{H}_0$ and then the phase offset is corrected using pilot carriers, obtaining
$$X_1(d) = \frac{Y_1(d)}{\hat{H}_0(d)} e^{j\theta_1} \qquad d \in D, \quad \theta_1 = \angle \sum_{p \in P} Y_1^*(p) X_1(p) H_0(p) \tag{5.14}$$
where D is the data carriers set, P is pilot carriers set and * denote the complex conjugate.
- Once the symbol X_1 is obtained, the receiver decodes the symbol with the help of the convolutional coding and then re-encodes the bits to correct the channel equalization parameters: indeed, this regenerated signal represents the

symbol originally transmitted by the sender and can be used as a reference to update the channel estimate using all available subcarriers by comparing the constellation points actually received with the regenerated ones.

$$X_1(d) \xrightarrow{\text{REGENERATE}} \hat{X}_1(d), \quad \hat{H}_1(d) = \frac{Y_1(d)}{\hat{X}_1(d)} \tag{5.15}$$

- The corrected channel estimate $\hat{H}_1(d)$ is finally used to equalize the reception of the next symbol and, recursively, the channel equalization is updated at every symbol received.

5.4 Experimental results

We tested the performance of our OFDM modem in simulated and real channels, under different configurations of the number of subcarriers N, prefix length N_{CP}, pulse shaping filter roll-off factor α_p and filter length L_p. We set the length of the OFDM packet to 4, 8, 16 or 32 symbols which, depending on the parameters and modulation used, can carry from few hundred bytes to a few kilobytes. The bits to be transmitted are generated randomly, consistently with the number of OFDM symbols chosen, and we apply a convolutional coding with rate equal to 1/2 to the data (i.e. only half of the bits are information, while the rest are redundancy bits generated by the convolutional code). The bits are then interleaved and mapped using a QPSK constellation to the various data subcarriers.

To obtain the performance of our OFDM modem in simulated channels we used the Watermark simulator. Instead, the performances in real channel were obtained executing experiments indoor, in an aquarium, and outdoor directly at-sea. For this purpose the commercial Evologics modem were used as hardware front-end. These modem provide an interface to operate in sdm mode allowing to pass it arbitrary waveform to be transmitted. We used a model that operates in the 18-34kHz band, for an available bandwidth of 16kHz with a sampling frequency of 250kHz.

5.4.1 Experiments with simulated channels

In order to plan the simulation experiments, we obtained each delay-spread and Doppler-spread of the five available channels in Watermark. Indeed, Watermark provide an estimate of the impulse responses of them, and allowed us to compute an estimate of the two amounts. The knowledge of the delay-spread and the Doppler-spread allowed us to have a rough estimate of the number of subcarriers and of the length of the cyclic prefix that can be used.

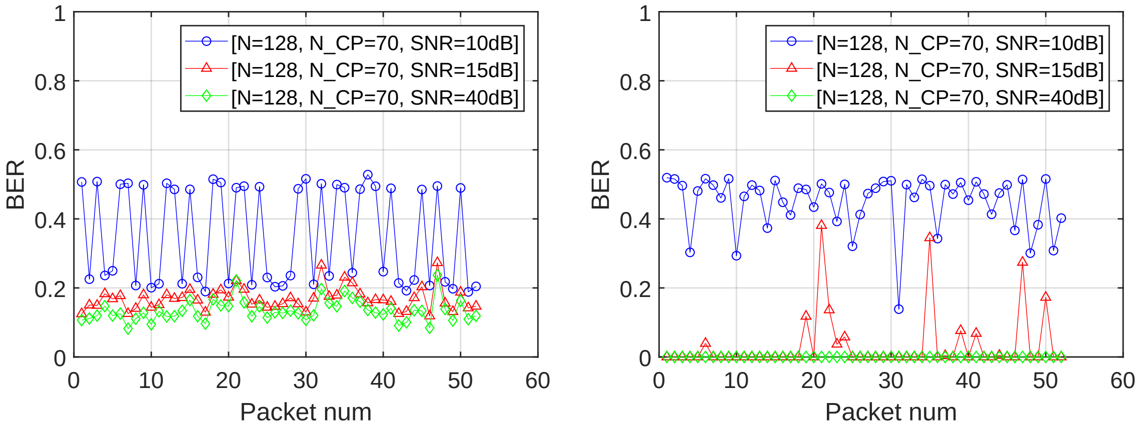
The computation results showed that only the Norwegian channel NOF1 and the French channel BCH1 allow the transmission of OFDM modulated signals. The other channels, would require a cyclic prefix greater than the number of subcarriers, since the Doppler-spread force to use only few subcarriers to avoid ICI. The simulation experiments were therefore executed using only the two aforementioned channels varying the parameters around the obtained rough estimate values. Indeed, we change the number of subcarriers from a minimum of 64 to a maximum of 256 (in power of 2) and the cyclic prefix in the range from 30 to 110 samples (never exceeding the number of subcarriers). More in detail, the experiments were executed changing all configuration parameters as follow:

- Roll-off factor of the RRC filter with values of 0.15, 0.20 and 0.25
- RRC filter length with values of 6, 9 and 12
- Number of subcarriers with values of 64, 128 and 256
- Length of the cyclic prefix with values between 50 and 110, step 20

As already said the roll-off factor defines the bandwidth used by the signal. Smaller roll-off values result in higher spectral efficiency, as more subcarriers can be used for data, but requires longer filtering delays (i.e. higher filter length). However, changing the filter roll-off factor and the filter length parameters have less impact compared to number of subcarriers and cyclic prefix length.

For each experiment, we transmit at least 100 packets on each sounding, collecting thousands of channel traces in total. We show here a subset of the results on NOF1 channel. In the experiments, we also added an increasing level of AWGN noise, with a residual Signal-to-Noise Ratio (SNR) from 40 dB down to 10 dB. As already specified, the SNR level shown does not take into account the noise added to the signal by the NOF1 channel itself.

Figure 5.7 shows the BER obtained in Watermark's NOF1 channel using packets of 16 symbols, a number of subcarriers equal to 128, and a cyclic prefix length equal



(a) Traditional receiver

(b) Improved receiver

Figure 5.7: OFDM results in Watermark with packet length of 16 symbols and different levels of SNR. The improved (decision-feedback) receiver significantly reduces the BER if the SNR is 15 dB or above.

to 70. The RRC filter was set with a roll-off factor of 0.2 and a filter length equal to 6. From the figure, we can see a comparison between the performance of the traditional receiver, and the improved receiver using the feedback chain for updating the channel estimate. It is clear that even with high SNR, the impact of the channel can be relevant using a traditional OFDM receiver, causing about 15% BER even with a high SNR. Instead, the improved OFDM receiver significantly reduces the BER down to almost zero if the SNR is 15 dB or above. Note that a cyclic prefix length of 70 is slightly below the ISI limit, but the improved receiver still works if the SNR is high enough. However, if the SNR decreases, the proposed decision-feedback receiver will suffer from error propagation: indeed, the channel estimates used for detection are corrupted by channel noise, and poor estimates lead to high error rates which in turn cause a failure of subsequent channel estimates. This is evident for the 10 dB example shown in figure 5.7 (b), and improves with higher SNR ratios. Nevertheless, increasing the cyclic prefix length to 90 samples is enough to get reliable performance even with packets that are 32 symbols long, as shown in figure 5.8. However, increasing it further does not provide much benefit.

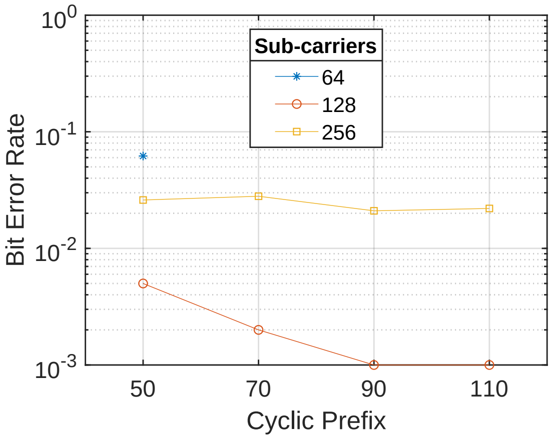


Figure 5.8: Impact of the OFDM cyclic prefix length with different number of subcarriers and packet length of 32 symbols.

5.4.2 Experiments in aquarium

As intermediate step, we executed some experimentation in indoor environments, specially to test the real-time implementation. We chose to perform tests both in air and in a small aquarium. Since these environments can be considered as static channels, we expect for them a very negligible Doppler-spread. Instead, the two different environments are characterized by different multipath phenomena. Indeed, in the air the incidence of reflections was very low, since the transducers were placed very close. Conversely, in the aquarium, there are the nearby walls and the water surface that cause reflections. However, the multipath components are located close to each other leading to a not much high delay-spread [43].

The tests executed in air are not reported, since they have a null BER pratically with the majority of the tested configurations. Indeed, as we expected, they achieved very good performance using a number of subcarriers up to 4096 and with a very small cyclic prefix. We also changed the modulation scheme of the subcarriers to a less robust 16-QAM. As we expected the average BER increases, although some error-free configuration reaches a speed of 20kpbs (throughput) with a spectral efficiency of about 3.2 bit/s/Hz.

Figure 5.9, shows instead the aquarium environment that we used. More in detail, for these experiments we varied the parameters covering a wide range of

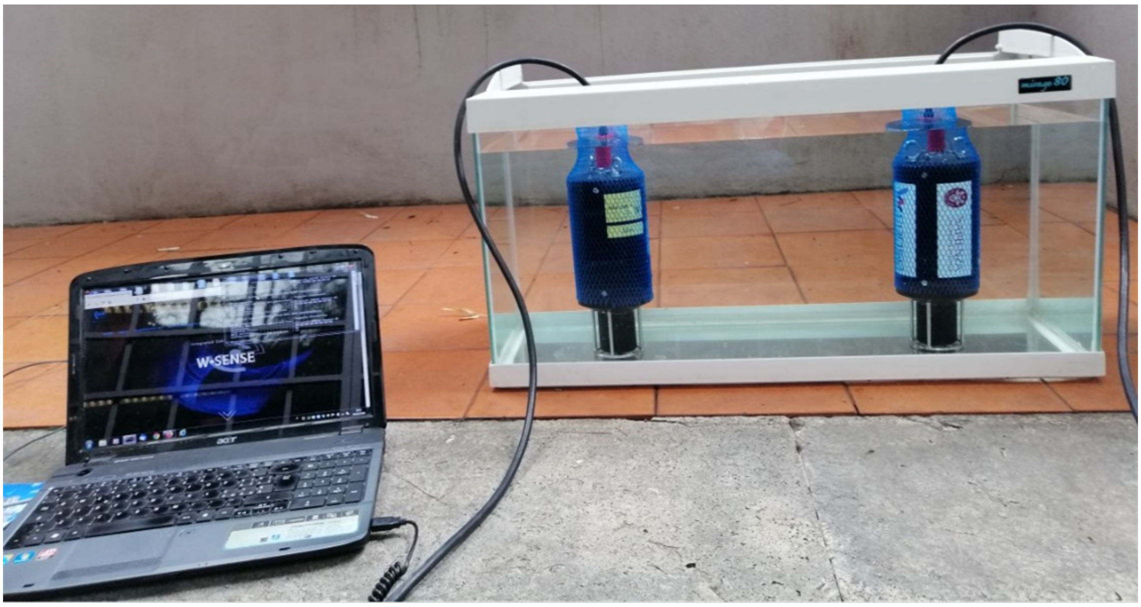


Figure 5.9: Aquarium test environment.

values as follows:

- Roll-off factor of the RRC filter with values equal to 0.20, 0.25 and 0.30
- RRC filter length with values of 6, 9, 12
- Number of subcarriers with values between 32 and 1024, step power of 2
- Length of the cyclic prefix with values between 16 and 240, step 32

Figure 5.10 show a subset of the results obtained by transmitting 32 OFDM symbols with RRC filter roll-off factor equal to 0.20 (figure 5.10a) and 0.30 (figure 5.10b). Both configurations use a filter length equal to 9. We can see as the most stable configurations use 1024 subcarriers. To deal with multipath reflection a cyclic prefix with a length greater then 48 is enough. In any case, the BER obtained is often below 5% in most of the configurations used. The maximum transmission speed obtained without errors was about 10kbps (throughput) with a spectral efficiency of about 1.6 bit/s/Hz.

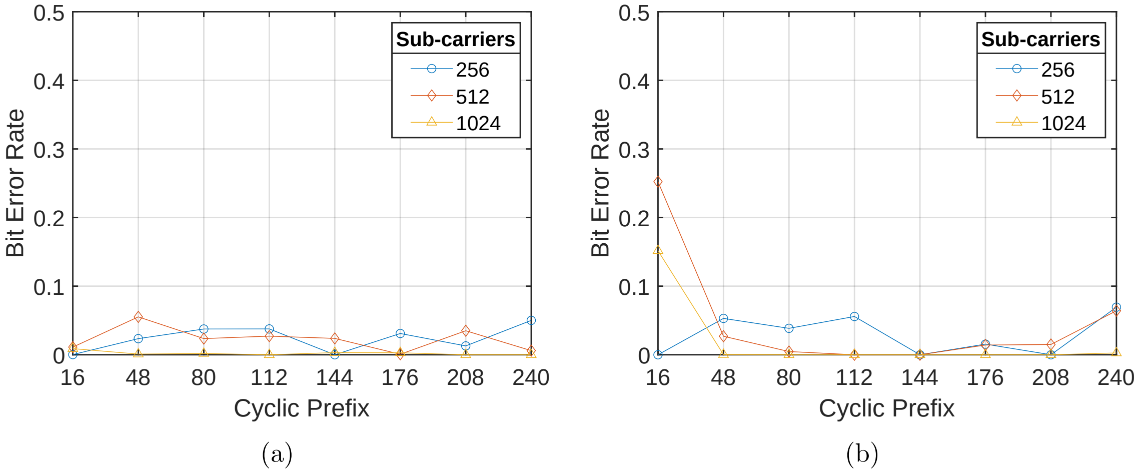


Figure 5.10: Average BER obtained indoor for OFDM transmissions with 32 symbols packets as a function of cyclic prefix and different number of subcarriers.

5.4.3 Experiments at sea

Finally, we tested our OFDM implementation by repeating the experiments at sea in a touristic harbor at Santa Marinella, close to Rome, Italy. We placed the transmitter and the receiver approximately 30m apart, between to two floating docks shown in figure 5.11. The water depth was 3m and the transducers were placed about 1m below sea level.

In this case, we didn't have a rough estimate of the delay-spread and Doppler-spread for this scenario. We started, as a first approximation, considering the values obtained in the Watermark BCH1 channel, since it provides a real channel measurements in an harbor. Moreover, since the transducers have a bandwidth of 16 kHz (thrice as much as the BCH1 Watermark channel), we also increased the range of OFDM parameters to be tested. More in detail, the following configurations were tested:

- Roll-off factor of the RRC filter with values equal to 0.15, 0.20 and 0.25
- RRC filter length with values of 6 and 9
- Number of subcarriers with values of 256, 512 and 1024
- Length of the cyclic prefix with values between 16 and 240, step 32

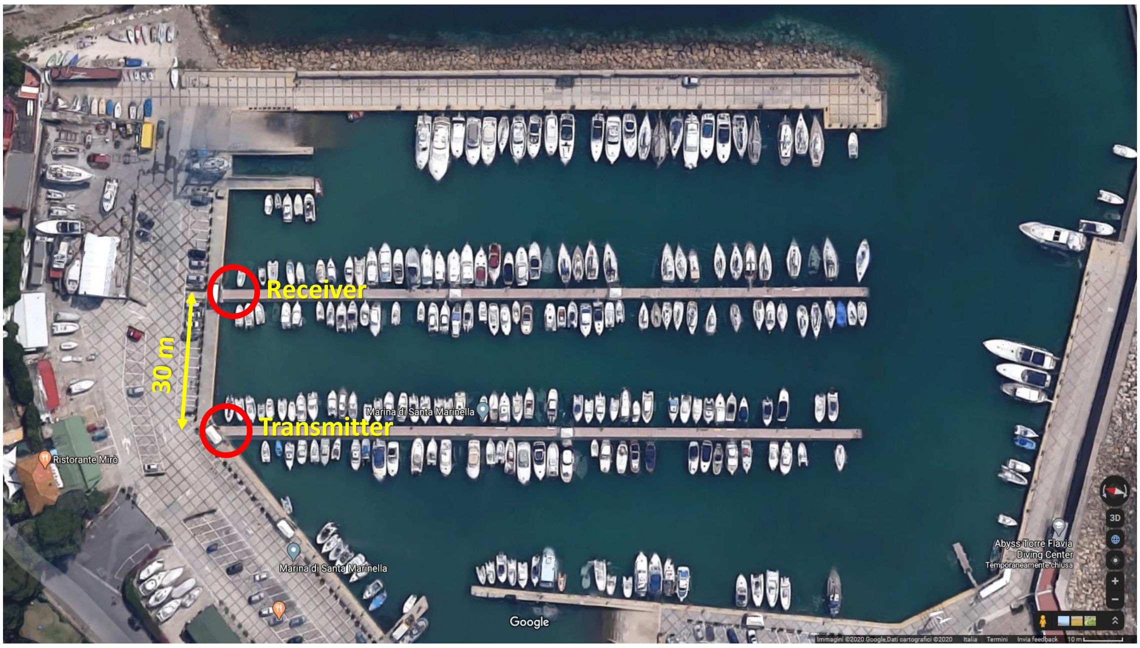

Receiver
30 m
Transmitter
Marina di Santa Marinella
Ristorante Mirò
Marina di Santa Marinella
Abyss Torre Flavia Diving Center Temporaneamente chiusa
Google

Figure 5.11: Experimental setup in a touristic harbor in Santa Marinella, close to Rome, Italy.

For a worst case scenario employing 32 symbols packets, figure 5.12 shows that in this scenario, good performance can be obtained even with 1024 subcarriers, achieving a spectral efficiency of 1.4 bit/s/Hz (corresponding to about 20kbps uncoded) with a BER lower than 5% even with packets 32 OFDM symbols long. This in line with the results of [35], although the BER is slightly higher due to the harsher experimental environment. More in detail, the figure refers to an experiment where the RRC filter was set with a roll-off factor of 0.2 and a filter length equal to 9. As we can see, in the more stable configuration with 1024 subcarriers, the cyclic prefix should be kept greater than 144 in order to minimize the impact of multipath propagation in this scenario.

Table 5.1 shows some representative results obtained during the tests. In addition to the parameters relating to the various configurations used, the BER obtained, the size of the transmitted packet and the transmission speed are specified. The

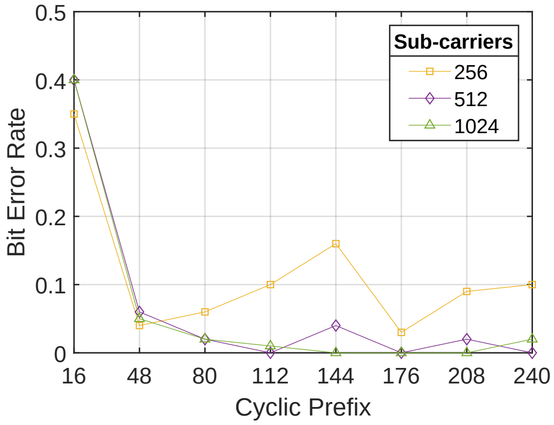


Figure 5.12: Average BER obtained at sea for coded OFDM with 32 symbols packets as a function of cyclic prefix and different number of subcarriers.

latter is a function of the configuration used and represents the throughput, i.e. the real information speed reached.

Table 5.1: Representative results obtained during transmissions at sea.

RRC Roll-Off Factor	RRC Filter Length	Subcarriers Number	Cyclic Prefix Length	BER	Packet Length (Bytes)	Bitrate (kbps)
0.20	9	1024	144	0.00	3240	9.15
0.20	6	1024	208	0.00	3240	8.67
0.20	6	1024	112	0.00	3240	9.41
0.20	6	1024	176	0.00	3240	8.90
0.20	6	1024	240	0.00	3240	8.45
0.20	9	1024	176	0.00	3240	8.90
0.15	6	1024	112	0.00	3448	10.56
0.15	9	1024	208	0.00	3448	9.74
0.20	9	1024	208	0.00	3240	8.67
0.25	9	1024	176	0.00	3048	7.96
0.15	6	1024	144	0.01	3448	10.27
0.15	6	1024	208	0.01	3448	9.74
0.20	9	1024	112	0.01	3240	9.41
0.15	9	1024	144	0.01	3448	10.27
0.25	6	1024	208	0.01	3048	7.75
0.25	6	1024	144	0.01	3048	8.18
0.15	6	1024	176	0.01	3448	10.00
0.25	6	1024	240	0.01	3048	7.56
0.15	6	1024	80	0.01	3448	10.87
0.20	9	1024	240	0.02	3240	8.45

Chapter 6

Channel-Aware adaptive modem

6.1 Introduction

Different solutions have been proposed for dealing with the Doppler effect, by estimating and compensating at the receiver the time compression or dilatation effect due to the Doppler on a per-packet (e.g. section 3.3) or per-symbol basis (e.g. [17]). However, most of these approaches significantly reduce the data rate or increase the computational complexity, and therefore the power consumption, of the receiver.

In this work, we investigate on a simple strategy for coping with the Doppler effect: rather than implementing a complete Doppler compensation technique at the receiver, we periodically estimate the channel spreading parameters and dynamically adjust the OFDM parameters in order to be robust under the maximum expected Doppler shift and multipath delay. While usual OFDM adaptations work on the choice of constellations with different cardinality, in our case we adapt the number of subcarriers to be deployed over the channel bandwidth and the length of the cyclic prefix.

Differently from approaches viewed in literature, our solution does not try to estimate and correct the exact Doppler scale over time, but it estimates the worst expected Doppler shift for tuning the OFDM parameters, and compensates symbol by symbol the average clock drift. This design choice leads to a solution with a complexity much lower than [20], which works well under the assumption that, even if the instantaneous Doppler shift and multi-path Delay parameters can vary relatively fast, the overall channel spreading parameters will be slowly time-varying.

The basic assumption for our design is that, for many applications, the underwater channel is good enough to allow the set-up of an OFDM-link and an opportunistic (periodic) calibration of the OFDM modulation parameters is possible in scenarios with mild temporal variability of environmental parameters due for example to particular weather conditions, temperature and salinity of the water, as well as the site-specific propagation environment.

To demonstrate the effectiveness of our approach, we designed a low-complexity estimation technique for time-varying channels, based on the transmission of predefined linear modulated frequency chirps, and an intelligent module for tuning the OFDM parameters as a function of the channel spreading estimates. The modules have been validated in different simulation platforms for underwater communications and in real experiments at sea. For the experimental validation, we exploited the software-defined-modem developed by our research group, which has been extended for running: i) the intelligent module for channel-adaptation, ii) an OFDM modulator exposing a configuration interface to the intelligent module; iii) a JANUS modulator, providing a robust feedback (and back-up) channel for notifying the channel spreading estimates to the transmitter.

As detailed in the next section, we extended our platform for automatically tuning the number of OFDM subcarriers and cyclic prefix length, as a function of the estimated channel conditions. This capability is different from previous OFDM modems, which typically work on the adaptation of the modulation parameters, only according to the reception errors [35].

The rest of the work is organized as follows: first we present the characteristics of the proposed system architecture in section 6.2 and the channel estimation method in section 6.3. Then we analyze some reference underwater scenarios (both in simulation and real experiments) in section 6.4.

6.2 High-level system design

Figure 6.1 summarizes the high-level design of the envisioned adaptive modem. The design has been motivated by the need of addressing very heterogeneous propagation

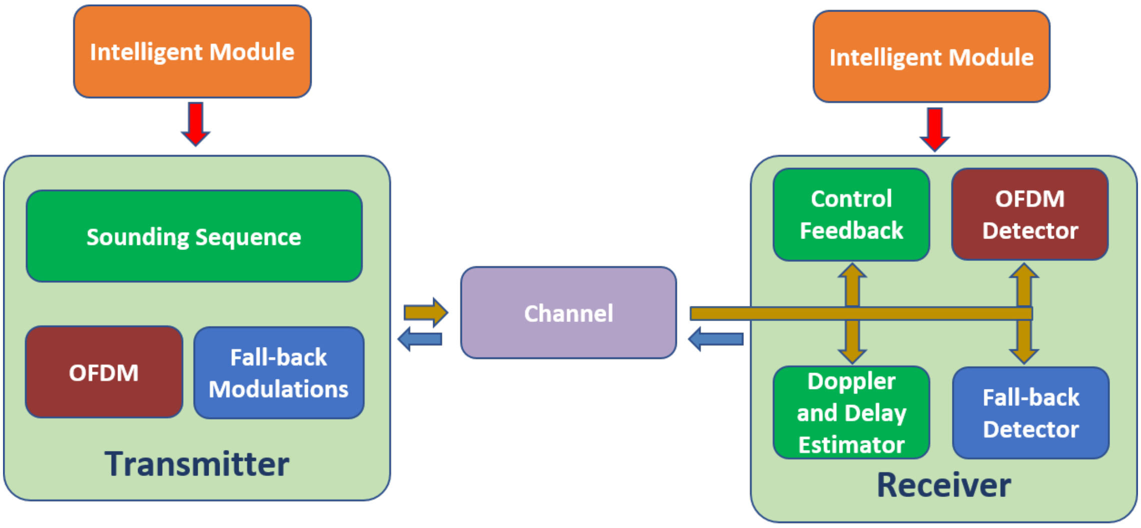


Figure 6.1: High-level block scheme of the proposed communication system.

scenarios: i) the case of *quasi-static or underspread channels*, for which modulations with high spectral efficiencies can be considered; ii) the case of *harsh or overspread channels*, for which only robust (low-rate) modulations can achieve a non-null transmission capacity. To cope with this heterogeneity, different forms of adaptations are supported. First, the modem is equipped with two different transceivers: an OFDM modulator, able to optimize the usage of the transmission bandwidth for benign channels, and a fall-back modulator, providing a robust back-up link in case of difficult channel conditions. Second, the OFDM modulator can change the number of subcarriers and the length of the cyclic prefix, in order to respond to the variability of multipath and Doppler effects, which may be encountered in different sites and applications.

Decisions on the selection of the best modulator, as well as on the tuning of the modulator parameters, are taken by an intelligent module after estimating the operating conditions of the system. These conditions are summarized in terms of channel spreading parameters, i.e. analysis of the variability of signal interference and distortions due to multi-path reflections and Doppler effects. The analysis is performed on the receiver side on the basis of special sounding sequences transmitted at the system start-up or any time the intelligent module needs an update (e.g. when the communication link is experiencing high error rates). The fall-back modulator is used as a feedback channel from the receiver to the transmitter for sending back the results of the analysis.

6.2.1 Intelligent module

Generally, in OFDM communications the maximum symbol duration must be related to the coherence time of the channel (inversely proportional to the Doppler spread), which indicates the period of time the channel can be considered as stationary. Increasing the number of subcarriers, improves the spectral efficiency and thus the overall bit rate of the transmission. However, if the number of subcarriers is too high, channel variations might cause errors in the demodulation, increasing the Bit-Error-Rate (BER). On the other hand, the duration of the cyclic prefix must be related to the delay spread of the channel, which measures the time duration of the multipath. The cyclic prefix must be large enough so that any interference phenomena due to multipath affects the next symbol only on the cyclic prefix, leaving the remaining part of the symbol unaffected.

Decisions of the intelligent module are based on the characterization of the channel behavior in terms of maximum observable Doppler shift and multipath delay. These parameters are represented by the Doppler spread and Delay spread parameters, whose estimation is detailed in the next section. Assuming that these parameters are available, the intelligent module computes the following bounds:

$$N_{CP} \geq \frac{\text{Delay spread}}{T_s} \tag{6.1}$$

$$N + N_{CP} \leq \frac{0.08}{\text{Doppler spread} \cdot T_s} \tag{6.2}$$

where N is the number of subcarriers, N_{CP} is the cyclic prefix length, T_s is the signaling period and the 0.08 term is derived by allowing (approximately) an 8% variation of the channel impulse response. Note that, when the Delay spread is too large compared to the channel coherence time, the above inequalities would lead to an unfeasible system with $N < 0$. In this case, OFDM cannot be used with a arbitrarily low BER since loss of orthogonality is unavoidable. Therefore, the intelligent module selects the fall-back modulator. Otherwise, OFDM is chosen and configured with the best possible transmission rate.

6.2.2 Fall-back modem

Different approaches have been envisioned for providing robustness in underwater channels, such as long symbol times, inter-symbol gaps and spread-spectrum techniques. Long symbol times or gap intervals guarantee the extinction of the symbol reflections before transmitting a new symbol, while spread-spectrum solutions, including frequency hopping, increase robustness to noise and inter-symbol interference. Among the different options for supporting the fall-back modem, in our implementation we chose the JANUS standard, in order to facilitate interoperability. In particular, we have integrated the WSense and Rome La Sapienza implementation of JANUS which complies with NATO STANAG standard.

6.3 Estimation of Channel Spreading Parameters

A key element of our system is represented by the estimation of the channel behavior. For channel behavior we do not mean the channel impulse response at a given time, but rather a simple characterization of the variability of this response as time varies.

6.3.1 Channel estimation for time-invariant channels

We start by deriving a channel estimation strategy for a linear time-invariant channel whose input-output relation may be written as

$$y_n = \sum_k h_k x_{n-k} + w_n \tag{6.3}$$

where x_n and y_n are respectively the input and output sequences, h_n is the channel impulse response and w_n is an equivalent noise including actual interference or noise and other measurement errors.

With the assumption that $h_k = 0$ for $k < 0 \vee k > L$, this equation may be rewritten in matrix form as follows:

$$y_n = \begin{bmatrix} x_n & x_{n-1} & x_{n-2} & \cdots & x_{n-L} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_L \end{bmatrix} + w_n. \tag{6.4}$$

A subsequence $(y_n, y_{n+1}, y_{n+2}, ..., y_{n+M})$ of channel outputs can be used to build the following linear system:

$$\begin{bmatrix} y_n \\ y_{n+1} \\ y_{n+2} \\ \vdots \\ y_{n+M} \end{bmatrix} = \begin{bmatrix} x_n & x_{n-1} & x_{n-2} & \cdots & x_{n-L} \\ x_{n+1} & x_n & x_{n-1} & \cdots & x_{n+1-L} \\ x_{n+2} & x_{n+1} & x_n & \cdots & x_{n+2-L} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n+M} & x_{n+M-1} & x_{n+M-2} & \cdots & x_{n+M-L} \end{bmatrix} \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_L \end{bmatrix} + \begin{bmatrix} w_n \\ w_{n+1} \\ w_{n+2} \\ \vdots \\ w_{n+M} \end{bmatrix} \tag{6.5}$$

or, in matrix form,

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{w}. \tag{6.6}$$

Provided the matrix $\mathbf{X}$ is known, this system can be solved for a Least Squares estimate of the channel impulse response $\mathbf{h}$ as follows:

$$\hat{\mathbf{h}} = \mathbf{X}^{+}\mathbf{y} \tag{6.7}$$

where $\mathbf{X}^{+}$ is the Moore-Penrose pseudoinverse of $\mathbf{X}$.

We assume that $\mathbf{X}$ is known since its components are given by samples of the transmitted sequence, called *sounding sequence*. When $M = L$, this matrix is square, and the sounding sequence may be chosen in such a way to optimize the channel estimation error and/or simplify the computation. Common choices for x_n are pseudo-noise or linear frequency modulated signals as reported in [37].

6.3.2 Sounding sequences for channel probing

We are interested in a channel estimation strategy which can be executed in real-time by a low-cost platform. This requires low-complexity methods for: (i) detection of the presence of the transmitted sounding signal while unsynchronized, (ii) estimation of the channel impulse response, (iii) estimation of the channel spreading features we are interested in.

For reasons that will be clear shortly, we choose to employ a sounding sequence signal made of periodic repetitions of chirp-like signals. The baseband signal we employ is the following:

$$s(t) = \sum_{l=0}^{N_c} \sum_{n=0}^{N_s-1} c_n p(t - (n + lN_s)T_s) \tag{6.8}$$

where N_s is the number of samples used and $N_c + 1$ is the number of repetitions of the sequence $c_n = \exp(-\jmath\pi(n - n^2/N_s))$. The signaling period T_s is the inverse of the channel bandwidth, while $p(t)$ is a root-raised-cosine signaling pulse which is zero-ISI Nyquist at rate $1/T_s$.

The received baseband signal will be given by (time-varying) convolution between $s(t)$ and the channel impulse response. The received sequence y_n is obtained at the receiver side by matched filtering of the received signal, and sampling at rate $1/T_s$, i.e. by projection of the received signal onto $p(t - nT_s)$:

$$y_n = \langle r(t), p(t - nT_s) \rangle \tag{6.9}$$

where $r(t) = \int s(\tau)h(t,\tau)d\tau \cong \int s(\tau)h(t-\tau)d\tau$, $h(t,\tau)$ is the time-varying channel impulse response (i.e. channel output for an input impulse at time τ) and $h(t)$ is the time-invariant estimate we can measure at a given time. Note that a Carrier Frequency Offset or phase drift in the received signal will appear as a time-varying common phase term in the estimated channel impulse response.

6.3.3 Low complexity channel estimation

If the channel coherence time is not too short, i.e. smaller than the probing signal duration, the receiver can detect its presence by simple correlation of two consecutive

N_s-samples subsequences. Specifically, following the Schwarz inequality, the receiver has to compute their inner product and compare it to the square root of the product of their energies.

The parameter N_s determines both the period of the probing signal and the channel-impulse response sampling rate. We obtain a channel estimate every N_sT_s received samples, and this means that the measurable Doppler range is $\pm 1/2N_sT_s$, while the maximum measurable delay spread is limited by N_sT_s (assuming that there is a direct path of maximum amplitude, otherwise it must be reduced by $1/2$). Therefore, N_s should be chosen the smallest possible without incurring in time-aliasing issues and can be adjusted iteratively by means of a feedback channel as devised e.g. in [35].

An estimate is obtained simply by computing (6.7) for a choice of $M = L = N_s - 1$. Choosing $M = L$ in (6.6) yields a square linear system which has the important property of being circulant by construction. A circulant matrix is diagonalized by the DFT matrix and the computation of the (pseudo-)inverse becomes trivial by means of the Fast Fourier Transform.

Specifically, denoted with $\mathbf{F}$ the square $N_s \times N_s$ matrix with elements

$$F_{nm} = \frac{1}{\sqrt{N_s}} e^{-\jmath 2\pi \frac{nm}{N_s}}, \tag{6.10}$$

the following equality holds for our choice of the probing signal c_n:

$$\mathbf{F}^H \mathbf{X} \mathbf{F} = \sqrt{M} \mathrm{diag}(\mathbf{c}^*) e^{-\jmath\pi(M-1)/4} \tag{6.11}$$

where the vector $\mathbf{c}$ is given by $\mathbf{c} = [c_0, c_1, \ldots, c_{M-1}]^T$.

Using this equality in the computation of (6.7) yields:

$$\hat{\mathbf{h}} = \left(\mathbf{X}^H \mathbf{X}\right)^{-1} \mathbf{X}^H \mathbf{y} = \frac{e^{\jmath\pi(M-1)/4}}{\sqrt{M}} \mathbf{F} \mathrm{diag}(\mathbf{c}) \mathbf{F}^H \mathbf{y}. \tag{6.12}$$

This expression can be further simplified by observing that $\mathbf{F}\mathbf{c} = \mathbf{c}^* e^{\jmath\pi(M-1)/4}$, which yields, after a few manipulations:

$$\hat{\mathbf{h}} = \frac{1}{\sqrt{M}} \mathrm{diag}(\mathbf{c}^*) \mathbf{F}^H (\mathrm{diag}(\mathbf{c}^*) \mathbf{y}). \tag{6.13}$$

This means that our choice of the sounding sequence yields an extremely low complexity channel estimation algorithm with can be performed in real-time with low-cost hardware.

The estimates we obtain every N_s samples are still affected by synchronization errors. The random time delay can be ignored since we are only interested in measuring the Delay Spread, while Carrier Frequency Offset(s) and the average Doppler caused by relative motion between transmitter and receiver can be compensated canceling the phase term of the direct path This phase term can be estimated by searching for phase of the estimated sample with maximum amplitude (which is the less affected by noise). Finally, clock drifts can be measured and compensated as described in [37].

6.3.4 Channel spreading parameter estimation

For evaluating the temporal variability of the channel impulse response, we repeat the estimation process for N_c repetitions of the sounding sequence. By analyzing the whole set of N_c channel estimates, the receiver can compute the channel spreading parameters we are looking for. The delay spread is estimated by finding the common smallest delay interval accounting for α_{DS} (tipically the 95%) of the channel impulse response energy.

In order to estimate the Doppler spread, we first compute the channel frequency response (by means of an FFT) for each estimate of the N_c channel impulse responses. Then we compute the autocorrelation of each frequency sample, along the channel-estimate-index dimension, i.e. we obtain N_s autocorrelations each of $2N_c - 1$ points. Last, we compute the DFT of the average of these autocorrelations and obtain the Doppler Spectrum estimate. We use the obtained Doppler Spectrum estimate to measure the Doppler spread as half of the interval accounting for α_{BD} (again, typically 95%) of the overall power spectrum, i.e. we measure only positive frequencies.

6.3.5 Computational complexity analysis

The proposed method requires the estimation of the channel impulse response (by means of equation 6.13), which is based on the well-known FFT algorithm. Since the estimation is done on N_s samples and is repeated N_c times, the channel estimation mechanism described in section 6.3.3 has computational complexity $O(N_c \cdot$

$N_s log(N_s))$. Once the channel impulse response is measured, estimation of the Delay spread is trivial and does not increase the order of the complexity. To compute the Doppler spread as described in section 6.3.4, computation of the N_c channel frequency response measurements requires as many FFTs on N_s points (same complexity as above), and the N_s autocorrelations of the frequency samples can also be computed by means of FFTs on $2N_c - 1$ points. However, since usually $N_s \gg N_c$, the overall computational complexity of the proposed method remains $O(N_c \cdot N_s log(N_s))$. Note that, compared to other strategies which aim at estimating channel distortions on every single packet, the proposed method might be used only seldomly on stable channels (or whenever the channel undergoes significant changes), potentially reducing the time, frequency and power resources required for this purpose. Indeed, most of the power consumption is absorbed by the transducer in TX/RX mode, which is two or three orders of magnitude higher than Idle mode [44].

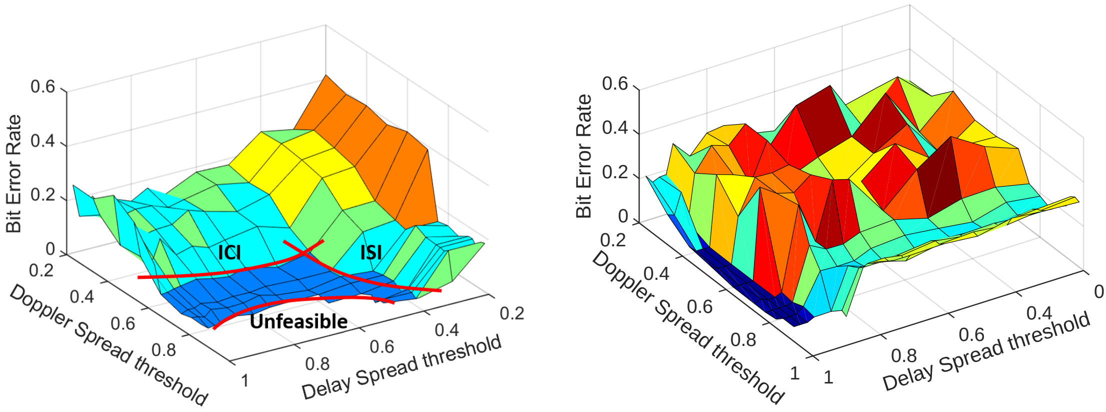
6.4 Analysis of exemplary underwater channels

In order to validate our technique for estimating the channel spreading parameters, we used two channel simulators: an in-house developed channel simulator which is similar to [45], and the well-known Watermark channel simulator. Finally, we employ our channel estimation technique in two sea environments and analysed the feasibility (or not) of the OFDM modulation in different sites.

6.4.1 Ad-hoc time-varying channels

We implemented a custom-made simulator for underwater channels, in which the time-varying channel responses can be specified in terms of number of reflected paths N_p, as well as per-path attenuation and temporal variability of the propagation delay. Although this channel specification can be somehow simplistic, it allows us to exactly model the channel responses over time and therefore refer to a ground truth for assessing the performance of channel spreading estimates.

More into details, the linear time-varying (LTV) model of the impulse response



(a) LTV1 channel

(b) NOF1 channel

Figure 6.2: Bit Error Rate performance of coded OFDM with decision feedback on the LTV1 and NOF1 channels, as a function of the delay and Doppler spread estimation thresholds.

obey to the following equation:

$$h(t,\tau) = \sum_{p=1}^{N_p} \rho_p \delta(t - \tau - d_p(t)/c) \tag{6.14}$$

where N_p is the number of paths included in the multi-path channel response, and we chose, for simplicity, constant amplitudes ρ_p and a constant sound speed c. The functions $d(t)$ can be tweaked at will in order to model relative movements between the transmitter, the receiver, and the scattering sources.

A first model we considered, denoted by LTV1, uses $d_p(t)/c = \tau_{0,p} + \tau_{1,p} cos(\omega_p t + \phi_p)$. With this choice, it is possible to model the communication channel between users which are stationary on average, and affected by a sinusoidal time-varying wave movement. An argument similar to the evaluation of the bandwidth of a Frequency Modulated signal leads to the following approximation for the Doppler spread of such a channel: $\max_p[\omega_p(f_0\tau_{1,p} + 1/2\pi)]$, where f_0 is the center frequency of the transmitted signal. Conversely, the delay spread can be computed directly in terms of the τ_0 and τ_1 parameters.

In order to evaluate the impact of the α_{DS} and α_{BD} estimation parameters introduced in the previous section - respectively the Delay Spread and Doppler Spread truncation errors - we ran simulations of a coded OFDM modem over the

Table 6.1: LTV1 channel model.

$N_p = 3$
$\rho^2 = [0.68, 0.24, 0.08]$
$\tau_0 = [1, 4, 15]$ (ms)
$\tau_1 = 1$ (ms)
$\omega = 0.06\pi$
$\phi = U(-\pi, \pi)$

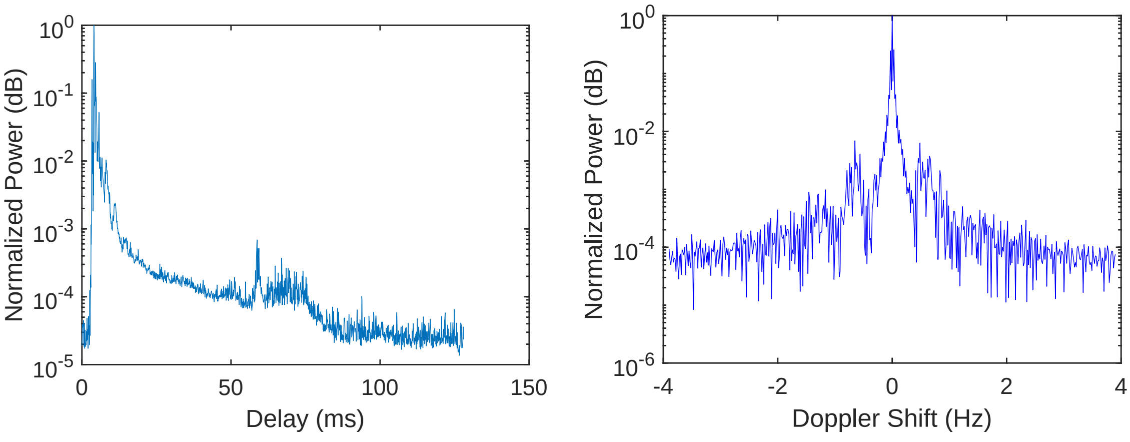
LTV1 channel model with three main paths as reported in Table 6.1. The OFDM modulation is calibrated by using the design criteria of our intelligent module, i.e. by configuring $N_{CP} = DelaySpread/T_s$ and $N = 0.08/(DopplerSpread \cdot Ts) - N_{CP}$.

Figure 6.2(a) shows the Bit Error Rate performance of coded OFDM on the LTV1 channel and a noise-free scenario as a function of different truncation errors on the evaluation of Delay and Doppler spread values. The noise-free assumption allows to assess the impact of inter-symbol interference (ISI) and inter-carrier interference (ICI). Inspection of the figure shows that, in order to limit the effects of these two interference terms, it is necessary to select a high enough threshold (i.e. a low enough truncation error). However, if the choice of these parameters is too high (or in overspread scenarios), the constraints on the OFDM lead to an unfeasible modulation design. The choice of the α_{DS} and α_{BD} parameters thus depends both on the application scenario and the tolerable interference levels, and will be a function of the modulation and coding set.

6.4.2 Watermark channels

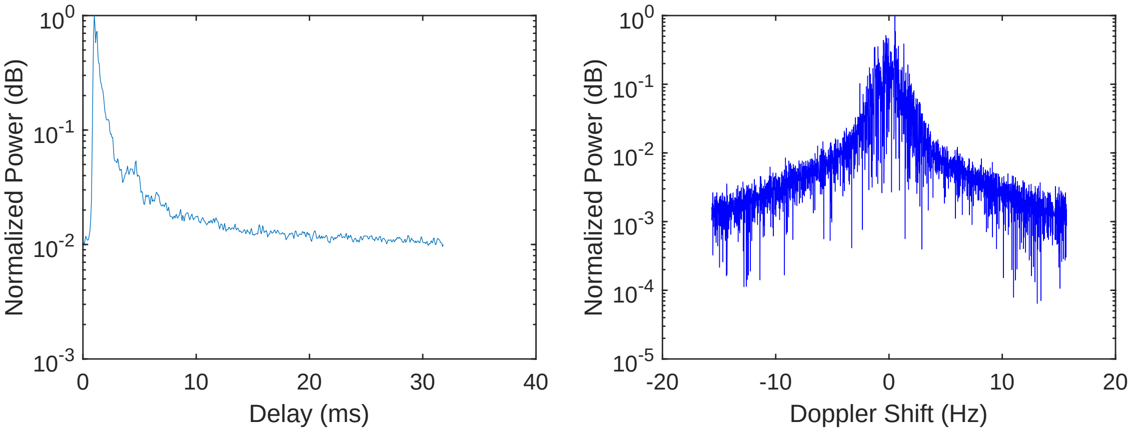
We applied our technique for estimating the channel spreading parameters to other underwater simulated channels, based on the recordings of real channel traces. In particular, we worked on the Watermark simulator, which provides several time-varying channel traces measured in different real sites.

We analysed the channel characteristics of Watermark's NOF1 and NCS1 traces, which both have 8 kHz bandwidth in the range 10-18 kHz and are measured between



(a) (b)

Figure 6.3: Impulse response and Doppler spectrum for a sample sounding of Watermark channel NOF.



(a) (b)

Figure 6.4: Impulse response and Doppler spectrum for a sample sounding of Watermark channel NCS.

two stationary nodes. In particular, the NOF1 dataset is constituted of 60 different soundings recorded in a shallow stretch of water, with a total play time of about 33 min. Each of the 60 soundings is long 32.9 s and was measured transmitting 258 consecutive chirps signals lasting 128 ms (2048 samples at 16 kHz sampling rate). For example, figure 6.3 reports one channel impulse response and the Doppler spectrum measured for one of these 60 soundings. Across the entire dataset, the estimated

Delay spread (considering 95% of the impulse response energy) ranged between 10 and 80 ms and the Doppler spread $\in [0.6, 2.2]$ Hz.

The NCS1 dataset is also constituted by 60 different soundings but measured on Norway's continental shelf using pseudonoise sequences. Each measurement is 32.6 seconds long, consisting of 1024 probing sequences 31.9 ms long (510 samples at 16 kHz sampling rate), yielding a total play time of almost 33 minutes. As reported in [37], NCS1 is more challenging than NOF1, because the true channel is characterized by long delay and Doppler tails, which are aliased in the measurement (although the fraction of energy that is aliased is not necessarily large). For example, figure 6.4 shows the channel impulse response and Doppler spectrum of a sample sounding. Across the entire dataset, the estimated Delay spread was around 31.6 ms (close to the measurement limit) and the Doppler spread was between 8.7 and 10.7 Hz. The NCS1 channel is thus an interesting representative test for harsh environments.

Figure 6.2(b) shows the Bit Error Rate performance of coded OFDM, designed according to our intelligent module, on the NOF1 channel. Again, inspection of the figure shows that, in order to obtain a good BER performance, it is necessary to choose a relatively high threshold α_{DS} for Delay Spread estimation (over 90%), while a relatively low threshold α_{BD} is sufficient (not lower than 40%). The figure for the NCS1 channel instead is not shown, as OFDM either had a high BER (when using low thresholds which led to high interference), or resulted unfeasible. As we will show, the fall-back modulation can be successfully used in such scenarios.

6.4.3 Sea channels experiments

Finally, we tested the proposed estimation technique in the Littoral Ocean Observatory Network (LOON) test bed [46], deployed in the gulf of La Spezia, in Italy. Implemented by the NATO STO Centre for Maritime Research and Experimentation (CMRE), this testbed is envisioned to foster cooperative development of underwater communications technologies and consists of tripods with acoustic communications equipment plus oceanographic and meteorological sensors all cabled to shore and remotely accessible. This allows researchers to conduct underwater acoustic experi-

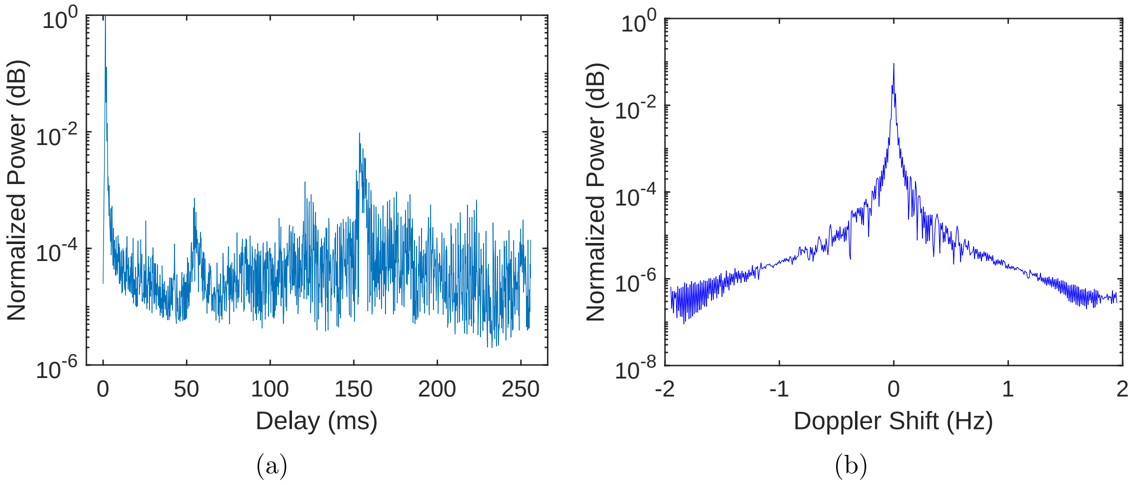


Figure 6.5: Impulse response and Doppler spectrum at sea using 258 chirps.

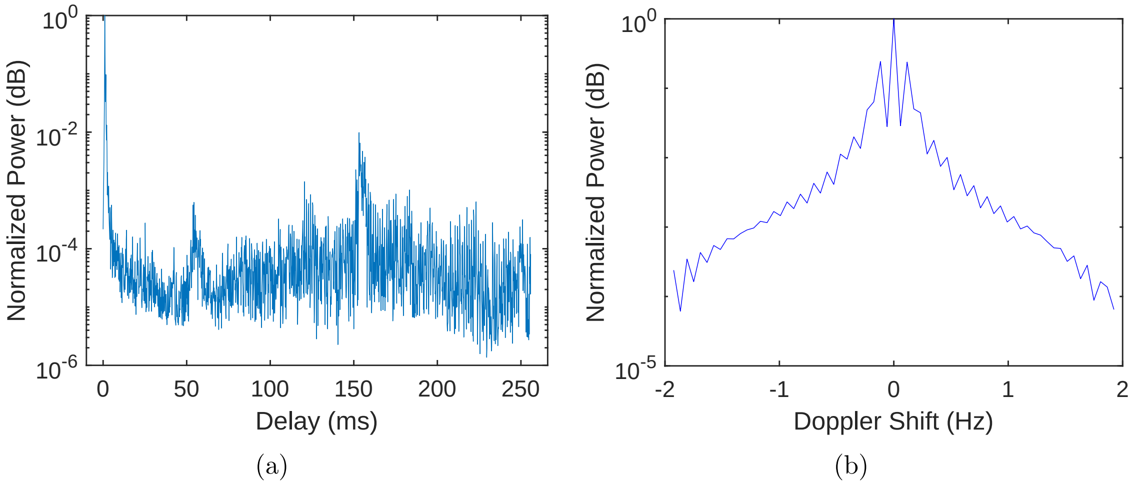


Figure 6.6: Impulse response and Doppler spectrum at sea using only 34 chirps.

mentation in a persistent way with no mobilization time[1]. The LOON is composed by 4 bottom-mounted tripods, with distances between nodes ranging from ∼200 to ∼700 m. The tripods are fitted with different acoustic capabilities based on the Woods Hole Micro-Modem, EvoLogics S2C 18/34 and arbitrary waveform transmis-

[1] This work acknowledges the use of data that was made possible thanks to the BLUE-INTEROP project that was selected at the 2nd TNA open call of the EUMarineRobots project, funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 731103. The BLUE-INTEROP project was performed by CMRE and WSENSE Ltd.

sion and recording capability.

Similar to Watermark probing signals, for the measurements on the LOON testbed we employ trains of 258 consecutive chirps, with 6 kHz bandwidth and central frequency of 11 kHz. We tested different chirp durations of 64, 128 and 256 ms and varied the root-raised-cosine filter length between 6 and 48 samples (with steps of 6 samples). For example, figure 6.5 shows the channel impulse response and Doppler spectrum measured on one of these measurements. Overall, the estimated Delay spread (considering 95% of the energy) varied widely between 0.8 and 225 ms, while the Doppler spread was always below 0.1 Hz.

While such measurements lead to accurate estimates of the channel parameters, they require up to over one minute to be accomplished. In practice, in less harsh environments (i.e. when the channel allows the use of OFDM modulation) it is often possible to reduce this measurement time, employing shorter chirp trains and accepting a more coarse Doppler estimation. For example, figure 6.6 shows the results obtained in the same scenario using only 34 chirps. In this scenario, while the corresponding Delay spread is not seriously affected, the Doppler spread increases to 0.4 Hz. This can be partially compensated by lowering the Doppler threshold or, for harsher scenarios, increasing again the number of estimates. Similar results were obtained in a touristic harbor at Santa Marinella, close to Rome, Italy.

Table 6.2 summarizes the (best) channel parameters obtained and the corresponding minimum cyclic prefix length N_{CP} and maximum number of subcarriers N according to the OFDM constraints described in section 6.2.1. From the table, it is clear that high performance OFDM can be obtained in the LOON environment while, on the other extreme, it is not possible to use OFDM on the NCS channel even in the best-case scenario.

6.5 Fall-back modulator performance

Finally, we validated the fall-back modulator performance in harsh channel conditions, when the OFDM modulator does not work or exhibits very poor performance.

Table 6.2: Design of OFDM parameters for tested channels.

Channel	Delay sp. (ms)	Doppler sp. (Hz)	N	N_{CP}
NOF1	13.5	0.6	128	90
NCS1	31.6	8.6	–	–
LOON	0.8	0.1	4096	10
Harbor	19.8	0.9	512	260

We used Watermark's NCS1 channel which was shown unfeasible for every possible combination of OFDM parameters.

The modulation of choice for harsh environments, is the binary FSK with frequency hopping used by the JANUS modulator. Indeed, the FSK modulation allows a very simple implementation of transmitter and receiver operations and the frequency hopping allows to skip the interference created by the signal reflections in the frequency domain, thus allowing a higher rate at the expense of spectral efficiency. Demodulation is performed by following the pre-defined hopping sequence, and differentiating the instantaneous received frequency by running the matched filter.

As described previously, Watermark NCS1 is a challenging scenario, with long Delay spread (higher than 30ms) and Doppler spread higher than 8Hz, which are subject to measurement aliasing (as reported in [37]). We also simulated the effect of an additive AWGN noise, with a tunable power, leading to a residual Signal-to-Noise Ratio (SNR) ranging from 10 dB to 20 dB.

Figure 6.7 shows the BER obtained by our JANUS implementation slightly modified to adapt to the NCS1 channel. From the figure, it is clear that the Fall-back modulation is free from errors when the residual SNR is above 15-20 dB. We also exploited our software-based implementation for removing the frequency hopping mechanism. In this case, the BER was close to 0.5 for all the simulated packets and SNR values. Indeed, the JANUS symbol lasts 6.26 ms, which is much lower than the Delay Spread of the NCS channel (31.6 ms). The frequency hopping strategy is thus essential on this channel, despite the other Doppler correction and Trellis code mechanisms implemented in JANUS for robust demodulation.

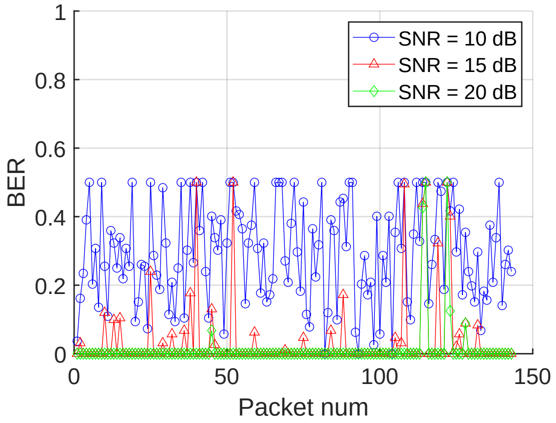


Figure 6.7: JANUS results under the Watermark NCS1 channel in different SNR conditions.

Chapter 7

Conclusions

In this work we designed and implemented a channel-aware adaptive system able to switch its configuration as a function of the underwater channel conditions. The system supports both low rate and high rate communications.

Our work started studying how to correct Doppler effects in JANUS standard transmissions, a low rate solution characterized by robustness and reliability in harsh underwater environments. In particular, we exploit the JANUS preamble to compute the CAF and compensate Doppler distortions up to $\pm 5m/s$ without requiring any modification to the standard. Validated through realistic simulations and in field experiments, the proposed method is able to compensate and correctly receive 90% of the packets in Watermark simulated channels and 75% in real at-sea channel even with severe Doppler, demonstrating the effectiveness of our technique.

Then, we designed and developed FLUMO, a flexible SDA modem. FLUMO is composed of three different parts, i) a digital platform; ii) an analog front-end board; and iii) a piezoelectric transducer. FLUMO presents two innovative characteristics. First the SDA modulation/demodulation is completely decoupled from the rest of the system. This separation is very helpful for the validation and testing of both of the SDA and HW of the modem, e.g. with externally modulated signals contained in a wav file. Second, FLUMO provides a flexible and tunable system to better fit the underwater channel conditions, in order to reach optimal performances in each scenario. In order to validate the implemented modem, we performed several experiment campaigns in two controlled environments (indoor and outdoor), as well as co-simulation tests to validate the modem in realistic underwater conditions, using Watermark. We also compared the performance of the proposed modem with a JANUS implementation. Although our proposed modulation scheme achieves good performance in benign channels, in harsh environments the frequency hopping strategy used by JANUS is needed to deal with multipath. In these environments, thanks to the flexibility of FLUMO, a custom implementation of an FH-FSK emulating JANUS achieves similar performances.

As a next step, we implemented, through the liquid-dsp library, a complete OFDM transceiver and we integrated it into our SDA modem as a high rate solution. In the implemented transceiver we can choice the parameters of the OFDM modulation like number of subcarriers, cyclic prefix length or, for example, the number of pilot subcarriers, as well as the modulation scheme of subcarriers, FEC scheme, and so on. A key feature of the implementation, pertain the receiver that use a feedback chain to update the channel estimate with the help of the convolutional coding, at every received symbol. In order to validate our implementation we used the Watermark simulator as well as real experiments indoor and outdoor. In the experiments the improved OFDM receiver achieved better performance compared to traditional receiver, reaching a spectral efficiency of 1.4 bit/s/Hz (QPSK modulation on the subcarriers), with low BER even at sea.

Finally, we devised a simple strategy to exploit high rate OFDM communications (when the channel allows it) and dynamically adjust the parameters to cope with the expected Doppler shift and multipath delay. Contrary to previous approaches, which tailor particular environments or employ complex Doppler correction techniques, we proposed a low-complexity estimation technique for time-varying channels and an intelligent module for tuning the OFDM parameters as a function of the channel spreading estimates. The proposed approach has been validated in different simulation platforms for underwater communications and in real experiments at sea, representing a variety of harsh and mild scenarios. Results show that OFDM achieves good performance in the latter conditions, while a fall-back modulation like JANUS can be used in the former.

As a future work, our system can be extended with different modulations (S2C, hybrid-carrier), thus increasing the possibility to adapt it to the various underwater scenarios.

List of Publications

[1] G. E. Galioto, D. Garlisi, D. Croce, L. Mistretta, R. Badalamenti, I. Tinnirello, C. G. Giaconia, C. Petrioli, P. Gjanci, "FLUMO: FLexible Underwater MOdem," OCEANS 2019, Marseille, France, 2019, pp. 1-7.

[2] C. Baldone, G. E. Galioto, D. Croce, I. Tinnirello, C. Petrioli, "Doppler Estimation and Correction in Underwater Industrial Internet of Things", IMEKO TC-4 International Symposium 2020 - Special session on Wireless technologies, signal processing algorithms and measurement techniques for the Industrial Internet of Things, Palermo, Italy, 2020.

[3] C. Baldone, G. E. Galioto, D. Croce, I. Tinnirello, C. Petrioli, "Doppler Estimation and Correction for JANUS Underwater Communications," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 2020.

[4] S. Mangione, G. E. Galioto, D. Croce, I. Tinnirello and C. Petrioli, "A Channel-Aware Adaptive Modem for Underwater Acoustic Communications," in IEEE Access, vol. 9, pp. 76340-76353, 2021, doi: 10.1109/ACCESS.2021.3082766.

[5] G. Galioto, I. Tinnirello, D. Croce, F. Inderst, F. Pascucci and L. Giarré, "Sensor Fusion Localization and Navigation for Visually Impaired People," 2018 European Control Conference (ECC), Limassol, 2018, pp. 3191-3196, doi: 10.23919/ECC.2018.8550373.

[6] D. Croce, L. Giarré, F. Pascucci, I. Tinnirello, G. E. Galioto, D. Garlisi, A. Lo Valvo, "An Indoor and Outdoor Navigation System for Visually Impaired People," in IEEE Access, vol. 7, pp. 170406-170418, 2019, doi: 10.1109/ACCESS.2019.2955046.

Bibliography

[1] J. Heidemann, M. Stojanovic, and M. Zorzi, “Underwater sensor networks: Applications, advances and challenges,” in Philosophical Transactions of the Royal Society A, vol. 370, August 2012, pp. 158-175

[2] C. Cardia, P. Gjanci, C. Petrioli, G. Saturni, D. Spaccini, and D. Tomaselli, “The internet of underwater things: From nemo to underwater whatsapp,” in Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing, ser. Mobihoc 2019. New York, NY, USA: ACM, 2019, pp. 409-410.

[3] M. Stojanovic and J. Preisig, “Underwater acoustic communication channels: Propagation models and statistical characterization,” in IEEE Communications Magazine, vol. 47, no. 1, pp. 84-89, January 2009. doi: 10.1109/M-COM.2009.4752682

[4] J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen and K. McCoy, “The JANUS underwater communications standard,” 2014 Underwater Communications and Networking (UComms), Sestri Levante, 2014, pp. 1-4.

[5] Konstantin G. Kebkal and Rudolf Bannasch. “Sweep-spread carrier for underwater communication over acoustic channels with strong multipath propagation.” The Journal of the Acoustical Society of America 112.5 (2002): 2043-2052.

[6] B. S. Sharif, J. Neasham, O. R. Hinton and A. E. Adams, “A computationally efficient Doppler compensation system for underwater acoustic communi-

cations," in IEEE Journal of Oceanic Engineering, vol. 25, no. 1, pp. 52-61, Jan. 2000.

[7] M. Johnson, L. Freitag and M. Stojanovic, "Improved Doppler tracking and correction for underwater acoustic communications," 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, Munich, 1997, pp. 575-578 vol.1.

[8] I. Nissen, G. Zappa and J. Potter, "Doppler compensation for JANUS applied to data collected in the Baltic sea" in Proc. 4th International Conference and Exhibition on Underwater Acoustic Measurements: Technologies and Results pp. 20-24 Jun. 2011.

[9] Stojanovic, M., & Freitag, L. (2013). Recent trends in underwater acoustic communications. Marine Technology Society Journal, 47(5), 45-50. https://doi.org/10.4031/MTSJ.47.5.9.

[10] M. Stojanovic, "Low complexity OFDM detector for underwater channels," in Proc. MTS/IEEE OCEANS Conf., Boston, MA, Sep. 18-21, 2006, CD-ROM.

[11] A. Radosevic, R. Ahmed, T. M. Duman, J. G. Proakis and M. Stojanovic, "Adaptive OFDM Modulation for Underwater Acoustic Communications: Design Considerations and Experimental Results," in IEEE Journal of Oceanic Engineering, vol. 39, no. 2, pp. 357-370,

[12] P. J. Gendron, "Orthogonal frequency division multiplexing with on-off-keying: Noncoherent performance bounds, receiver design and experimental results," U.S. Navy J. Underwater Acoust., vol. 56, no. 2, pp. 267-300, Apr. 2006. April 2014, doi: 10.1109/JOE.2013.2253212.

[13] Wan, L., Zhou, H., Xu, X., Huang, Y., Zhou, S., Shi, Z., et al. (2015). Adaptive modulation and coding for underwater acoustic OFDM. IEEE Journal of Oceanic Engineering, 40(2), 327-336.

[14] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Non-uniform Doppler compensation for zero-padded OFDM over fast-varying underwater acoustic channels," in Proc. IEEE OCEANS Eur. Conf., 2007, pp. 1-6.

[15] F. Mason, C. R. Berger, S. Zhou, and P. Willett, “Detection, synchronization, and Doppler scale estimation with multicarrier waveforms in underwater acoustic communication,” IEEE J. Sel. Areas Commun., vol. 26, no. 9, pp. 1638-1649, Dec. 2008.

[16] B. Li, S. Zhou, M. Stojanovic, L. Freitag and P. Willett, “Multicarrier Communication Over Underwater Acoustic Channels With Nonuniform Doppler Shifts,” in IEEE Journal of Oceanic Engineering, vol. 33, no. 2, pp. 198-209, April 2008, doi: 10.1109/JOE.2008.920471.

[17] N. Parrish, S. Roy, and P. Arabshahi, “Symbol by symbol Doppler rate estimation for highly mobile underwater OFDM,” in Proc. 4th ACM Int. Workshop Underwater Netw., Berkeley, CA, USA, Nov. 2009, pp. 1-8.

[18] T. Arikan, T. Riedl, A. Singer, and J. Younce, “Comparison of OFDM and single-carrier schemes for Doppler tolerant acoustic communications,” in Proc. IEEE OCEANS Conf., Genova, Italy, 2015, pp. 1-7.

[19] Y. V. Zakharov and A. K. Morozov, “OFDM Transmission Without Guard Interval in Fast-Varying Underwater Acoustic Channels,” in IEEE Journal of Oceanic Engineering, vol. 40, no. 1, pp. 144-158, Jan. 2015.

[20] J. Li, Y. V. Zakharov and B. Henson, “Multibranch Autocorrelation Method for Doppler Estimation in Underwater Acoustic Channels,” in IEEE Journal of Oceanic Engineering, vol. 43, no. 4, pp. 1099-1113, Oct. 2018, doi: 10.1109/JOE.2017.2761478.

[21] R. Jurdak, P. Aguiar, P. Baldi, and C. V. Lopes, “Software modems for underwater sensor networks,” in Proc. IEEE/OES Oceans, Aberdeen, Scotland, Jun. 2007.

[22] I. Vasilescu, C. Detweiler, and D. Rus, “AquaNodes: An underwater sensor network,” in Proc. ACM WUWNet, MontrÃ©al, Canada, Sep. 2007.

[23] B. Benson, Y. Li, B. Faunce, K. Domond, D. Kimball, C. Schurgers, and R. Kastner, “Design of a low-cost underwater acoustic modem,” IEEE Embedded Syst. Lett., vol. 2, no. 3, pp. 58-61, Sep. 2010.

[24] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, “The WHOI Micro-Modem: An acoustic communications and navigation system for multiple platforms,” in Proc. MTS/IEEE Oceans, Washington DC, USA, 2005.

[25] N. Ahmed, W. bin Abbas, and A. A. Syed, “A low-cost and flexible underwater platform to promote experiments in UWSN research,” in Proc. ACM WUWNet, Los Angeles, CA, Nov. 2012.

[26] G. Cario, A. Casavola, M. Lupia, and C. Rosace, “SeaModem: a low-cost underwater acoustic modem for shallow water communication,” in Proc. MTS/IEEE Oceans, Genova, Italy, May 2015.

[27] R. Petroccia, G. Cario, M. Lupia, V. Djapic, and C. Petrioli, “First in-field experiments with a “bilingual” underwater acoustic modem supporting the JANUS standard,” in Proc. MTS/IEEE Oceans, Genova, Italy, May 2015.

[28] (2016). Evologics Underwater Acoustic Modems. [Online]. Available: https://www.evologics.de

[29] M. Chitre, I. Topor, and T.-B. Koay, “The UNET-2 modem – an extensible tool for underwater networking research,” in Proceedings of OCEANS 2012 MTS/IEEE Yeosu, (Yeosu, South Korea), May 2012.

[30] EvoLogics SDM mode. http://github.com/EvoLogics/sdmsh

[31] UNetStack, http://www.unetstack.net

[32] H. S. Dol, P. Casari, T. van der Zwan and R. Otnes, “Software-Defined Underwater Acoustic Modems: Historical Review and the NILUS Approach,” in IEEE Journal of Oceanic Engineering, vol. 42, no. 3, pp. 722-737, July 2017, doi: 10.1109/JOE.2016.2598412.

[33] B. Li et al., “MIMO-OFDM for high-rate underwater acoustic communications,” IEEE J. Ocean. Eng., vol. 34, no. 4, pp. 634-644, Oct. 2009.

[34] S. Shahabudeen, M. Chitre, M. Motani, and A. Siah, “Unified simulation and implementation software framework for underwater MAC protocol development,” in Proc.MTS/IEEE OCEANS, Biloxi,MS, USA, Oct. 2009

[35] E. Demirors, G. Sklivanitis, G. E. Santagati, T. Melodia and S. N. Batalama, "A High-Rate Software-Defined Underwater Acoustic Modem With Real-Time Adaptation Capabilities," in IEEE Access, vol. 6, pp. 18602-18615, 2018, doi: 10.1109/ACCESS.2018.2815026.

[36] E. Demirors et al., "The SEANet Project: Toward a Programmable Internet of Underwater Things," 2018 Fourth Underwater Communications and Networking Conference (UComms), Lerici, Italy, 2018, pp. 1-5, doi: 10.1109/UComms.2018.8493207.

[37] Paul A. van Walree, Francois-Xavier Socheleau, Roald Otnes, Trond Jenserud. The Watermark benchmark for underwater acoustic modulation schemes. IEEE Journal of Oceanic Engineering 2017, Vol. 42.(4) pp. 1007-1018.

[38] P. Qarabaqi and M. Stojanovic, "Statistical Characterization and Computationally Efficient Modeling of a Class of Underwater Acoustic Communication Channels," in IEEE Journal of Oceanic Engineering, vol. 38, no. 4, pp. 701-717, Oct. 2013.

[39] R. Jacob, T. Thomas and A. Unnikrishnan, "Fast computation of wide-band ambiguity function and matched filtering in active sonars," International Symposium on Ocean Electronics, Kochi, India, 2011.

[40] http://www.liquidsdr.org

[41] http://www.redpitaya.com

[42] Obradovic, Vuk, Okiljevic, Predrag, Kozic, Nadica, Ivkovic, Dejan. (2016). Practical implementation of digital down conversion for wideband direction finder on FPGA. Scientific Technical Review. 66. 40-46.

[43] Hovannes Kulhandjian and Tommaso Melodia. 2014. "Modeling Underwater Acoustic Channels in Short-range Shallow Water Environments." In Proceedings of the International Conference on Underwater Networks & Systems (WUWNET '14). Association for Computing Machinery, New York, NY, USA, Article 26, 1-5. DOI:https://doi.org/10.1145/2671490.2674560

[44] S. Sendra, J. Lloret, J. M. Jimenez and L. Parra, "Underwater Acoustic Modems," in IEEE Sensors Journal, vol. 16, no. 11, pp. 4063-4071, June1, 2016, doi: 10.1109/JSEN.2015.2434890.

[45] B. Henson, J. Li, Y. V. Zakharov and C. Liu, "Waymark baseband underwater acoustic propagation model," 2014 Underwater Communications and Networking (UComms), Sestri Levante, 2014, pp. 1-5, doi: 10.1109/UComms.2014.7017132.

[46] J. Alves, J. Potter, P. Guerrini, G. Zappa and K. LePage, "The LOON in 2014: Test bed description," 2014 Underwater Communications and Networking (UComms), Sestri Levante, 2014, pp. 1-4, doi: 10.1109/UComms.2014.7017141.