Towards Underwater Video Transmission

Master of Science Thesis

Author: Laura Dubreuil Vall
Advisor: Milica Stojanovic

Telecommunications Engineering, 2011

Per Aspera Ad Astra
(Through adversity to the stars. Latin proverb)

Contents

List of Figures

List of Tables

Collaborations

This project was started in 2009 at the Sea Grant College Program from the Massachusetts Institute of Tehcnology –MIT– (Cambridge, Massachusetts, USA) by the graduate student Jordi Ribas Oliva (ribas@mit.edu), under the supervision of Dr. Milica Stojanovic (millitsa@mit.edu).

From October 2010 to May 2011, the project was continued by the author of the current thesis, Laura Dubreuil Vall (lauradv@mit.edu). The research involved in this thesis was conducted at the Communications and Digital Signal Processing (CDSP) Center, based at the Department of Electrical and Computer Engineering at Northeastern University (Boston, Massachusetts, USA). The supervisor of the thesis was Dr. Milica Stojanovic, who works closely with the Sea Grant College Program at MIT.

Starting on June 2011, this project will be continued at the Sea Grant College Program at MIT by Laura Dubreuil Vall, under the supervision of Dr. Chryssostomos Chryssostomidis (chrys@mit.edu).

Acknowledgments

I would like to express my gratitude to my supervisor, Dr. Milica Stojanovic, for her valuable leadership, support and patience. Her enthusiasm and inspiration provided a constructive guidance, with countless e-mails and and worthy discussions. I would also like to thank the other member of the committee, Dr. Chryssostomos Chryssostomidis, for his interest in the thesis and for offering me the opportunity to continue this project at the Sea Grant College Program from the Massachusetts Institute of Technology (MIT).

I would not have been able to complete this journey without the aid and support of countless people in Northeastern University: Parastoo Qarabaqi, Fatemeh Fazel and Yashar M. Aval are only a few of them. Our group meetings have enriched my knowledge as well as provided me with different ways to approach scientific problems. Special thanks go to Rameez Ahmed, who kindly lent me his in-air testbed and helped me with the experimental part of the project. Maytee Zambrano also deserves a special acknowledgment for her valuable friendship along these months. I expand my thanks to all the staff in the Electrical and Computer Engineering department for their assistance and kindness, specially to Joan Pratt.

I am indebted with Daniel Sura for the time he spent on reading this thesis. I would like to thank him for his valuable comments, insightful discussions and, specially, his helpful advices on my experience at MIT. I must also acknowledge Jordi Ribas Oliva, the former student who started this project and provided me with interesting suggestions and tips on how to improve it.

My gratitude is also due to Demetris Galatopoullos for reading this document and making valuable suggestions. He has always been a source of enthusiasm, good advice and collaboration. Jorgina Busquets Corominas also deserves special thanks for her help in editing and proof-reading this thesis. Her friendship and insightful discussions have been a life motivation since high school. I am tempted to individually thank all of my friends but, as the list might be long and for fear to omit someone, I will simply and genuinely say: thank you all for your support and trust.

My most sincere thanks, Jordi, for being there through thick and thin. Your unfailing help, care and loyalty have been an essential source of energy for me, as well as our trips of a lifetime through America.

Last but not least, I would like to deeply thank my family, specially my parents, who supported and encouraged me from the beginning of this experience. None of this would have been possible without their help through endless Skypes and visits. Moltes gràcies Ladis i Joan. This thesis is for you.

Laura Dubreuil Vall
April 2011

Resum

Aquest projecte consisteix en la implementació d'un sistema acústic OFDM (*Orthogonal Frequency-Division Multiplexing*; Multiplexació per Divisió de Freqüències Ortogonals) per a transmissions de vídeo submarines. El sistema està dissenyat per a enllaços de curta distància, amb retards breus i un ample de banda relativament gran. La recerca realitzada en aquest projecte es divideix en quatre parts diferenciades:

1. Recerca en tècniques de compressió de vídeo.
2. Recerca en tècniques de modulació i detecció.
3. Disseny de tècniques de processament de senyal destinades a compensar l'efecte Doppler.
4. Recerca en tècniques per reduir la codificació de canal.

Pel que fa al primer punt (1), cal destacar que és important trobar un mètode de compressió de vídeo adequat que redueixi substancialment la quantitat d'informació a transmetre, per tal d'aprofitar al màxim la capacitat de transmissió de dades del canal submarí. La tècnica seleccionada ha estat el codificador VLBV (*Very Low Bitrate Video*) de l'estàndard MPEG-4, dissenyat per treballar amb seqüències d'imatges de vídeo rectangulars amb una alta eficiència de codificació, robustesa enfront els errors de canal, baixa latència i baixa complexitat per a aplicacions a temps real.

Pel que es refereix al segon punt (2), la tècnica de modulació OFDM ha estat seleccionada per les seves excel · lents propietats per sistemes de comunicació d'alta velocitat en canals acústics submarins, caracteritzats per ser selectius en freqüència. No obstant, els problemes de sincronització i desplaçaments de freqüència causats pel moviment entre transmissor i receptor –l'anomenat efecte Doppler– ha creat la necessitat de dissenyar un algorisme eficient (3) que faci front a aquesta distorsió.

Pel que fa a l'últim punt (4), l'estàndard MPEG-4 ofereix un conjunt d'eines de protecció enfront els errors que ha estat incorporat al codificador de vídeo del sistema. Aquestes eines faciliten una comunicació segura en canals *wireless* sorollosos, oferint una protecció addicional contra les ràfegues d'error i mantenint una bona qualitat visual per al vídeo descodificat. D'aquesta manera, es pot reduir la codificació de canal aplicada a la informació transmesa, incrementant així la velocitat de transferència de dades del sistema.

Resumen

Este proyecto consiste en la implementación de un sistema acústico OFDM (*Orthogonal Frequency-Division Multiplexing*; Multiplexación por División de Frecuencias Ortogonales) para transmisiones de video submarinas. El sistema esta diseñado para enlaces de corta distancia, con retardos breves y anchos de banda relativamente grandes. La investigación realizada en este proyecto se divide en cuatro partes diferenciadas:

1. Investigación en técnicas de compresión de video.
2. Investigación en técnicas de modulación y detección.
3. Diseño de técnicas de procesado se señal destinadas a compensar el efecto Doppler.
4. Investigación en técnicas para reducir la codificación de canal.

Por lo que al punto (1) se refiere, cabe destacar que es importante encontrar un método de compresión adecuado que reduzca sustancialmente la cantidad de información a transmitir, con el objetivo de aprovechar al máximo la capacidad de transmisión de datos del canal submarino. La técnica seleccionada es el codificador VLBV (*Very Low Bitrate Video*) del estándar MPEG-4, diseñado para trabajar con secuencia de imágenes de video rectangulares con un alta eficiencia de codificación, robustez enfrente de errores de canal, baja latencia y baja complejidad para aplicaciones en tiempo real.

En cuanto al segundo punto (2), la técnica de modulación OFDM ha sido seleccionada por sus excelentes habilidades para sistemas de comunicación de alta velocidad en canales acústicos submarinos, caracterizados por ser selectivos en frecuencia. Sin embargo, los problemas de sincronización y desplazamientos de frecuencia causados por el movimiento entre transmisor y receptor –el conocido efecto Doppler– ha creado la necesidad de diseñar un algoritmo eficiente (3) que aborde esta distorsión.

En cuanto al último punto (4) se refiere, el estándar MPEG-4 ofrece un conjunto de herramientas de protección contra errores que ha sido incorporado al codificador de video del sistema. Estas herramientas facilitan una comunicación segura en canales *wireless* ruidosos, ofreciendo una protección adicional contra las ráfagas de errores y manteniendo una buena calidad visual para el video decodificado. De esta forma, se puede reducir la codificación de canal aplicada a la información transmitida, incrementando así la velocidad de transferencia de datos del sistema.

Abstract

This project addresses the implementation of an acoustic OFDM (Orthogonal Frequency-Division Multiplexing) system for underwater video transmissions aimed to work over short distance links, with short delays and relatively high available bandwidth. This work includes research on four main approaches:

1. Research on video compression techniques.
2. Research on modulation and detection techniques.
3. Design of signal processing techniques aimed to compensate for the Doppler effect caused by motion.
4. Research on techniques to reduce the channel coding applied to the video data.

Regarding (1), it is important to find a suitable video compression technique that substantially reduces the amount of video data to be transmitted in order to properly utilize the limited bit rate capacity of the underwater channel. The chosen compression technique has been the VLBV (Very Low Bitrate Video) coder offered by the MPEG-4 standard due to its efficient low bit rate compression capabilities, aimed to work with conventional rectangular image sequences with high coding efficiency, high error robustness, low latency and low complexity for real-time applications.

Concerning (2), the OFDM modulation technique has been considered for its excellent capabilities applicable to high speed communication systems in wireless acoustic underwater channels, characterized by frequency selectivity. However, the motion-induced Doppler distortion makes it important to design an efficient algorithm (3) that deals with the synchronization problems and frequency shifts caused by the accentuated Doppler effect in underwater channels.

With regard to (4), a set of error resilience tools offered by the MPEG-4 standard has been implemented at the video encoder. These tools enable robust video communication over noisy wireless channels by offering additional protection over error bursts and maintaining the visual quality of the decoded video. This allows the conventional channel coding applied to the video data to be reduced, thus increasing the available data rate of the system.

Chapter 1

Introduction

1.1 Motivation

Underwater video recording has historically been used for many purposes. Many examples are to be found in the field of marine biology, such as marine biological studies [1, 2], sediment studies [3], tidal microtopography [4], and even sports, entertainment, education and more.

For many years, the only available methods to obtain underwater visual observations were either diving in deep water or using waterscopes. Its obvious limitations regarding depth, observation time, temperature and accessibility created a need for a more efficient system. In the mid 1950s, underwater video recording using submersible Remotely Operated Vehicles (ROVs) was first introduced by the British Royal Navy and provided a considerable contribution in the field of underwater imaging. However, the ROV's critical dependency on a tether link with the vessel in the surface is also a significant technological obstacle limiting the vehicle's maneuverability. Thus, real-time underwater wireless video transmission has recently become a demanding communication application for subsea operations in a wide range of fields:

- Underwater defense applications, such as port security and hull inspections, in which wireless video transmission capabilities could be employed in small Unmanned Undersea Vehicle (UUV) systems.
- Real-time environmental monitoring of marine life and reef.
- Enabling Autonomous Underwater Vehicles (AUVs) with video transmission for supervisory control, surveillance, and intervention procedures in offshore oilfield environments.
- Deployment of AUV platforms in situations when ROVs are expensive to run or limited in availability such as in post hurricane or storm subsea equipment inspections.
- Video transmission to and from moving and stationary structures, ships, and submarines for defense operations such as battle space preparations and security.
- Open ocean exploration and sonar surveys where high resolution sidescan sonar imagery from an AUV could be compressed and sent as a video for real-time review.

The implementation of an underwater high-speed wireless communication link is currently an area of active research. Recently, experiments using electromagnetic waves in the blue-green range of the visible spectrum showed promising results [5]. However, most of these systems are unsuitable for connecting mobile platforms due to the need for precise optical alignment between transmitter and receiver and the fact that optical waves are considerably affected by scattering, allowing only extreme-intensity optical waves to propagate in the water. Distance is also a major issue that limits applicability to short-range (<100 m) high-rate data transmissions.

In contrast, sound waves travel exceptionally well through the water due to its high density: unlike in the air, the molecules are closer to each other and the bounds between them are tighter, so it takes less time for them to pass the sound to each other. Low frequency sounds (~20 Hz) may even be detected across an entire ocean basin [6], while ultrasound waves are usable at ranges of hundreds to thousands of meters. Industry practice has settled on acoustic transmission as the key to effective real-time communication with sub-sea assets.

Reliance on this technology has recently created a need for the design of better wireless underwater video transmission systems in terms of visual quality and transmission rate. This project deals with both issues, proposing a system that implements them and demonstrating its feasibility.

1.2 Historical background

In the 1850s, William Thompson made the first attempt in the field of underwater photography with a pole mounted camera. Several successful attempts were made over the next decades, but it was not until 1890 that the first scientific results from an underwater camera were published by the French naturalist Louis Boutan [7], who devised the underwater flash and other related equipment. Underwater video is a descendant to underwater photography, and has existed since the 1940's. The first scientific results were reported by Harvey Barnes in Nature [8].

The combination of underwater video recording and wireless transmission started in the 1990s, when a group of Japanese scientists succeeded in transmitting a video of a slowly-moving object at a depth of 6,500 m [9]. Actually, the video was a sequence of independently recorded images played at a frame rate of a few frames every 10 seconds. In 1991, another long-range acoustic system was developed and successfully tested in 2,000 m deep water [10]. In 2000, the Portguese project called ASMIOV [11] accomplished the transmission of a sequence of still images at 2 frames/second. In 2002, a Japanese group designed an experimental system [12] that employed the MPEG-4 standard (Section 2.2) for video compression, and supported a frame rate of 10 frames/second.

The last decade has witnessed a proliferation of different wireless video transmission systems. For instance, The Deep Submergence Laboratory at the Woods Hole Oceanographic Institution (WHOI) has been performing research on a different approach to convey a moving picture, called moisaicing. It consists of the reconstruction of the video by piecing together the images transmitted from a moving camera using additional transmitted information about time and position. Fortkey Ltd., a company based in United Kingdom, also made use of mosaicing in an underwater pipe-monitoring system.

Recently, companies such as LinkQuest Inc. (San Diego, CA, U.S.), Sea-Eye Underwater Ltd. (Ashkelon, Israel) and Applied Ocean Systems (San Diego, CA, U.S.), have been pushing the limits of the existing technologies by continuously innovating and developing commercial acoustic modems suitable for underwater video transmissions.

1.3 Goals

This project builds upon the underwater video transmission research from the Master's Thesis "Underwater Wireless Video Transmission using Acoustic OFDM" [13]. In that thesis, the basic structure of the system was designed and several modulation parameters were tested in the underwater experiments conducted at MIT Sailing Pavilion. The current thesis accomplishes several improvements over the previous design by including enhancements to the algorithms and additional features.

The goal of the project is to design a wireless video transmission system aimed to work in an underwater environment. This approach faces several challenging problems. On the one hand, video signals typically have large information content, which requires a large bit rate to be transmitted in real-time. On the other hand, the underwater acoustic channel has a limited bandwidth that can only support low bit rates. In order to find a trade-off between the video bit rate requirements and the limited bandwidth of the channel (Figure 1.1), research has been conducted using two different approaches:

1. Research on advanced video compression techniques to reduce the large bit rate of the uncompressed video data in order to fit the limited bandwidth of the underwater channel.

2. Research on efficient modulation techniques designed to increase the bit rate supported by the system, such as Orthogonal Frequency-Division Multiplexing (OFDM).

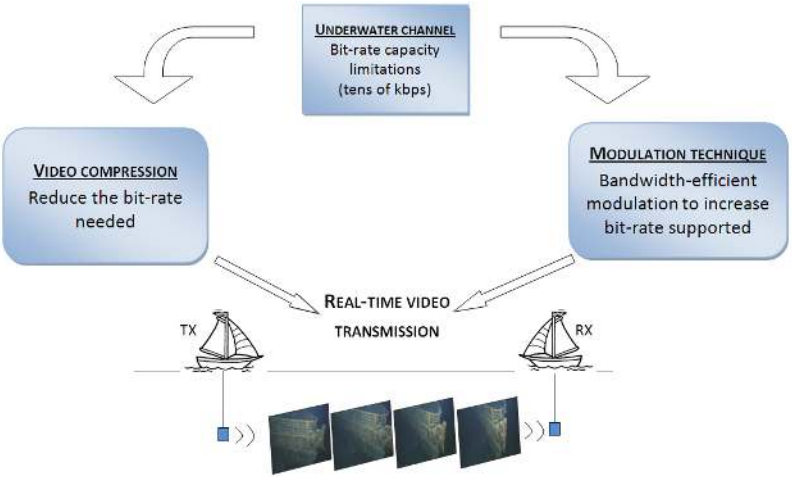

UNDERWATER CHANNEL
Bit-rate capacity limitations (tens of kbps)
VIDEO COMPRESSION
Reduce the bit-rate needed
MODULATION TECHNIQUE
Bandwidth-efficient modulation to increase bit-rate supported
REAL-TIME VIDEO TRANSMISSION
TX
RX

Figure 1.1: Problem statement.

Research on signal processing techniques has also been addressed in order to deal with the motion-induced Doppler effect. Moreover, a set of error resilience tools included at the video encoder has relaxed the performance requirements of the modulation system, while maintaining the visual quality of the decoded video. Finally, the performance of the system has been tested in a set of simulations and in-air experiments.

The final goal is to prove the feasibility of high data rate video transmissions in real-time, while providing an acceptable visual quality for supervision and monitoring tasks.

1.4 Outline

The first chapter of this thesis is a general introduction to the field of underwater video transmission. Chapter 2 contains a detailed description of the video compression fundamentals, along with an explanation of the functionailites of the MPEG-4 standard and the Compressed Sensing approach. The physical characteristics of the underwater channel are addressed in Chapter 3, as well as a review of the existing techniques for the communication link. The general structure of the system is described in Chapter 4, whereas the contributions made to the system are explained in Chapter 5. The simulations and experiments results are detailed in Chapter 6. Finally, Chapter 7 concludes the analysis and provides future directions.

Chapter 2

Image and video compression

In any underwater video compression method, there is a compromise between resolution and compression rate. The limited bit rate capacity of the underwater channel allows typical bit rates that can range from 50 bits per second (bps) to tens of kilobits per second (kbps). This fact makes it important to find a powerful compression technique that can reduce the initial bit rate requirements for real-time underwater video transmissions. As a numerical example, one can think of a low resolution video (176 ×144 pixels) with a frame rate of 10 fps. If it is transmitted without any compression, it will require about 2.5 Mbps of bit rate capacity.

This Section discusses the basics of data compression, while focusing on image and video compression. The MPEG-4 standard is extensively analyzed, since it is one of the latest compression methods that provide low bit rate compression tools. Another compression method, named Compressed Sensing, is also explained in detail.

2.1 Compression fundamentals

Compression is often referred to as coding, where coding is a very general term encompassing any special representation of data which satisfies a given need. Information theory is defined as the study of efficient coding in terms of transmission speed and error probability. Data, image or video compression may be viewed as a branch of information theory in which the primary objective is to minimize the amount of data to be transmitted.

When considering the image processing realm, a picture is represented by a two-dimensional array of samples or pixels (Figure 2.1). Each pixel can be a vector of any dimension representing the visual information of the image. If it is a one-dimensional vector, it will represent luminance for gray-scale images, whereas in a three-dimensional vector, each component will represent color components for color images.

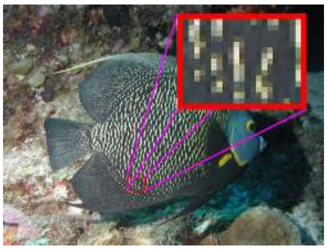


Figure 2.1: Pixels that form an image. Source: Jorgen van der Velde, www.deepocean.net

Color representations are typically modeled by an abstract mathematical model. For instance, the Red Green Blue (RGB) components system represents an image by three vectors, each of which contain the red, green and blue intensities of the image (Figure 2.2). The Cyan Magenta Yellow Key black (CMYK) is another color model that represents each pixel with a four-dimensional vector indicating the intensity of the named colors in the image. The YUV space components system encodes an image by taking human perception into account and defines it by its luminance Y, and two color difference components or chrominances: U=B-Y and V=R-Y, where B indicates the blue intensity and R, the red intensity.

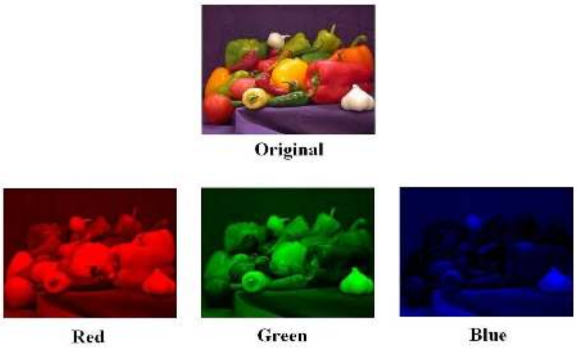


Figure 2.2: RGB decomposition. Source: Rick Rosson. *RGB Image Decomposition* (The MathWorks).

The resolution (width × height) and the color depth (number of bits used to represent each pixel) define the quality of an uncompressed digital picture. The higher the resolution and the greater the number of bits for each pixel, the higher the quality, as the edges are smoothed and the image looks more natural.

A video is a sequence of still images that, when correctly timed and displayed, convey the picture and motion information. An important quality measurement of a video is the frame rate, which is defined as the number of pictures (frames) displayed per second. It determines the subjective motion sensation: a higher frame rate translates into smoother motion, although it requires the transmission of more data. The level of compression of a video is usually determined by the bit rate, i.e., the total number of bits used to represent the video divided by the time duration of the video (bits/second).

2.1.1 Lossless and lossy compression methods

As a general classification, compression methods can be divided into lossless and lossy types. In the lossless type, no data is lost during the compression process, i.e., the decompressed file is identical to the data before compression. Software applications and databases are compressed using lossless techniques, since a change in a single bit of data can render them useless or produce incorrect results. Well-known lossless compression methods include ZIP and RAR.

The lossy processes reduce the file size by permanently eliminating redundant information, so that only a part of the original data is retained and reproduced. Most video and audio encoders compress data in ways that are lossy, but produce very high compression levels. Properly compressed video and audio files are nearly indistinguishable from the original to the human eye or ear.

Discrete cosine transform (DCT) is a lossy compression algorithm that transforms the spatial domain of a picture into the two-dimensional frequency domain by means of the discrete cosine transform, discarding those frequencies which do not affect the way in which human eye perceives the image (Figure 2.3).

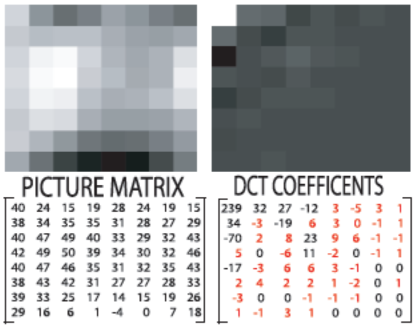
Vector quantization (VQ) is also a lossy compression method that considers an array of data instead of individual values. It can then generalize what it sees, compressing redundant data, while at the same time retaining the desired object or data stream's original intent.

Fractal compression is a form of VQ and is also a lossy compression. Compression is performed by locating self-similar Sections of an image, and using a fractal algorithm to generate the Sections.

2.1.2 Entropy, source and hybrid video coding

For their use in multimedia systems, image and video compression techniques can be classified into entropy coding, source coding and hybrid coding.

Entropy coding is a lossless process. The data to be compressed is considered a sequence of bits, and the semantics of the data are ignored by disregarding the specific characteristics of the stream. This coding scheme is used to compress data in file systems or still images in the Telefax standards.



$$\begin{bmatrix} 40 & 24 & 15 & 19 & 28 & 24 & 19 & 15 \\ 38 & 34 & 35 & 35 & 31 & 28 & 27 & 29 \\ 40 & 47 & 49 & 40 & 33 & 29 & 32 & 43 \\ 42 & 49 & 50 & 39 & 34 & 30 & 32 & 46 \\ 40 & 47 & 46 & 35 & 31 & 32 & 35 & 43 \\ 38 & 43 & 42 & 31 & 27 & 27 & 28 & 33 \\ 39 & 33 & 25 & 17 & 14 & 15 & 19 & 26 \\ 29 & 16 & 6 & 1 & -4 & 0 & 7 & 18 \end{bmatrix} \quad \begin{bmatrix} 239 & 32 & 27 & -12 & 3 & -5 & 3 & 1 \\ 34 & -3 & -19 & 6 & 3 & 0 & -1 & 1 \\ -70 & 2 & 8 & 23 & 9 & 6 & -1 & -1 \\ 5 & 0 & -6 & 11 & -2 & 0 & -1 & 1 \\ -17 & -3 & 6 & 6 & 3 & -1 & 0 & 0 \\ 2 & 4 & 2 & 2 & 1 & -2 & 0 & 1 \\ -3 & 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 1 & -1 & 3 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2.3: Result of transforming an image into its DCT representation. Numbers in red represent the coefficients that fall below a specified threshold and, thus, do not affect the image as the human eye perceives it. Source: Danny Blanco, Elliot Ng, Charlie Ice, Bryan Grandy. *Compression - Dropping the DCT Coefficients*.

In contrast, source coding takes advantage of the properties of the human eye and ear, and is a lossy process. Unlike the entropy coding, it takes into account the semantics of the data. There exists a one-way relation between the original data stream and the coded data stream, and they appear identical to the human perception. For example, in the case of speech, a transformation from the time domain to the frequency domain, followed by the encoding of the formants[1], substantially reduces the amount of data. Another example of source coding with specific semantic knowledge is the use of spatial redundancies in still images used in the DCT-based compression techniques.

Most multimedia systems combine both methods into hybrid coding techniques, such as the MPEG-4 coding standard.

2.2 The MPEG-4 standard

In this thesis, several video compression methods have been analyzed in order to find an appropiate technique that meets the limited bit rate capacity of the underwater transmission link. Among them, the commercially-available MPEG-4 has been chosen for its good compression capabilities. In particular, it provides high compression rates down to 64 kbps by combining spatial and temporal techniques, while providing good visual quality for the compressed video. In addition, it offers a set of error resilience tools that will be useful for the current application.

2.2.1 General description

MPEG-4 is a set of complex coding tools developed by the Moving Picture Experts Group (MPEG) to represent multimedia content consisting of natural and synthetic audio, visual, and object data. MPEG-4 has the same compression algorithm as MPEG-1 and MPEG-2, two similar compression techniques also developed by the MPEG group. The difference lies on the fact that MPEG-4 was expanded to support an object-oriented coding, low bit rate encoding, 3D content, text and other media types.

2.2.2 MPEG-4 object-oriented hierarchy

MPEG-4 standard is an object-oriented compression technique. Simply put, this means that it breaks down each video frame into objects, which are rendered in terms of lines and shapes following a mathematical theory.

The high level syntax used in the MPEG-4 standard consists of four hierarchically organized classes based on the objects present in the video (Figure 2.4):

- Video Session - Each Video Session (VS) consists of one or more Video Objects (VO), corresponding to the various objects in the scene.

[1] In the speech processing realm, formants are defined as the maxima in the voice spectrum.

- Video Object - Each one of these VOs can have several scalability layers (spatial, temporal, or Signal to Noise Ratio), corresponding to different Video Object Layers (VOL).
- Video Object Layer - Each VOL consists of an ordered sequence of snapshots in time, called Video Object Planes (VOP).
- Video Object Plane - Each VOP is a snapshot in time of a VO for a certain VOL. They are equivalent to the known concept of “frames”. There are three types of VOPs: Intra (I), Predictive (P) and Bi-directional (B) VOPs, and they are arranged in a periodic pattern referred to as Group of Pictures (GOP), e.g., IBBPBBPBBPBB. The I-VOPs are coded independently, irrespective of any other VOP. The P-VOPs are coded using motion information from a past reference I-VOP or P-VOP. B-VOPs are encoded by means of motion information from past and future P-VOPs or B-VOPs (Figure 2.5).

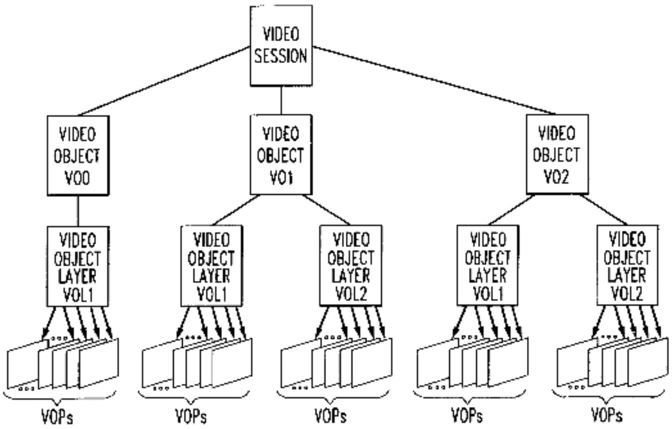


Figure 2.4: MPEG-4 file structure. Source: Barin Geoffry Haskell, Atul Puri, Robert Lewis Schmidt. *Generalized scalability for video coder based on video objects.*

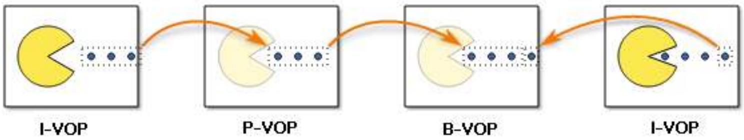


Figure 2.5: Types and recovery of VOPs. Source: Wikimedia Commons.

The first step in encoding one arbitrary VO is to find a rectangular bounding box that completely contains the object to be encoded. This bounding box is divided into blocks of 16 × 16 pixels called macroblocks (MB). Each one of these MBs is characterized by its shape, motion and texture, which have to be encoded.

Regarding the shape information, each MB is classified according to three possible classes: transparent (MB outside the object but inside the bounding box), opaque (MB completely inside the object) or border (MB over the border). The border MBs are the ones for which actual shape coding is required.

The motion information is encoded by means of motion vectors, which tell the decoder which block of pixels in the previous VOP is closest to the current one and, therefore, will be used for the prediction of the texture. The decoder will also use the motion vectors for motion compensation.

Finally, the texture data can be encoded in any of two modes: intra coding for I-VOPs and inter coding for P-VOPs and B-VOPs. If intra coding is chosen, the given MB is encoded by itself (with no temporal prediction), exploiting only the spatial redundancies. If inter coding is performed, then temporal prediction and motion compensation are used to explore the temporal redundancy, and the differences between the current and the prediction MB are encoded. Both the absolute and the differential texture values are then encoded using DCT coefficients.

Finally, the motion information, the DCT coefficients and some added overhead information are encoded using run-length schemes and Variable Length Coding (VLC) techniques in order to achieve a higher degree of compression. The information for shape, motion, and texture is multiplexed at the MB level, which means that for each MB the shape information is sent first, then the motion data, and finally the texture data. The block diagram of this process is illustrated in Figure 2.6.

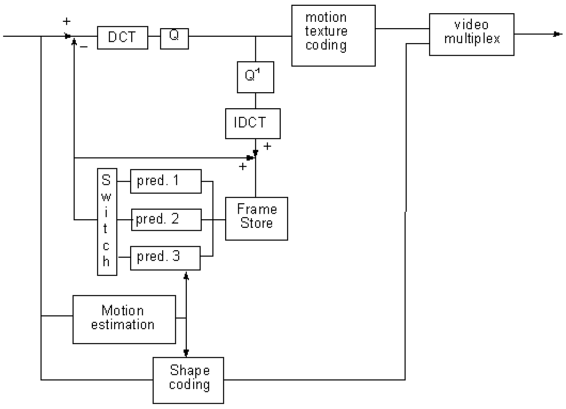
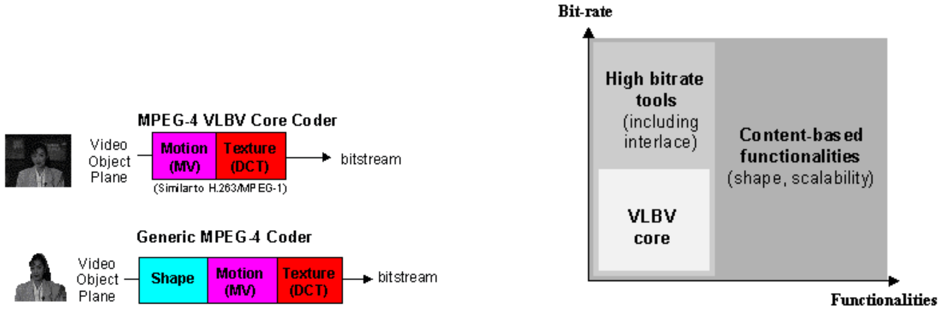


Figure 2.6: Block diagram of the MPEG-4 Video Coder. Source: http://mpeg.chiariglione.org

2.2.3 Coder type and profile

As mentioned in the previous Section, the coding and representation of video content is accomplished through an object-oriented approach. This approach is especially useful for modern interactive multimedia systems, where the content-based functionalities have an important role. However, the videos used in the current project do not require any specific technologies to encode complex shapes or transparencies. For these kind of videos with lighter content-based requirements, MPEG-4 provides the Very Low bit rate Video (VLBV) coder. It is similar to the conventional rectangular video coding, which is considered as a special case of the object-oriented approach. It involves motion prediction/compensation followed by texture coding. Since the VLBV coder assumes that the Video Objects are the rectangular frames that form the video, it does not encode any kind of shape information (Figure 2.7- *a*).



(a) The VLBV Core and the Generic MPEG-4 Coder.

(b) Classification of the MPEG-4 Image and Video Coding Algorithms and Tools.

Figure 2.7: MPEG-4 functionalities and coders. Source: http://mpeg.chiariglione.org

The VLBV coder provides algorithms and tools for applications operating at bit rates typically between 5 and 64 kbps, supporting image sequences with low spatial resolution and low frame rates (Figure 2.7-*b*). The basic functionalities supported by the VLBV Core include coding of conventional rectangular image sequences with high coding efficiency and high error robustness, low latency and low complexity for real-time multimedia communications applications.

Profiling is another consideration that must be taken into account when selecting an appropiate MPEG-4 scheme. The specifications for the coding tools provided by MPEG-4 are divided into three parts: Systems, Visual and Audio. Subsets of these groups, named Profiles, have been identified, each one having different toolsets for a variety of applications. For each of these Profiles, one or more Levels have been set, limiting some parameters of the tools present in a profile. A hierarchical flowchart is shown in Figure 2.8.

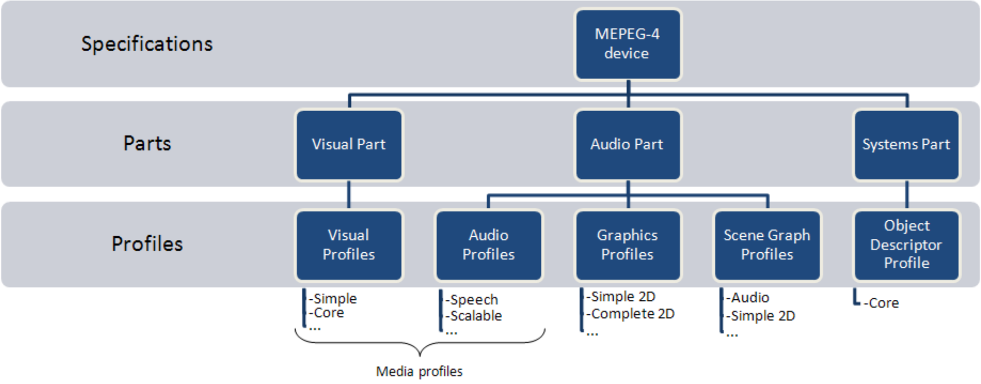


Figure 2.8: An overview of the profile structure in MPEG-4.

In the current project, no audio data has been encoded and only the Visual Simple Profile at Level 0 has been specified. This level supports a maximum total resolution of a Quarter Common International Format (QCIF) picture (176×144 pixels). There are four levels for the Simple Profile with bit rates ranging from 64 to 384 kbps. This profile also provides several error resilience tools that will be useful for the purposes of the current application (Section 5.2). It was created with low complexity applications in mind such as network cameras, video conferencing, multimedia gateways, applications of surveillance, video servers and consumer electronics.

2.3 A special case of compression: Video Compressed Sensing

Although the MPEG-4 standard has been deemed as the most appropiate for our current system requirements, the Compressed Sensing (CS) method has been extensively analyzed for its recent good results [14, 15].

2.3.1 Compressed Sensing fundamentals

Compressed Sensing, also known as Compressive Sensing, Compressive Sampling and Sparse Sampling, aims to reconstruct signals and images from significantly fewer measurements than than the traditional necessary by utilizing the prior knowledge that the signals are sparse or compressible [16].

Let us consider an image x of N pixels which can be expanded in an orthonormal basis $\Psi_{N\times N}$, such as wavelets:

$$x = \Psi s \tag{2.1}$$

where s is the K-sparse coefficient vector of x in the Ψ domain (it only has $K \ll N$ non-zero coefficients).

Let us define the *coherence* between two domains as the largest correlation between any two elements of its orthonormal basis:

$$\mu(\Psi, \Phi) = \sqrt{N} \cdot \max |\langle \Psi, \Phi \rangle| \tag{2.2}$$

If Ψ and Φ contain correlated elements, the coherence is large. Otherwise, it is small. As for how large and how small, it follows from linear algebra that $\mu(\Psi, \Phi) \in [1, \sqrt{n}]$.

Compressed Sensing states that if there exists another domain Φ such that it has minimum coherence with the Ψ domain (i.e., x is not sparse in the Φ domain), then the following algorithm can be used to recover the image from only M samples ($K < M \ll N$) of the original image x:

1. Take $M = CK\log(N)$ samples in the Φ domain, for some positive constant C[2]:

$$y = \Phi x \tag{2.3}$$

   where $y = [y_1, y_2, ..., y_M]$ is the vector containing the M measurements and $\Phi_{M\times N}$ is the matrix containing the sampling vectors of the incoherent domain.

2. Recover the original image by solving the convex problem:

$$\min_{\tilde{s}} ||\tilde{s}||_1 \qquad \text{subject to} \qquad y = \Phi\tilde{x} = \Phi\Psi\tilde{s} \tag{2.4}$$

   where $\|\|_1$ denotes the $l_1 -$ norm, defined as the sum of the absolute values of the vector.

3. The recovered image would be $\tilde{x} = \Psi\tilde{s}$.

2.3.2 Compressed Sensing applied to video compression

Since underwater video images are usually sparse in some basis, the Compressed Sensing approach can be applied to a sequence of video frames. In this thesis, an algorithm that takes advantage of the sparsity of underwater video images has been designed and compared to the MPEG-4 approach.

Let us consider a video consisting of F frames with N pixels per frame. The key steps of the algorithm are:

1. Presume that each frame x is K-sparse ($K \ll N$) in the wavelet domain Ψ.

2. Take $M = CK\log(N)$ samples of each frame in an incoherent domain Φ. In this algorithm, the Hadamard matrix has been used as it is incoherent with most orthonormal basis [17].

3. Recover each frame using the Gradient Projection for Sparse Reconstruction[3] (GPSR) algorithm [18], similar to the $l_1 -$ norm convex problem:

$$\min_{\tilde{s}} \frac{1}{2}||y - \Phi\Psi\tilde{s}||_2^2 + \tau||\tilde{s}||_1 \tag{2.5}$$

   where $\|\|_2$ denotes the $l_2 -$ norm, defined as the sum of the squared absolute values of the vector.

4. Each recovered frame can be expressed as $\tilde{x} = \Psi\tilde{s}$.

2.3.3 Video Compressed Sensing versus MPEG-4

In order to assess the performance of the designed algorithm in terms of video bit rate savings, it has been compared to the MPEG-4 compression method:

- Let us simplify the **MPEG-4** object-oriented approach and consider that it divides and classifies the frames (not VOPs) into I, B and P frames:
  - *I-frames* are fully encoded using the Discrete Cosine or Discrete Wavelet transforms. Since images are usually K-sparse in these domains, each I-frame is efficiently encoded using only K coefficients.
  - As explained in Section 2.2, *P-frames and B-frames* are not fully encoded but rather predicted using motion vectors from past and future frames.

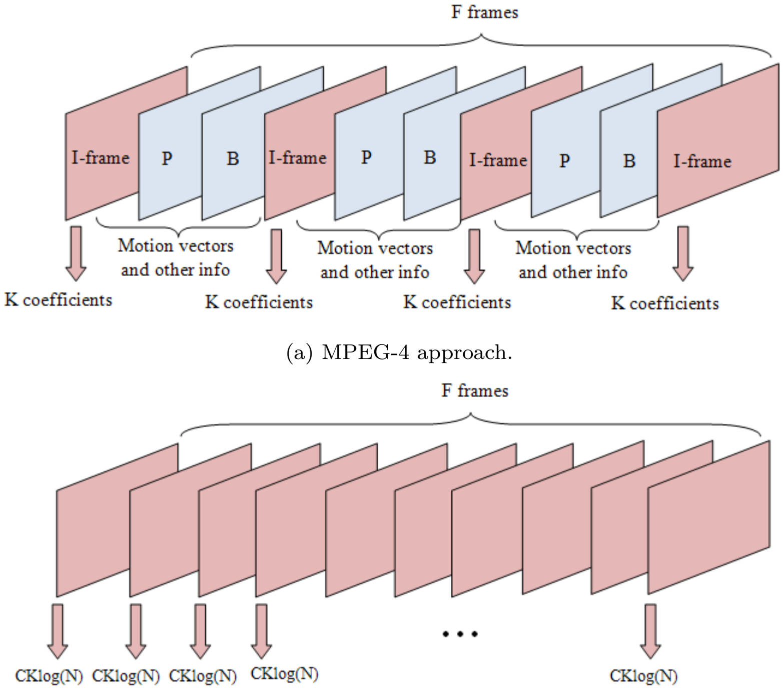
[2] C is usually between 3 and 4.

[3] The GPSR algorithm has been used instead of the $l_1 -$ norm in order to reduce the computation time of the recovering algorithm. See [18] for more details.

For the whole video we will have $\alpha F \cdot K$ coefficients, where α indicates the portion of I-frames in the video (usually around 10%), along with some motion vectors and other information that are negligible in terms of size compared to the $\alpha F \cdot K$ coefficients.

- The algorithm using the **Compressed Sensing** approach obtains $M = CK \log(N)$ samples of each frame, which corresponds to $F \cdot CK \log(N)$ non-zero coefficients for the whole video.

The amount coefficients that contain the video compressed using MPEG-4, $\alpha F \cdot K$, is far below the number of coefficients that contain the video compressed using Compressed Sensing; $F \cdot CK \log(N)$. As expected, MPEG-4 provides a much better compression ratio because it exploits the temporal correlation between frames (Figure 2.9).



(a) MPEG-4 approach.

(b) Compressed Sensing approach.

Figure 2.9: MPEG-4 versus Compressed Sensing.

As a numeric example, a video file consisting of 46 frames with $N = 320 \times 480 = 153,600$ pixels per frame has been compressed using both methods. The results are the following:

- Using a compression ratio of 15%, the Compressed Sensing algorithm could recover the original video with a Peak Signal-to-Noise Ratio (PSNR) of 31 dB.
- The same file was compressed using the MPEG-4 standard with a compression ratio of 0.45%. The video could be recovered with the same value of $PSNR = 31$ dB.

Future work in this field includes research on the sparsity in the temporal domain of underwater video sequences. If Compressed Sensing can be applied to both the temporal and spatial domains, a new compression approach could be designed by combining the Compressed Sensing principles with the motion estimation features of the MPEG-4 standard.

2.3.4 Applications of Video Compressed Sensing

Although Compressed Sensing theory has not proved a useful approach for bit rate savings purposes, it has other applications in the realm of sampling techniques:

- It has the ability to capture a video using only few measurements, which may be useful in applications where it is computationally difficult or expensive to obtain raw samples. For instance, Magnetic Resonance Imaging (MRI) is an essential medical imaging tool burdened by an inherently slow data acquisition process. The application of Compressed Sensing to MRI has the potential for significant scan time reductions, with benefits for patients and health care economics. Note that this is not the case of the current system, in which a large amount of video data has already been recorded and needs to be reduced.

- Multi-view imaging: multicamera arrays are used to capture a scene from different angles. Several limitations such as bandwidth, power and storage could be overcome with Compressed Sensing, treating each signal captured by each camera as a frame of the video sequence.

Chapter 3

Communication link

Once the video has been compressed, it has to be transmitted acoustically to the receiver through an underwater wireless communication link. The first Section of this chapter presents an overview of the existing techniques for underwater video transmission, covering the radio frequency (RF), optical and acoustic waves. The last Section addresses the physical characteristics of the underwater channel.

3.1 Existing techniques for underwater wireless transmission

3.1.1 Radio Frequency waves

Radio frequency (RF) waves are electromagnetic waves in the frequency band below 300 GHz. An electromagnetic wave is a wave of energy that has a frequency within the electromagnetic spectrum and propagates as a periodic disturbance of the electromagnetic field when an electric charge oscillates or accelerates. Underwater radio frequency communications have been investigated since the very early days of radio [19], and received considerable attention during the 1970s. Even so, few underwater RF systems have been developed due to the highly conducting nature of sea water. Some of them are in use for short ranges, such as WFS modems [20, 21].

3.1.2 Optical waves

Optical waves are electromagnetic waves that have wavelengths between 400 nm (blue light) and 700 nm (red light). Due to their very short wavelength, high frequency and high speed ($3 \cdot 10^8$ m/s), optical waves offer the potential for very high speed communication underwater –up to 1 Gbps. However, optical waves used as wireless communication carriers are generally limited to very short distances because of severe water absorption at optical frequencies and strong backscatter from suspended particles [22]. Research conducted at MIT [23] and the optical/acoustic modem designed by the Woods Hole Oceanographic Institution (WHOI) showed good results in short-range high-rate optical communication. However, no commercial optical modems for underwater environments have been developed so far.

3.1.3 Acoustic waves

Acoustic waves are caused from variations of pressure in a medium. Due to the greater density of water, they travel five times faster in water than they do in air, but are about five orders of magnitude slower than electromagnetic waves. They have been widely used in underwater communication systems due to the relatively low attenuation of sound in water. However, acoustic waves can be adversely affected by absorption loss, spreading loss, ambient noise, and severe multipath, as explained in Section 3.2. In spite of these impairments, acoustic waves remain as the most robust and feasible carrier to date for wireless underwater communications. For this reason, the current system has been designed to work with acoustic waves.

3.2 Underwater channel

The underwater acoustic channel is considered to be one of the most difficult media for wireless communications. Its main drawbacks are the frequency-dependent attenuation that affects especially the higher frequencies, the dependency of the bandwidth on the distance (usual designs operate within a few tens of kHz), the non-negligible background noise and the non-uniform Doppler effect that is more accentuated than in radio channels. The channel frequency response is not constant and strong fades are present as a consequence of the multipath propagation. In addition, the low speed propagation of the sound (1,500 m/s) causes high propagation delays. In this Section, each of these characteristics are explained in detail.

3.2.1 Acoustic propagation

3.2.1.1 Attenuation

The attenuation or path loss in an underwater channel over a distance l for a signal of frequency f is given by the expression

$$A(l,f) = l^k a(f)^l \tag{3.1}$$

where k is the propagation constant, whose values are usually between 1-2, and $a(f)$ is the absorption coefficient, which is an increasing function of the frequency (Figure 3.1). The cause for the frequency-dependence is the energy absorption of the pressure waves and the spreading loss that increases with distance.

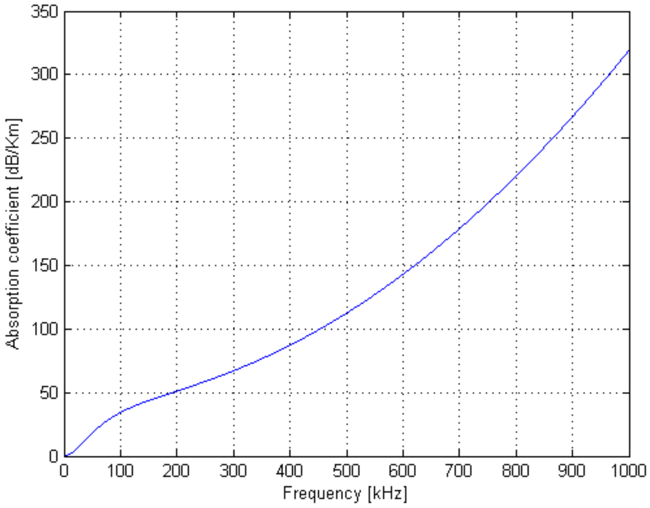


Figure 3.1: Absorption coefficient as a function of frequency.

The expression of the absorption coefficient for frequencies above a few hundred Hz can be expressed using the Thorp's formula [24], which gives $a(f)$ in dB/Km for f in kHz:

$$10\log a(f) = 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f^2} + 2.75\cdot 10^{-4}f^2 + 0.003 \tag{3.2}$$

For lower frequencies, the expression is:

$$10\log a(f) = 0.11\frac{f^2}{1+f^2} + 0.011f^2 + 0.002 \tag{3.3}$$

The absorption coefficient increases rapidly with frequency, thus limiting the available bandwidth for a given distance. Although it is common to use reduced bandwidths, the system is inherently wideband due to the low carrier frequencies employed in acoustic communications.

3.2.1.2 Noise

The ambient noise in an underwater acoustic channel can be modeled taking into account four sources: turbulence, shipping, waves and thermal noise. Most of the sources may be described as having a continuous spectrum and Gaussian statistics. The following empirical formulae give the power spectral density (PSD) of the four noise components described above in dB re μ Pa per Hz as a function of frequency in kHz:

$$\begin{aligned}10\log N_t(f) &= 17 - 30\log f \\ 10\log N_s(f) &= 40 + 20(s-0.5) + 26\log(f) - 60\log(f-0.03) \\ 10\log N_w(f) &= 50 + 7.5w^{1/2} + 20\log f \\ 10\log N_{th}(f) &= -15 + 20\log f \end{aligned} \tag{3.4}$$

where the shipping activity s ranges from 0 to 1, for low and high activity, respectively, and w is the wind speed measured in m/s. Turbulence noise only affects the very low frequencies region, $f < 10$ Hz. Noise caused by distant shipping is dominant in the frequency region 10-100 Hz. The surface motion caused by wind-driven waves is the major contribution to the noise in the frequency region 100 Hz - 100 kHz, which is the one used by most acoustic systems. Finally, thermal noise becomes dominant for $f > 100$ kHz, and it is not so accentuated for shallow water channels.

The overall PSD of the ambient noise, $N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f)$, is illustrated in Figure 3.2, for the cases of no wind and wind at 10 m/s, with different shipping activity factors. Note that the noise increases in the low frequency range, limiting the useful bandwidth from below. As the PSD decays linearly on the logarithmic scale (in a certain frequency range), the overall ambient noise can be approximated by a power spectral density that decays at approximately 18 dB/decade:

$$N(f) \approx 10^5 \cdot f^{-18} \tag{3.5}$$

This approximation is also shown in Figure 3.2.

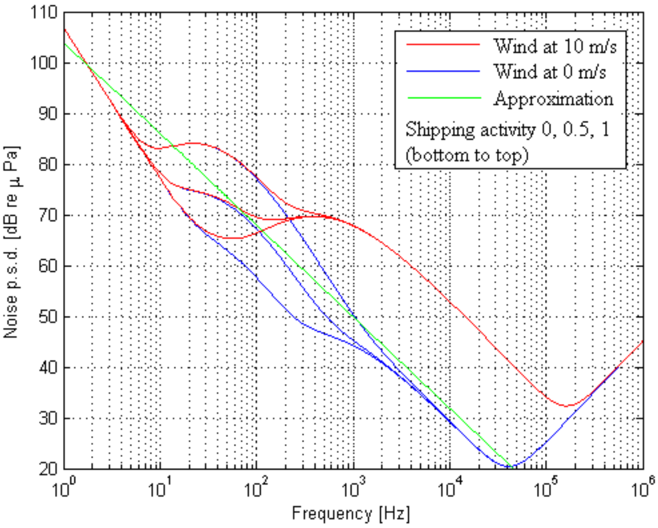


Figure 3.2: Power spectral density of the ambient noise, $10\log N(f)$.

3.2.1.3 Propagation delay

The nominal speed of sound in water is $1,500$ m/s; 200,000 times lower than the speed of electromagnetic waves in open-air. Thus, the delays experienced in an underwater channel are much higher than in open-air environments, making it difficult to use feedback information to correct channel distortions. The order of typical propagation delays in acoustic underwater links is several seconds, while the coherence time[1] in an underwater channel can be of 100 milliseconds. In contrast, the propagation delay in an open-air channel can be in the order of microseconds.

[1] Time duration over which the channel impulse response is considered to be not varying.

3.2.1.4 Multipath propagation

A common problem in underwater acoustic links is the phenomenon called multipath propagation, which results in acoustic signals reaching the receiver by two or more paths. Hence, more than one pulse is detected at the receiver, each of them with different amplitudes, phases and instants of arrival. This effect can be modeled by the next expression for the channel response:

$$c(\tau, t) = \sum_{p} A_p(t)\delta(\tau - \tau_p(t)) \tag{3.6}$$

where $A_p(t)$ are the paths' amplitudes and $\tau_p(t)$ are the time-varying paths' delays (Section 5.1).

The mechanisms of the multipath formation in the ocean depend on the channel geometry, the signal frequency, the sound speed profile (sound speed increases with temperature, pressure and salinity), the range of transmission and the depth of the water. Two fundamental mechanisms of multipath propagation are the reflection at the boundaries (bottom, surface and any objects in the water) and the refraction, i.e., the tendency of acoustic waves to travel along the axis of lowest sound speed. The former occurs more commonly in shallow waters, while the latter is typically experienced in deep water environments. Figure 3.3 illustrates these mechanisms.

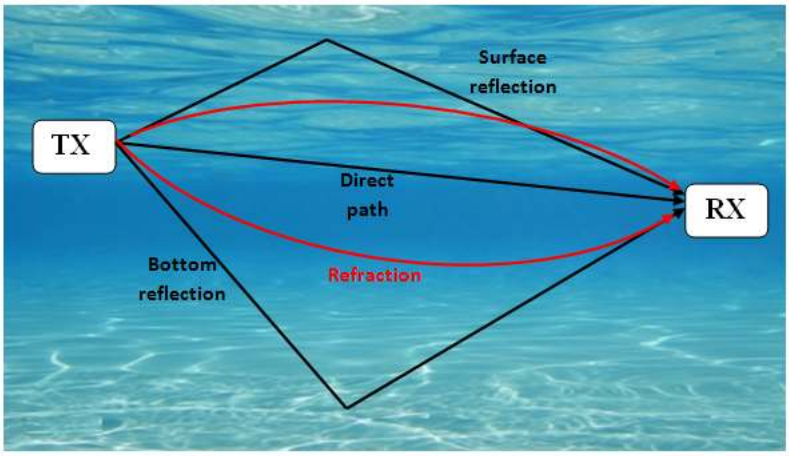


Figure 3.3: Multipath effects.

Regardless of its origin, multipath propagation creates unwanted signal echoes, resulting in Inter-Symbol Interference (ISI) in a digital communication system. To avoid the ISI, a guard time of length at least equal to the multipath spread must be inserted between successively transmitted symbols. However, this reduces the overall symbol rate, which is already limited by the system bandwidth.

3.2.1.5 Doppler effect

Underwater acoustic communications systems usually have to deal with a non-negligible Doppler effect caused by the relative motion of the transmitter-receiver pair. It is known that the Doppler effect is proportional to the ratio $a = v_r/c$, where v_r is the relative velocity between the transmitter and the receiver and c is the signal propagation speed. In radio wireless communications, where the speed of the electromagnetic signals is $3 \cdot 10^8$ m/s, the Doppler factor takes values around $a = 10^{-7}$, which is negligible. However, in an underwater wireless acoustic system, the Doppler coefficient can reach non-negligible values around $a = 10^{-4}$ due to the lower velocity of acoustic signals ($1,500$ m/s). Moreover, the fact that underwater systems are wideband causes many Doppler shifts for different frequency components of the transmitted signal, known as frequency spreading.

Thus, it is important for underwater acoustic systems to include techniques that deal with the Doppler effect. One of the contributions of this thesis is the incorporation of an algorithm at the receiver side aimed to counteract the Doppler effects (Section 5.1).

3.2.2 Resource allocation

Unlike the radio spectrum, the underwater acoustic spectrum has not yet been regulated by the Federal Commission of Communications (FCC), although some restrictions have been introduced by the Marine Mammal Protection Act (MMPA) [25]. However, the acoustic path characterization and the ambient noise characteristics do not leave many possibilities to allocate the system frequency band. Considering optimal signal energy allocation, such band is defined so that the channel capacity is maximized [26, 27].

In this Section, the frequency allocation methodology is introduced in order to assess the importance of correctly selecting the center frequency f_c and the system bandwidth B.

3.2.2.1 The AN product and the SNR

Taking into account the attenuation over a distance l for a frequency f (Section 3.2.1.1) and the spectrum of the ambient noise (Section 3.2.1.2), one can derive an expression for the narrow-band Signal-to-Noise Ratio (SNR) observed at the receiver:

$$SNR(l,f) = \frac{P/A(l,f)}{N(f)\Delta f} = \frac{S(l,f)}{N(f)A(l,f)} \tag{3.7}$$

where P is the average power of the transmitted signal, Δf is a narrow band around the frequency f, $S(f) = P/\Delta f$ is the PSD of the transmitted signal and $N(f)$ is the PSD of the ambient noise. In this expression, directivity indices and losses other than the path loss are not taken into account. The AN product, $A(l,f)N(f)$, determines the frequency-dependent part of the SNR. Its inverse, $1/(A(l,f)N(f))$, is illustrated in Figure 3.4 for different distances.

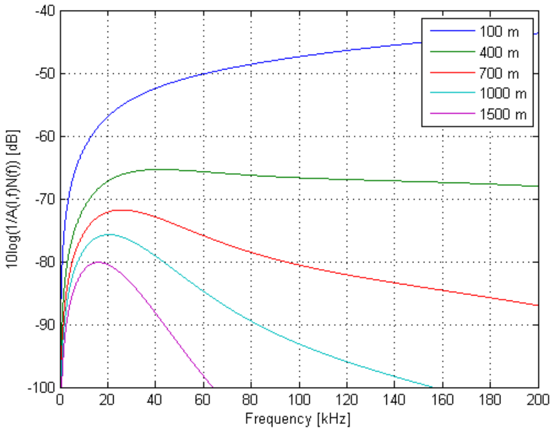


Figure 3.4: Inverse of the frequency-dependent part of the SNR, $1/(A(l,f)N(f))$. Practical spreading factor $k = 1.5$. $N(f)$ was calculated using approximation (3.5).

3.2.2.2 Optimal frequency

From Figure 3.4, one can conclude that, given a certain transmission distance l, there exists an optimal frequency $f_0(l)$ for which the maximal narrow-band SNR is achieved. This optimal frequency is illustrated in Figure 3.5 as a function of the transmitter-receiver distance.

Optimally, some transmission bandwidth around $f_0(l)$ is chosen when implementing a communication system. The transmission power is then adjusted in order to achieve the desired SNR value throughout the selected frequency band. In practice, the transmitting and receiving responses of the transducers and hydrophones should be taken into account and, thus, the optimal transmission frequency may vary.

3.2.2.3 3 dB bandwidth definition

In the communications field, the 3 dB bandwidth is defined as the range of frequencies around $f_0(l)$ for which $SNR(l,f) > SNR(l,f_0(l))/2$. From expression 3.7 and considering a constant PSD of the

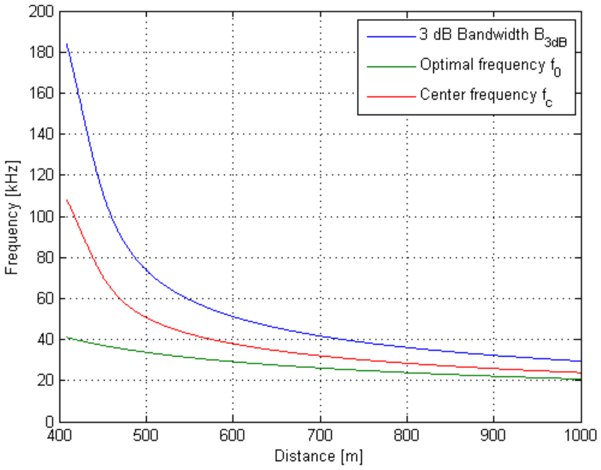


Figure 3.5: Optimal frequency $f_0(l)$ considering the inverse of the AN product, $1/(A(l,f)N(f))$. The 3dB bandwidth, $B_{3dB}(l)$, and the center frequency of this bandwidth, $f_c(l)$, are also shown. $N(f)$ was calculated using approximation (3.5).

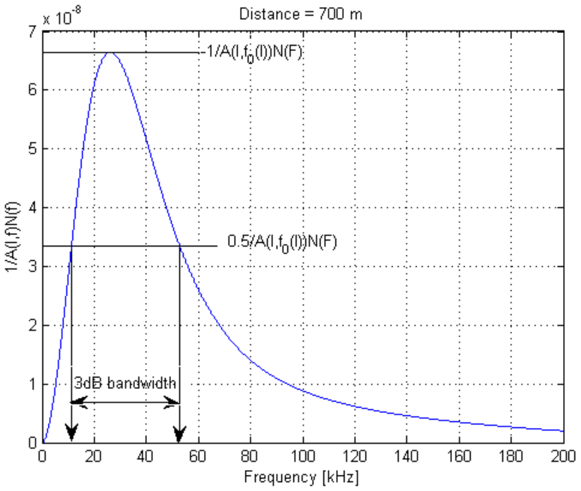


Figure 3.6: 3 dB bandwidth for a distance of $l = 700$ m.

transmitted signal $S(f)$, the previous relation can be expressed as $A(l,f)N(f) < 2A(l,f_0(l))N(f_0(l))$. This concept is illustrated in Figure 3.6 for a distance of 700 m. The 3 dB bandwidth, $B_{3dB}(l)$, and the center frequency of the frequency band, $f_c(l)$, are shown in Figure 3.5 as a function of the transmission distance. As expected, these two parameters and the optimal frequency, $f_0(l)$, show similar trends. Note that the achievable bandwidth decreases with the distance l.

3.2.2.4 Transmission power

Assuming that the PSD of the transmitted signal has been set within the 3 dB bandwidth, the transmission power $P(l)$ can be adjusted so as to provide a target narrow-band SNR_0, corresponding to the 3 dB bandwidth $B_{3dB}(l)$. Denoting the PSD of the transmitted signal at a certain distance l by $S_l(f)$, the total necessary power to be transmitted is:

$$P(l) = \int_{B_{3dB(l)}} S_l(f)df = SNR_0 B_{3dB}(l) \frac{\int_{B_{3dB(l)}} N(f)df}{\int_{B_{3dB(l)}} A^{-1}(l,f)df} \tag{3.8}$$

where the transmitted signal PSD is considered constant in the signal bandwidth.

Chapter 4

OFDM system

In telecommunications, modulation is the process of conveying a signal, for instance a digital bit stream or an audio signal, inside another signal that can be physically transmitted. Taking into account parameters such as implementation complexity, robustness against channel effects and supported data rate, the best modulation scheme must be chosen for each specific system.

In this chapter, a review of the existing modulation techniques is first discussed, covering the single and multi-carrier methods. Next, the OFDM theoretical principles are explained, as it has been the modulation chosen for this project. Finally, a detailed description of the designed OFDM system can be found at the end of the chapter.

4.1 Modulation techniques: single versus multi-carrier

Single-carrier methods modulate information onto a single carrier using frequency, phase, or amplitude adjustment of the carrier. This is a good technique for channel tracking purposes. Single-carrier equalization [28] is a method that has been used in many systems that have attempted image or video transmission through an underwater acoustic channel. This technique has also been used in previous work done by the MIT Sea Grant College Program [29], where pre-packaged video images were transmitted at the very high bit rate of 150 kbps, but only over a stable, 10 m long vertical path.

Multi-carrier schemes extend the concept of single carrier modulation by dividing the transmitted data into multiple subcarriers within the same bandwidth. They can also be used as Frequency-Division Multiplexing (FDM) techniques by assigning each subcarrier to independent data sources (or users). Such techniques offer an advantage over single-carrier modulations in terms of narrowband frequency interference, as only some frequency sub-bands will be affected. Another advantage of the multi-carrier systems is the ease of implementation of adaptive modulation, which consists of using a different modulation for each of the subcarriers depending on the channel conditions in each frequency band. However, signals modulated using a multi-carrier scheme present a higher Peak-to-Average Ratio (Section 4.3). Another disadvantage of these systems is that they usually require a guard band between modulated subcarriers to prevent the spectrum of one subcarrier from interfering with another. These guard bands lower the system's effective information rate when compared to a single carrier system with similar modulation. Current National Television Systems Committee (NTSC) television and Frequency Modulation (FM) stereo multiplex are examples of this technique.

4.2 OFDM fundamentals

OFDM stands for Orthogonal Frequency-Division Multiplexing and is a special case of FDM. It can be treated as a combination of modulation techniques (a mapping of the information on changes in the carrier phase, frequency, amplitude or combination) and multiplexing techniques.

The principles of OFDM modulations have been in existence for several decades. However, in recent years these techniques have quickly moved out of textbooks and research laboratories and been put into practice in modern communications systems. They are well-suited for systems in which the channel characteristics make it difficult to maintain adequate communication link performance, such as the Asynchronous Digital Subscriber Line (ADSL) or the European digital television, which is based on the Digital Video Broadcast-Terrestrial (DVB-T) standard. It is also being considered for the fourth generation of cellular wireless standards.

OFDM's application to underwater acoustic systems has been addressed recently as an alternative to deal with the frequency-selective underwater channel. It relies on the idea of dividing the total available bandwidth into many closely-spaced orthogonal sub-bands, such that the channel transfer function appears constant (ideal) within each sub-band. Each of these sub-bands carries a portion of the signal to be transmitted. Subsequently, each of these bit streams is divided into several groups of bits, which are mapped into symbols using a basic modulation scheme such as the Quadrature Amplitude Modulation (QAM) or the Phase Shift Keying (PSK) (Figure 4.1).

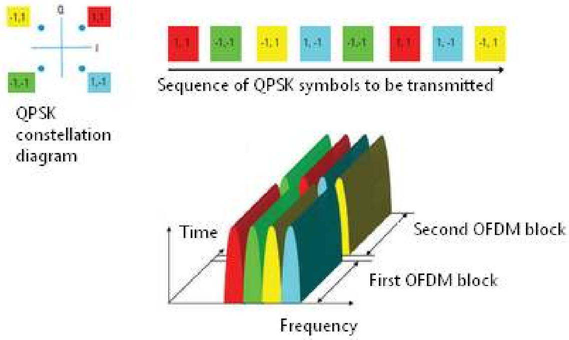


Figure 4.1: Example of an OFDM scheme in time and frequency for a sequence of QPSK symbols. Source: *OFDM and SC-FDMA*. http://3g4g.blogspot.com/2009/02/ofdm-and-sc-fdma.html

Note that the channel equalization in OFDM systems is simple because each of the sub-bands is considered to be constant. Moreover, if differential encoding is performed in the frequency domain, equalization becomes simpler because these schemes are insensitive to slowly changing amplitude and phase distortion (see *Differentially coherent detector* on page 49).

An important concept when considering OFDM systems is the orthogonality between subcarriers. In order to avoid Inter-Carrier Interference (ICI), the subcarrier frequencies are chosen so that they are orthogonal to each other, which means that in the center frequency of the subcarrier k the other subcarriers' amplitude is null (Figure 4.2). Since inter-carrier guard bands or separate filters for each subchannel are not required, the design for both the transmitter and receiver is simplified. The orthogonality between subcarriers also allows to efficiently implement the modulator and demodulator of the system using the Fast Fourier Transform (FFT) algorithm on the receiver side and the Inverse Fast Fourier Transform (IFFT) on the transmitter side (Section 4.3 and 4.4).

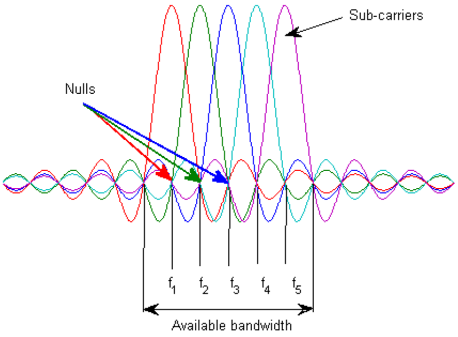


Figure 4.2: Bandwidth utilization for an OFDM signal.

In order for the subcarriers to be orthogonal, it is required that the subcarrier spacing is $\triangle f = \alpha/T$ Hz (equation 4.1), where T seconds is the useful symbol duration, and α is a positive integer, typically equal to 1.

$$\langle \phi_k, \phi_m \rangle = \int_0^T e^{j2\pi(k-m)\triangle ft} dt = \frac{e^{j2\pi(k-m)} - 1}{2\pi(k-m)\triangle f} = C \cdot \text{sinc}(k-m) = C \cdot \delta(k-m) \tag{4.1}$$

where C is a constant. Therefore, with K subcarriers, the total passband bandwidth is $B = K \cdot \Delta f$.

The orthogonality between subcarriers allows high spectral efficiency, since most of the available frequency band can be utilized. However, OFDM systems require very accurate frequency synchronization between the receiver and the transmitter, because any slight frequency deviation can violate the orthogonality. The frequency offsets in underwater environments are caused by Doppler shifts due to motion between the transmitter and receiver. While Doppler shifts may be compensated for at the receiver, the situation is worsened when combined with multipath, as reflections will appear at various frequency offsets, making it much harder to correct the resulting signal.

One of the advantages of the OFDM scheme is that it is robust against Inter-Symbol Interference (ISI) because a guard interval is introduced between adjacent OFDM blocks, making it possible to handle time-spreading.

In summary, the main advantages and disadvantages of the OFDM modulation technique are highlighted below.

Advantages

- Robustness against frequency selectivity.
- High spectral efficiency.
- Efficient implementation using FFT, avoiding the need for complex subchannel filters.
- Ease of implementation of adaptive modulation.

Disadvantages

- Sensitivity to Doppler shifts and time variation (Doppler spread).
- High Peak-to-Average Ratio (PAR), requiring linear transmitter circuitry, which suffers from poor power efficiency. In order to deal with this problem, several PAR reducing techniques have been included in this system (Section 4.3).
- Loss of efficiency caused by guard interval.

4.3 Transmitter

In this Section, each part of the transmitter block is described in detail, as well as the theoretical framework of the system. The model used on the transmitter side consists of a typical OFDM system with K subchannels, and its block diagram is illustrated in Figure 4.3. For an extended description of the general structure of the transmitter and the receiver, refer to [13].

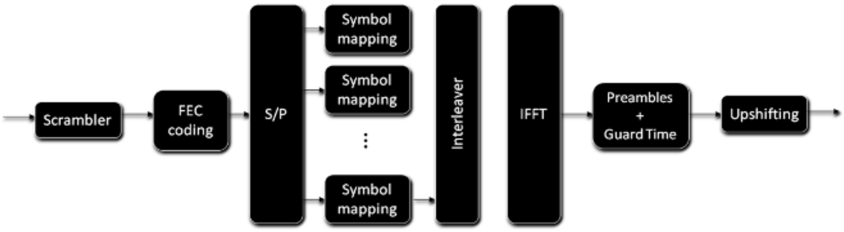


Figure 4.3: Transmitter scheme.

Scrambler

The input signal to the transmitter is an MPEG-4 coded video file, which is read in the form of a long bit stream. The first step consists of randomizing the 0/1 sequences in order to eliminate the correlation and, thus, uniformize the power allocation. At the output of the scrambler, there will be a similar amount of '0' and '1' values, avoiding long '0' or '1' bursts that can result in the same codeword when passed through the FEC coder.

Scrambling is implemented by applying a logical XOR operation between a pseudo-random scrambling sequence and the bit stream coming from the MPEG-4 coder. The pseudo-random scrambling sequence is known by the descrambling block at the receiver, which can decode the received sequence by re-applying a logical XOR.

The scrambling algorithm, in addition to the security and protection of the established communication, has also proved to reduce the Peak-to-Average Ratio of the resulting OFDM signal, as it decorrelates the constellation points that come out from the subcarrier mapper (see *Symbol interleaving to reduce Peak-to-Average Ratio* on page 46).

FEC coding

The next step is to apply a Forward Error Correction (FEC) coding to the data. FEC coding is an error control system for data transmission, whereby the sender systematically generates and adds redundant data to the transmitted signal, so that the random errors in the received bit stream can be corrected by the decoder. In order to make best use of FEC coding, the possible error bursts due to poor channel conditions or highly attenuated frequency bands must be minimized. For this reason, an interleaver is included after the FFT block.

The FEC codes used in this project are the Bose-Chaudhuri-Hocquenghem (BCH) codes [30]. They belong to the category of FEC block codes, which operate on fixed-size blocks (packets) of bits or symbols of predetermined size. They have been selected for the ease in which they can be decoded, their practical implementation as they come in a MATLAB built-in package, and because they have been used in previous OFDM system implementations for underwater environments. Although there might be scenarios where the BCH codes are not likely to be the best candidates, they showed good results in the tests conducted at MIT Sailing Pavilion [13]. In the practical implementation, a set of BCH codes with different redundancies and correction capabilities has been tested (Table 6.2 in Section 6.1.4.2).

The code rate of a BCH code is defined as

$$r = \frac{L}{N} \tag{4.2}$$

where L is the length of the original word and N is the length of the codeword.

The ratio that assesses the channel utilization given a BCH code is $N/(N+1)$, where N is the length of the codeword. The system is designed so that the ratio $K/(N+1)$ is an integer, where K is the total number of subcarriers. This ratio also represents the number of subcarriers that are not used, which are equi-spaced and may be used to carry pilots.

In [13], the BCH(63,18) code was selected as the best candidate, since it was the one with minimal redundancy to achieve the targeted decoding performance. For this reason, this code has been used in all the simulations and in-air experiments of the current project. However, the code rate can be increased by including the MPEG-4 Error Resilience Tools at the video encoder (Section 5.2)

Subcarrier mapping

Two types of modulations are used to modulate each of the subcarriers:

- **Phase Shift Keying** conveys data by changing, or modulating, the phase of a reference signal (the carrier wave). Two types of PSK modulations are used in this project: Quadrature PSK (QPSK) and 8-PSK. Note that, if differential detection is performed with QPSK, the mapped points need to be further processed, including a 45° shift of the constellation in order to maintain its correct position (Section 4.4).

- **Quadrature Amplitude Modulation** conveys data by changing the amplitude of the reference signal. The 16-QAM, 32-QAM and 64-QAM are the QAM types that have been implemented in this system.

In practice, the higher the density of points in the constellation, the higher the probability that the symbols are poorly detected in the receiver because of the amplitude scaling in the channel. This is the reason why the performance of the different modulation types was tested in [13], and 8-PSK was chosen as the most appropiate one, as it provided the highest data rate while meeting the detection requirements. Thus, unless otherwise stated, the 8-PSK constellation is the one used in the practical implementation.

Frequency interleaving

In the channel coding, if the number of errors within a codeword exceeds the error-correcting code's capability, it subsequently fails to recover the original word. Frequency interleaving ameliorates this problem by shuffling the different parts of a codeword (in this case, the symbols) across the the K subcarriers, thereby creating a more uniform distribution of errors and reducing the impact of the highly attenuated or distorted frequency bands. The interleaving process consists of writing the symbols in the interleaving matrix by rows, and obtaining the output sequence by reading the matrix by columns, as shown in Figure 4.4.

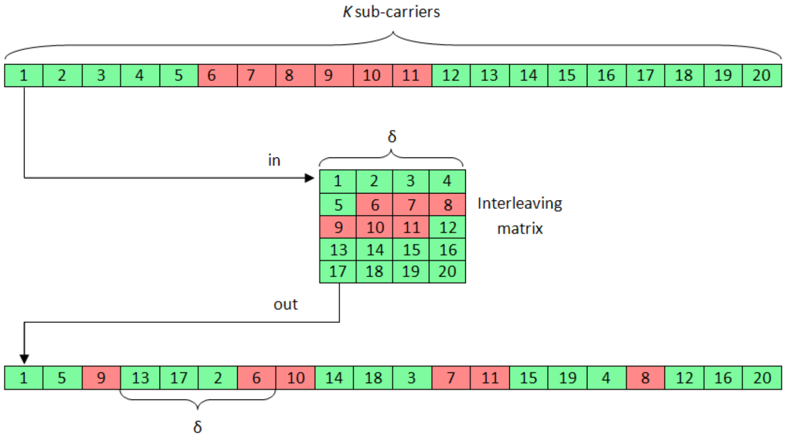


Figure 4.4: Interleaving concept. The subcarriers in red color represent the distorted frequency band.

The interleaving depth of the matrix is defined as the number of symbols in the interleaved sequence that have been placed between two adjacent symbols in the original sequence. It is calculated as

$$\delta = K \cdot \frac{N}{(N+1)\cdot n_d} \tag{4.3}$$

where K is the total number of subcarriers, N is the length of the codeword, and n_d is a factor that indicates a certain number of symbols, corresponding to an integer number of codewords.

IFFT modulation

After interleaving, the Inverse Fast Fourier Transform (IFFT) is applied to the OFDM temporal blocks.

According to the Nyquist sampling theorem, a sampled bandlimited signal can be perfectly reconstructed from an infinite sequence of samples if the sampling rate, f_s, exceeds $2f_h$ samples per second, where f_h is the highest frequency of the passband signal. However, an oversampling ratio of between 3 and 4 is advised.

To account for the oversampling, the IFFT algorithm does not operate with a number of samples K, but rather with a number of samples γK, where γ is the oversampling ratio, and $(\gamma-1)K$ zeros are appended to the K subcarriers' symbols. Thus, the IFFT algorithm applies the following equation to each OFDM block:

$$y(l) = \sum_{k=0}^{N_s-1} \check{d}(k) e^{j2\pi kl/N_s}, \quad l = 0, ..., N_s - 1 \tag{4.4}$$

where $y(l)$ is the resulting sampled signal; N_s is the number of samples during a symbol time, given by Tf_s; and $\breve{d}(k)$ is a vector of N_s samples, which contains the subcarriers' symbols in the first K positions and the $(\gamma - 1)K$ appended zeros.

Symbol interleaving to reduce Peak-to-Average Ratio

The waveform shape is determined by the output of the IFFT applied to the subcarriers' symbols, which in turn depend on the sequence of bits coming out from the FEC coder. Therefore, the shape of the signal can vary considerably from block to block.

The main problem is the existence of peaks in the signal that forces the transmitter to reduce the average power. Thus, the reduction of the signal variation through time is an important issue. Such variation is evaluated by the Peak-to-Average Ratio (PAR), defined as the ratio between the maximal power of the signal and its average power:

$$PAR \equiv \frac{\max_{0 \leq t < T} |y(t)|^2}{\frac{1}{T}\int_0^T |y(t)|^2 \, dt} \tag{4.5}$$

where $y(t)$ is the signal coming out from the IFFT block and T is the symbol duration time. It is also commonly expressed in dB as $10\log(PAR)$.

In this project, four PAR reduction techniques have been included: clipping and filtering, symbol interleaving, in-band tone reservation and out-of-band tone insertion. Depending on the chosen technique, they take place before the IFFT block or after it. The symbol interleaving technique is the one that has been used in the simulations and experiments for this project, since it showed the best performance in [13].

This technique consists of decorrelating the complex points that have a direct impact on the time-domain shape of the signal by reordering the symbols through the K subcarriers before sending them to the IFFT block. The difference between this interleaving process and the previous one is that a one-to-one random rearrangement is performed, without using any interleaving matrix (Figure 4.5).

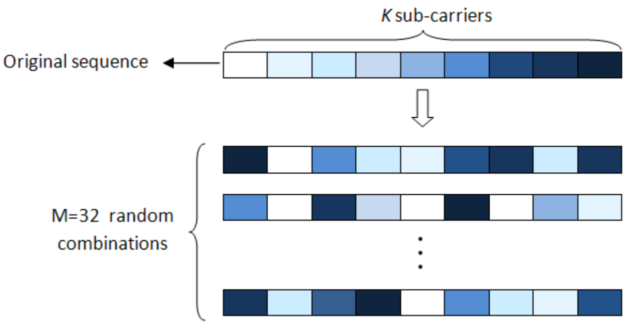


Figure 4.5: Interleaving technique for PAR reduction.

As each OFDM signal is different, it may require a different interleaver to provide the best PAR reduction. This is why M randomly generated combinations are used. The pilot symbols inserted in the signal allow the receiver to infer which interleaver was used by calculating Mean Square Error (MSE) between the pilot symbols retrieved from each interleaver and the transmitted pilot symbols (a parameter that is known by the receiver). The interleaver that provides the lowest MSE is the one that is finally selected.

Synchronization and guard time

In order to correctly locate the OFDM blocks for demodulation, it is extremely important to synchronize the received signal in the time domain.

Synchronization is performed by including a synchronization preamble at the beginning of the signal, which is based on a pseudo-random sequence of bits whose length is 127 bits. It is quadrature modulated using the center frequency of the desired band and its time duration is given by $127/B$ seconds, where B is the total bandwidth. It has good correlation properties: only a main peak is the result of its autocorrelation. In order to resynchronize and compensate for the Doppler effect several times within a single file transmission, the synchronization preamble is repeated periodically in the signal (Figure 4.6).

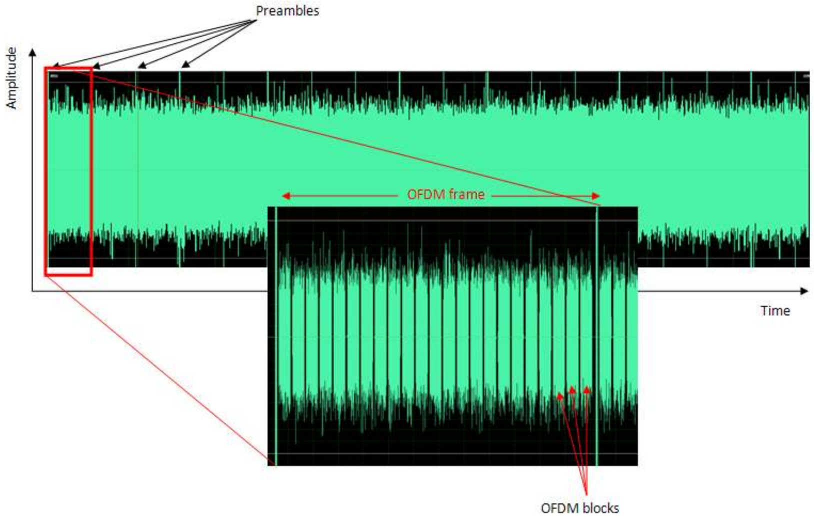


Figure 4.6: Structure of a transmitted signal with $K = 16384$, $T_g = 16$ ms and $B = 115$ kHz.

The other function of this block is to add a guard time, T_g, at the end of each OFDM block. The guard time absorbs the multipath propagation effects, which would cause an overlap adjacent blocks. As typical values for the multipath spread in the underwater channel are around 10 ms, a guard time of 16 ms has been chosen for the current implementation.

At the output of this block, the signal can be modeled as $u(t) = \sum_{k=0}^{K-1} u_k(t)$, where the following apply:

- $u_k(t) = \sum_n d_k e^{jk\triangle\omega_0(t-nT')} g(t - nT')$
- The signal $g(t)$ describes the zero-padding operation as

$$g(t) = \begin{cases} 1, & t \in [0, T] \\ 0, & t \in [T, T_g] \end{cases} \tag{4.6}$$

  where T is the duration of one OFDM symbol, T_g is the guard time and $T' = T + T_g$.
- $\triangle\omega = 2\pi f$, where $\triangle f = 1/T$ is the subcarrier spacing.
- $\omega_k = \omega_0 + \triangle\omega$, $k = 0, ..., K - 1$ are the K subcarriers, where ω_0 is the frequency of the lower subcarrier.
- $d_k(n)$ is the symbol of the k subcarrier at the n OFDM block.

Frequency adjustment

The final step in the transmitter consists of shifting the signal to the desired frequency band in order to avoid the noise present at low frequencies. This is done by multiplying the signal by $e^{j\omega_0 t}$.

The signal coming out of the transmitter can be modeled as

$$s(t) = Re\left\{u(t)e^{j\omega_0 t}\right\} \tag{4.7}$$

4.4 Receiver

This Section contains a detailed description of the elements at the receiver side, whose block diagram is illustrated in Figure 4.7.

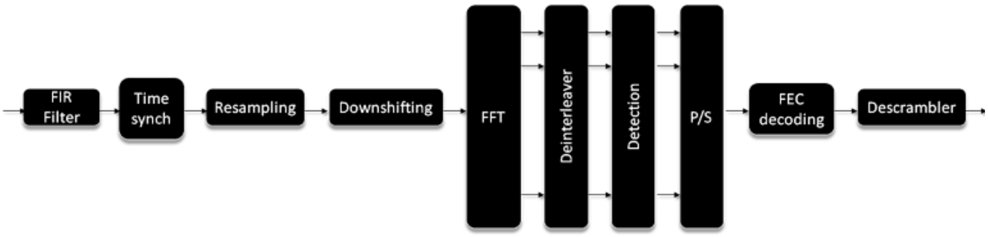


Figure 4.7: Receiver scheme.

FIR filtering

Once the OFDM signal is received, a bandpass FIR filtering is applied in order to minimize the out-of-band noise and interference, so that the signal is clean before applying the synchronization technique.

Time synchronization and resampling algorithm

Synchronization is performed by calculating the cross-correlation between the received signal and the synchronization preamble. As discussed in Section 4.3, the key-property of the selected preamble is that it shows a high peak in its autocorrelation, which enables the receiver to identify with high probability when the synchronization preamble is received. Figure 4.8 shows the transmitted synchronization preamble, along with an example of the received signal and the cross-correlation between them. In the cross-correlation plot, there are three peaks with decreasing amplitudes. These are the reflections caused by the multipath propagation. The position of the peak with the highest amplitude is considered to be the point in which the preamble is located.

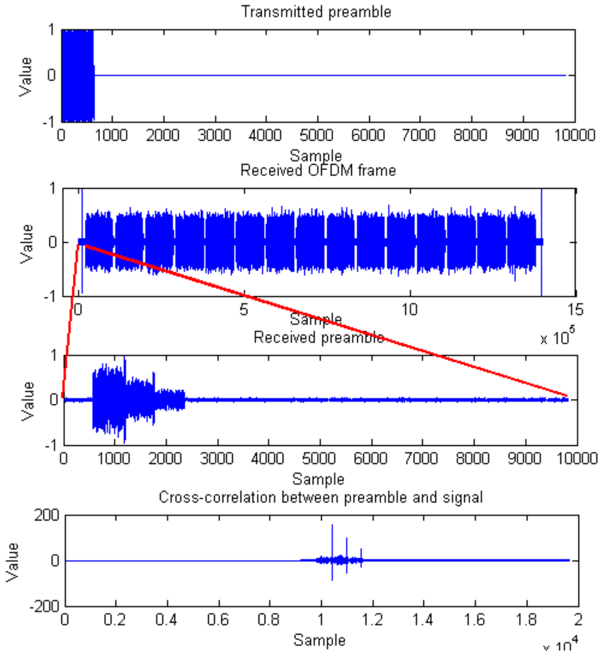


Figure 4.8: Transmitted preamble (first), received OFDM frame (second), zoom of the received preamble (third) and cross-correlation between transmitted preamble and received preamble (fourth).

After synchronization, a Doppler resampling algorithm is applied (Section 5.1). The first step of the algorithm consists of locating the next synchronization preamble by again computing the cross-correlation of the received signal and the preamble. Note that the theoretical time (or the number of samples) between two adjacent preambles, T_{tx}, is known by the receiver. When the preamble is detected, the receiver compares the time between the received preambles, T_{rx}, with the theoretical T_{tx}. If they are not equal, the current OFDM frame is resampled so that it recovers its original length. In the current implementation, the resampling process is performed with the linear interpolation method [31].

After that, the receiver algorithm starts demodulating all the symbols in the current OFDM frame. Every time the receiver detects another preamble, the process is repeated again.

FFT demodulation

Once the received signal has been correctly synchronized, it is downshifted to baseband by multiplying it by $e^{-j\omega_0 t}$. Then, the Fast Fourier Transform (FFT) algorithm is used to retrieve the symbols before channel treatment. The processing applied to each OFDM block is

$$y(k) = \sum_{l=0}^{N_s-1} v(l) e^{-j2\pi kl/N_s}, \quad l = 0, ..., N_s - 1 \tag{4.8}$$

where N_s is the number of samples during a symbol time and $v(l)$ is the received signal in its baseband form.

As mentioned in Section 4.3, the signal is oversampled by a factor γ, which means that the received signal after FFT demodulation includes the K retrieved subcarriers plus $(\gamma - 1)K$ appended zeros. That is why only the first useful K values are chosen and passed to the post-FFT detection algorithms.

Detection process

After the frequency deinterleaving, the detection algorithm retrieves the constellation points. Two types of detectors are included in this system: a differentially coherent detector based on a differential detection among subcarriers, and a coherent detector based on the calculation of phase and channel estimates. Although the differentially coherent detector is the one selected for the practical implementation for its ability to deal with fast motion, both detection methods are explained in depth.

- **Differentially coherent detector**

  Unlike the coherent detector, the differential detector does not rely on the retrieval of phase and channel estimates nor on coherence between adjacent OFDM blocks. Thus, this technique can deal with sudden channel changes or high motion, which could cause a coherent detector to lose track of the channel.

  At the transmitter side, the differentially encoded symbols, $d_k(n)$, are calculated for each block $b_k(n)$, as

  $$d_k(n) = d_{k-1}(n) b_k(n), \quad l = 0, ..., K - 1 \tag{4.9}$$

  where the first subcarrier symbol is initialized as $d_{-1}(n) = 1$. In the practical implementation, a pilot is relocated on that first subcarrier in order to avoid the loss in data rate. Note that only PSK constellations can be used with this type of detector because the applied algorithm does not take into account amplitude differences. When using a QPSK modulation, it is important to take into account that the input symbols, $b_k(n)$, need to be on a + constellation to assure that only four possible complex points are the output of the differential coder. To accomplish this, a 45° shift is applied to the mapped symbols.

  At the receiver side, the received symbols for each subcarrier k in the block n can be represented, in the absence of noise and ICI, as

  $$y_k(n) = H_k(n) d_k(n) e^{j\theta_k(n)} \tag{4.10}$$

  where $H_k(n)$ represents the channel response (a complex point containing the amplitude attenuation and the phase shift) and θ_k represents the residual Doppler shift after the resampling algorithm. In order to extract the mapped symbols, $b_k(n)$, it is assumed that the channel responses, $H_k(n)$, and the phase shifts, θ_k, are similar between adjacent subcarriers:

$$\frac{y_k(n)}{y_{k-1}(n)} = \frac{H_k(n)d_k(n)e^{j\theta_k(n)}}{H_{k-1}(n)d_{k-1(n)}e^{j\theta_{k-1}(n)}} \approx \frac{d_k(n)}{d_{k-1}(n)} = b_k(n) \tag{4.11}$$

When spatial diversity is used (Section 4.4.1), the received signals are combined using maximal-ratio combining (Section 4.4.1).

- **Coherent detector**

This method is based on the *Low complexity OFDM Detector for Underwater Acoustic Channels* [32]. It focuses on low complexity post-FFT signal processing using adaptive channel estimation. Non-uniform Doppler compensation across sub-bands is performed using a single adaptively estimated parameter representing the Doppler rate.

When using spatial diversity with M receiving elements, the received signals for each receiver are stored as an M-element vector, $\mathbf{y}_k(n)$. We assume that a phase estimate $\hat{\theta}_k(n–1)$ and a channel frequency-domain estimate $\hat{\mathbf{H}}_k(n–1)$ (see *Sparse Channel Estimation* on the facing page) were previously calculated for each receiving element and are available at the beginning of the n block. When the receiving elements are closely spaced, the motion-induced frequency offset is assumed to be equal for all the elements. Upon analyzing the received signal $\mathbf{y}_k(n)$, a first data estimate can be calculated using MMSE combining of the different receivers:

$$\hat{d}_{k1}(n) = \hat{\gamma}_k(n-1)\hat{\mathbf{H}}'_k(n–1)\mathbf{y}_k(n)e^{-j\hat{\theta}(n-1)} \tag{4.12}$$

Although the channel may change slowly, thus allowing us to use the past estimate $\hat{\mathbf{H}}_k(n–1)$, the phase could change considerably over a block duration. Thus, using the old estimate $\hat{\theta}_k(n–1)$ may lead to an angular offset between $\hat{d}_{k1}(n)$ and the true value $d_k(n)$. In order to form an updated estimate $\hat{\theta}_k(n)$, which will then be used to make the actual data symbol estimate $d_k(n)$, the angular offset in $\hat{d}_{k1}(n)$ is measured by

$$\triangle\hat{\theta}_k(n) = \left\langle \hat{d}_{k1}(n)d^*_k(n) \right\rangle \tag{4.13}$$

The true data symbols $d_k(n)$ are needed to calculate $\triangle\hat{\theta}_k(n)$. There are two ways to obtain them:

1. By using only the pilot symbols –which are already known by the receiver– to calculate the angular offset $\triangle\hat{\theta}_k(n)$.
2. By making tentative decisions over all the symbols (not only the pilot ones). To do so, the phase $\hat{\theta}(n–1)$ and the Doppler rate $\hat{a}(n–1)$ are used to make a prediction on the phase:

$$\breve{\theta}_k(n) = \hat{\theta}_k(n–1) + \hat{a}(n-1)\omega_k T' \tag{4.14}$$

This prediction is then used to form the estimate $\hat{d}_{k2}(n)$:

$$\hat{d}_{k2}(n) = \hat{\gamma}_k(n-1)\hat{\mathbf{H}}'_k(n–1)\mathbf{y}_k(n)e^{-j\breve{\theta}(n-1)} = \hat{d}_{k1}(n)e^{-j\hat{a}(n-1)\omega_k T'} \tag{4.15}$$

The next step is to calculate the reference for the angular offset measurement, $\overline{d}_k(n)$, given by

$$\overline{d}_k(n) = \begin{cases} d_k(n) & \text{if } k \in \tilde{K}(n) \\ \text{decision}\left[\hat{d}_{k2}(n)\right] & \text{otherwise} \end{cases} \tag{4.16}$$

where $\tilde{K}(n)$ is the pool of pilot subcarriers.

The angular offset, calculated using either of the previous methods, is used to estimate the Doppler coefficient by calculating the mean over the considered channels:

$$\hat{a}(n) = \frac{1}{K}\sum_{k=0}^{K-1}\frac{\triangle\hat{\theta}_k(n)}{\omega_k T'} \tag{4.17}$$

The final phase estimate is calculated:

$$\hat{\theta}_k(n) = \hat{\theta}_k(n–1) + \hat{a}(n)\omega_k T' \tag{4.18}$$

Next, the algorithm estimates the data symbols $\hat{d}_k(n)$:

$$\hat{d}_k(n) = \hat{\gamma}_k(n-1)\hat{\mathbf{H}}'_k(n{-}1)\mathbf{y}_k(n)e^{-j\hat{\theta}(n)} = \hat{d}_{k1}(n)e^{-j\hat{a}(n)\omega_k T'} \tag{4.19}$$

The final symbol decisions, $\tilde{d}_k(n)$, are made by detecting blocks of multiple codewords, so that the data bits can be directly retrieved and some errors corrected (coding-oriented decision):

$$\tilde{d}_k(n) = \begin{cases} d_k(n) & \text{if } k \in \tilde{K}(n) \\ \text{decision}\left[\hat{d}_k(n)\right] & \text{otherwise} \end{cases} \tag{4.20}$$

These symbols are then used to update the channel estimates, $\hat{\mathbf{H}}_k(n)$, and phase estimates, $\hat{\theta}_k(n)$, for the next block. Specifically, the *Sparse Channel Estimation* [32] algorithm is used in the time domain. The first step in this technique is to calculate the frequency-domain channel estimates from the retrieved values, $\mathbf{y}_k(n)$, the phase offsets, $\hat{\theta}(n)$, and the set of data decisions and pilots $\tilde{d}_k(n)$:

$$\mathbf{X}_k(n) = \mathbf{y}_k(n)e^{-j\hat{\theta}(n)}\tilde{d}^*_k(n) \tag{4.21}$$

Then, the IFFT algorithm is applied to get the first time-domain channel estimates $\mathbf{x}_l(n)$ from $\mathbf{X}_k(n)$:

$$\mathbf{x}_l(n) = \frac{1}{K}\sum_{k=0}^{K-1}\mathbf{X}_k(n)e^{j2\pi kl/K}, \quad k = 0, ..., K-1 \tag{4.22}$$

Sparsing is applied by truncating the time-domain channel coefficients. Out of the total number of coefficients, the ones kept are those whose magnitude is above some selected threshold α_c, while the rest are set to zero. The new time-domain channel response, $\tilde{\mathbf{x}}_l(n)$, is used to calculate the time-domain channel estimates by applying the following algorithm:

$$\hat{\mathbf{h}}_l(n+1) = \lambda\hat{\mathbf{h}}_l(n) + (1-\lambda)\tilde{\mathbf{x}}_l(n) \tag{4.23}$$

The FFT algorithm is applied to get the final frequency-domain channel estimates used in the next block:

$$\hat{\mathbf{H}}_k(n) = \sum_{l=0}^{K-1}\hat{\mathbf{h}}_l(n)e^{-j2\pi kl/K}, \quad l = 0, ..., K-1 \tag{4.24}$$

Finally, the normalization parameters $\hat{\gamma}_k(n)$ from the received values $\mathbf{y}_k(n)$ are assessed. A high SNR regime is supposed, so the noise variance σ_z^2 is omitted:

$$\hat{\gamma}_k(n) = \left(\sigma_z^2 + \hat{\mathbf{H}}'{}_k(n)\hat{\mathbf{H}}_k(n)\right)^{-1} \approx \left(\hat{\mathbf{H}}'{}_k(n)\hat{\mathbf{H}}_k(n)\right)^{-1} \tag{4.25}$$

The algorithm is initialized using $\hat{\mathbf{H}}_k(1) = \mathbf{y}_k(1)d^*_k(1)$, $\hat{a}(1) = 0$ and $\hat{\theta}_k(1) = 0$. For a correct performance, the receiver must know the transmitted symbols $d_k(1)$ to accurately initialize the channel estimates, so a training sequence is transmitted in the first OFDM block.

After the detection process, the symbols are transformed into their corresponding bits, and the FEC decoding and descrambling algorithms are applied.

Performance analysis

One of the parameters that has been used to quantify the performance of the system is the Mean Square Error (MSE), which indicates the distance from the detected symbols (before a decision is made by the detector) to the transmitter complex points in the constellation. The MSE over the K subcarriers (frequency domain) and MSE over the N received OFDM blocks (time domain) are evaluated:

$$MSE_{time}(n) = \frac{1}{K}\sum_{k=0}^{K-1}\left|d_k(n) - \hat{d}_k(n)\right|^2 \tag{4.26}$$

$$MSE_{frequency}(k) = \frac{1}{N}\sum_{n=1}^{N}\left|d_k(n) - \hat{d}_k(n)\right|^2 \tag{4.27}$$

The other parameters used to evaluate the system performance are the Bit Error Rate (BER) and the Symbol Error Rate (SER). The BER is calculated as the total number of wrong bits divided by the number of transmitted bits. In order to have an indication of whether the transmitted sequence is correctly retrieved, it is assessed for both the coded sequence (before the FEC decoding is applied) and for the decoded sequence.

The SER is defined as the total number of wrong symbols (before the FEC decoding is performed) divided by the total number of symbols in the transmission. Like the MSE, this parameter is both assessed over the K subcarriers (frequency domain) and the N received OFDM blocks (time domain).

In the experiments conducted in [13], the system required a BER below 10^{-3} to obtain a qualitatively acceptable detection. The MSE achieved excellent performance when it was below -5 dB; otherwise the results varied depending on the FEC coding.

4.4.1 Spatial diversity

It is well-known that each of the signal bounces due to multipath effects of an underwater channel can introduce phase shifts, time delays, attenuation and distortion that can destructively interfere with another one at the receiver. The use of multiple receivers, known as *spatial diversity*, is especially effective at mitigating these effects. Since multiple receivers offer several observations of the same signal, each receiver will experience a different interference environment. Thus, if one receiver is experiencing a deep fade, it is possible that another has a sufficient signal. Collectively such a system can provide a robust link.

There are four ways to combine the received signals:

- **Switching**

  The signal from only one receiver is used for as long as the quality of that signal remains above some desired threshold. If the signal degrades, another receiver is used. This method is the easiest and least power-consuming type of diversity processing techniques, although some periods of fading and desynchronization may occur while the quality of one receiver degrades and another receiver link is established.

- **Selecting**

  This method also uses only one receiver's signal at any given time. The receiver chosen, however, is based on the best Signal-to-Noise Ratio among the received signals. This requires a pre-measurement to take place and, thus, all the receivers to have established connections beforehand, resulting in a higher power requirement.

- **Combining**

  In this scheme, all receivers maintain established connections at all times. The signals are combined and presented to the detector. Depending on the configuration of the system, the signals can be added directly (equal gain combining) or weighted and added coherently (maximal-ratio combining). This system provides the greatest resistance to fading but, since all the received paths must remain energized, it is also the most power-consuming one.

- **Dynamic Control**

  The systems using this technique are capable of choosing from the above processing schemes the most suitable one for each specific scenario. Although they are much more complex, they optimize the trade-off between the consumed power and the performance quality. Transitions between modes and/or receiver connections are signaled by a change in the perceived quality of the link. In situations of low fading, the system employs no diversity and uses the signal presented by a single receiver. As conditions degrade, the system can then use the more reliable but power-consuming modes described above.

For its simplicity, the combining scheme has been implemented in the current system (see the different types of detection methods, Section 4.4). In Section 6.2.3, the results of several experiments using spatial diversity are discussed.

4.5 Final system overview

The structure of the final system can be modeled by the scheme illustrated in Figure 4.9.

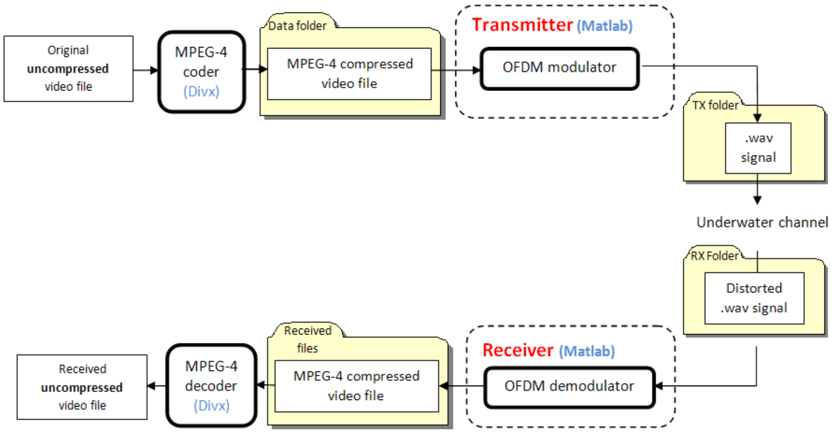


Figure 4.9: Off-line processing code scheme.

The modulation and demodulation algorithms have been developed in MATLAB due to the advantages that this application provides, such us the debugging of the program, the tuning of the chosen parameters and its good graphics interface. For the video coder and decoder software, the Divx 5.0.2 codec, altogether with the VirtualDub video editing software [33], have been used to compress the videos according to the MPEG-4 standard specifications.

Chapter 5

Contributions

The previous chapter focused on the basic structure of the modulator and demodulator algorithms, which were designed and analyzed in [13]. In this thesis, some parts of those algorithms have been refined and updated. However, the most important contributions to the system are the implementation of an algorithm that compensates for the Doppler effect and the inclusion of a set of error resilience tools at the video encoder.

5.1 Doppler compensation algorithm

5.1.1 Mathematical analysis

The Doppler effect is defined as the change in the frequency (or period) of a signal when there is a relative speed between the transmitter and receiver. When they get closer to each other, each successive wave crest is emitted from a closer position than the previous wave. Therefore, each wave takes slightly less time to reach the receiver than the previous wave. Thus, the time between the arrival of successive wave crests at the receiver is reduced, causing an increase in the frequency. While there is motion between the transmitter and the receiver, the distance between successive wavefronts is also reduced, so the waves are “compressed”.

Conversely, if the transmitter and/or the receiver are moving away from each other, each wave is emitted from a farther position than the previous wave. Therefore, the arrival time between successive waves is increased, thus reducing the frequency. The distance between successive wavefronts is increased, and the waves are “dilated” (Figure 5.1).

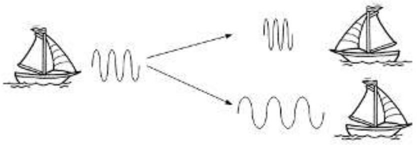


Figure 5.1: Doppler effects on the signal when the transmitter and the receiver are getting closer to each other (top) and when they are getting farther from each other (bottom).

The Doppler effect is modeled by introducing the Doppler scaling factor, $a = v_r/c$, in the multipath underwater channel response, as

$$c(\tau, t) = \sum_p A_p(t)\delta(\tau - \tau_p + a_p t) \tag{5.1}$$

where $A_p(t)$ are the paths’ amplitudes, τ_p are the paths’ delays and a_p are the Doppler factors of each path. During the development of the algorithm, we have adopted the following assumptions:

- All the paths have a similar Doppler factor: $a_p \approx a$.

- The nominal Doppler factor a, the path delays τ_p and the path taps A_p are nearly constant over the OFDM block duration $T' = T + T_g$.

For our specific OFDM system, the transmitted signal in passband for one OFDM block is modeled as

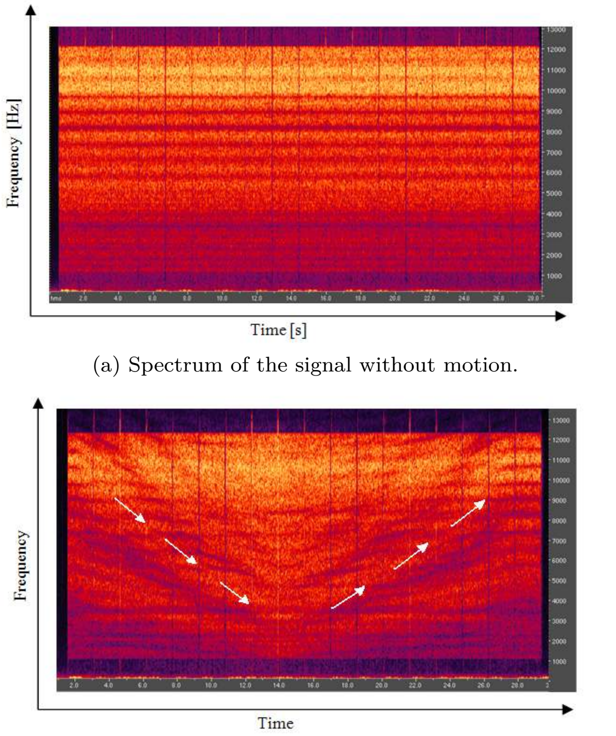
$$s(t) = Re\left\{\sum_k d_k e^{j\omega_k t} g(t)\right\},\ k = 0...K-1 \tag{5.2}$$

Once the signal has passed through the underwater channel, the received signal in passband is given by

$$r(t) = \sum_p A_p s(t - \tau_p + a_p t) = Re\left\{\sum_k d_k e^{j\omega_k (t+at-\tau_p)} \left[\sum_p A_p g(t + at - \tau_p)\right]\right\} + n(t) \tag{5.3}$$

where $n(t)$ denotes additive noise. Note that the received signal experiences two effects:

1. The signal is scaled in time, from T to $T(1+a)$, causing possible desynchronizations by the receiver.
2. Each subcarrier experiences a frequency shift $e^{ja\omega_k t}$ (Figure 5.2), which depends on the frequency value of the subcarrier. These shifts cause strong ICI if an effective Doppler compensation scheme is not performed prior to OFDM demodulation.



(a) Spectrum of the signal without motion.

(b) Spectrum with motion. During the first 14 seconds of the signal, the transmitter and the receiver are getting closer to each other; from then on, they are getting farther from each other.

Figure 5.2: Spectrum of the received signal as a function of time, showing the frequency shifts on the subcarriers.

5.1.2 Algorithm

In order to compensate for the Doppler effects, we have implemented an algorithm that estimates the Doppler coefficient and resamples the received signal according to this value. It relies on the preambles of the transmitted signal, located before and after an arbitrary number of OFDM blocks (one OFDM frame), as previously shown in Figure 4.6 (page 47).

The algorithm has two stages:

1. For each OFDM frame, and via synchronization with two adjacent preambles, the receiver estimates the time duration of one packet as T_{rx}. The time duration of this packet at the transmitter side is T_{tx}, which is a known value at the receiver. By comparing T_{rx} with T_{tx}, the receiver infers how much the received signal has been compressed or dilated by the channel:

$$T_{rx} = T_{tx}(1+b) \qquad \Rightarrow \qquad b = \frac{T_{rx}}{T_{tx}} - 1 \tag{5.4}$$

   This requires a nearly constant moving speed between adjacent preambles. If a sudden speed change is applied to the transmitter or receiver, this algorithm will not be able to track the variations of the Doppler factor that may occur in a single OFDM frame. Instead, the estimated Doppler factor will be the average along this frame, degrading the performance of the receiver. However, this should not be an issue for typical AUV motion during inspection procedures, which are slow and steady.

2. The signal is resampled by the estimated factor $1+b$:

$$\tilde{z}(t) = \tilde{y}\left(\frac{t}{1+b}\right) = Re\left\{\sum_k d_k e^{j\omega_k\left(t\frac{1+a}{1+b}-\tau_p\right)}\left[\sum_p A_p g\left(\frac{1+a}{1+b}t-\tau_p\right)\right]\right\} + \tilde{n}(t) \tag{5.5}$$

The goal is to make b as close as possible to a. If they are not exactly equal, there will remain a residual shift in each subcarrier, denoted by $\epsilon = \frac{a-b}{1+b}$:

$$\tilde{z}(t) = Re\left\{\sum_k d_k e^{j\omega_k(1+\epsilon)t}\left[\sum_p A_p g((1+\epsilon)t-\tau_p)e^{-j\omega_k\cdot\tau_p}\right]\right\} + \tilde{n}(t) \tag{5.6}$$

Since the residual frequency shift is similar between adjacent subcarriers, it will not affect the differentially coherent detector.

A couple of remarks must be made about (1) the point in which the Doppler compensation algorithm is applied and (2) the time between preambles.

Regarding (1), the Doppler compensation algorithm is applied after converting the signal to baseband. However, it could also be applied before downshifting the signal by just taking into account the frequency offset $e^{j\omega_0 t}$, without any difference in the performance.

As for (2), it is important to take into account the fact that the time between preambles needs to be short enough to avoid missing any strong variation of the Doppler coefficient in one OFDM frame. However, the minimum time between preambles is determined by

$$\frac{1}{f_s} < \min\left|\frac{v_r}{c_{water}}\right| \cdot T_{frame} = \min|a| \cdot T_{frame} \tag{5.7}$$

where T_{frame} denotes the time between adjacent preambles, f_s is the sampling frequency, v_r is the relative speed between the transmitter and the receiver and $c_{water} = 1,500$ m/s is the speed of sound waves in water. This expression can be considered as the resolution of the system when measuring the Doppler factor, i.e., the algorithm will not be able to detect any variation of the Doppler coefficient smaller than $\frac{1/f_s}{T_{frame}}$.

5.2 MPEG-4 Error-resilient coding

The underwater acoustic channel is typically noisy and suffers from a number of channel degradations such as bit errors and burst errors, whose effects on transmission of a compressed video can be very severe. Moreover, predictive coding techniques such as motion compensation applied by the MPEG-4 standard make matters worse by quickly propagating the effects of the channel across the video sequence and rapidly degrading the video quality. This renders the video unusable unless the video encoder and decoder take appropiate remedial steps.

In order to minimize the impact of transmission errors and increase the portion of corrupted bit streams that can be correctly decoded, a set of error-robust coding techniques provided by the MPEG-4 standard has been included in the video coder and decoder of the current system. These techniques are integrated with the core encoding algorithm and have minimal impact on the coding efficiency. The complexity introduced by these tools is minimal and they perform a trade-off between robustness and efficiency. As a consequence, they are meant to be used together with, not in replacement of, traditional technologies, such as channel coding, which are outside the scope of the MPEG-4 standard. In this scenario, the combined use of external channel coding techniques and error resilience tools reduces the overhead while maintaining the same error resilience efficacy. Such overhead, introduced by the heavy use of channel coding techniques, is reduced by allowing these techniques to leave residual errors in the data stream. The error resilience efficacy is maintained by letting the error resilience tools take care of such residual errors. This is actually the goal of the error resilience tools in this system: to reduce the redundancy added to the data when applying the channel coding at the OFDM modulator.

The Packet-based Resynchronization and Data Partitioning tools are the specific ones that have been included in this system.

Packet-based Resynchronization

When using Variable Length Coding (VLC), the length of the codeword is implicit, and transmission errors typically lead to an incorrect number of bits being used when decoding. This causes the video decoder to lose synchronization with the encoder, causing some of the bits following the corrupted ones to be erroneously decoded, degrading the quality of the decoded video data.

This resynchronization tool is aimed to reestablish the synchronization between the decoder and the bit stream after an error has been detected. This is achieved by means of resynchronization markers, implemented as unique codes (i.e., a sequence of bits that can not be emulated by any code used by the encoder). At the encoder side, these markers are inserted into the bit stream prior to transmission. When an error is detected at the decoder side, the decoder searches for the next resynchronization marker. Once it is found, the synchronization is reestablished, allowing the correct decoding of the remaining bits. This process is illustrated in Figure 5.3. The data between the resynchronization marker prior to the detected error and the newly found resynchronization marker is generally discarded, but some of this data may also be used.

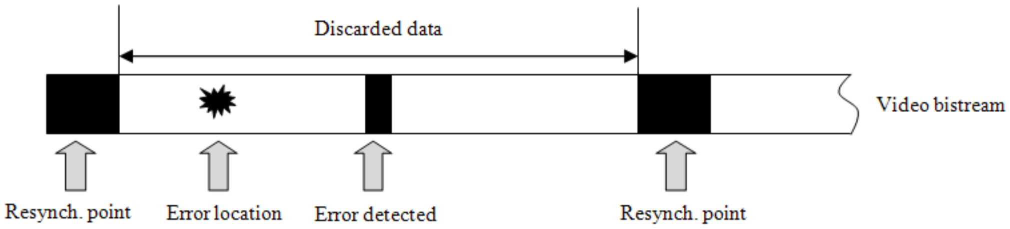
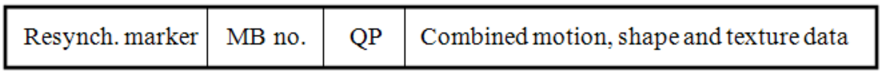


Figure 5.3: Error detection procedure when using Packet-based Resynchronization.

This resynchronization mode is usually combined with the approach of dividing each VOP into independent resynchronization packets in order to avoid error propagation from one packet to the other. Each video packet consists of an integer number of consecutive coded macroblocks preceded by a video packet header, which contains the resynchronization marker (Figure 5.4).



Resynch. marker	MB no.	QP	Combined motion, shape and texture data

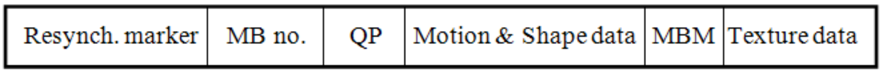
Figure 5.4: Organization of the data within a video packet when using Packet-based Resynchronization, without Data Partitioning.

In addition to inserting resynchronization markers at the beginning of each video packet, the encoder removes all the dependencies that exist between data belonging to two consecutive video packets. This assures that, even if one of the video packets is corrupted by transmission errors, all of the remaining packets can be decoded. To remove these dependencies, the encoder adds two additional fields to the video packet header: the absolute macroblock number of the first macroblock in the video packet (MB no.), which indicates the spatial position of the macroblock in the VOP; and the quantization step for the same first macroblock in the packet (QP). These additional fields are referred to as the video packet header.

Data Partitioning (DP)

After detecting an error in the bit stream and resynchronizing to the next resynchronization marker, the decoder has isolated the erroneous data between the two resynchronization markers. Typical video decoders discard all these erroneous macroblocks. One of the main reasons for this is that between two resynchronization markers, the motion and texture data for each macroblock are coded together. Hence, when the decoder detects an error, regardless of whether the error occurred in the motion or texture part, all the data in the video needs to be discarded.

To deal with this problem, MPEG-4 allows the user to combine Packet-based Resynchronization with the Data partitioning tool, which divides and rearranges the elements in each video packet into two groups, according to their importance in the decoding process. The most important syntax elements are the motion and shape information, which are placed in the first partition of the packet (Figure 5.5). The second partition contains the texture information, which is considered to be less important for the decoding algorithm. These partitions are separated by a Motion Boundary Marker (MBM). If the first partition is recovered, it is possible for the decoder to reconstruct the whole packet, even if the second partition is damaged or lost due to transmission errors.



Resynch. marker	MB no.	QP	Motion & Shape data	MBM	Texture data

Figure 5.5: Organization of the data within a video packet when combining Packet-based resynchronization and Data Partitioning.

Three types of situations can be identified when decoding the packet:

- If an error is detected in the motion Section, the decoder replaces all the macroblocks in the current packet with skipped blocks until the next resynchronization marker is successfully read.
- If the error is detected in the texture information (and no errors are detected in the motion Section), only the texture part is discarded and the motion vectors are used to perform motion compensation.
- If there are no detected errors in the motion or texture Section but the resynchronization marker is not found at the end of decoding all the macroblocks of the current packet, only the texture part of all the macroblocks needs to be discarded. Motion compensation is still applied because the decoder has a higher confidence in the motion vectors since it got the MBM.

In summary, the advantages of these two error resilience tools are twofold. First, we have a more stringent check on the validity of the motion data, because the decoder needs to get the MBM in order to consider the motion data to be valid. In case there are no errors in the motion or texture parts but the decoder does not end on the correct position for the next resynchronization marker, the motion data can be salvaged as validated by the detection of the MBM.

Chapter 6

Experimental results

This chapter contains the results of the simulations and in-air experiments designed to validate the correct performance of the developed system under a variety of situations. In this thesis, only the performance of the Doppler compensation algorithm and the MPEG-4 error resilience tools is discussed, while the basic simulations and underwater experiments that lead to the general design of the system can be found in [13].

6.1 Simulation tests

The first experimental tests correspond to a set of software simulations aimed to provide the preliminary results for checking the system performance under different constraints.

The feasibility of real-time transmission in terms of video bit rate is analyzed in the first simulation. Next, different motion scenarios are simulated in order to test the designed Doppler resampling algorithm. Finally, the performance of the MPEG-4 error resilience tools is analyzed.

6.1.1 Simulated channel

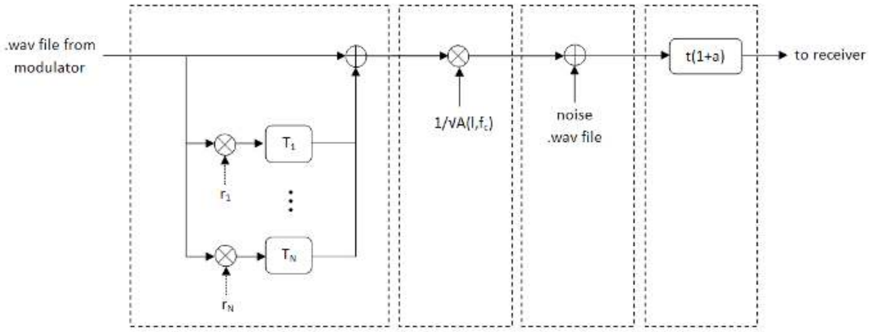


Figure 6.1: Block diagram of the channel simulator.

The software application that simulates the underwater channel (Figure 6.1) has been implemented in MATLAB and has four stages:

- The **multipath** propagation is simulated by adding the generated OFDM signal to itself, shifting it in time and scaling it by a given reflection coefficient. Three arrivals have been considered: at 0 ms (direct path), 1 ms and 2 ms, with reflection coefficients of 1, 0.6 and 0.3, respectively. This geometry corresponds to an example of an underwater channel response.
- The **attenuation** is then applied to the resulting signal. The $A(l, f_c)$ coefficient is calculated using equation 3.1, where f_c is the center frequency of the useful bandwidth. In all the simulations, a distance of $l = 200$ m has been considered, as it was the range that provided the best results in [13].

- A sequence of Additive White Gaussian (AWG) **noise** samples is added to the signal. The average power of the noise is adjusted to meet the SNR value introduced by the user. Since the experiments conducted in [13] showed values around 15 dB and 30 dB, a SNR of 15 dB has been considered in all the simulations.

- The signal is finally **scaled** in time as follows:

$$r'(t) = r\left(t(1+a)\right) \tag{6.1}$$

  where a is the Doppler factor introduced by the user.

6.1.2 MPEG-4 compression capabilities

The performance of the MPEG-4 standard in terms of video bit rate compression has been tested by compressing a video from National Oceanic and Atmospheric Administration's (NOAA) website containing a ROV mission during the discovery of the Titanic shipwreck. Some frames of this video are shown in Figure 6.2.

Figure 6.2: Selected frames of the Titanic video used to test the performance of the MPEG-4 codec.

Table 6.1 shows a comparison of the different parameters for the uncompressed and compressed videos.

	Uncompressed	Compressed
Duration	20s	20s
bit rate	108,007 kbps	64 kbps
Resolution	480×320 pixels	176×144 pixels (QCIF)
Frame rate	30 fps	30 fps
Total number of frames	600	600
Bits / video frame	3600 kbits	2133 bits

Table 6.1: Video compression parameters.

The modulation parameters are detailed below.

- Number of subcarriers: $K = 16384$
- Bandwidth: $40 - 155kHz$ $(B = 115kHz)$
- OFDM block duration: $T = 1/\triangle f = 142.6$ ms
- Channel coding: BCH(63,18) $(1/r = 3.5)$
- Subcarrier modulation: 8-PSK (3 bits/symbol)
- Guard time: $T_g = 16$ ms

According to these parameters, the useful data bit rate is calculated as

$$\begin{aligned}\text{data rate} &= r \cdot \#\text{symbols/second}\cdot\#\text{bits/symbol} = r \cdot \frac{K}{T+T_g} \cdot \#\text{bits/symbol} = \\ &= r \cdot \frac{K}{\frac{K}{B}+T_g} \cdot \#\text{bits/symbol} = \frac{103.39 \cdot 3}{63/18} = 88.6\,\text{kbps}\end{aligned} \tag{6.2}$$

Thus, the real-time frame rate of the modulated video is $\frac{88.6\,\text{kbps}}{2133\,\text{bits/frame}} = 41.5$ fps. Since this value is higher than the frame rate of the uncompressed video, 30 fps, a real-time transmission is theoretically affordable, assuming that the signal processing can happen real-time.

6.1.3 Doppler resampling algorithm

The performance of the developed resampling algorithm is tested by a set of four simulations. The OFDM modulation parameters are shown below.

- Sampling frequency: $f_s = 575$ kHz
- Number of subcarriers: $K = 16384$
- Bandwidth: $40 - 155kHz$ $(B = 115kHz)$

- Subcarrier spacing: $\triangle f = B/K = 7$ Hz
- OFDM block duration: $T = 1/\triangle f = 142.6$ ms
- Channel coding: BCH(63,18) $(1/r = 3.5)$
- Subcarrier modulation: 8-PSK
- Time between preambles: $T_{frame} = 2.5$ s
- Guard time: $T_g = 16$ ms
- Detection method: differentially coherent detection
- Number of receiving elements: 1 receiver

The selected frequency bandwidth, sampling frequency and number of subcarriers are the same as in the underwater experiments conducted in [13].

Initially, the upper bound thresholds for the BER and the MSE will be the ones set in [13] (BER$_{max} = 10^{-3}$ and MSE$_{max} = -5$ dB). However, the results of the in-air experiments will prove that these requirements can be relaxed.

It is worth to mention that further experiments will likely need to use a bit more conservative values, especially for the number of subcarriers and the available bandwidth.

Simulation 1: ideal case

In order to have a reference of what can be expected in the "ideal" case, a channel with no Doppler effect ($a = 0$) has been simulated (Figure 6.3). The system performance is illustrated in Figure 6.4.

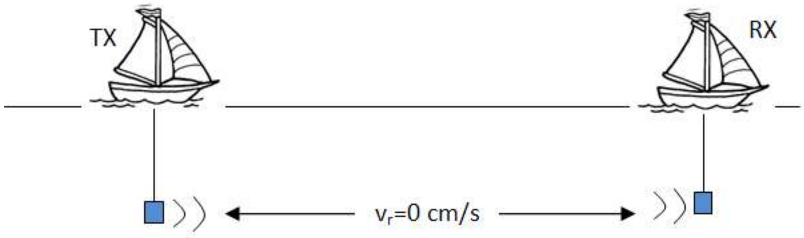


Figure 6.3: Scenario under calm conditions.

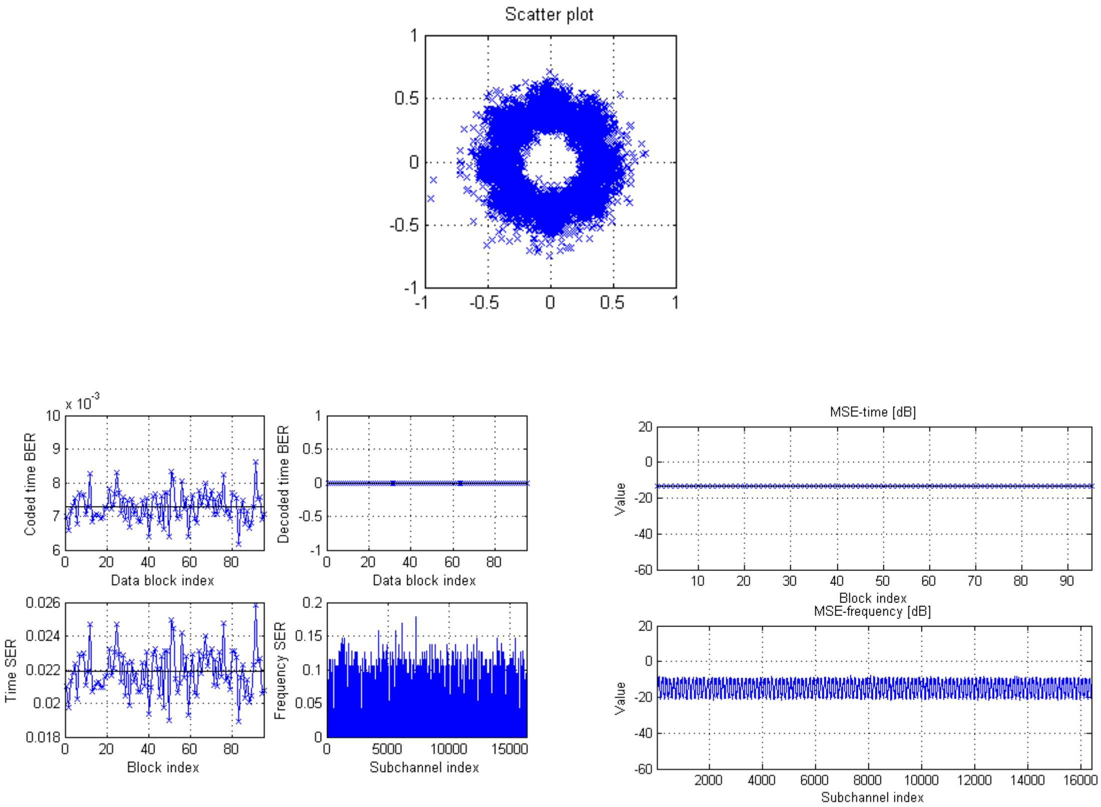


Figure 6.4: System performance in the ideal case.

As expected, the average MSE and the average BER for the decoded sequence, indicated with black lines, are far below -5 dB and 10^{-3}, respectively. Thus, the received video could be perfectly decoded with no distortion.

Simulation 2: uniform speed

This simulation evaluates the system performance when the signal has been dilated by a constant Doppler factor of $a = 3 \cdot 10^{-4}$, which corresponds to a relative speed between the transmitter and the receiver of $v_r = a \cdot c_w = 47$ cm/s, which is equivalent to 1 knot (Figure 6.5).

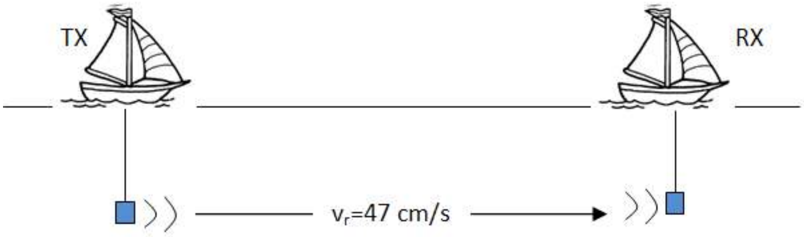


Figure 6.5: Situation recreated with uniform speed.

Figure 6.6 illustrates the performance results when the Doppler compensation algorithm has **not** been applied. Note that the average BER for the decoded sequence is around 0.5, which indicates that around 50% of the total bits have been wrongly detected. Hence, the received video was not decodable.

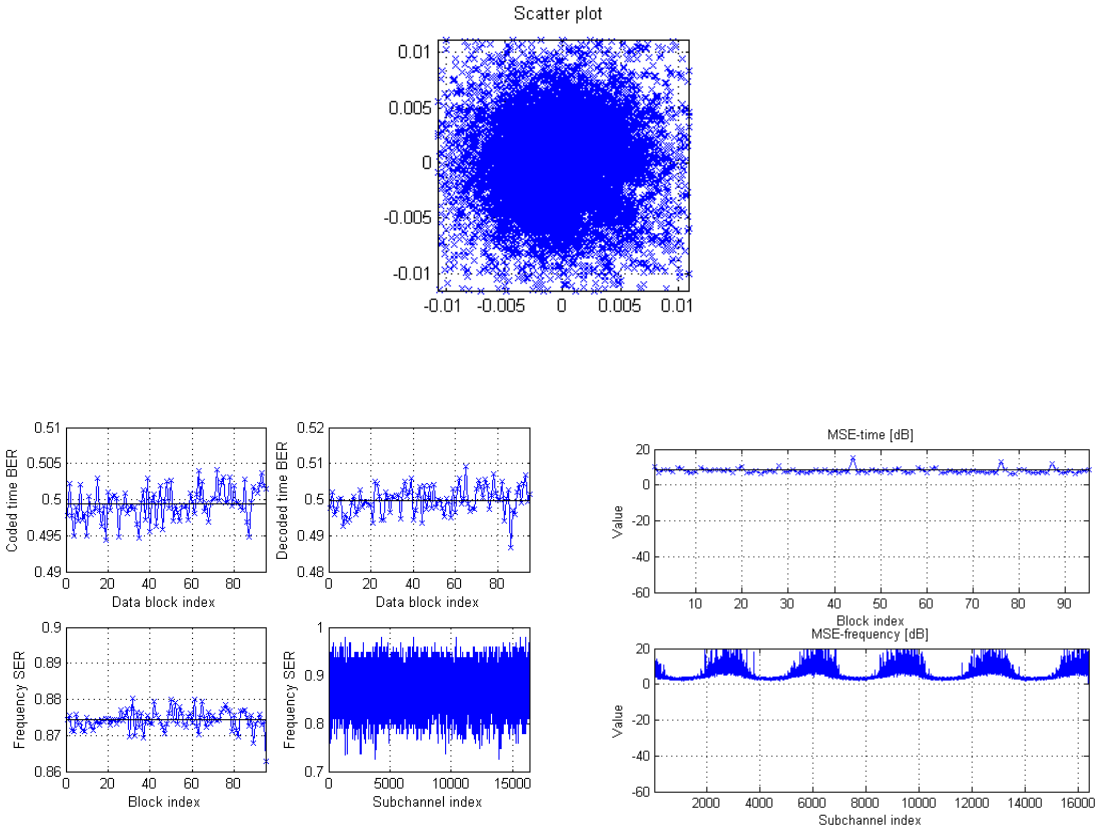


Figure 6.6: System performance for a constant Doppler factor $a = 3 \cdot 10^{-4}$ **without** Doppler compensation.

Figure 6.7 illustrates the performance results when the Doppler compensation algorithm has been applied. The system provides excellent performance in terms of BER and MSE, which show about the same performance as in the ideal case. As expected, the received video could be perfectly decoded with no distortion. However, this simulation does not correspond to a realistic situation, in which the Doppler coefficient may vary slightly along the transmission.

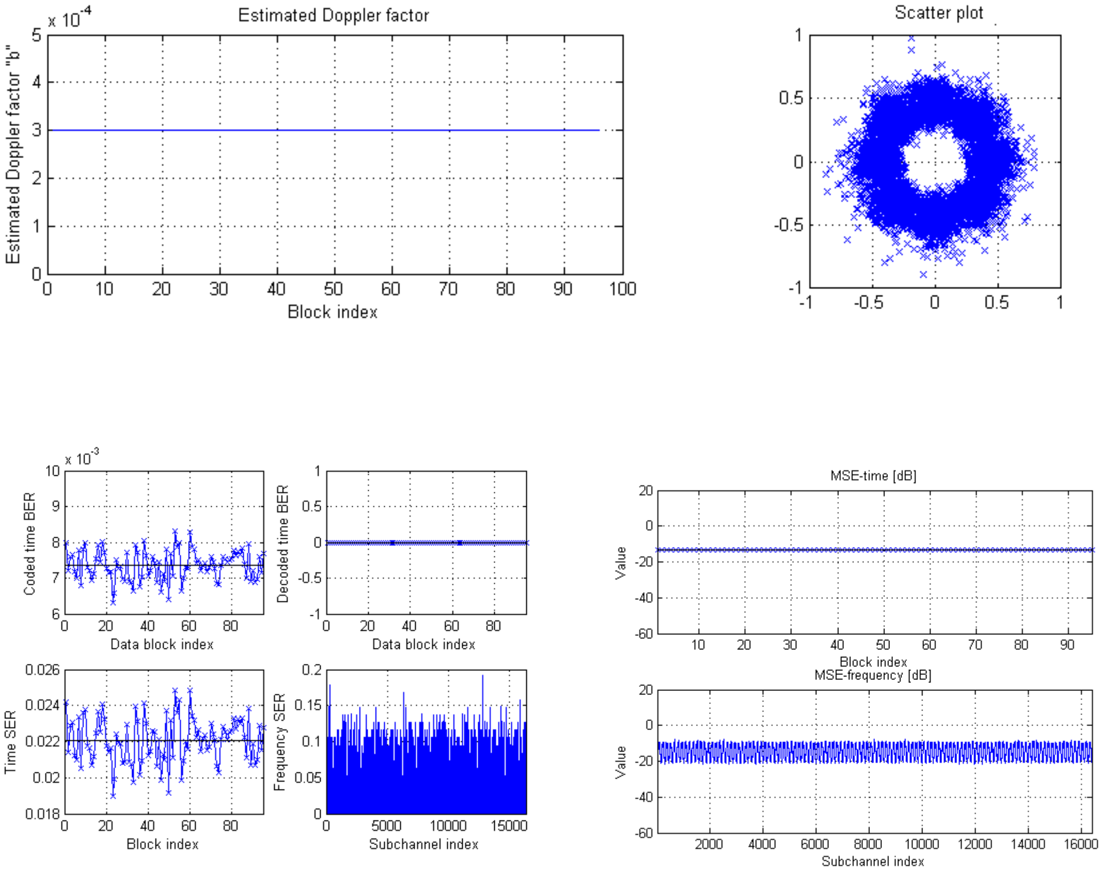


Figure 6.7: System performance for a constant Doppler factor $a = 3 \cdot 10^{-4}$ **with** Doppler compensation.

Simulation 3: slight acceleration

This case corresponds to a more realistic situation in which there is a slight acceleration between the transmitter and the receiver along the signal transmission, which lasts 16 seconds. The value of the Doppler coefficient at the beginning of the transmission is $a = 3 \cdot 10^{-4}$, and it shifts along the signal until reaching $a = 4 \cdot 10^{-4}$ at the end of the transmission, which corresponds to $v_r = 60$ cm/s.

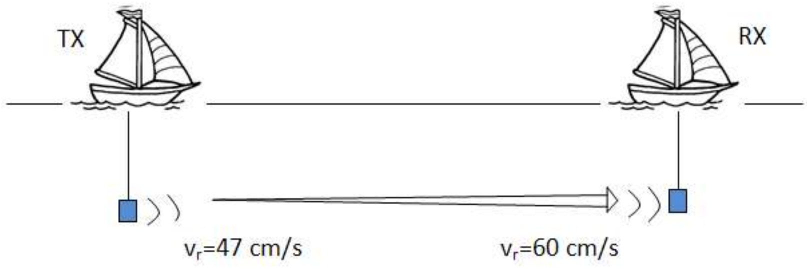


Figure 6.8: Situation recreated with a slight acceleration.

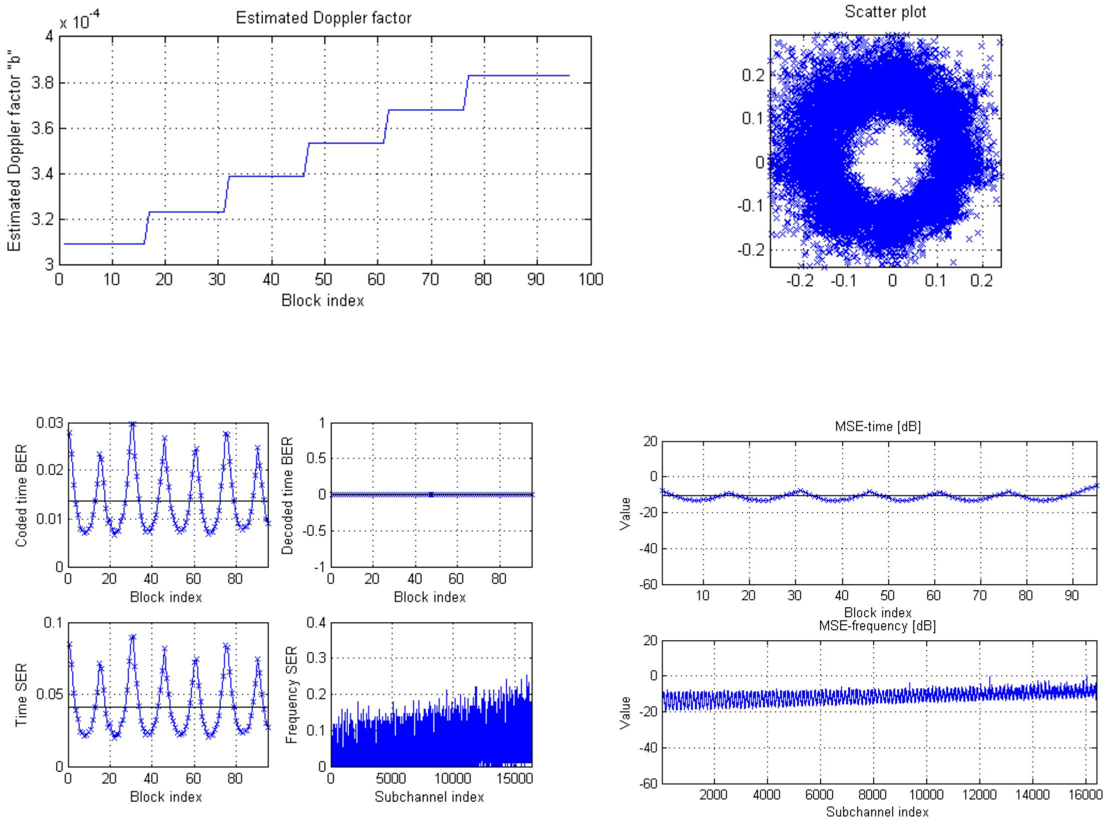


Figure 6.9: System performance when there is a slight variation in the speed.

Since the Doppler factor does not vary drastically between adjacent preambles, the algorithm is able to detect its variations along the signal. Again, the average values for the BER and MSE remain below the targeted thresholds, thus allowing the video decoder to decode the video with no distortion. It must be noted that the increasing trend of the frequency-SER and frequency-MSE curves for higher frequencies is due to the linear frequency-dependence of the Doppler frequency shift at each subcarrier: $a \cdot f_k$ (Section 5.1).

Simulation 4: sudden change in speed

Let us study now a case in which there is a sudden change in the relative speed between the transmitter and the receiver (Figure 6.10). As in the previous simulation, the Doppler coefficient shifts from $a = 3 \cdot 10^{-4}$ to $a = 4 \cdot 10^{-4}$, but in this case the change occurs in just one OFDM frame (block indices 31-45).

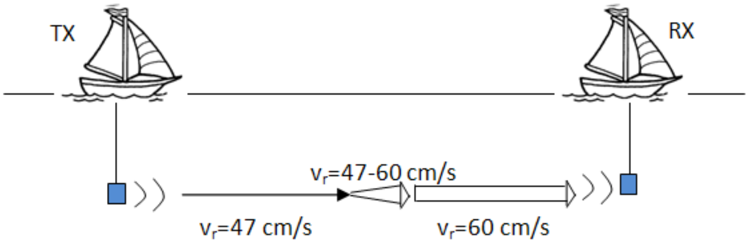


Figure 6.10: Situation recreated with a sudden change in speed.

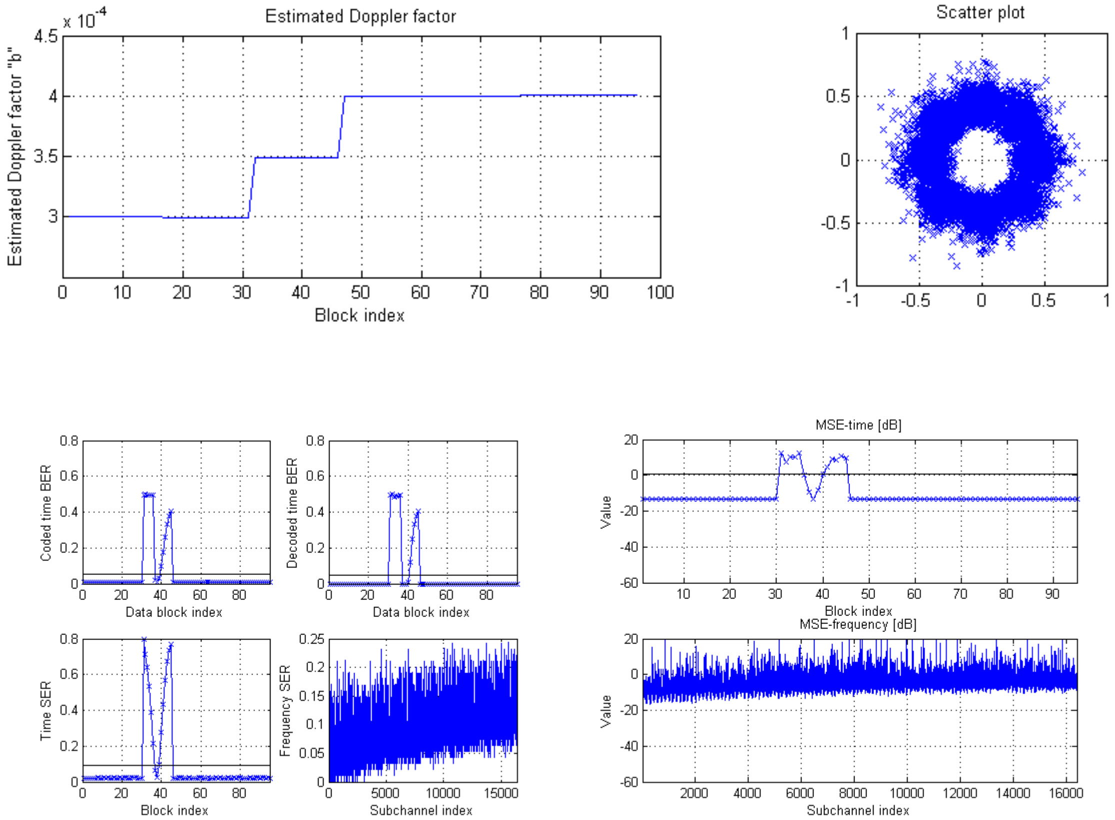


Figure 6.11: System performance in the case of a sudden change in speed.

In the frame where the speed change occurs, the BER for the decoded sequence increases to 0.5 and the MSE increases to 10 dB. This is due to the fact that the algorithm rescales the signal by an **average** Doppler factor calculated between two adjacent preambles ($b = 3.5 \cdot 10^{-4}$) and it is not able to track the speed changes that may occur in between. Notice also that in the middle of the affected OFDM frame, the BER decreases to 0 and the MSE decreases to -13 dB. This is because, at this block, the actual Doppler factor reaches the value $a = 3.5 \cdot 10^{-4}$, which coincides with the average Doppler factor estimated by the algorithm.

The data that was damaged during the affected OFDM frame lead to just a few damaged frames in the decoded video (Figure 6.12). However, if more sudden speed changes occur during the file transmission, the received video may have a high visual distortion or may be rendered partially undecodable.

By decreasing the time between preambles, the performance of the algorithm can be improved in cases with sudden motion. Nevertheless, it has to be assured that the inequality $1/f_s < \min|a| \cdot T_{frame}$ is satisfied (Section 5.1.2).

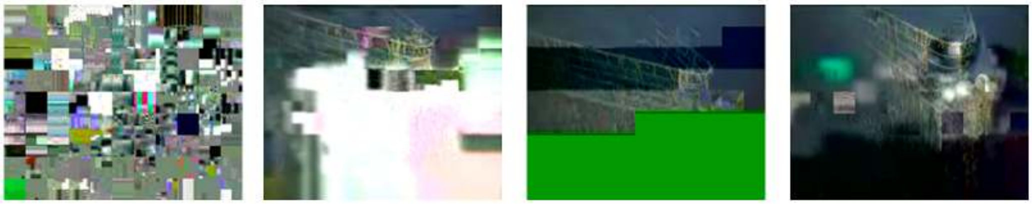


Figure 6.12: Damaged frames due to sudden changes in the speed between the transmitter and the receiver.

For the current application, aimed to work in AUV inspections, this should not be an issue because motion is typically slow and steady due to sensitive equipment in close proximity.

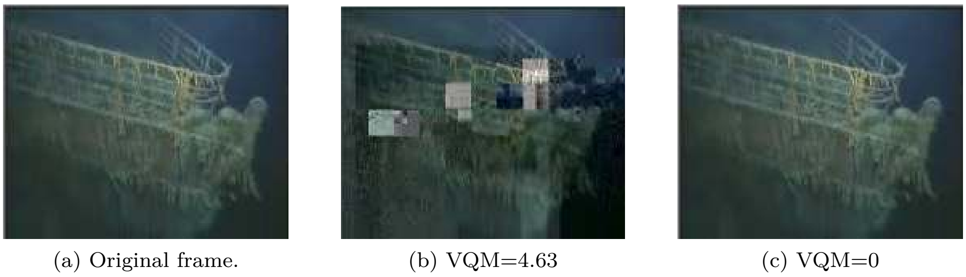
6.1.4 MPEG-4 error resilience tools

6.1.4.1 Video Quality Metric

The main goal of the error resilience toolset is to reduce the channel coding applied to the video data at the OFDM modulator, while maintaining the visual quality of the decoded video. In other words, they provide a better visual quality for a given code rate and BER.

Thus, a specific metric that quantifies the “human perception” needs to be chosen to decide whether the received video has an acceptable visual quality. For instance, it could be evaluated using the pixel-based Mean Square Error (MSE), which is a dominant metric in practice. However, the MSE does not take into account the spatio-temporal property of human’s visual perception and this is the reason why these types of metrics fail under many circumstances. In this thesis, the DCT-based Video Quality Metric (VQM) [34] has been chosen to evaluate the current system, as it exploits the property of visual perception.

The VQM indicates how much has the video been distorted: the higher VQM, the higher the distortion and the worse the quality; the lower VQM, the lower distortion and the better the quality (Figure 6.13). It is based on a simplified human spatio-temporal contrast sensitivity model illustrated in Figure 6.14.



(a) Original frame.

(b) VQM=4.63

(c) VQM=0

Figure 6.13: Video Quality Metric block diagram.

The conclusion after analyzing some videos with different VQM values is that the VQM for the decoded video must be below 1.5 in order to have an acceptable visual quality.

6.1.4.2 Simulations results

A portion of 5 seconds of the Titanic video has been compressed with and without using error resilience tools. The encoded video has been modulated with the same parameters used in Section 6.1.2. The characteristics of the tested BCH codes are summarized in Table 6.2.

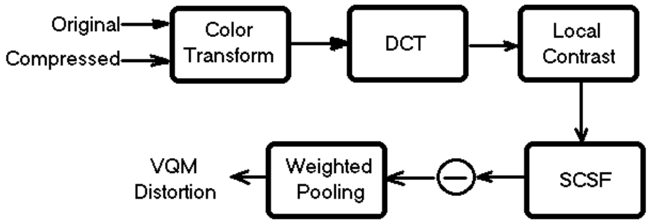


Figure 6.14: VQM block diagram. SCSF is the Spatial Contrast Sensitivity Function.

	BCH codes								
	127/120	**31/26**	**15/11**	**63/45**	**7/4**	**15/7**	**63/18**	**31/6**	**63/10**
Inverse code rate ($1/r$)	1.06	1.19	1.36	1.4	1.75	2.15	3.5	5.17	6.3
Ch. utilization	99.2%	96.9%	93.7%	98.4%	87.5%	93.7%	98.4%	96.9%	98.4%
Correction bits	1	1	1	3	1	2	10	7	13
Data bit rate (kbps)	293	260	228	222	177	145	88.6	60	49.2

Table 6.2: BCH coding characteristics. The data bit rates are calculated with $K = 16384$, $B = 115$ kHz and $T_g = 16$ ms.

Figure 6.15 shows the average[1] VQM as a function of the inverse code rate, $1/r$, for different values of the Signal-to-Noise Ratio (SNR). As expected, for lower values of SNR, higher inverse code rates are required in order to obtain an acceptable VQM.

When we are using error resilience tools, the inverse code rate required to meet a target VQM is lower than in the case in which we are not using error resilience tools. The BER and MSE thresholds are also improved by the use of error resilience tools, as demonstrated in Section 6.2. Unless otherwise indicated, the videos used in all simulations and experiments are encoded using error resilience tools.

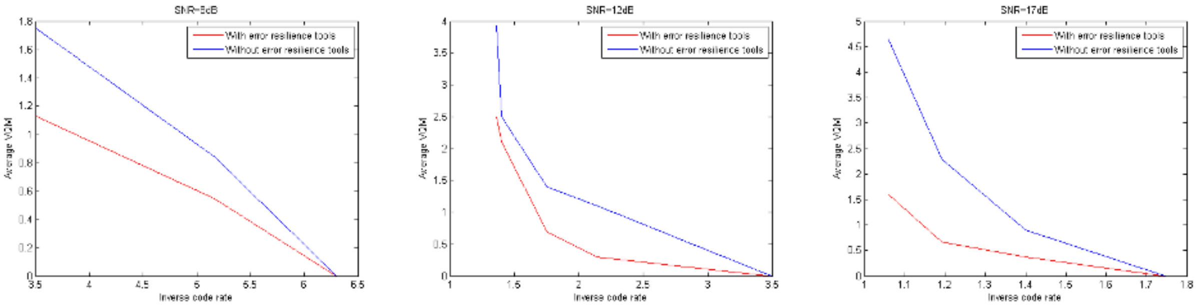


Figure 6.15: Average VQM as a function of the inverse code rate for SNR=8 dB (left), SNR=12 dB (middle) and SNR=17 dB (right).

6.2 In-air experiments

Because of the lack of widely accepted statistical channel models for the underwater channel, the results obtained by software simulations do not provide full justice to the actual underwater channel. Hence, proof-of-concept for various communication techniques is usually left to the experimental tests. But underwater tests are time-consuming and costly. In contrast, the in-air experiments allow testing the algorithms in a known channel, making it easier to conduct them in a laboratory environment, without depending on the weather conditions that make underwater experiments a challenge. Such an approach is expected to provide a more realistic version of the acoustic channel than software simulations. In addition, the indoor testbed provides an easy way of checking the performance under several channel scenarios, which can be recreated by moving the transmitter and/or receiver.

[1] The values of VQM have been obtained by averaging five repetitions for each value of the inverse code rate.

6.2.1 Deployment

The transmitter and receiver block diagrams are shown in Figure 6.16.

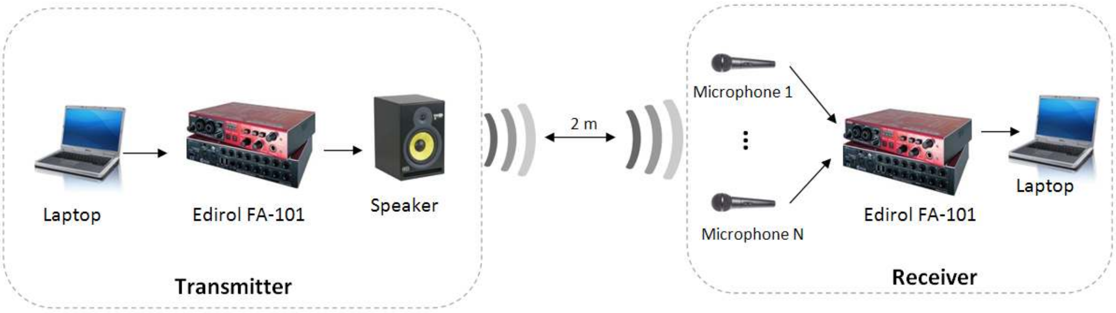


Figure 6.16: System deployment.

First, the MATLAB application generates the OFDM signal at the transmitting side laptop. It is then sent to the speaker using the EDIROL FA-101 board as an interface [35]. The receiving element is a microphone (or more than one in the case of spatial diversity). The EDIROL FA-101 board at the receiver side is the interface with the MATLAB application, which records and demodulates the received signal.

The sampling frequency of the soundcard is $f_s = 44.1$ kHz. In order to meet with the oversampling requirements ($f_h = f_s/4$), the frequency bandwidth has been set between 1 and 12.025 kHz ($B = 11.025$ kHz), avoiding a narrow low frequency band from 0 to 1 kHz where there are high noise components.

6.2.2 Doppler resampling algorithm

This set of in-air experiments demonstrate the correct performance of the Doppler compensation algorithm. The modulation parameters that have been used are detailed below.

- Sampling frequency: $f_s = 575$ kHz
- Number of subcarriers: $K = 4096$
- Bandwidth: $1 - 12.025kHz$ ($B = 11.025kHz$)
- Subcarrier spacing: $\triangle f = B/K = 2.69$ Hz
- OFDM block duration: $T = 1/\triangle f = 371$ ms
- Guard time: $T_g = 16$ ms
- Channel coding: BCH(63,18) ($1/r = 3.5$)
- Subcarrier modulation: 8-PSK
- Time between preambles: $T_{frame} = 1.15$ s
- Detection method: differentially coherent detection
- Number of receiving elements: 2 receivers

The relative speed between the transmitter and the receivers in all the experiments is around 10 cm/s, which corresponds to a Doppler coefficient $a = 0.1/c_{air} = 3 \cdot 10^{-4}$. In an underwater environment, this value would correspond to a relative speed of $v_r = a \cdot c_{water} = 47$ cm/s (∼1 knot).

The following must be mentioned about the guard time, T_g: as the speed of sound underwater is approximately five times the speed of sound in air, the multipath that is absorbed in a given time period for an underwater channel is longer than the absorbed multipath in air. For the same reason, the relative speed that can be tolerated in an underwater environment implementation is five times the applied motion in air.

Experiment 1: ideal case

The first experiment corresponds to the case in which there is no motion between the transmitter and the receiver (Figure 6.17). These results provide a reference of what can be expected in the ideal case.

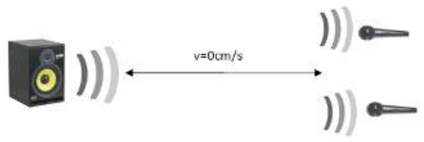


Figure 6.17: Experiment layout in the case of no motion.

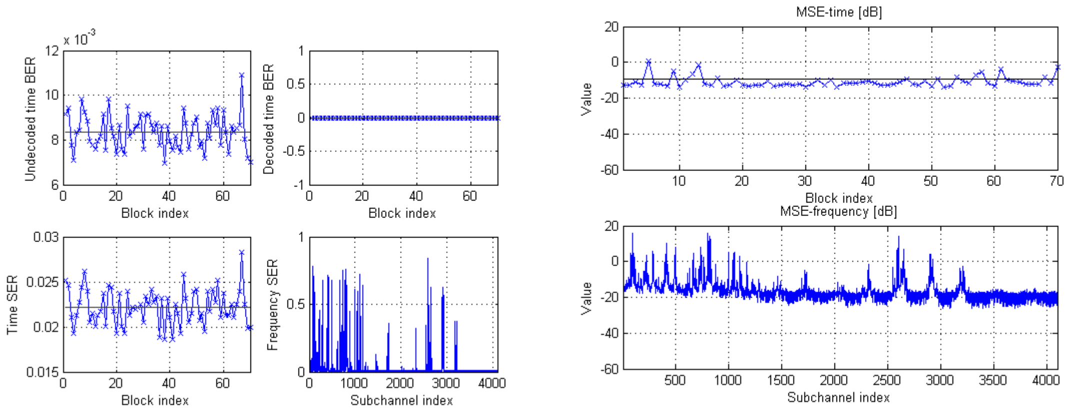


Figure 6.18: System performance in the ideal case.

As expected, the average BER for the decoded sequence remains at 0 along the transmission. The excellent performance is also demonstrated by the low average MSE $= -9$ dB. However, the frequency-MSE plot shows higher values for low frequencies due to the presence of high noise components at the low frequency bandwidth. For further details, see *Simulation 4: channel estimates* on page 77.

Note in the scatter plot that the detected symbols have been scaled in amplitude. This does not affect the detection process because the symbol regions are determined by angular areas, as shown in Figure 6.19.

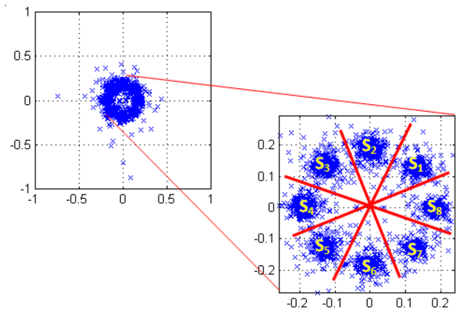


Figure 6.19: Scatter plot in the ideal case.

Experiment 2: uniform speed

In this experiment, the transmitter is moved with approximately uniform speed towards the receivers, followed by a deceleration and a final static period. The experiment layout is shown in Figure 6.20.

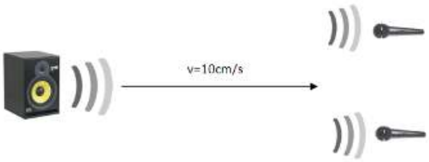


Figure 6.20: Experiment layout in the case of uniform speed.

When the Doppler compensation algorithm is not applied (Figure 6.21), the average BER for the decoded sequence remains at 0.5 when there is motion between the transmitter and the receiver. The MSE-time curve also shows high values during the motion period. As a result, the received video could not be decoded.

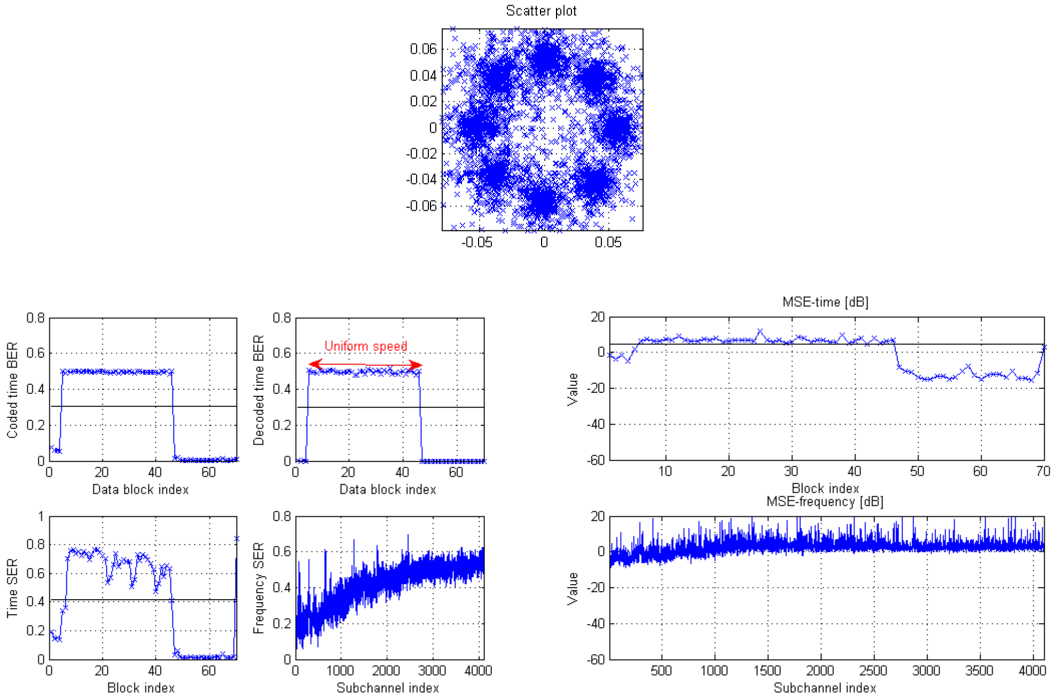


Figure 6.21: System performance when the compensation algorithm is **not** applied in the case of uniform speed.

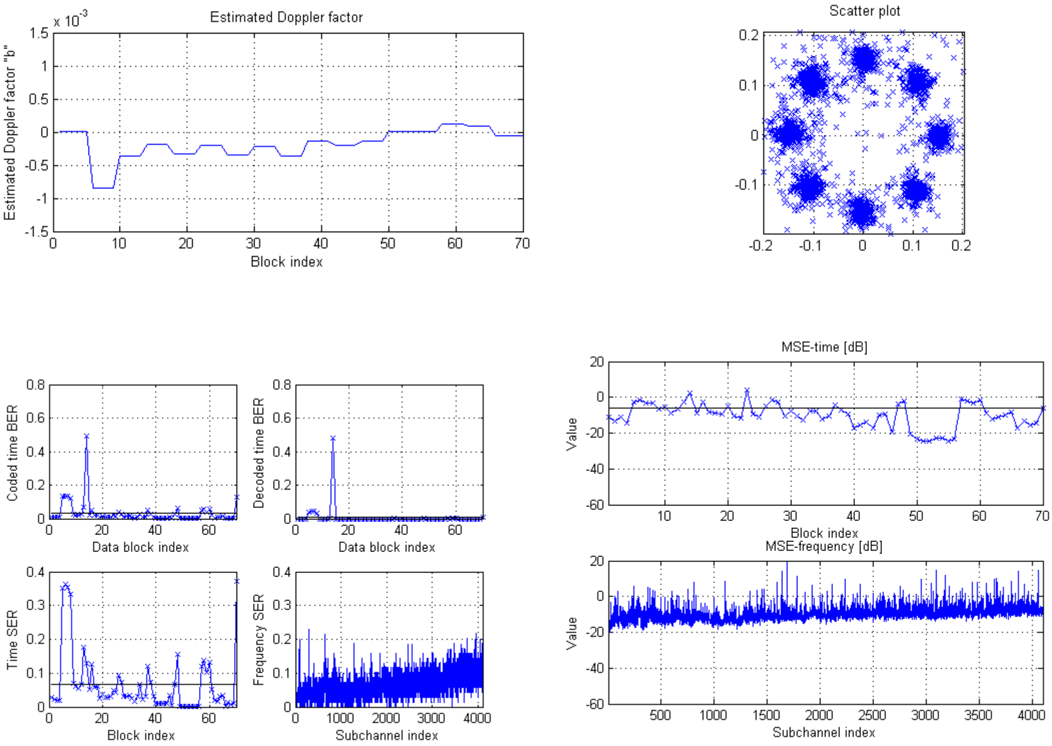


Figure 6.22: System performance when the compensation algorithm is applied in the case of uniform speed.

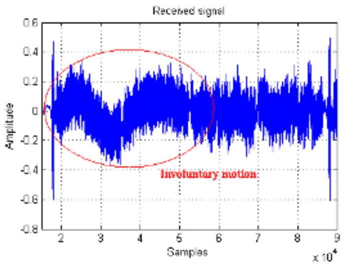


Figure 6.23: Effect of the fast motion at the beginning of the transmission.

When the Doppler compensation algorithm is applied (Figure 6.22), the BER for the decoded sequence shows a high peak at block 15, which is due to the fast motion applied to the transmitter when it was moved from its initial position to start the motion period (Figure 6.23). Since the Doppler coefficient only varied during 1 millisecond (much less than the duration of one OFDM frame), the algorithm was not able to detect this short period of motion.

The average BER for the decoded sequence along the rest of the signal remains around $9 \cdot 10^{-3}$. As this value is above the maximum permitted threshold, 10^{-3}, the video should present high levels of distortion or could even be undecodable. The simplest solution to improve these results would be to increase the channel coding, which in turn would decrease the data rate of the system (equation 6.2).

However, in this case the video decoder was able to decode the video with almost no distortion (VQM=0.8). This is explained by the use of MPEG-4 error resilience tools at the video decoder, which overcame those residual errors that the channel coding could not correct. Therefore, the video visual quality is maintained, while the BER requirements are relaxed.

It must be also noted that, unlike Experiment 1, the frequency-MSE curve shows an increasing trend for high frequencies. This is because the frequency-dependent shifts caused by motion, $a \cdot f_k$, have a stronger effect than the high noise components at low frequencies.

Experiment 3: change in speed

In this case, the transmitter is moved with uniform speed towards the receivers, followed by a decrease in velocity to a full stop. Then, the transmitter is moved with uniform speed in the opposite direction, moving away from the receivers. The experiment layout is shown in Figure 6.24.

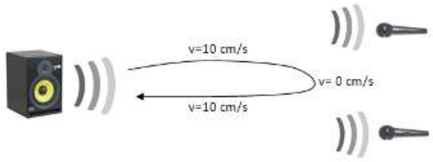


Figure 6.24: Experiment layout in the case of a change of speed.

When the compensation algorithm is not applied (Figure 6.25), the BER for the decoded sequence reaches values around 0.5 when there is motion between the transmitter and the receiver. The time-MSE curve also shows a similar trend. As expected, the received video was rendered undecodable.

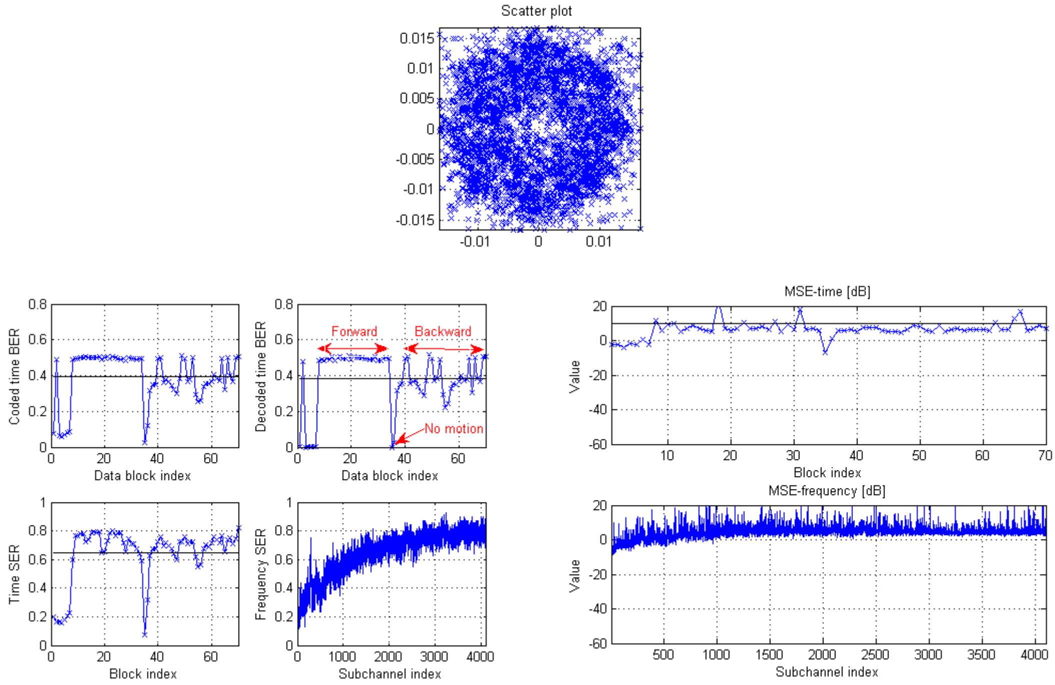


Figure 6.25: System performance when the compensation algorithm is **not** applied in the case of a change in speed.

Figure 6.26 illustrates the system performance when the Doppler compensation algorithm is applied. Unlike the scenario in *Simulation 4: sudden change in speed*, in this case the algorithm is able to follow the changes in the speed because the variations in the Doppler factor between adjacent OFDM frames are very small.

The BER for the decoded sequence shows again a high peak at block 10, which can also be explained by the fast movement at the beginning of the motion period. For the rest of the transmission, the BER for the decoded sequence remains around $1.3 \cdot 10^{-2}$. Again, this value is far above 10^{-3}, but the received video could be successfully decoded with low distortion (VQM=1) due to the use of error resilience tools at the video decoder.

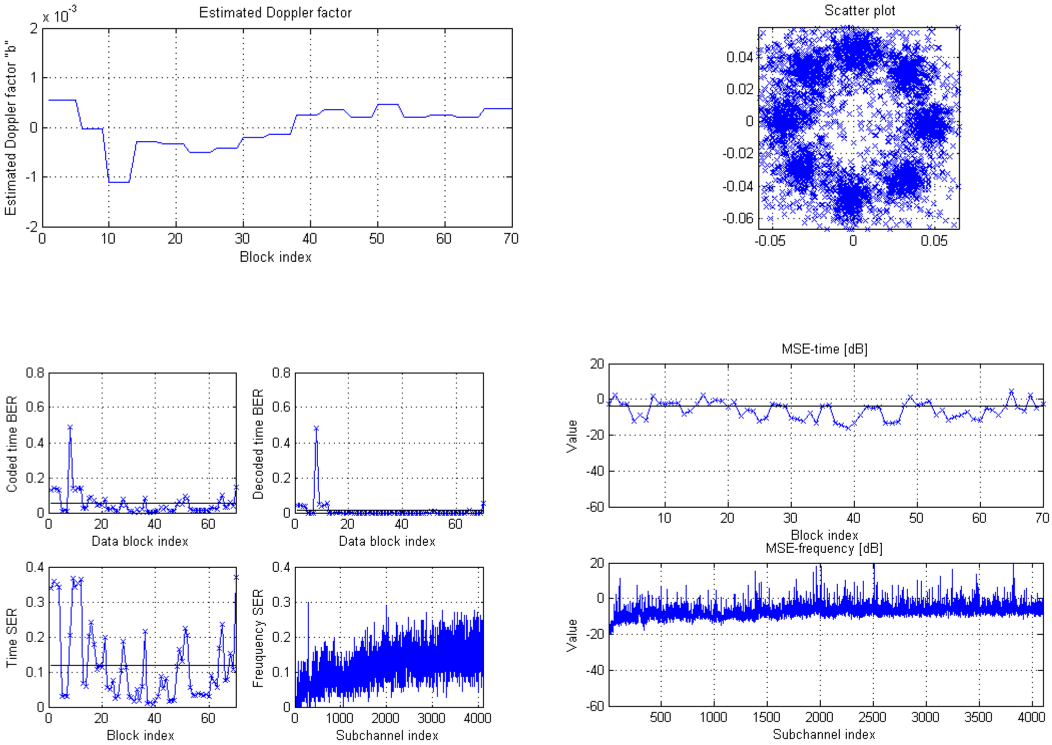


Figure 6.26: System performance when the compensation algorithm is applied in the case of a change in speed.

A final comment should be made regarding the performance of the Doppler compensation algorithm. Although it provides good results in most experiments with motion, sometimes it fails in measuring the Doppler factor. Specifically, when the second or third multipath arrival has a higher path amplitude than the direct path, the highest peak in the cross-correlation between the received signal and the transmitted preamble is not the correct point at which the preamble is located (Figure 6.27). Thus, future implementations should include an improved method to correctly locate the preambles.

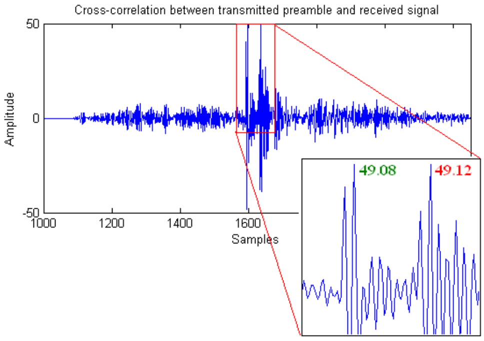


Figure 6.27: Sometimes the point in which the preamble is located does not coincide with the highest peak.

Experiment 4: channel estimation

The last experiment is designed to obtain a channel response estimation, both in time and frequency domains, by introducing pilot symbols along the signal. The relative speed between the transmitter and the receivers is approximately uniform along the transmission.

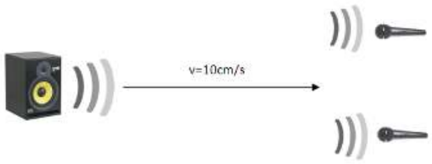


Figure 6.28: Experiment layout in the case of uniform speed.

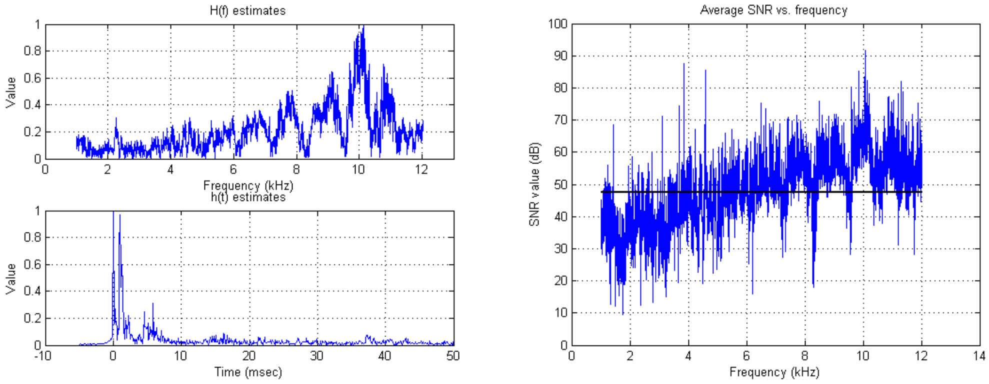


Figure 6.29: Channel estimates (left) and SNR in dB versus frequency (right).

Channel estimates show a high attenuation for low frequencies and the SNR shows an increasing trend for high frequencies. This is due to the presence of high noise components at the frequency band from 1 to 7 kHz. This fact agrees with the trend of the MSE-frequency in the case where there is no motion between the transmitter and the receivers (*Experiment 1: ideal case* 6.2.2). Thus, in future implementations, the operational frequency bandwidth should be shifted to higher frequencies in order to avoid this highly attenuated band.

In the time-domain channel estimation, the reflections due to the multipath propagation can be identified. They arrive at 1 ms (reflection on the wall) and 3 ms (reflection in the ceiling).

6.2.3 Multiple receivers

The last set of experiments aim to demonstrate the performance improvement with the use of multiple receivers (microphones) at the receiver side. The detected symbols from each receiver are averaged at the differentially coherent detector to obtain the final symbols. These results correspond to a motion case with uniform speed around 10 cm/s.

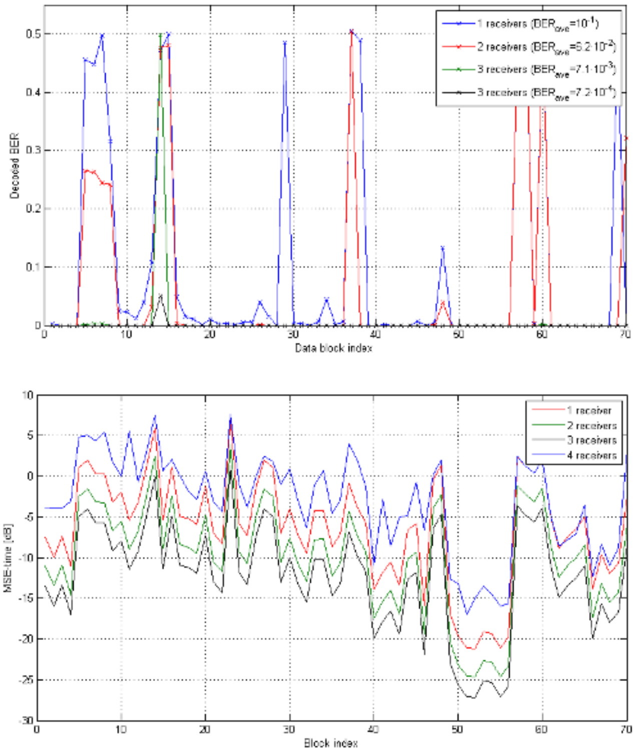


Figure 6.30: Time-BER for the decoded sequence (top) and time-MSE in dB (bottom).

The performance improvement when using 2, 3 and 4 receivers can be seen in the average time-BER plot. The MSE curves are also lower when spatial diversity is used, which represents a more accurate retrieval of the constellation.

Although in this project spatial diversity has only been implemented in the receiver side, it has also proven valuable for transmitter systems (transmission diversity) [36].

Chapter 7

Conclusions and future work

Based on the experimental results, a number of conclusions may be drawn regarding the design of the system and its implementation in a real-time environment.

The MPEG-4 standard appears to be the best video encoder for the low bit rate capacity of the underwater channel, reaching compression bit rates down to 64 kbps. With the selected OFDM parameters, the supported bit rate of the implemented system exceeds this value. Hence, although in this project the signal processing is performed off-line, the system can handle real-time transmission.

The developed Doppler compensation algorithm provides the system with an efficient technique to deal with the motion-induced Doppler effects. It has proven to correctly counteract the synchronization problems and compensate for the frequency shifts. However, some modifications to the algorithm are required in order to accurately find the correct position of the preambles in the received signal.

The advantages of the MPEG-4 error resilience tools are twofold. In cases in which the system performance provides high BER (above 10^{-3}), the error resilience tools allow the video decoder to decode the video with low distortion, avoiding the need to increase the channel coding. When a low BER is obtained, the channel coding applied to the video data can be reduced without losing visual quality, so that the data rate of the system is increased. In future implementations, additional error resilience tools may be included, such as Reversible Variable Lengths Codes (RVLCs), Header Extension Code (HEC) and New Prediction (NEWPRED) [37].

The experiments using several receiving elements demonstrate the performance improvement in terms of BER and MSE. Future work may include some testing with multiple transmitting and receiving elements (MIMO) in order to achieve a higher data rate, or a better detection performance.

Further testing needs to be conducted in an underwater environment, after which the system may be transformed into a real-time implementation. This implementation will involve several modifications:

- The implementation of the modulator (and demodulator) needs to be transformed into a “real-time code”, in the sense that both applications need to be continuously accepting (delivering) the data coming out of (entering to) the MPEG-4 encoder (decoder). To do so, it will be necessary to re-write the code in a faster platform than MATLAB, such as in a C language.
- The MPEG-4 encoder and decoder need to be coupled with the OFDM modulator and demodulator, respectively. This can be performed by placing a buffer between the two applications, so that it continuously receives the data from one side and delivers it to the other side in real-time.
- The MPEG-4 decoder needs to be coupled with the video player, so that the demodulated data can be converted to a live video.
- Finally, the timing constraints of the system and the channel latency must be taken into account, so that the video player can stream the video without any pause.

Figure 7.1 illustrates the structure of a potential real-time system with the features mentioned above.

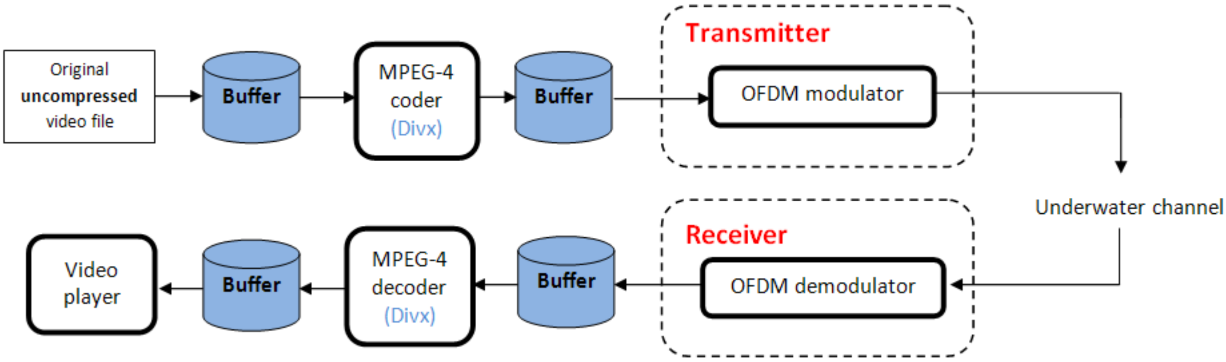


Figure 7.1: Structure of the real-time code.

Bibliography

[1] M. E. Field; C. H. Nelson; D. A. Cacchione and D. E. Drake. Sand waves on an epicontinental shelf: Northern bering sea. *Marine Geology*, vol. 42:1-4:pages 233–258, 1981.

[2] C. J. Smith and K.-N. Papadopoulou. Burrow density and stock size fluctuations of nephrops norvegicus in a semi-enclosed bay. *ICES Journal of Marine Science*, vol. 60:pp. 798–805, 2003.

[3] P.D. Osborne and B. Greenwood. Frequency dependent cross-shore suspended sediment transport. 2. a barred shoreface. *Marine Geology*, vol. 106:pp. 25–51, 1991.

[4] L. Lund-Hansen; E. Larsen; K. Jensen; K. Mouritsen; C. Christiansen; T. Andersen and G. Vølund. A new video and digital camera system for studies of the dynamics of microtopographic features on tidal flats. *Marine Georesources and Geotechnology*, vol. 22: 1-2:pp. 115–122, 2004.

[5] L.V. Pei; H.E. Junhua; Zhou Renkui and Liu Haiying. The application of underwater optics and its development. *Proceedings of the SPIE*, Volume 6837:pp. 68370I, 2008.

[6] P.N. Mikhalevsky; A.N. Gavrilov and A.B. Baggeroer. The transarctic acoustic propagation experiment and climate monitoring in the arctic. *IEEE Journal of Oceanic Engineering*, 24:2:pp. 183–201, 1999.

[7] Louis Boutan. Mémoire sur la photographie sous-marine. *Archives de Zoologie Expérimentale et Générale*, vol. 1, 3ème sér.:pages 281–324, 1893.

[8] H. Barnes. Underwater television and marine biology. *Nature*, vol. 169:pp. 477–479, 1952.

[9] M.Suzuki and T.Sasaki. Digital acoustic image transmission system for deep sea research submersible. *Proc. IEEE Oceans'92 Conference*, 1992.

[10] G. Ayela and J. Coudeville. Tiva: A long range, high baud rate image/data acoustic transmission system for underwater applications. *Proc. Underwater Defense Technol. Conf., Paris, France*, 1991.

[11] J. Gomes; V. Barosso; G. Ayela and P. Coince. An overview of the asimov acoustic communication system. *Proc. IEEE Oceans'00 Conference, Providence, RI*, pages pp. 278–283, 2000.

[12] H.Ando J.Kojima, T.Ura and K.Asakawa. Highspeed acoustic data link transmitting moving pictures for autonomous underwater vehicles. *Proc. IEEE Intl. Symp. on Underwater Technology*, pages pp. 278–283, 2002.

[13] J. Ribas and Dr. M. Stojanovic (advisor). Underwater wireless video transmission using acoustic ofdm. Master's thesis, Massachusetts Institute of Technology, 2009.

[14] E.Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *http://www-stat.stanford.edu/ candes/papers/OptimalRecovery.pdf*, 2004.

[15] David L. Donoho. Compressed sensing. *Department of Statistics, Stanford University*, 2004.

[16] Justin Romberg. Imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):pp. 14–20, March 2008.

[17] L. Gan; T. Do and T. D. Tran. Fast compressive imaging using scrambled hadamard ensemble. *Proc. EUSIPCO*, 2008.

[18] M. A. T. Figueiredo; R. D. Nowak and S. J. Wright. Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, vol. 1,no. 4:pp. 586–597, 2007.

[19] J.B. Lindsay. Mr. lindsay's marine telegraph. *Dundee Advertiser*, 12th April 1853.

[20] WFS Defense. *Seatext® Product Datasheet.* http://www.wfs-tech.com/.

[21] WFS Defense. *Seatooth® Product Datasheet.* http://www.wfs-tech.com.

[22] L. Liu; S. Zhou and J.-H. Cui. Prospects and problems of wireless communication for underwater sensor networks. *Wireless Communications and Mobile Computing*, pages pp. 977–994, 2008.

[23] Heather Brundage. Designing a wireless underwater optical communication system. Master's thesis, Massachusetts Institute of Technology, 2006.

[24] L.Berkhovskikh and Y.Lysanov. Fundamentals of ocean acoustics. *New York: Springer*, 1982.

[25] Marine Mammal Commission. The marine mammal protection act of 1972 as amended. *http://www.nmfs.noaa.gov/pr/pdfs/laws/mmpa.pdf*, 1972.

[26] M. Stojanovic. Capacity of a relay acoustic channel. *Proc. IEEE Oceans Conf.*, 2007.

[27] M. Stojanovic. On the relationship between capacity and distance in an underwater acoustic communication channel. *Proc. First ACM International Workshop on Underwater Networks (WUWNet)*, 2006.

[28] J.Catipovic; M.Stojanovic and J.Proakis. Phase coherent digital communications for underwater acoustic channels. *IEEE Journal of Oceanic Engineering*, vol.13, No.1:pp.100–111, 1994.

[29] C. Pelekanakis; M. Stojanovic and L. Freitag. High rate acoustic link for underwater video transmission. *Proc. IEEE Oceans'03 Conference*, 2003.

[30] H.Wallace. Error detection and correction using the bch code. 2001.

[31] E. Meijering. A chronology of interpolation: from ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, vol. 90 (3):pp. 319–342, 2002.

[32] M. Stojanovic. Ofdm for underwater acoustic communications: Adaptive synchronization and sparse channel estimation. *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP'08)*, 2008.

[33] Virtualdub. *http://www.virtualdub.org/.*

[34] Feng Xiao. Dct-based video quality evaluation. *Final Project for EE392J, Standford University.*, Winter 2000.

[35] Roland®. Edirol fa-101. *http://www.rolandus.com/products.*

[36] B. Li; J. Huang; S. Zhou; K. Ball; M. Stojanovic; L. Freitag and P. Willett. Mimo-ofdm for high rate underwater acoustic communications. *IEEE Journal of Oceanic Engineering*, vol.34, No.4:pp.634–645, October 2009.

[37] Fernando C. N. Pereira and Touradj Ebrahimi. *The MPEG-4 book.* Prentice Hall PTR, 2002.