NTNU – Trondheim
Norwegian University of
Science and Technology

CHANNEL ESTIMATION IN UNDERWATER ACOUSTICAL OFDM SYSTEMS

COURSE: TTT-4511 – SPECIALIZATION PROJECT

STUDENT: SONIA MENDIETA MORETON

NTNU, 2016

INDEX

TABLE OF FIGURES

1. INTRODUCTION

1.1. PROJECT OBJECTIVES

The underwater acoustical channel may produce severe changes in a signal that passes through it. For a communication system that requires high bit-rates it would be a challenge the design the system due to the need of the compensation of the signal degradation suffer in the underwater acoustical channel.

The main features of the channel are that the acoustical waves will travel slowly through the channel and that the water channel is inherently scattering. Therefore, the main consequences will be:

- The delay spread tends to be large.
- The coherence bandwidth tends to be small.
- The channel will be frequency selective for wideband systems.
- The Doppler spread is much larger than in radio communication and cannot be neglected.
- The coherence time will be small.

In order to compensate the signal degradation, adaptive channel estimation in the receiver is needed. To estimate the channel, some pilot symbols must be transmitted. The number of transmitted pilot symbols will be a trade-off between the need of high-bit rates and the need of an accurate demodulator.

In the present document it is explained in detail the design of an OFDM system implemented in an underwater acoustic channel.

First, an introduction to OFDM systems is presented. Then, the main characteristics of an underwater acoustic channel are introduced. Afterwards, the system implementation is presented in detail, including the transmitted signal, an underwater acoustic channel simulator, the received signal, and an OFDM Low-Complexity detector including the signal after the FFT demodulator, the channel estimation, and the data estimation and decision.

The present work has been done in collaboration with the company Water Linked AS.

2. OFDM SYSTEMS

2.1. INTRODUCTION TO OFDM

The OFDM (Orthogonal Frequency-Division Multiplexing) technique consists of dividing the available spectrum into several subchannels, allocated in different subcarriers. The main advantage of this technique is that every subchannel can be considered as a narrowband system. Therefore, if the frequency response of a channel varies fast and is frequency-selective, then a wideband system can be modeled as a set of several narrowband systems where the equalization is easier. Figure 1 shows the spectrum of an OFDM system. In order to be orthogonal, the subcarrier spacing Δf must be the inverse of the duration of a symbol (i.e. 1/T). The main purpose of the orthogonality is preventing the system of ICI (i.e. inter-carrier interference).

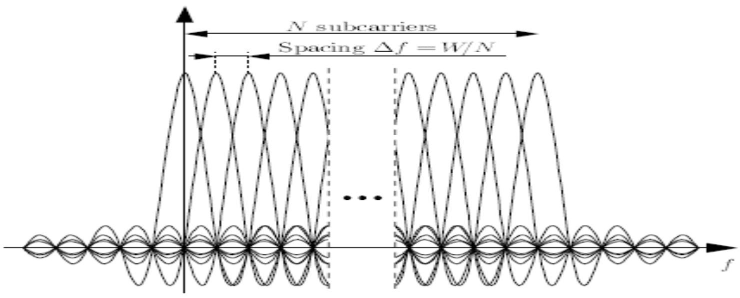

Figure 1. OFDM Spectrum with N subcarriers

2.2. CONTINUOUS-TIME OFDM SYSTEM MODEL

The continuous-time OFDM system model can be seen in Figure 2, including the transmitter, the channel and the receiver. In this case, the system is not employing digital modulation and demodulation. Nowadays, all the processing is digital.

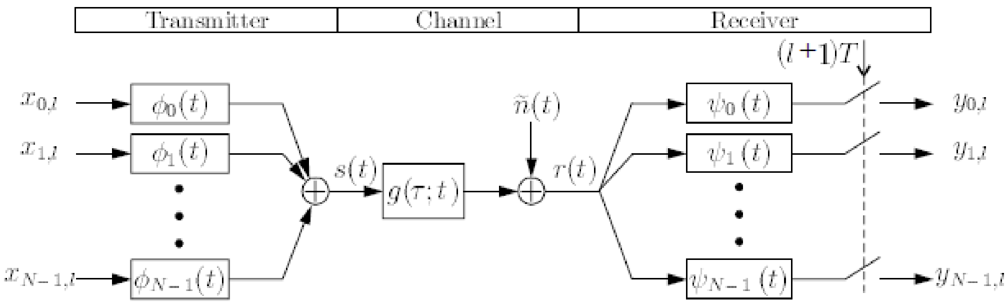

Figure 2. Continuous-time OFDM system model

In order to combat the multipath propagation effect a guard time is included between the transmission of every OFDM symbol and the next one. This guard interval has to be larger than the channel impulse duration. There are several techniques regarding the filling of the guard interval in order to take advantage of it. The most common is the cyclic prefix technique, which consists of filling the guard interval with a copy of the last part of the OFDM symbol, as shown in Figure 3. In this way, the transmitted signal becomes periodic, which makes easier to avoid the inter-symbol and inter-carrier interference.

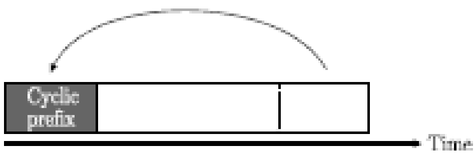

Figure 3. Cyclic prefix technique

Other techniques are also used as Zero-Padding, which is used when saving transmission power is an important requirement.

Regarding the block diagram of Figure 2, it is assumed that the system has N subcarriers, a bandwidth W in Hz, and a symbol length T seconds. The length of the cyclic prefix is T_{cp}, included in T. The waveforms of the transmitter filters are rectangular pulses modulated in the frequency corresponding to each subcarrier and can be expressed as follows:

$$\phi_k(t) = \left\{ \begin{array}{ll} \frac{1}{\sqrt{T - \mathrm{T_{cp}}}} e^{j2\pi \frac{W}{N} k(t - \mathrm{T_{cp}})}, & if\ t \in [0, T] \\ 0, & otherwise \end{array} \right\}$$

Where T = N/W + T_{cp}.

The transmitted base band signal for the OFDM symbol l is the following:

$$s_l(t) = \sum_{0}^{N-1} x_{k,l} \phi_k(t - lT)$$

Where $x_{k,l}$ are the input complex numbers from a set of signal constellation points. Then, the complete transmitted signal of the OFDM system will be:

$$s(t) = \sum_{l=-\infty}^{+\infty} s_l(t)$$

The next part of the diagram to be explained is the channel, which has an impulse response $g(\tau;t)$. In this case, we consider a time-varying channel. In order to reduce the multipath impact (i.e. spread of the transmitted symbol and, in consequence, inter-symbol interference), the length of the cyclic prefix must be equal or higher than the duration of the impulse response of the channel (i.e. $T_{cp} \geq \tau$). Thus, the received signal can be expressed as:

$$r(t) = g(\tau;t) * s(t) = \int_0^{T_{cp}} g(\tau;t)s(t-\tau)d\tau + \tilde{n}(t)$$

Where $\tilde{n}(t)$ is the channel noise, which is consider as complex, additive, white and Gaussian for simplicity.

Then, in the part of the receiver, the cyclic prefix is removed and the received signal r(t) is the input of a set of matched filters to the last part (i.e. $t \in [\mathrm{T_{cp}}, T]$) of the transmitter waveforms $\phi_k(t)$, which can be expressed as follows:

$$\varphi_k(t) = \begin{Bmatrix} \phi_k^*(T-t), & if\ t \in [0, T-\mathrm{T_{cp}}] \\ 0, & otherwise \end{Bmatrix}$$

Therefore, since the transmitter filters are orthogonal, the output of the receiver filter bank for one OFDM symbol is the following:

$$y_k = h_k x_k + n_k$$

Where n_k is the noise after the filter bank which can be modeled as a random variable and with its corresponding statistics and h_k is the channel frequency response in the kth-subcarrier. Assuming that the channel response is constant over one OFDM symbol (i.e. during T seconds), it can be written as:

$$h_k = G\left(k\frac{W}{N}\right) = \int_0^{Tcp} g(\tau)e^{-j2\pi k\tau W/N} d\tau$$

2.3. DISCRETE-TIME OFDM SYSTEM MODEL

On the other hand, the equivalent discrete-time model of the OFDM system can be seen in Figure 4.

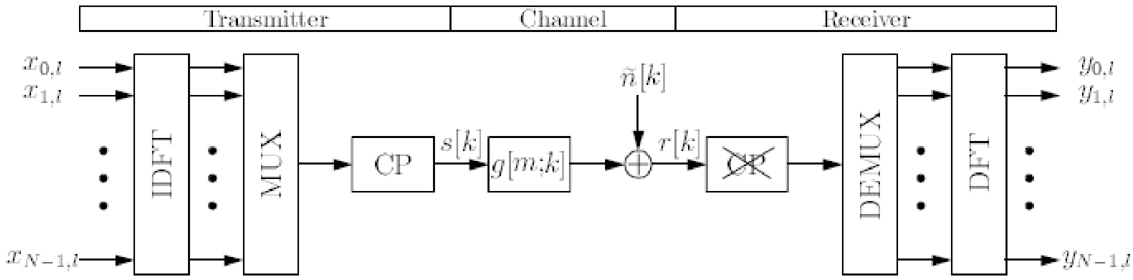

Figure 4. Discrete-time OFDM system model

In this case, in order to make the processing digital, the modulation and demodulation are replaced by the IDFT (*Inverse Discrete Fourier Transform*) and the DFT (*Discrete Fourier Transform*), respectively. Additionally, the channel turns to be a discrete-time convolution. The integration of the cyclic prefix makes that the system can be describe in frequency domain as:

$$y_l = x_l \cdot h_l + n_l$$

Where y_l is the demodulated signal for the OFDM symbol l that contains N received data points, x_l are the N transmitted constellation points, $h_l = DFT(g_l)$ is the frequency response of the channel and n_l is uncorrelated noise.

2.4. ADVANTAGES OF OFDM SYSTEMS

The use of OFDM systems has the following advantages with respect to other systems:

- All sub-channels can be considered as narrowband channels, experiencing almost flat fading.
- Robustness in the case of frequency-selective fading.
- Easier equalization than in single-carrier systems.
- Effective use of the frequency spectrum.
- Elimination of the effects of the inter-symbol interference (ISI) and inter-carrier interference (ICI) due to the integration of cyclic prefix or zero-padding.
- Orthogonality maintain due to the CP even in a time-dispersive channel.
- Computationally more efficient due to the use of FFT (Fast-Fourier Transform).

3. UNDERWATER CHANNEL CHARACTERISTICS

In this section, some of the characteristics of an underwater acoustic channel will be discussed. The main effects that will suffer the transmitted signal when passing through the underwater channel will be attenuation, noise addition, multipath propagation and Doppler Effect. There are no standardized models for the underwater acoustic channel fading so the statistical properties are usually identified from measurements in real environment. It is worth noting that the underwater acoustic channel is time-varying so that all the effects will affect in a different way each OFDM symbol, assuming that the frequency response is constant during one OFDM symbol. These effects will be explained in detail in the following sections.

Moreover, the following characteristics must be taking into account when designing an underwater acoustic system:

- The propagation of the signals is in the form of acoustic waves.
- The approximate speed of sound in water is c = 1500 m/s.
- The channel capacity is limited, and it depends on the distance.
- Propagation at low frequencies, in the order of KHz.
- The system will be wideband, because the bandwidth is not negligible compared to the center frequency (i.e. it is not true that: $f_c \gg W$).
- The signal will be affected by time-varying attenuation, noise, multipath propagation, Doppler Effect, and other aspects.
- The coherence time of the channel is in the order of hundreds of milliseconds.

3.1. ATTENUATION AND NOISE

The transmitted signal will be attenuated when travelling through the channel. The attenuation increases with the signal frequency. Thus, in OFDM systems, signal with higher frequency will be more attenuated than signals at lower frequencies. Attenuation effect is mainly due to absorption, which is modeled by an absorption loss depending on the operating frequency. It can be seen in Figure 5 a comparison of the attenuation for several types of elements with frequency.

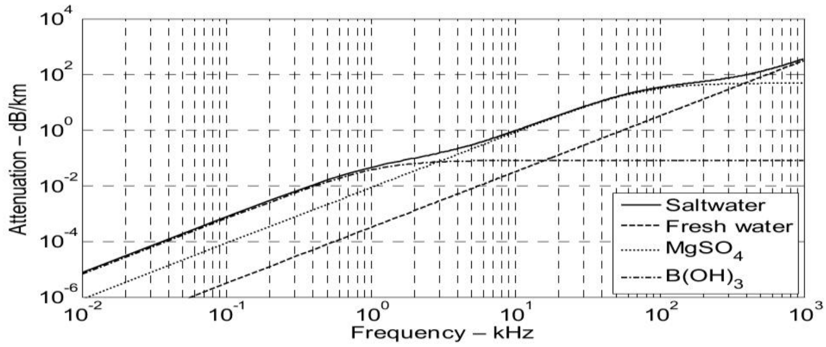

Figure 5. Attenuation in Underwater Acoustic Channel

On the other hand, the background noise is often characterized as additive, Gaussian and white noise (AWGN) but it is not really white. Indeed, it has a decaying power spectral density. Moreover, depending on the source, the noise can contain non-Gaussian components that have to be considered.

3.2. MULTIPATH PROPAGATION

Due to the reflection of the transmitted signal in the surface or the bottom of the sea or other objects like fish or boats, as seen in Figure 6, the signal will be propagated over multiple paths, depending on the physical environment. The acoustic rays will follow Snell's law.

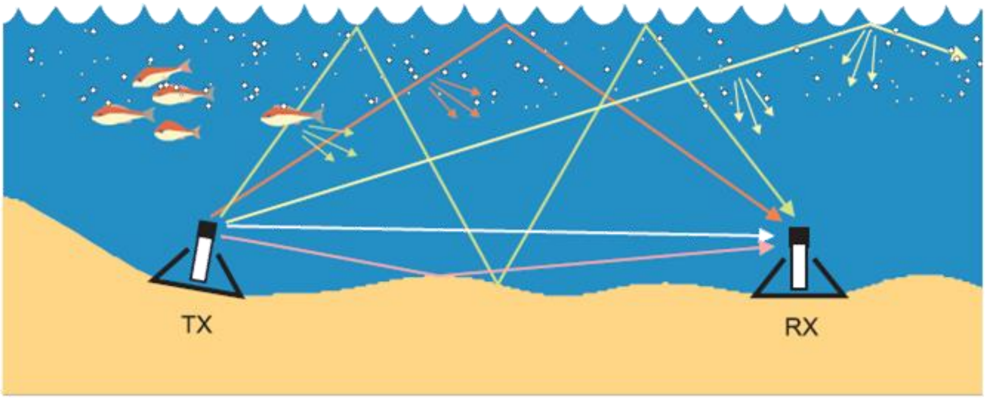

Figure 6. Multipath Propagation in Underwater Acoustic Channel

The delay spreading due to multipath propagation in underwater acoustic channels is in the order of tens of milliseconds, which leads to the need of a long time guard interval and a longer OFDM symbol period.

Due to multipath propagation, the receiver will observe several signals (when transmitting just one signal). The channel response will consist of several paths with a gain and a delay associated to each one of them. The path delays can be obtained for each path p applying the following formula:

$$\tau_p = \frac{L_p}{c}$$

Where L_p is the distance covered by acoustic wave in the path p and c is the speed of sound in the water (i.e. c = 1500 m/s).

Therefore, in the presence of multipath, the channel impulse response can be written as:

$$h(t) = \sum_p A_p \delta(t - \tau_p)$$

3.3. DOPPLER EFFECT

Some motion of the transmitter or the receiver introduces changes in the channel response. Some underwater systems have intentionally moving transmitter or received, for example, when using autonomous underwater vehicles (AUVs) moving at several tens of m/s. On the other hand, some other systems will have fixed location of the transmitter and receiver, but due to waves, currents, and other environmental factors, they will be in motion too with comparable relative velocities. Thus, there will always be some motion present in underwater acoustic systems so that it always must be taken into account.

This motion produces Doppler Effect, which causes frequency shifting and frequency spreading. The effect of frequency shifting for an OFDM system can be appreciated in Figure 7.

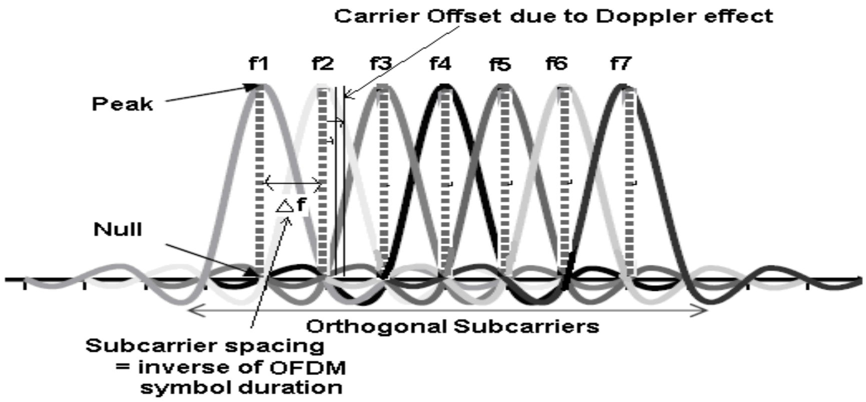

Figure 7. Doppler Effect in an OFDM system

Therefore, when transmitting a signal in a frequency f_k, the signal observed at the receiver in the presence of Doppler effect, will be one located at a frequency $f_k + a \cdot f_k$, where a is called the Doppler rate. The Doppler rate can be expressed as the ratio between the relative velocity due to motion of the transmitter or receiver to the speed of sound by $a = v/c$. This parameter is usually much less than 1.

The value of a in the case of underwater acoustic channels is not negligible compared to other types of channels (as radio channels), so it must be taken into account and the phase and delays must be tracked. In multicarrier systems (as OFDM), the Doppler Effect creates severe distortion leading to possible inter-carrier interference (ICI). In an acoustic system, the Doppler shift will be different for each subcarrier since the system is a wideband system, so that there will be non-uniform Doppler distortion across the signal bandwidth.

4. CHANNEL ESTIMATION

Usually, the impulse response of the channel for wireless systems is not known at the receiver. Thus, the receiver has to be able to estimate the channel in some way in order demodulate correctly the transmitted data symbols.

An underwater acoustic channel has complex characteristics that must be taken into account when designing the channel estimator. It is important to remember that the impulse response of this type of channels is time variant. Hence, the channel must be estimated dynamically and adaptively every certain period of time in order to extract the data correctly during the whole transmission.

In order to initialize the channel estimator and to track correctly the parameters, the transmission of pilot symbols (i.e. reference symbols that are known in the receiver) is needed. An example of pilot symbol insertion into an OFDM system can be seen in Figure 8, where the pilot symbols are located in all subcarriers during one OFDM symbol interval.

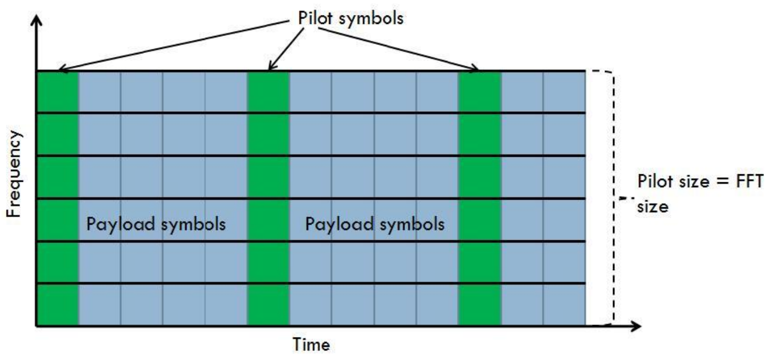

Figure 8. Example of pilot symbol insertion into an OFDM system

Regarding OFDM systems, there are two main problems when designing the channel estimator: the choice of the pilot symbols (in time and frequency) and the need of a low complexity estimator and good channel tracking ability.

The number of transmitted pilot symbols is a trade-off between the need of a high data rate and the need of reliable channel estimation (low bit-error rate).

5. SYSTEM IMPLEMENTATION

A complete OFDM system has been implemented using MATLAB® software. The system consists of an OFDM transmitter, an underwater acoustic channel, and an OFDM receiver which estimates the channel adaptively and demodulates the received signal.

In the following sections, the different parts of the system are explained. First, a set of system requirements are listed. Then, the transmitted signal is presented. Afterwards, a channel simulator is modeled, according to the characteristics of an underwater acoustic channel. Later, an equivalent model of received signal is presented. And, finally, the detector algorithm is explained and its performance is presented.

5.1. SYSTEM REQUIREMENTS

5.1.1. TRANSMITTER

- The transmitter and the receiver are perfectly synchronized.
- The cyclic prefix is larger than the impulse response of the channel.

5.1.2. CHANNEL

- The channel noise is Additive, White and Gaussian (AWGN).
- The noise components are assumed to be of equal variance and uncorrelated.
- The fading is slow enough that the channel can be considered constant over the duration of one OFDM symbol.
- The frequency shift produced by the Doppler Effect is much less than the subcarrier spacing (i.e. $\Delta f \gg af_k$).
- The channel produces non-uniform Doppler distortion across subbands.
- Wideband OFDM system, because the bandwidth is comparable to the center frequency ($B{\sim}f_c$).
- The path gains, delays, and the Doppler rate are assumed to be constant over the duration of one OFDM symbol, but may change from one block to another.
- The Doppler rate is assumed equal for all paths.
- Transmitter/Receiver motion is the dominant source of Doppler distortion.
- The angles of multipath arrivals differ little.
- The channel gain varies slowly from one OFDM symbol to another.

5.1.3. RECEIVER

- The sampling time is optimum.

5.2. TRANSMITTED SIGNAL

The transmitted signal is an OFDM signal using an IFFT modulator. A rectangular pulse shaping is used in the modulation in order to implement efficiently the modulation and detection. The transmitter follows the block diagram of Figure 4, which has been explained in Section 2.3.

The OFDM transmitted signal has K subchannels, where the input data stream is converted from serial to parallel and then located in the K subcarriers ($k = 0, \dots, K-1$) and then transmitted in form of OFDM symbols. The number N of OFDM symbols depends on the length of the input data stream.

Then, the transmitted OFDM signal can be expressed as follows:

$$s(t) = Re\left\{\sum_{k=0}^{K-1} u_k(t) e^{j\omega_0 t}\right\}$$

Where

$$u_k(t) = \sum_n d_k(n) e^{jk\Delta\omega(t-nT')} g(t-nT')$$

The function $g(t-nT')$ is a rectangular pulse of width T' and unit amplitude. The parameter T' in this case is the duration of the transmitted OFDM symbol including the guard time interval (i.e. $T' = T + T_{CP}$). T_{CP} is the guard interval and is longer than the multipath spread. The term $\Delta\omega$ is equal to $2\pi\Delta f$, where $\Delta f = 1/T$ is the carrier spacing.

In this case, the input data symbols $d_k(n)$ (i.e. the n^{th} OFDM symbol at the k^{th} subcarrier) correspond to a set of signals from a QPSK constellation using Gray coding, as shown in Figure 9. The data symbols of the QPSK constellation are transmitted with the same energy and are equiprobable.

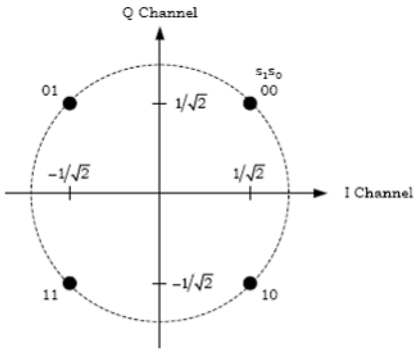

Figure 9. QPSK Constellation using Gray coding

As an example, a transmitted OFDM signal has been simulated with the following characteristics:

Number of bits to transmit	1024
Number of subcarriers	8
Bandwidth	24 KHz
Δf	3 KHz
T	0.33 ms
Tcp	0.05 ms
CP length	2 symbols

The different steps follow in the transmitted for the first OFDM symbol are shown in the following figures. Then, in Figure 22 and in Figure 23 the complete transmitted OFDM digital signal is shown.

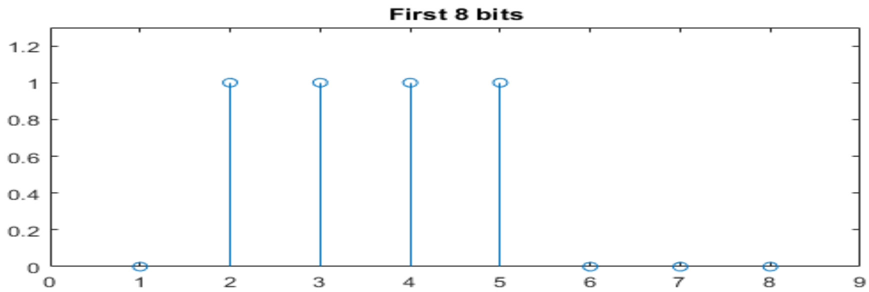

Figure 10. First 8 transmitted bits: 01111000

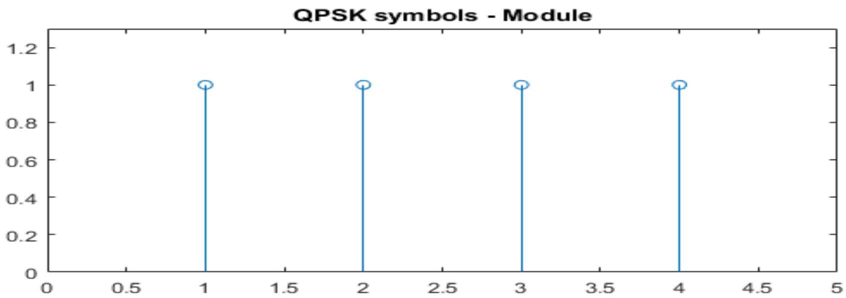

Figure 11. Module of the first 4 QPSK transmitted symbols

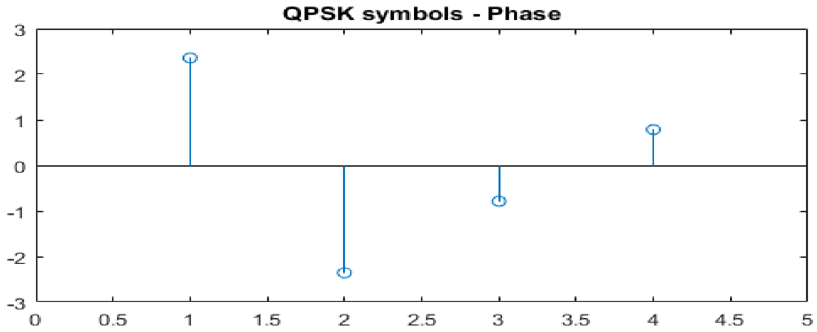

Figure 12. Phase of the first 4 QPSK transmitted symbols

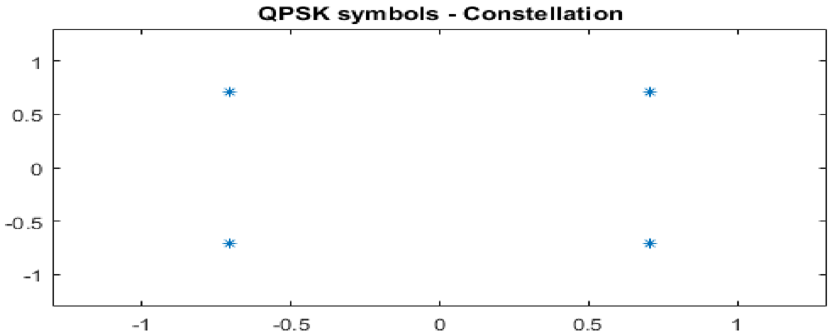

Figure 13. First 4 QPSK transmitted symbols as a constellation

Then, in order to show the effect of the cyclic prefix, we extend the analysis to the first 8 QPSK transmitted symbols, each one located in a different subcarrier.

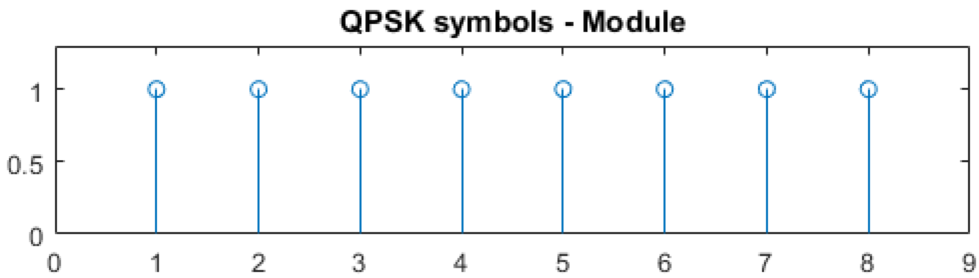

Figure 14. Module of first 8 QPSK transmitted symbols

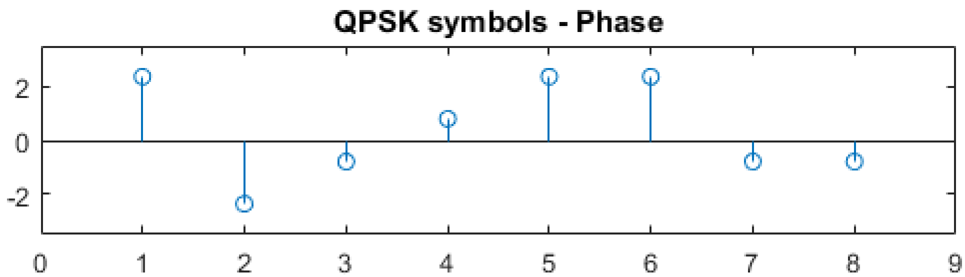

Figure 15. Phase of first 8 QPSK transmitted symbols

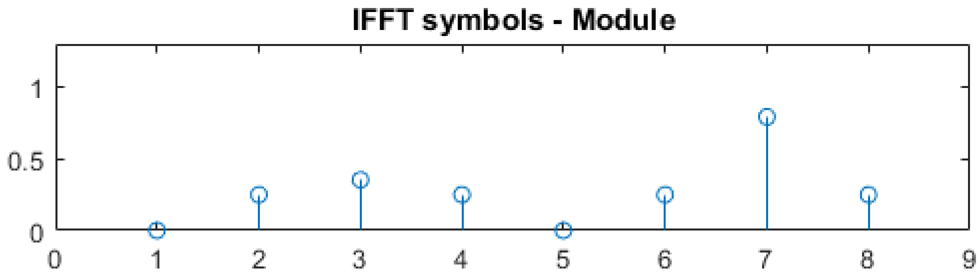

Figure 16. Module of first IFFT stream with 8 samples

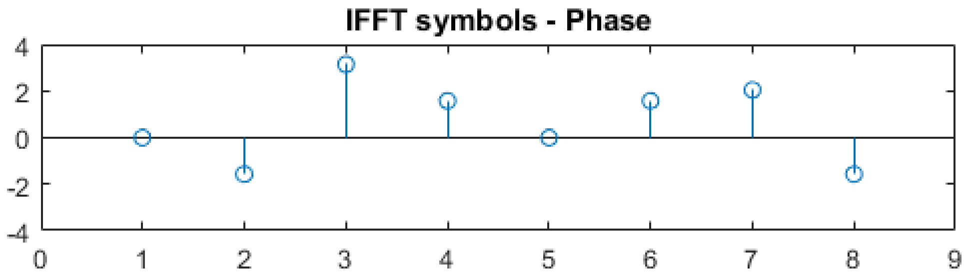

Figure 17. Phase of first IFFT stream with 8 samples

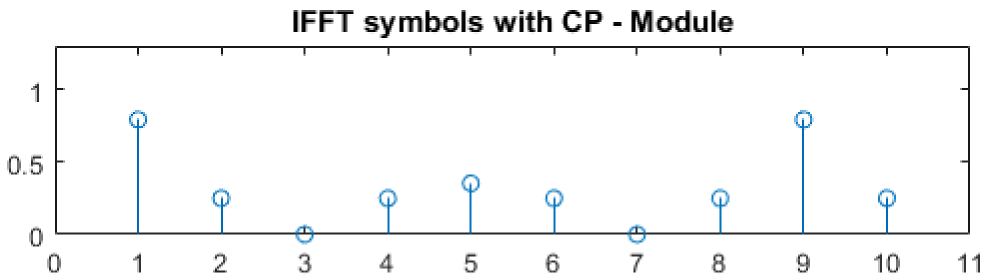

Figure 18. Module of first IFFT stream with CP

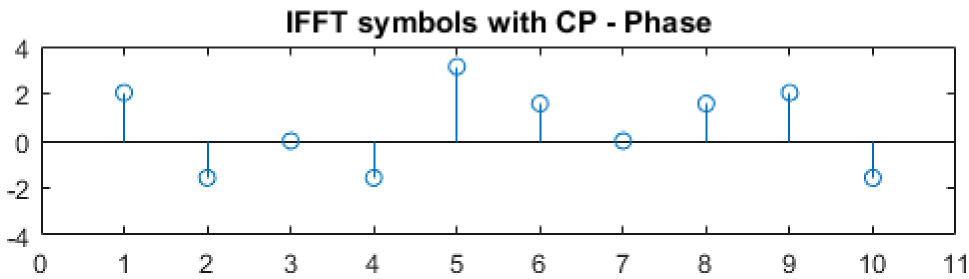

Figure 19. Phase of first IFFT stream with CP

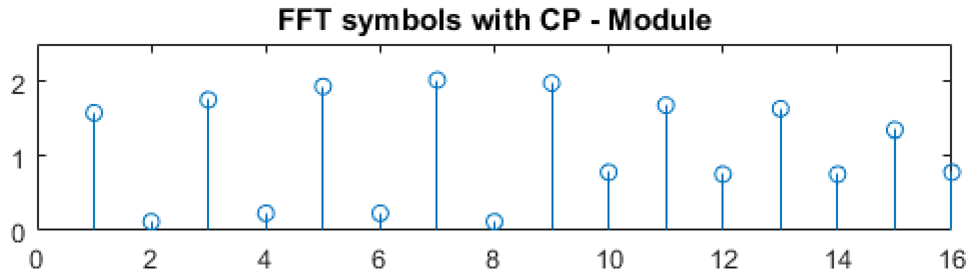

Figure 20. Discrete spectrum of the first 8 symbols with CP – Module

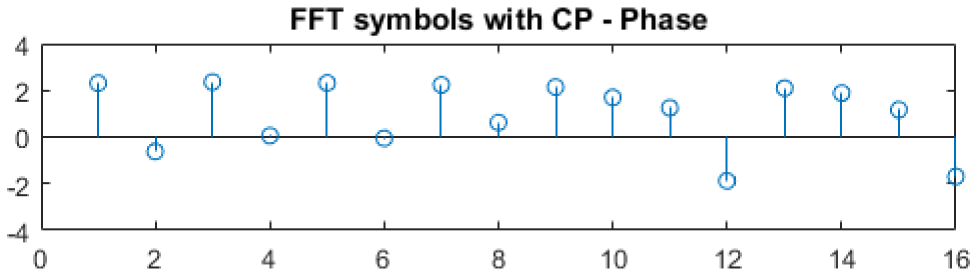

Figure 21. Discrete spectrum of the first 8 symbols with CP – Phase

For the whole data stream, the complete signal is shown in the following figures. The signal is represented by its samples.

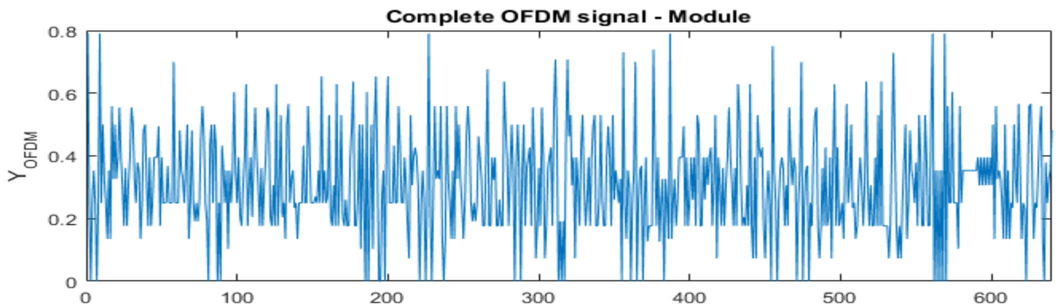

Figure 22. Module of Complete transmitted OFDM signal

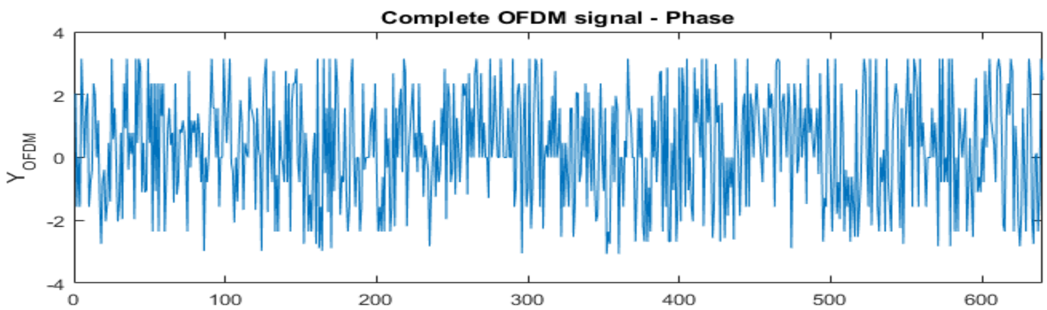

Figure 23. Phase of Complete transmitted OFDM signal

5.3. RECEIVED SIGNAL AND DEMODULATOR FOR IDEAL CHANNEL

In the absence of distortion produced by the non-ideal channel, the received signal is the same as the transmitted OFDM signal. Thus, the process of demodulation is made following the inverse steps of the transmitter. In the following figures can be seen the well functioning of the demodulator with an ideal channel. The decision of the estimated symbols is made according to the ML (*Maximum Likelihood)* criterion, as shown in Figure 24.

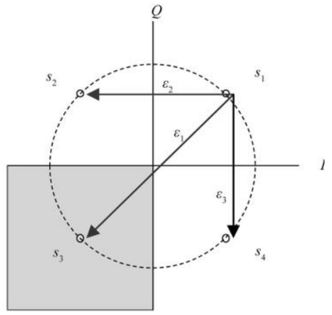

Figure 24. Rule of decision of estimated symbols - ML criterion

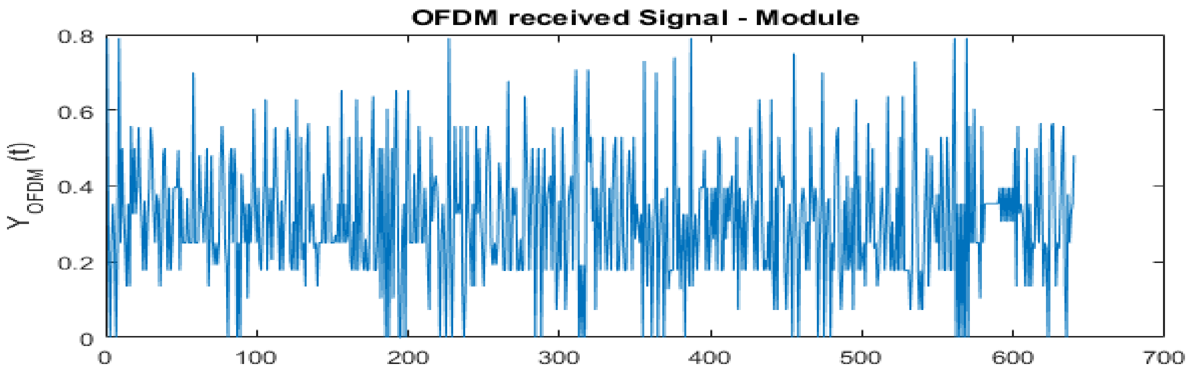

Figure 25. Received OFDM Signal – Module

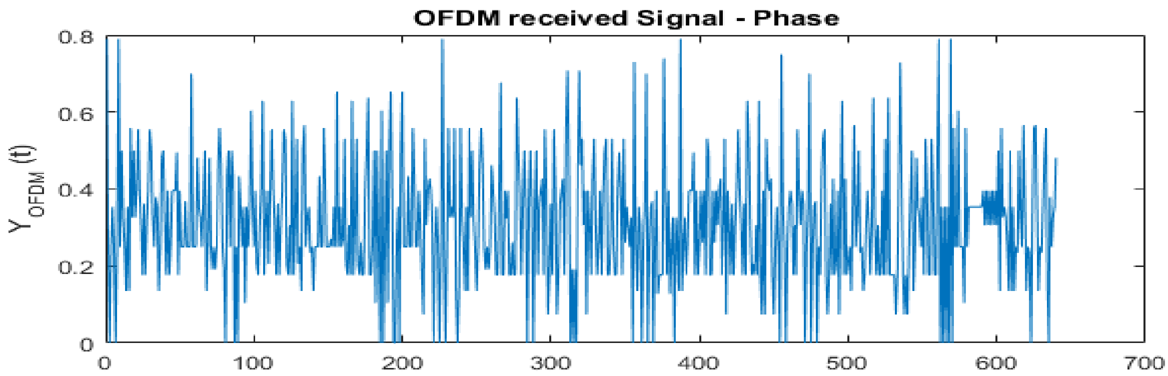

Figure 26. Received OFDM Signal – Phase

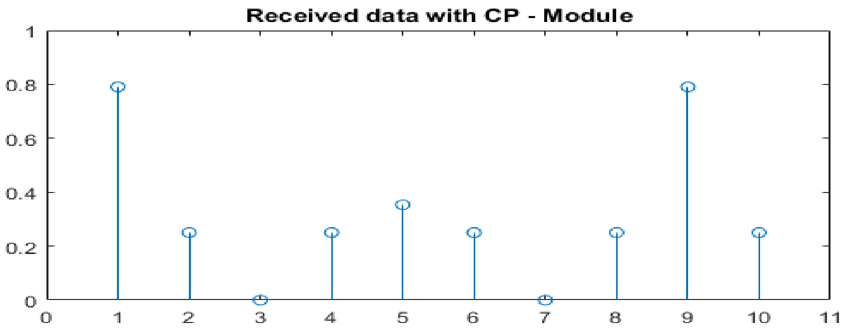

Figure 27. Module of the first 10 samples of received data

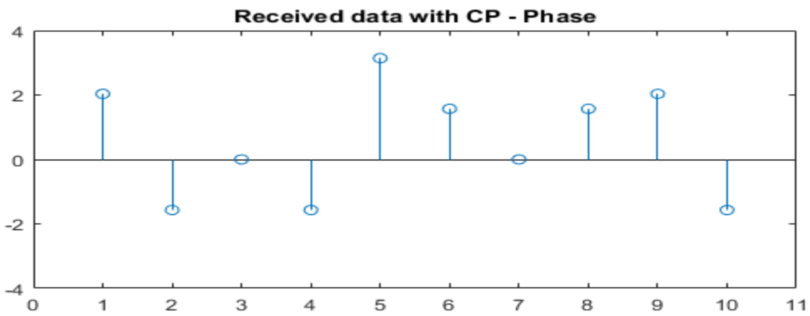

Figure 28. Phase of the first 10 samples of received data

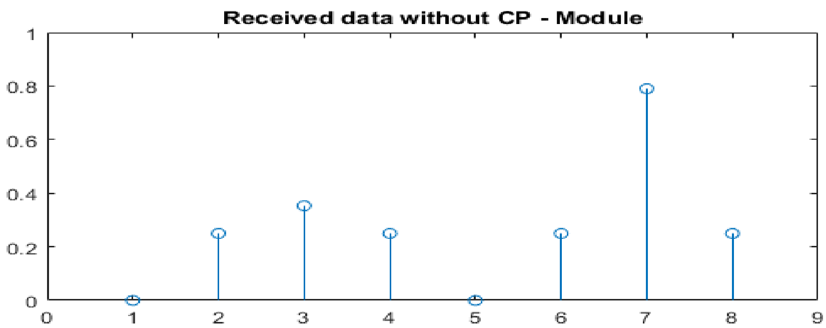

Figure 29. Module of First 8 samples of received data after removing CP

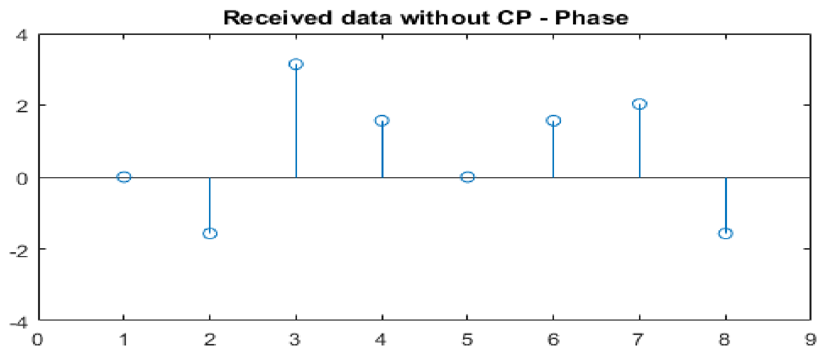

Figure 30. Phase of First 8 samples of received data after removing CP

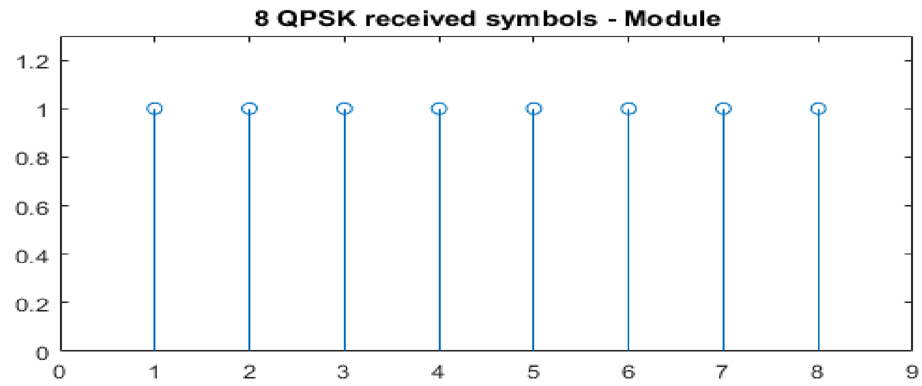

Figure 31. Module of First 8 QPSK received symbols after FFT demodulation

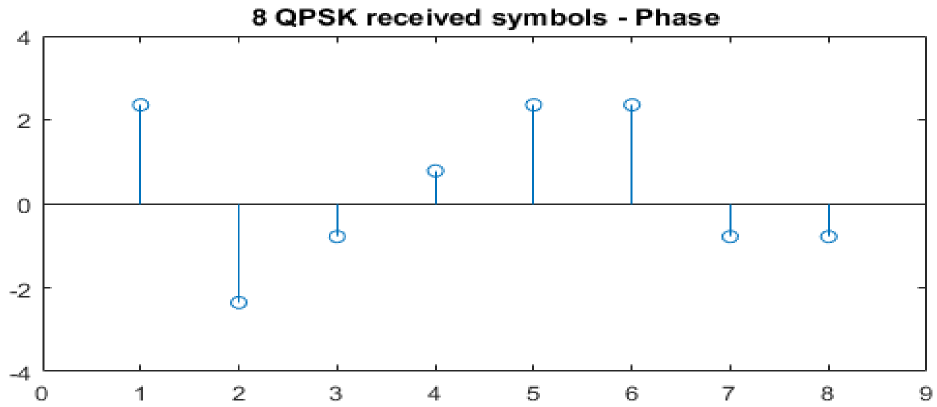

Figure 32. Phase of First 8 QPSK received symbols after FFT demodulation

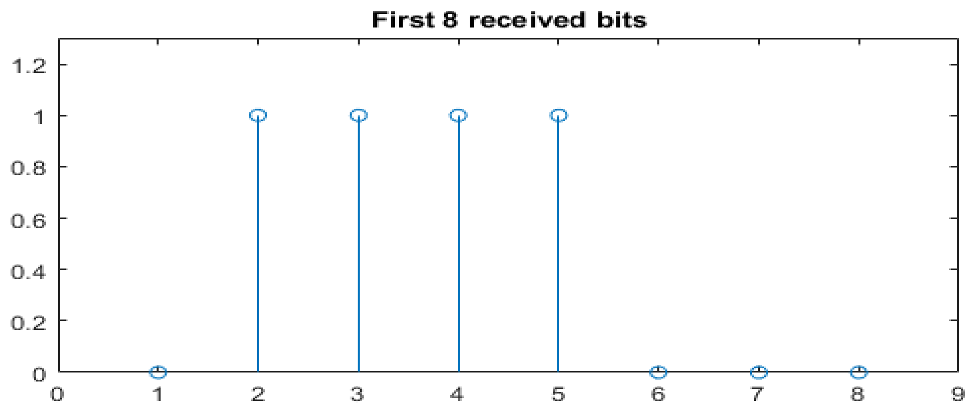

Figure 33. First 8 demodulated bits from the first 4 received data symbols

As it can be seen, the demodulated bits are exactly the same than the transmitted bits. Therefore, the demodulator is working correctly for an ideal channel.

5.4. CHANNEL SIMULATION

The signal $s(t)$ will pass through an underwater acoustical channel whose impulse response can be written as follows:

$$c(t,n) = \sum_{p} A_p(n)\delta(t + a(n)t - \tau_p(n))$$

The impulse response $c(t,n)$ is the one observed during the *n*th OFDM symbol, in the interval $t \in (nT', nT' + T')$. $A_p(n)$ represent the path gains, $a(n)$ is the Doppler rate and $\tau_p(n)$ is delay spread for each path.

The corresponding transfer function is given by:

$$C(f,n) = \sum_{p} A_p(n)e^{-j2\pi f\tau_p(n)/(1+a(n))}$$

In order to simplify the implementation, a channel simulator has been implemented with the following characteristics:

Maximum number of paths	4
Delay of first path	0 s
Maximum delay spread	0.2 ms
Transmitter CP interval	0.3 ms
Channel gain of first path	1
Maximum channel gains	1±0.06
Maximum Doppler rate	±0.001
σ_z^2 Noise	10^{-8}

An example of channel impulse response without Doppler can be seen in Figure 34 where several paths gains and delays can be appreciated.

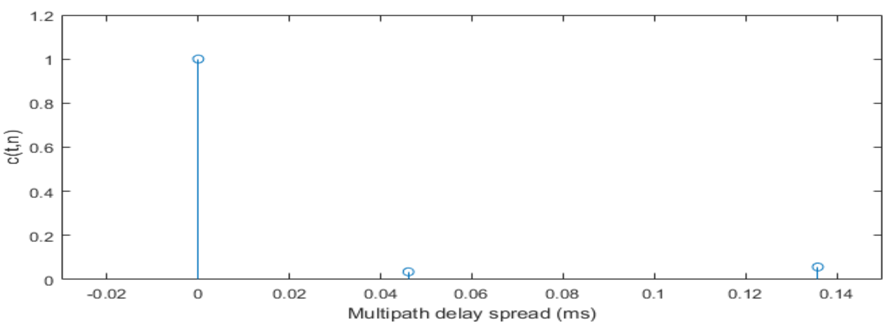

Figure 34. Channel impulse response with multipath

Due to the channel properties, the channel gain varies slowly from symbol to symbol but the Doppler rate may vary faster, which will produce a change on the phase that cannot be compensated with the Cyclic Prefix so that an accurate tracking of the phase is needed.

Another example of the effect of the channel is the following, where 4 different subcarriers have been used, in a channel with 2 paths, with T = 1 ms and Tcp = 0.2 ms and a Doppler rate of a = 0.001. In the following figures can be seen the effect of the properties of the channel in the transmitted signal.

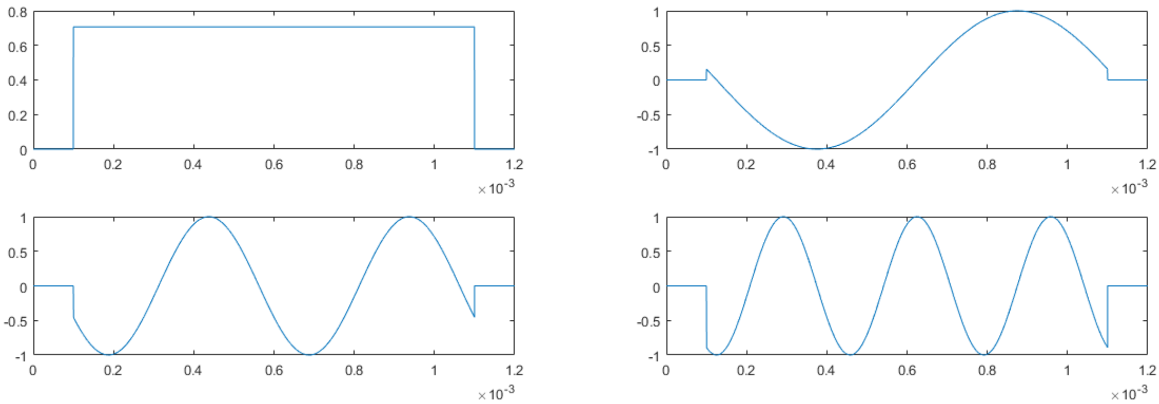

Figure 35. Modulation of 4 subcarriers - Example of channel

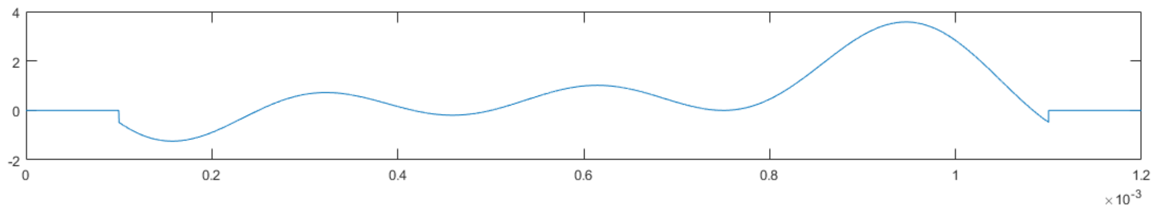

Figure 36. Modulated OFDM Signal with 4 subchannels

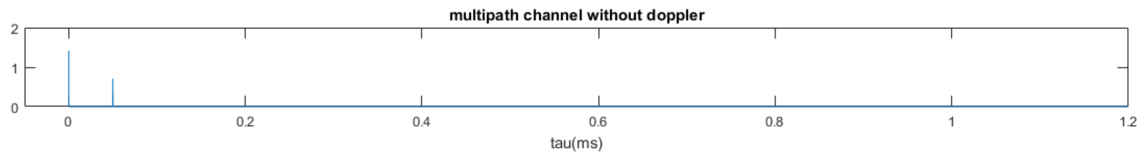

Figure 37. Delay Spread of multipath propagation without Doppler

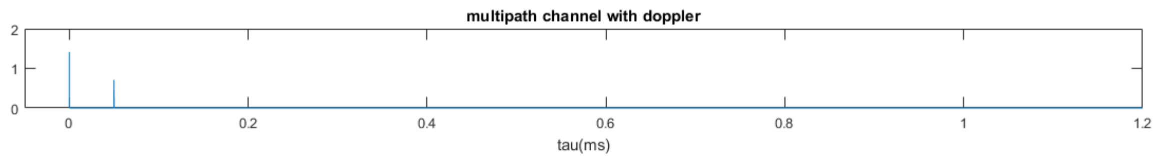

Figure 38. Delay Spread of multipath propagation with Doppler

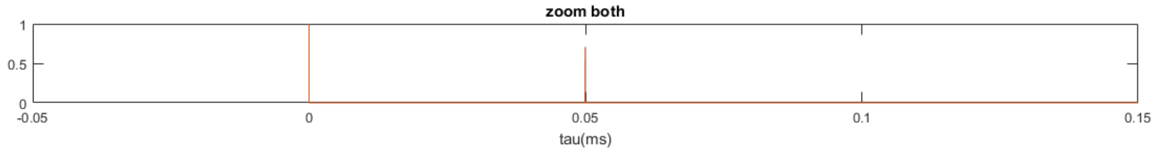

Figure 39. Zoom to the delay spread of both paths

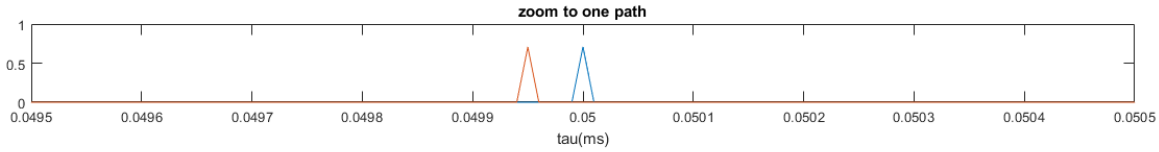

Figure 40. Doppler effect in the delay spread of one path

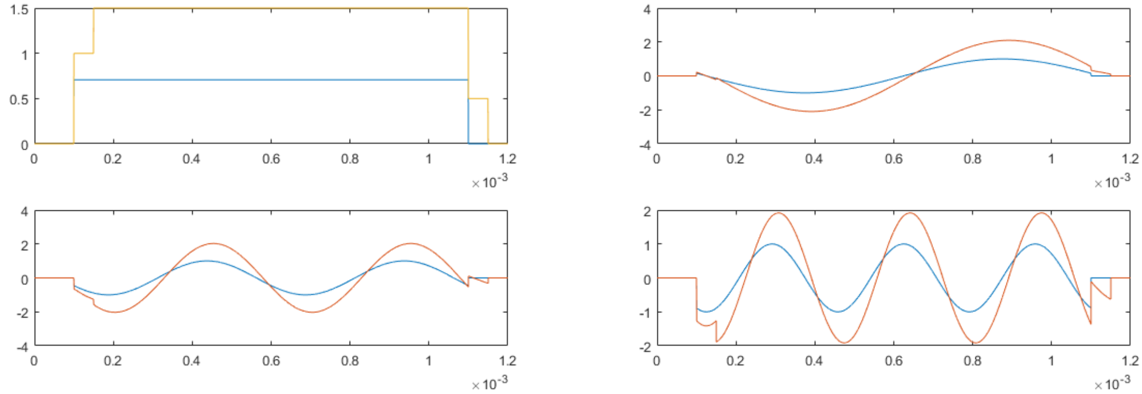

Figure 41. Multipath and Doppler effect in each subcarrier of the transmitted OFDM signal

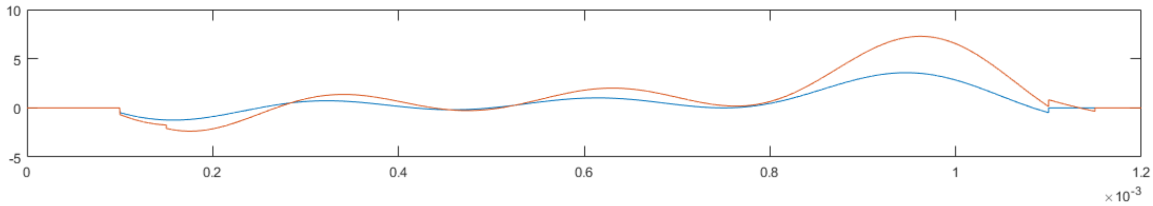

Figure 42. Multipath and Doppler effect in the transmitted OFDM signal

As it can be seen in the figures above, the channel will affect the signal in amplitude and in phase. Thus, if this effect is strong enough to change the phase or the amplitude in a rough way, a good channel estimator is needed.

5.5. RECEIVED SIGNAL FOR UNDERWATER ACOUSTIC CHANNEL

As the demodulation is in the frequency domain, we can express the received signal in terms of the transmitted data symbols and the frequency response of the channel. In order to correctly estimate the channel, the first K symbols transmitted where pilot symbols, corresponding to the bits 00 in the QPSK constellation (i.e. $d_k(1) = \frac{1}{\sqrt{2}} + j\frac{1}{\sqrt{2}}$).

Therefore, at the output of the FFT demodulator, after removing the cyclic prefix, the received signal during the *n*th interval in the *k*th subcarrier can also be expressed as follows:

$$y_k(n) = c_k(n)d_k(n)e^{j\theta_k(n)} + z_k(n)$$

Where $c_k(n) = C(f_k, n)$ is the channel frequency response in the *k*th subcarrier during the *n*th interval and $\theta_k(n) = a(n)\omega_k T'$ is the phase offset. In this case we are assuming that the residual Doppler shift is much less than the subcarrier spacing (i.e. $a(n)f_k \ll \Delta f$), resulting in a low ICI.

The first term of the expression contains the desired information on the symbol $d_k(n)$, and the second term includes the residual ICI and noise.

Since the channel amplitudes vary slowly, the major distortion of the received signal remains in the time-varying phase offset $\theta_k(n)$, which is not uniform across sub-bands.

In the following section the algorithm and working flow of a Low-Complexity detector is described, which is able to estimate the channel adaptively and decides the estimated symbol according to the ML decision rule.

5.6. DETECTOR ALGORITHM

The detector algorithm implemented in this project for OFDM signal detection in non-uniform Doppler-distorted and time-varying multipath channel is based on *[1]*. In the referenced document, the receiver performs the MMSE (*Minimum Mean Square Error*) combination of signals received across an array. In this case, for simplicity, we consider the reception of a single signal.

The use of IFFT/FFT modulator/demodulator allows the channel equalization in the frequency domain, which is easier than the equalization in time domain. As the channel is time-varying, the channel estimation must be adaptive.

OFDM systems are extremely sensitive to any frequency offset. In fact, a frequency offset can be only tolerable if it is much smaller than the carrier spacing (i.e. $\Delta f \gg af_k$). The frequency offset can lead to a loss of orthogonality between subcarriers and the resulting ICI. Therefore, assuming that the only shift in frequency is produced due to motion, one of the main tasks of the equalizer must be tracking continuously the Doppler Effect (concretely, the Doppler Rate $a_k(n)$) and compensate its non-uniform effect across sub-bands.

On the other hand, it is well known that the channel introduce severe multipath propagation that produce a delay spread. However, this effect is removed by the use of the Cyclic Prefix.

The MMSE estimator yields to a data estimate of:

$$\hat{d}_{k0} = \gamma_k(n)c_k^{'}(n)y_k(n)\, e^{-j\theta_k(n)}$$

Where $\gamma_k(n) = (\sigma_z^2 + c_k^{'}(n)c_k(n))^{-1}$ and the prime denotes conjugate in this case.

In practice, the channel $c_k(n)$ and the phase offset $\theta_k(n)$ are not known. Hence, they have to be estimated in some way.

On the one hand, the channel estimation can be expressed in terms of a phase estimate $\hat{\theta}_k(n)$ and a data symbol decision $\tilde{d}_k(n)$ instead of the true values. The channel estimation will also depend on a factor $\lambda < 1$. Then, the channel estimation can be written as:

$$\hat{c}_k(n) = \lambda\hat{c}_k(n-1) + (1-\lambda)y_k(n)e^{-j\hat{\theta}_k(n)}\tilde{d}_k^*(n)$$

As it can be seen from the expression above, the channel estimate in one interval will depend on the channel estimate of the previous interval and on a phase estimate of the current interval. This is because we are assuming that the channel varies slowly, but the phase may vary faster.

Once the channel estimate is available, it can be use to estimate the data symbol $\hat{d}_{k0}$ according to the abovementioned expression.

Then, as we are assuming that the channel gain varies slowly from one OFDM symbol to another, the previous estimate of the channel can be used to estimate the current data symbol, so that we are one step closer to calculate the term $\tilde{d}_k^*(n)$ of the channel estimate expression. Thus, the estimate of the current data symbol can be written as:

$$\hat{d}_k(n) = \hat{\gamma}_k(n-1)\hat{c}_k'(n-1)y_k(n)e^{-j\hat{\theta}_k(n)}$$

In order to compute the expression for $\hat{d}_k(n)$, an estimate of the phase offset $\hat{\theta}_k(n)$ is needed. The phase offset can be defined as:

$$\theta_k(n) = \theta_k(n-1) + a(n)\omega_k T'$$

And therefore its estimate is defined as:

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

Then, the estimate of the Doppler rate $\hat{a}(n)$ needed for the estimate of the phase offset is not trivial because it may change considerably from one OFDM symbol to another. Thus, in order to make a reliable estimation, the estimate of the data symbol must be performed in two ways:

- Estimation using $\hat{\theta}_k(n-1)$:

$$\hat{d}_{k1}(n) = \hat{\gamma}_k(n-1)\hat{c}_k'(n-1)y_k(n)e^{-j\hat{\theta}_k(n-1)}$$

- Estimation using a prediction $\breve{\theta}_k(n)$ of the phase depending on $\hat{a}(n-1)$:

$$\hat{d}_{k2}(n) = \hat{\gamma}_k(n-1)\hat{c}_k'(n-1)y_k(n)e^{-j\breve{\theta}_k(n)} = \hat{d}_{k1}(n)e^{-j\hat{a}(n-1)\omega_k T'}$$

Where

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

Afterwards, the angular offset of $\hat{d}_{k1}(n)$ is measured with the pilot symbols for the first OFDM symbol and with tentative decisions of $\hat{d}_{k2}(n)$.

The tentative decisions can be represented as:

$$\bar{d}_k(n) = \begin{cases} d_k(n), & if\ d_k(n)\ is\ a\ pilot\ symbol \\ decision\left[\hat{d}_{k2}(n)\right], & otherwise \end{cases}$$

Where the decision criterion has been explained in Section 5.3.

Then, the angular offset of $\hat{d}_{k1}(n)$ is calculated as:

$$\Delta\hat{\theta}_k(n) = \langle \hat{d}_{k1}(n)\bar{d}_k^*(n) \rangle$$

And consequently, as we assume that phase change is only caused by motion, the angular offset calculated will contain the Doppler information for the *k*th sub-band in the *n*th interval. Since the Doppler rate is constant during one OFDM symbol, it can be estimated as the arithmetic mean of the values in each subcarrier:

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

Then, returning to the equations where the estimate of the Doppler rate $\hat{a}(n)$ was needed, now we can substitute it and calculate the phase estimate:

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

Then the estimation of the current data symbol can be compute with the value of the phase estimate in the current interval and the estimate of the channel in the previous interval as:

$$\hat{d}_k(n) = \hat{\gamma}_k(n-1)\hat{c}_k'(n-1)y_k(n)e^{-j\hat{\theta}_k(n)} = \hat{d}_{k1}(n)e^{-j\hat{a}(n)\omega_k T'}$$

The final symbol decisions $\tilde{d}_k(n)$ needed to calculate the channel estimate for the current interval are based on the data estimates $\hat{d}_k(n)$ as:

$$\tilde{d}_k(n) = \begin{cases} d_k(n), & if\ d_k(n)\ is\ a\ pilot\ symbol \\ decision\left[\hat{d}_k(n)\right], & otherwise \end{cases}$$

Again, the decision rule is the same as in previous symbol decisions (i.e. ML rule).

And, finally, the channel estimate can be updated and the current data symbol can be correctly estimated and decided.

The algorithm is initialized with the following values: $\hat{c}_k(1) = y_k(1)d_k^*(1)$, $\hat{a}(1) = 0$ and $\hat{\theta}_k(1) = 0$.

5.7. PERFORMANCE ANALYSIS

In order to measure the quality of the algorithm, an OFDM system with the following characteristics has been tried:

Number of data bits transmitted	16384
Number of QPSK data symbols	8192
Number of Pilot Symbols	128
Number of Subcarriers	128
Number of intervals	65
Bandwidth	24 KHz
Guard interval Tcp	0.3 ms
Symbol duration T	5.3 ms
Maximum Doppler Rate	0.001
σ_z^2 Noise	10^{-8}

In the following figures the different steps are shown.

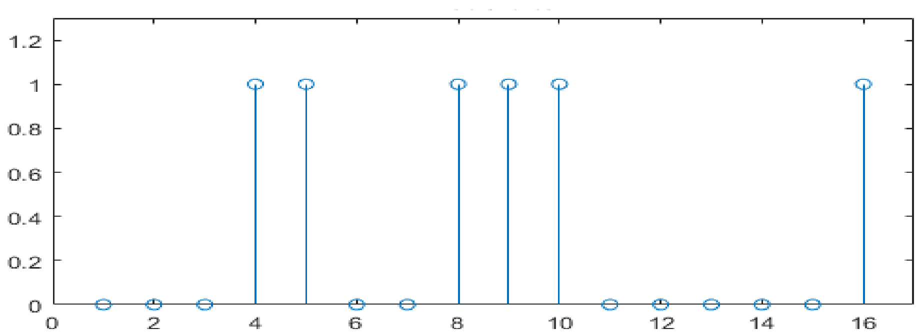

Figure 43. First 16 bits transmitted

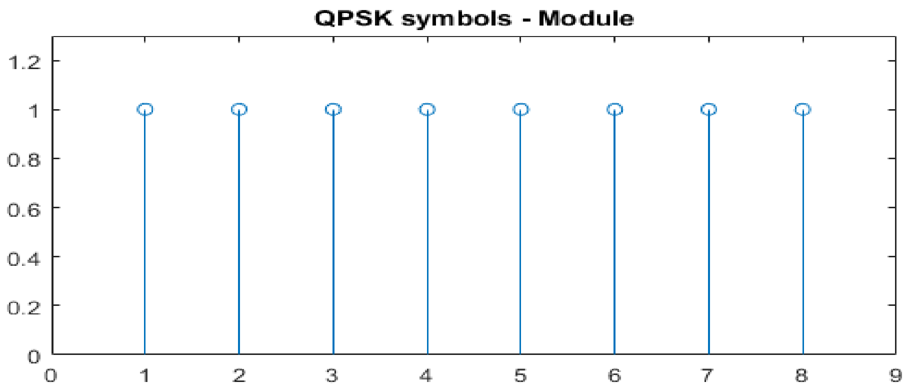

Figure 44. Module of first 8 QPSK transmitted symbols

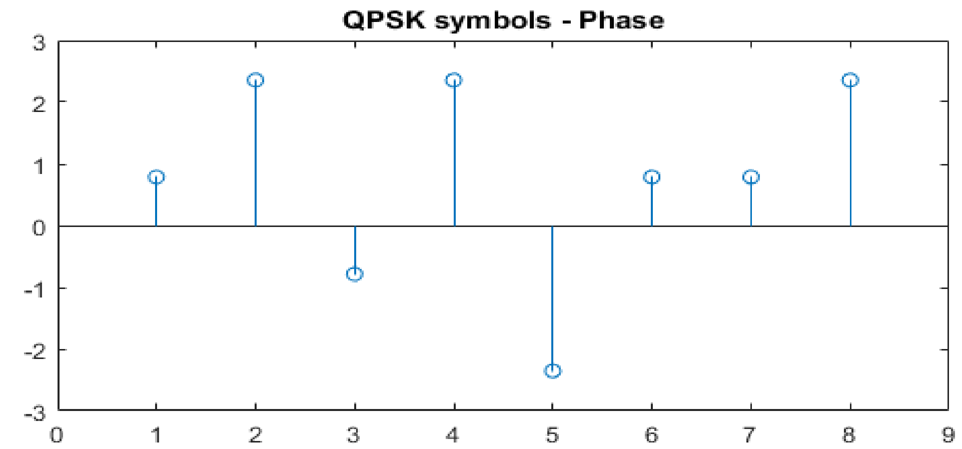

Figure 45. Phase of first 8 QPSK transmitted symbols

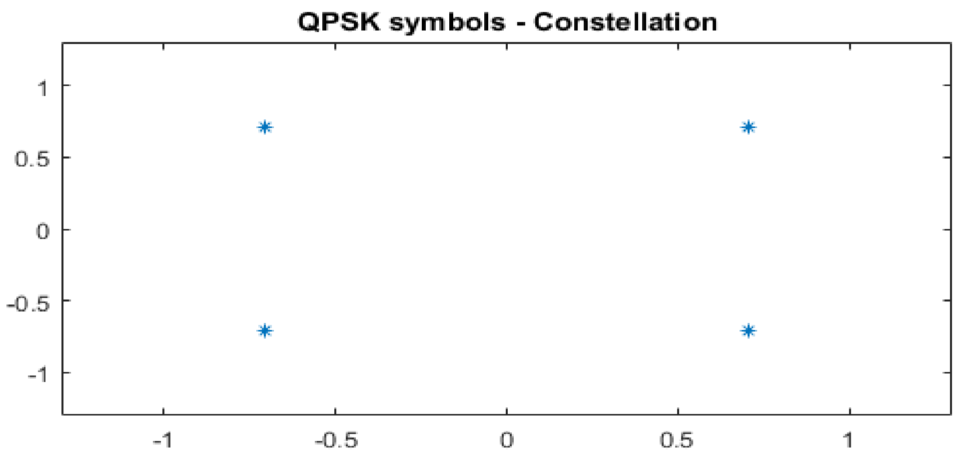

Figure 46. QPSK constellation of first 8 transmitted symbols

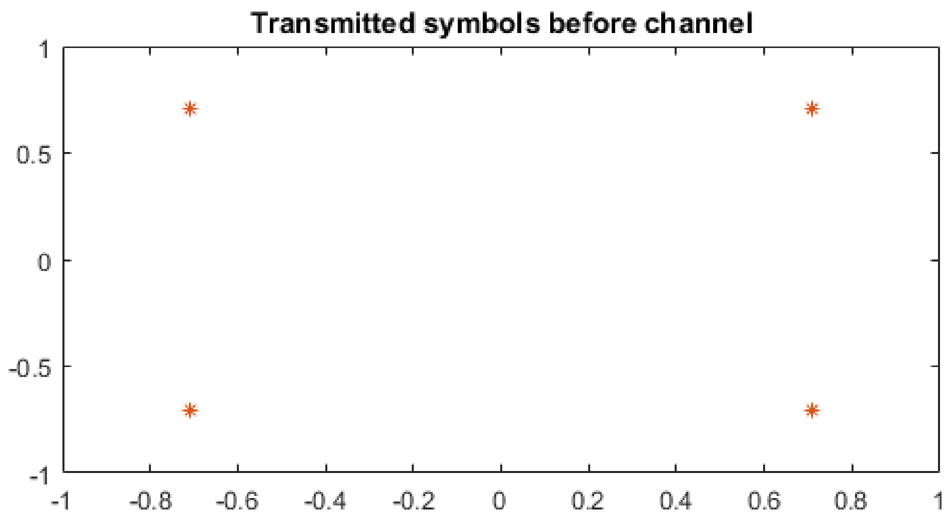

Figure 47. QPSK constellation of all transmitted symbols before the channel

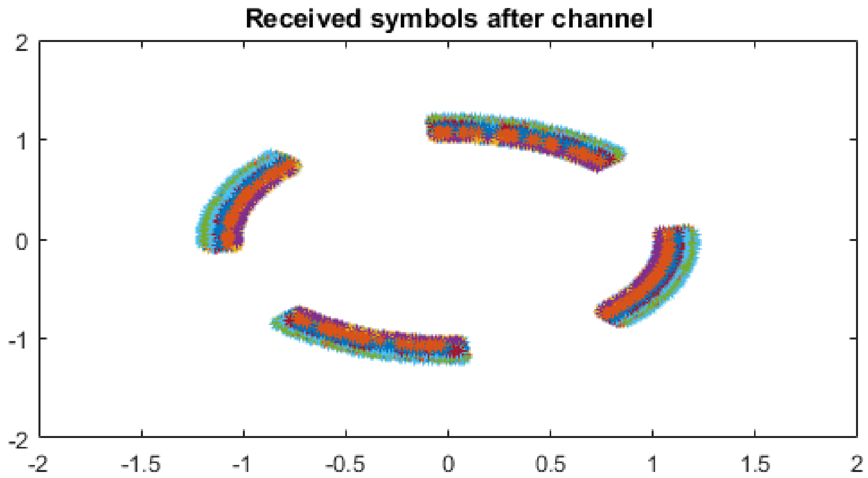

Figure 48. QPSK constellation of all received symbols after the channel

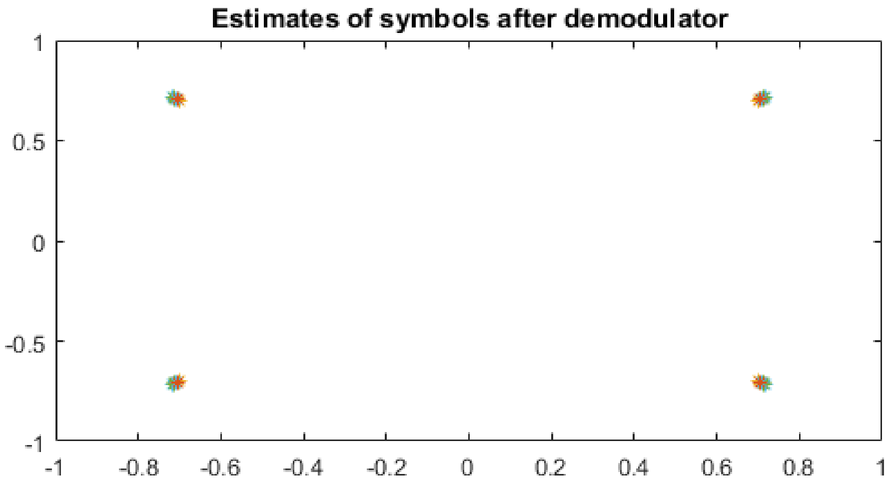

Figure 49. QPSK constellation of the estimates of the symbols after the demodulator

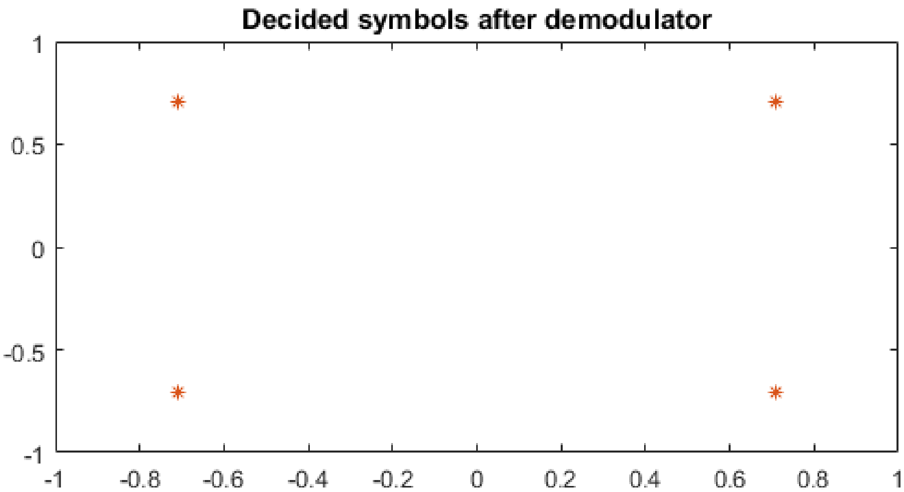

Figure 50. QPSK constellation of the decided symbols after the demodulator

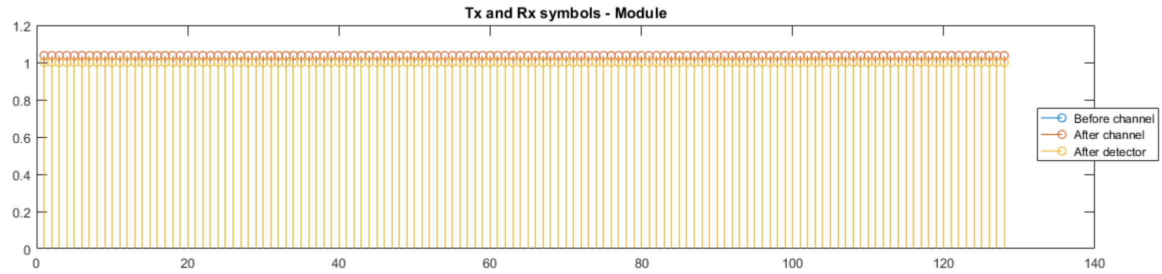

Figure 51. Module of Transmitted and Received pilot symbols

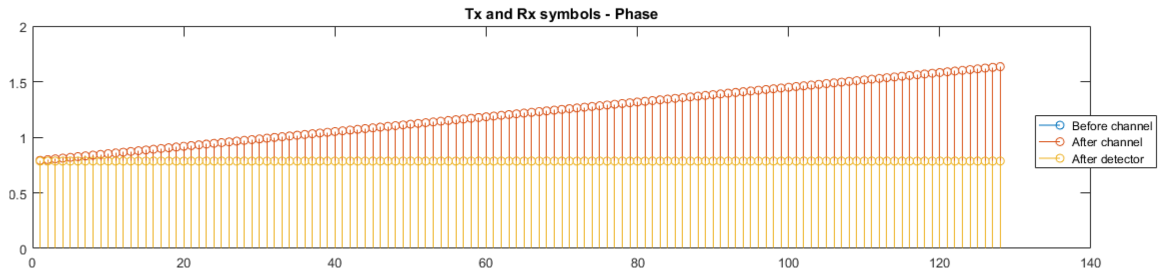

Figure 52. Phase of Transmitted and Received pilot symbols

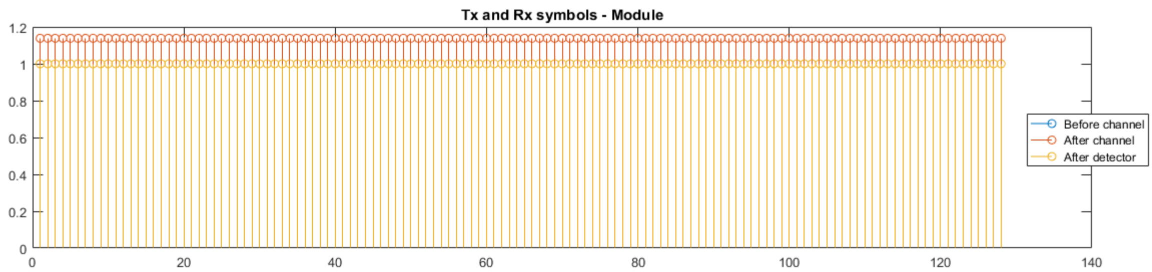

Figure 53. Module of First Tx and Rx data symbols in the first OFDM symbol

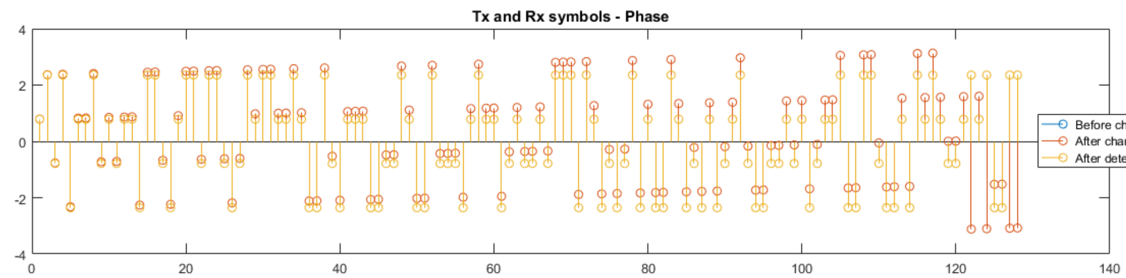

Figure 54. Phase of First Tx and Rx data symbols in the first OFDM symbol

It can be appreciated in Figure 52 the severe effect of the non-uniform Doppler distortion across sub-bands.

In the following figure, it can be seen the symbol errors that have appeared without including the channel estimation and the corresponding data decisions. It can be appreciated that there are no errors if the detector is included.

However, the algorithm could be improved and a better channel simulator can be used in order to simulate a more realistic environment.

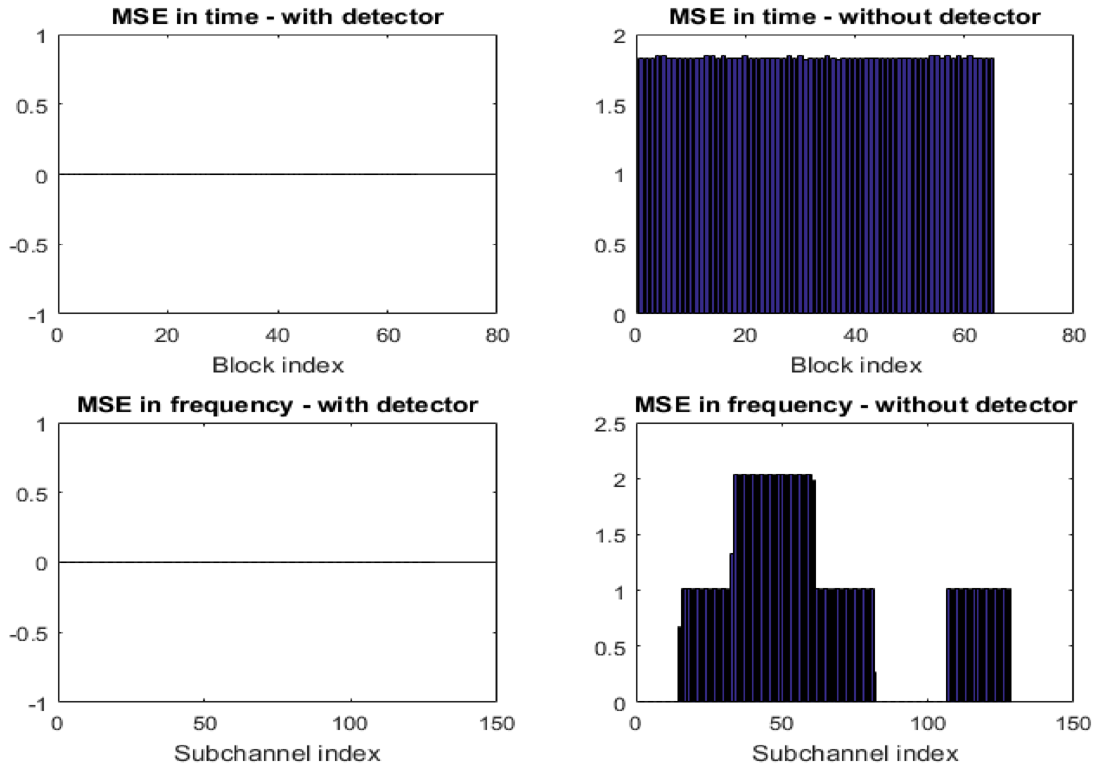

Figure 55. Performance analysis with and without detector

6. CONCLUSIONS

A low complexity detector which performs adaptive channel estimation for an OFDM system in an underwater acoustical channel has been implemented using MATLAB® software. The equalization is made in the frequency domain in order to reduce the complexity. The algorithm still has to be improved and a more sophisticated channel simulator has to be used in order to analyze a more realistic environment.

It has been seen that the major distortion was due to the time varying phase offset produced by the motion of the transmitter or the receiver. Thus, an accurate phase tracking is needed in order to compensate the fast channel variations. This phase has to be tracked for every subcarrier of the OFDM system since the system is considered wideband, and the underwater channel is frequency selective, inducing a non-uniform Doppler distortion across sub-bands.

7. FUTURE STEPS

In the following list some future work that can be done is presented:

- Improving the algorithm for a more realistic environment.
- A receiver array can be used in order to improve the channel estimation.
- An analysis of the correlation in time and in frequency of the channel can be performed in the receiver. In this case a real environment or a more complex channel simulator is needed.
- Try other modulation formats (M-PSK or M-QAM).
- Testing the algorithm in a real environment, which would need to:
    - Adjust the algorithm for the real test, by improving the algorithm and maybe change the programming language.
    - Build an interface between the software and the real hardware
    - Try different scenarios (shallow water, static transmitter and receiver, AUVs, etc.)
    - Study if more pilot symbols are needed depending on the properties of the real underwater acoustic channel.
    - Study the variations of the real channel during the transmission of one OFDM symbol.
    - Characterization of the noise based in measurements.

8. BIBLIOGRAPHY

[1] – Milica Stojanovic, "Low complexity OFDM detector for underwater Acoustic Channels", *IEEE Oceans'06 Conf.*, Sept. 2006.

[2] – Mandar Chitre, Shiraz Shahabudeen, Lee Freitag, Milica Stojanovic, "Recent advances in underwater acoustic communications and networking" *IEEE Oceans '08 Conf.*, Sept. 2008.

[3] – Trond Jenserud, Paul van Walree and Roald Otnes, "An underwater acoustic channel simulator", *NTNU*, 2012.

[4] – Paul van Walree, "Channel sounding for acoustic communications: Techniques and shallow water examples", *FFI*, 2011.

[5] – Baosheng Li, Jie Huang, Shengli Zhou, Keenan Ball, Milica Stojanovic, Lee Freitag, Peter Willett, "MIMO OFDM for High-Rate Underwater Acoustic Communications", *IEEE Journal of Oceanic Engineering, vol.34 No.4, pp.634-644*, Oct. 2009.

[6] – Milica Stojanovic, James Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization", *IEEE Communications Magazine, vol. 47, No.1, pp.84-89*, Jan. 2009.

[7] – O. Edfors, M. Sandell, J. –J. van de Beek, S.K. Wilson, P.O. Borjesson, "OFDM channel estimation by singular value decomposition", *IEEE Transactions on Communications, vol.46, No.7, pp.931-939*, Jul. 1998.

[8] – Parastoo Qarabaqi, Milica Stojanovic, "Statistical characterization and computationally efficient modeling of a class of underwater acoustical communication channels", *IEEE Journal of Oceanic Engineering, Vol. 38, No.4, pp.701-717*, Oct. 2013.

[9] – Yashar M. Aval, Milica Stojanovic, "Differentially coherent multichannel detection of acoustic OFDM signals", *IEEE Journal of Oceanic Engineering, Vol.40, No.2, pp.251-268*, April 2015.

[10] – J. –J. van de Beek, O. Edfors, M. Sandell, S.K. Wilson, P.O. Borjesson, "On channel estimation in OFDM systems", *IEEE, Vehicular Technology Conference, 1995, IEEE 45th*.

[11] – Sandell, M., & Edfors, O. (1996). "A comparative study of pilot-based channel estimators for wireless OFDM", *(Div. of Signal Processing, Research Report; Vol. TULEA 1996:19)*, Luleå University of Technology.

[12] – Ove Edfors, Magnus Sandell, Jan-Jaap van de Beek, Daniel Landstrom, Frank Sjoberg, "An introduction to orthogonal frequency division-multiplexing", Sept. 1996.

[13] – Hyungu Hwang, Hyoungjun Park, "Doppler frequency offset estimation in OFDM systems", *IEEE Wireless Pervasive Computing, 2009. ISWPC 2009. 4th International Symposium.*

[14] – Hao Zhou, A.V. Malipatil, Yih-Fang Huang, "Maximum-Likelihood Carrier Frequency Offset Estimation for OFDM Systems in Fading Channels", *IEEE Wireless Communications and Networking Conference*, 2006.

[15] – Byung-Chul Kim, I-Tai Lu, "Parameter study of OFDM underwater Communications System", *IEEE Oceans '00, MTS/IEEE Conference and Exhibition*.

ANNEX 1 – MATLAB CODE – OFDM SYSTEM WITH IDEAL CHANNEL

- *main1.m*

```
% Executable 1
tic
clc, clear all, close all

% Parameter setting

nbits = 1024;      % Number of bits to be transmitted
data = round(rand(1,nbits));  % total data in bits to be transmitted

Nsubcarriers = 8;    % Number of OFDM subcarriers - Power of 2
B = 24e3;    % Bandwidth
T = Nsubcarriers/B;     % OFDM symbol period without cyclic prefix
Tg = 0.5e-4;
delta_f = 1/T;  % Separation between subcarriers

cp_len = ceil(Nsubcarriers*Tg/T); % Length of cyclic prefix in symbols -
Depends on the channel

% TRANSMITTER

transmitted_signal = tx_system (data,Nsubcarriers,cp_len,T);  %
Transmitted signal before channel

received_signal = transmitted_signal;   % Received signal for ideal
channel

toc
tic
demodulated_data = rx_system(received_signal,Nsubcarriers,cp_len,nbits);
% Demodulated signal after receptor
toc
find(data~=demodulated_data)
```

- *tx_system.m*

```
function [ output_data ] = tx_system( input_data, Nsubcarriers,cp_len,T)
%computes the baseband transmitted digital data before the D/A converter
and the channel

% Modulation of input_data into symbols (QPSK)

data_symbols = QPSK(input_data_zp); % Conversion of bits into QPSK
symbols

% Plotting 4 first symbols

figure('name','Step 1 - bits to symbols QPSK')
    subplot(2,2,1)
    stem(1:8,input_data(1:8))
    axis([0 9 0 1.3])
    title('First 8 bits')
    subplot(2,2,2)
    plot(data_symbols(1,1:4),'*')
    axis([-1.3 1.3 -1.3 1.3])
    title('QPSK symbols - Constellation')
    subplot(2,2,3)
    stem(abs(data_symbols(1,1:4)))
    axis([0 5 0 1.3])
    title('QPSK symbols - Module')
    subplot(2,2,4)
    stem(angle(data_symbols(1,1:4)))
    axis([0 5 -3 3])
    title('QPSK symbols - Phase')

% Conversion of serial input data into parallel

nsymbols = length(data_symbols);
n = nsymbols/Nsubcarriers; % Number of symbols per subcarrier
cp_data = [];
no_cp_data = [];

for j = 1:n
    new_data(:,j) = data_symbols(Nsubcarriers*(j-1)+1:Nsubcarriers*j);

    % IFFT of the input symbols
    data_ifft(j,:) = IFFT(new_data(:,j),Nsubcarriers);
    no_cp_data = [no_cp_data data_ifft(j,:)];

    % Including cyclic prefix
    cp(j,:) = [data_ifft(j,end-cp_len+1:end) data_ifft(j,:)];
    cp_data = [cp_data cp(j,:)];
end
```

```
figure('name', 'First OFDM data symbol')
    subplot(2,2,1)
    stem(1:length(new_data(1:8,1)),abs(new_data(1:8,1)))
    title('QPSK - Module')
    subplot(2,2,2)
    stem(1:length(new_data(1:8,1)),angle(new_data(1:8,1)))
    title('QPSK - Angle')
    subplot(2,2,3)

stem(T/Nsubcarriers.*(1:length(data_ifft(1,1:8))),abs(data_ifft(1,1:8)))
    title('IFFT - Module')
    subplot(2,2,4)

stem(T/Nsubcarriers.*(1:length(data_ifft(1,1:8))),angle(data_ifft(1,1:8)
))
    title('IFFT - Angle')

figure('name', 'Step 2 - symbols QPSK to IFFT')
    subplot(4,2,1)
    stem(abs(data_symbols(1,1:Nsubcarriers)))
    axis([0 Nsubcarriers+1 0 1.3])
    title('QPSK symbols - Module')
    subplot(4,2,2)
    stem(angle(data_symbols(1,1:Nsubcarriers)))
    axis([0 Nsubcarriers+1 -3.5 3.5])
    title('QPSK symbols - Phase')
    subplot(4,2,3)
    stem(abs(data_ifft(1,1:Nsubcarriers)))
    axis([0 Nsubcarriers+1 0 1.3])
    title('IFFT symbols - Module')
    subplot(4,2,4)
    stem(angle(data_ifft(1,1:Nsubcarriers)))
    axis([0 Nsubcarriers+1 -4 4])
    title('IFFT symbols - Phase')
    subplot(4,2,5)
    stem(abs(cp_data(1,1:Nsubcarriers+cp_len)))
    axis([0 Nsubcarriers+cp_len+1 0 1.3])
    title('IFFT symbols with CP - Module')
    subplot(4,2,6)
    stem(angle(cp_data(1,1:Nsubcarriers+cp_len)))
    axis([0 Nsubcarriers+cp_len+1 -4 4])
    title('IFFT symbols with CP - Phase')

    g = 2^nextpow2(length(cp_data(1,1:Nsubcarriers+cp_len)));
    fft_cp_data = fft(cp_data(1,1:Nsubcarriers+cp_len),g);
    subplot(4,2,7)
    stem(abs(fft_cp_data))
    axis([0 g 0 2.5])
    title('FFT symbols with CP - Module')
    subplot(4,2,8)
    stem(angle(fft_cp_data))
    axis([0 g -4 4])
    title('FFT symbols with CP - Phase')

ofdm_signal = cp_data;

output_data = ofdm_signal;
```

```
figure('name', 'OFDM without cyclic prefix')
subplot(4,1,1)
plot(abs(no_cp_data))
axis([0 512 0 0.8])
title('Module IFFT')
subplot(4,1,2)
plot(angle(no_cp_data))
axis([0 512 -5 5])
title('Angle IFFT')

fft_no_cp_data = fft(no_cp_data);
subplot(4,1,3)
plot(abs(fft_no_cp_data))
axis([0 512 0 13])
title('Module FFT')
subplot(4,1,4)
plot(angle(fft_no_cp_data))
axis([0 512 -5 5])
title('Angle FFT')

figure('name','OFDM signal')
subplot(2,1,1)
plot(abs(ofdm_signal))
axis([0 n*(Nsubcarriers+cp_len) 0 0.8])
title('Complete OFDM signal - Module')
ylabel('Y_O_F_D_M')
subplot(2,1,2)
plot(angle(ofdm_signal))
axis([0 n*(Nsubcarriers+cp_len) -4 4])
title('Complete OFDM signal - Phase')
ylabel('Y_O_F_D_M')

figure('name', 'OFDM Spectrum')
subplot(4,1,1)
plot(abs(fft(ofdm_signal)))
title('Spectrum ofdm signal - Module')
subplot(4,1,2)
plot(angle(fft(ofdm_signal)))
title('Spectrum OFDM signal - Phase')
subplot(4,1,3)
plot(abs(fft(ofdm_signal,2^nextpow2(length(ofdm_signal)))))
title('Spectrum OFDM signal with more samples - Module')
subplot(4,1,4)
plot(angle(fft(ofdm_signal,2^nextpow2(length(ofdm_signal)))))
title('Spectrum OFDM signal with more samples - Angle')

end
```

- *QPSK.m*

```
function [ y ] = QPSK( x )
% Modulation of input x into QPSK symbols y

nbits = length(x);

    for n = 1:floor(nbits/2);

        bits = [x(2*(n-1)+1) x(2*(n-1)+2)];

        if bits(1) == 0 && bits(2) == 0
            y(n) = (+1+j)/sqrt(2);
        elseif bits(1) == 0 && bits(2) == 1
            y(n) = (-1+j)/sqrt(2);
        elseif bits(1) == 1 && bits(2) == 1
            y(n) = (-1-j)/sqrt(2);
        elseif bits(1) == 1 && bits(2) == 0
            y(n) = (+1-j)/sqrt(2);
        end
    end

end
```

- *IFFT.m*

```
function [ y ] = IFFT(x,N)
%IFFT computes de Inverse Fast Fourier Transform algorithm of N points

    y = ifft(x,N);

end
```

- *rx_system.m*

```
function [ y ] = rx_system( x, Nsubcarriers, cp_len, nbits )

% 'rx_system' demodulates the received signal after the channel
% transformation of the transmitted signal

% Plot received signal
figure('name', 'Received Signal')
subplot(2,1,1)
plot(abs(x))
ylabel('Y_O_F_D_M (t)')
title('OFDM received Signal - Module')
subplot(2,1,2)
plot(abs(x))
ylabel('Y_O_F_D_M (t)')
title('OFDM received Signal - Phase')

% Removing cyclic prefix
m = length(x)/(Nsubcarriers+cp_len);
for i = 1:m
    % Serial to parallel
    x_paralel_with_CP(i,:) = x((Nsubcarriers+cp_len)*(i-
1)+1:(Nsubcarriers+cp_len)*i);
end

x_paralel_without_CP = x_paralel_with_CP(:,cp_len+1:end);

% Plotting first OFDM symbol
figure('name','Removing cyclic prefix')
subplot(2,2,1)
stem(abs(x_paralel_with_CP(1,:)))
axis([0 Nsubcarriers+cp_len+1 0 1])
title('Received data with CP - Module')
subplot(2,2,2)
stem(angle(x_paralel_with_CP(1,:)))
axis([0 Nsubcarriers+cp_len+1 -4 4])
title('Received data with CP - Phase')
subplot(2,2,3)
stem(abs(x_paralel_without_CP(1,:)))
axis([0 Nsubcarriers+1 0 1])
title('Received data without CP - Module')
subplot(2,2,4)
stem(angle(x_paralel_without_CP(1,:)))
axis([0 Nsubcarriers+1 -4 4])
title('Received data without CP - Phase')

% FFT
data_qpsk = [];
for i = 1:m
    data_fft(:,i) = FFT(x_paralel_without_CP(i,:),Nsubcarriers);
    data_qpsk = [data_qpsk data_fft(:,i).'];
end

data_bits(1,:) = demod_QPSK(data_qpsk(1,:));
```

```
y = data_bits(1:nbits);

figure('name', 'First demodulated symbols')
subplot(2,2,1)
stem(abs(x_paralel_without_CP(1,:)))
axis([0 Nsubcarriers+1 0 1])
title('8 received symbols before FFT  - Module')
subplot(2,2,2)
stem(angle(x_paralel_without_CP(1,:)))
axis([0 Nsubcarriers+1 -4 4])
title('8 received symbols before FFT - Phase')
subplot(2,2,3)
stem(abs(data_qpsk(1,1:Nsubcarriers)))
axis([0 Nsubcarriers+1 0 1.3])
title('8 QPSK received symbols - Module')
subplot(2,2,4)
stem(angle(data_qpsk(1,1:Nsubcarriers)))
axis([0 Nsubcarriers+1 -4 4])
title('8 QPSK received symbols - Phase')

figure('name', 'First demodulated data')
subplot(2,2,1)
stem(abs(data_qpsk(1,1:4)))
axis([0 5 0 1.3])
title('First 4 QPSK received symbols - Module')
subplot(2,2,2)
stem(angle(data_qpsk(1,1:4)))
axis([0 5 -4 4])
title('First 4 QPSK received symbols - Phase')
subplot(2,2,3)
plot(data_qpsk(1,1:4),'*')
axis([-1 1 -1 1])
title('QPSK Constellation')
subplot(2,2,4)
stem(y(1:8))
axis([0 9 0 1.3])
title('First 8 received bits')

end
```

- *FFT.m*

```
function [ y ] = FFT(x,N)
%FFT computes de Fast Fourier Transform algorithm of N points
y = fft(x,N);

end
```

- *demod_QPSK.m*

```
function [ y ] = demod_QPSK( x )
%Demodulation of input x into corresponding bits y
nsymbols = length(x);
y = 0;
for symbol = 1:nsymbols
    if 0 < angle(x(symbol)) && angle(x(symbol))<= pi/2
        bits = [0 0];
    elseif pi/2 < angle(x(symbol)) && angle(x(symbol)) <= pi
        bits = [0 1];
    elseif - pi < angle(x(symbol)) && angle(x(symbol)) <= -pi/2
        bits = [1 1];
    elseif -pi/2 < angle(x(symbol)) && angle(x(symbol)) <= 0
        bits = [1 0];
    end
    y = [y bits];
end
y = y(2:end);
end
```

ANNEX 2 – MATLAB CODE – OFDM SYSTEM WITH UNDERWATER CHANNEL

- *main2.m*

```
% Executable
tic
clc, clear, close all

% Parameter setting

nbits =16384;      % Number of bits to be transmitted
data = round(rand(1,nbits));  % total data in bits to be transmit

Nsubcarriers =128;   % Number of OFDM subcarriers - Power of 2
pilot_bits = zeros(1,2*Nsubcarriers);
data = [pilot_bits data];
B = 24e3;   % Bandwidth
T = Nsubcarriers/B;    % OFDM symbol period without cyclic prefix
Tg = 3e-4;

cp_len = 2; % Length of cyclic prefix in symbols - Depends on the
channel
T_prima = T+Tg;    % Duration of one OFDM symbol (T' = T+Tg)

% System

data_symbols = tx_symbols(data,nbits, Nsubcarriers);
[received_symbols channel_response doppler_rate phase_shift] =
channel(data_symbols, Nsubcarriers,cp_len,T_prima,T);

pilot_symbols= data_symbols(:,1);
disp('Pilot symbols'), disp(data_symbols(:,1))
disp('Channel response in pilot symbols'), disp(channel_response(:,1))
disp('Received pilot symbols'), disp(received_symbols(:,1))
disp('First data symbols'), disp(data_symbols(:,2))
disp('Channel response to first data symbols'),
disp(channel_response(:,2))
disp('Received first data symbols'), disp(received_symbols(:,2))
disp('Doppler rate first data symbols'),disp(doppler_rate(:,2))
disp('Phase shift first data symbols'),disp(phase_shift(:,2))

% No detector

for fk = 1:Nsubcarriers
    decided_symbols_without_detector(fk,:) =
decision_QPSK(received_symbols(fk,:));
end

decided_symbols_without_detector_serie = [];
for L = 1:length(decided_symbols_without_detector(1,:))
    decided_symbols_without_detector_serie =
[decided_symbols_without_detector_serie
decided_symbols_without_detector(:,L).'];
end
```

```
 decided_bits_without_detector =
demod_QPSK(decided_symbols_without_detector_serie);

% Detector
[final_symbols, decided_bits, doppler_rate_estimate, final_estimation] =
detector_algorithm(received_symbols, pilot_symbols,
Nsubcarriers,T_prima,T);

find(decided_bits ~= data)
% Plots
figure('name', 'Final estimation of data symbols')
subplot(2,2,1)
plot(data_symbols,'*')
axis([-1 1 -1 1])
title('Transmitted symbols before channel')
subplot(2,2,2)
plot(received_symbols,'*')
axis([-2 2 -2 2])
title('Received symbols after channel')
subplot(2,2,3)
plot(final_estimation, '*')
axis([-1 1 -1 1])
title('Estimates of symbols after demodulator')
subplot(2,2,4)
plot(final_symbols, '*')
axis([-1 1 -1 1])
title('Decided symbols after demodulator')

figure('name', 'Transmitted and received pilot symbols')
subplot(2,1,1)
stem(abs(data_symbols(:,1)))
hold on
stem(abs(received_symbols(:,1)))
hold on
stem(abs(final_symbols(:,1)))
legend('Before channel','After channel','After detector')
title('Tx and Rx symbols - Module')
subplot(2,1,2)
stem(angle(data_symbols(:,1)))
hold on
stem(angle(received_symbols(:,1)))
hold on
stem(angle(final_symbols(:,1)))
legend('Before channel','After channel','After detector')
title('Tx and Rx symbols - Phase')
```

```
figure('name', 'Transmitted and received first OFDM data symbols')
subplot(2,1,1)
stem(abs(data_symbols(:,2)))
hold on
stem(abs(received_symbols(:,2)))
hold on
stem(abs(final_symbols(:,2)))
legend('Before channel','After channel','After detector')
title('Tx and Rx symbols - Module')
subplot(2,1,2)
stem(angle(data_symbols(:,2)))
hold on
stem(angle(received_symbols(:,2)))
hold on
stem(angle(final_symbols(:,2)))
legend('Before channel','After channel','After detector')
title('Tx and Rx symbols - Phase')

figure('name', 'Transmitted and received second OFDM data symbols')
subplot(2,1,1)
stem(abs(data_symbols(:,3)))
hold on
stem(abs(received_symbols(:,3)))
hold on
stem(abs(final_symbols(:,3)))
legend('Before channel','After channel','After detector')
title('Tx and Rx symbols - Module')
subplot(2,1,2)
stem(angle(data_symbols(:,3)))
hold on
stem(angle(received_symbols(:,3)))
hold on
stem(angle(final_symbols(:,3)))
legend('Before channel','After channel','After detector')
title('Tx and Rx symbols - Phase')

% MSE evaluation with detector
mse_t = mse_time(data_symbols, final_estimation);
mse_f = mse_frequency(data_symbols, final_estimation);

% MSE evaluation without detector
mse_t_no = mse_time(data_symbols, decided_symbols_without_detector);
mse_f_no = mse_frequency(data_symbols,
decided_symbols_without_detector);

n_blocks = length(data_symbols(1,:));

figure('name', 'MSE')

subplot(2,2,1)
bar(1:n_blocks, mse_t)
xlabel('Block index')
title('MSE in time - with detector')
```

```
subplot(2,2,2)
bar(1:n_blocks, mse_t_no)
xlabel('Block index')
title('MSE in time - without detector')

subplot(2,2,3)
bar(1:Nsubcarriers, mse_f)
xlabel('Subchannel index')
title('MSE in frequency - with detector')

subplot(2,2,4)
bar(1:Nsubcarriers, mse_f_no)
xlabel('Subchannel index')
title('MSE in frequency - without detector')

% Errors with detector
number_error_bits = length(find(data ~= decided_bits));
number_error_symbols = length(find(data_symbols ~= final_symbols));
per_bits = number_error_bits/length(data);
per_symbols = number_error_symbols/(0.5*length(data));

% Errors without detector
number_error_bits_no = length(find(data ~=
decided_bits_without_detector));
number_error_symbols_no = length(find(data_symbols ~=
decided_symbols_without_detector));
per_bits_no = number_error_bits_no/length(data);
per_symbols_no = number_error_symbols_no/(0.5*length(data));

n_BER = sprintf('Bit errors with detector: \t%d', number_error_bits);
n_SER = sprintf('Symbol errors with detector: \t%d',
number_error_symbols);
BER = sprintf('BER with detector: \t%d', per_bits);
SER = sprintf('SER with detector: \t%d', per_symbols);

n_BER_no = sprintf('Bit errors without detector: \t%d',
number_error_bits_no);
n_SER_no = sprintf('Symbol errors without detector: \t%d',
number_error_symbols_no);
BER_no = sprintf('BER without detector: \t%d', per_bits_no);
SER_no = sprintf('SER without detector: \t%d', per_symbols_no);

disp(n_BER)
disp(n_SER)
disp(BER)
disp(SER)
disp(sprintf('\n'))
disp(n_BER_no)
disp(n_SER_no)
disp(BER_no)
disp(SER_no)
toc
```

- *tx_symbols.m*

```
function [ output_symbols ] = tx_symbols( input_data,nbits, Nsubcarriers
)
% tx_symbols

 % Modulation of input_data into symbols (QPSK)

data_symbols = QPSK(input_data); % Conversion of bits into QPSK symbols

% Plotting 4 first symbols

figure('name','Step 1 - bits to symbols QPSK')
    subplot(2,2,1)
    stem(1:16,input_data(2*Nsubcarriers+1:2*Nsubcarriers+16))
    axis([0 17 0 1.3])
    title('First 8 bits')
    subplot(2,2,2)
    plot(data_symbols(1,Nsubcarriers+1:Nsubcarriers+8),'*')
    axis([-1.3 1.3 -1.3 1.3])
    title('QPSK symbols - Constellation')
    subplot(2,2,3)
    stem(abs(data_symbols(1,Nsubcarriers+1:Nsubcarriers+8)))
    axis([0 9 0 1.3])
    title('QPSK symbols - Module')
    subplot(2,2,4)
    stem(angle(data_symbols(1,Nsubcarriers+1:Nsubcarriers+8)))
    axis([0 9 -3 3])
    title('QPSK symbols - Phase')

% Conversion of serial input data into parallel

nsymbols = length(data_symbols);
n = nsymbols/Nsubcarriers; % Number of symbols per subcarrier


for j = 1:n
    new_data(:,j) = data_symbols(Nsubcarriers*(j-1)+1:Nsubcarriers*j);
end

output_symbols = new_data;

end
```

- *QPSK.m*

```
function [ y ] = QPSK( x )
%QPSK Modulator

nbits = length(x);

    for n = 1:floor(nbits/2);

        bits = [x(2*(n-1)+1) x(2*(n-1)+2)];

        if bits(1) == 0 && bits(2) == 0
            y(n) = (+1+j)/sqrt(2);
        elseif bits(1) == 0 && bits(2) == 1
            y(n) = (-1+j)/sqrt(2);
        elseif bits(1) == 1 && bits(2) == 1
            y(n) = (-1-j)/sqrt(2);
        elseif bits(1) == 1 && bits(2) == 0
            y(n) = (+1-j)/sqrt(2);
        end
    end

end
```

- *decision_QPSK.m*

```
function [ y ] = decision_QPSK( x )
% decision_QPSK
%   Decides the corresponding QPSK symbol from the received symbol

L  = length(x);

for nsymbol = 1:L
    if 0 < angle(x(nsymbol)) && angle(x(nsymbol))<= pi/2
        y(nsymbol) = (+1+j)/sqrt(2);
    elseif pi/2 < angle(x(nsymbol)) && angle(x(nsymbol)) <= pi
        y(nsymbol) = (-1+j)/sqrt(2);
    elseif - pi < angle(x(nsymbol)) && angle(x(nsymbol)) <= -pi/2
        y(nsymbol) = (-1-j)/sqrt(2);
    elseif -pi/2 < angle(x(nsymbol)) && angle(x(nsymbol)) <= 0
        y(nsymbol) = (+1-j)/sqrt(2);
    end
end

end
```

- *channel.m*

```
function [ output_channel_signal, channel_response,a, theta ] =
channel(input_signal, Nsubcarriers, cp_len, T_prima,T)
% channel
    % Time varying channel, with multipath, doppler rate and attenuation
    m = length(input_signal(1,:)); % Number of OFDM symbols

    % Channel parameters
    maxp = 4; % Maximum number of paths

    C = zeros(Nsubcarriers,m);

    for n = 1:m
            % Number of paths:
                p = 1+ceil((maxp-1)*rand); % Number of paths
             % Delay of multipath
                tau_p(1) = 0;
                Ap(1) =1;

            for k = 2:p
                % Delay in each path:
                   tau_p(k) = 2e-4*rand;
                % Channel gains in each path:
                    Ap(k) = 0.05+0.01*randn;
            end

             % Doppler rate (equal for each path):
                a(n) = 0.0001*randn;

        for fk = 1:Nsubcarriers
            % Theta
                theta(fk,n) = a(n)*2*pi*fk*T_prima/T;

            % Random noise
                z = 10e-8*randn;
                z = z+1i*z;

            for j = 1:p

                % Channel
                    Cp(j) = Ap(j)*exp(-1i*2*pi*(fk/T)*tau_p(j));

                    C(fk,n) = C(fk,n)+Cp(j);
            end

            y(fk,n) = C(fk,n)*input_signal(fk,n)*exp(1i*theta(fk,n))+z;
        end
    end

    output_channel_signal = y;
    channel_response = C;
end
```

- *demod_QPSK.m*

```
function [ y ] = demod_QPSK( x )
% demod_QPSK
%   Demodulates the received symbol

nsymbols = length(x);
y = 0;
for symbol = 1:nsymbols
    if 0 < angle(x(symbol)) && angle(x(symbol))<= pi/2
        bits = [0 0];
    elseif pi/2 < angle(x(symbol)) && angle(x(symbol)) <= pi
        bits = [0 1];
    elseif - pi < angle(x(symbol)) && angle(x(symbol)) <= -pi/2
        bits = [1 1];
    elseif -pi/2 < angle(x(symbol)) && angle(x(symbol)) <= 0
        bits = [1 0];
    end
    y = [y bits];
end
y = y(2:end);
end
```

- *FFT.m*

```
function [ y ] = FFT(x,N)
%FFT
y = fft(x,N);

end
```

- *IFFT.m*

```
function [ y ] = IFFT(x,N)
%IFFT computes de Inverse Fast Fourier Transform algorithm of N points

    y = ifft(x,N);

end
```

- *detector_algorithm.m*

```
function [ final_symbols, final_bits, doppler_rate_estimate,
final_estimation ] = detector_algorithm( received_symbols,
pilot_symbols, Nsubcarriers, T_prima,T)
% detector_algorithm
%   This function computes the detector algorithm after the FFT in the
receptor

% Constants
    m = length(received_symbols(1,:));  % Number of OFDM symbols
    lambda = 0.1;
    serial_symbols(1,:) = pilot_symbols.';

% Initialization
    d_aster(:,1) = pilot_symbols;   % Pilot symbols
    y = received_symbols;           % Received symbols
    C_est(:,1) = y(:,1).*conj(d_aster(:,1));      % Channel estimation
    a_est_part = 0;
    a_est(1) = 0;               % Doppler estimation
    theta_est(1:Nsubcarriers,1) = 0;    % Shift phase estimation (theta)
    theta_happy(:,1) = theta_est;      % Prediction of phase estimation
    sigma_z = 0;                              % Noise variance

    for h = 1:Nsubcarriers
        gamma_est(h,1) = 1/(sigma_z+conj(C_est(h,1)).*C_est(h,1));    %
Gamma estimation: (sigma_z^2+Ck(n)'*Ck(n))^-1
    end
    d_est_0(1:Nsubcarriers,1) = d_aster; % Estimation of current symbols
    d_est_1(1:Nsubcarriers,1) = d_aster; % First estimation of symbols
    d_est_2(1:Nsubcarriers,1) = d_aster; % Second estimation of symbols
    d_palo(1:Nsubcarriers,1) = d_aster;  % Tentative decision of second
estimation
    decision_d(1:Nsubcarriers,1) = d_aster; % Final decision of current
symbol
    delta_theta_est(1:Nsubcarriers,1) = 0;  % Angular offset estimation

    a_part(1) = 0;  % Summatory of doppler rate estimation

    d(1:Nsubcarriers,1) = pilot_symbols;

    % Recursive algorithm
for n = 2:m
    a_part(n) = 0;
    for fk = 1:Nsubcarriers

        % Prediction on the phase theta_est(fk,n)
        theta_happy(fk,n) = theta_est(fk,n-1)+
        a_est(n-1)*2*pi*fk*T_prima/T;

        % First estimation of symbols
        d_est_1(fk,n) = gamma_est(fk,n-1)*conj(C_est(fk,n-1))*y(fk,n)
*exp(-1i*theta_est(fk,n-1));

        % Second estimation of symbols
        d_est_2(fk,n) = gamma_est(fk,n-1)*conj(C_est(fk,n-1))*y(fk,n)
```

```
*exp(-1i*theta_happy(fk,n));

        % Tentative decisions
        d_palo(fk,n) = decision_QPSK(d_est_2(fk,n));

        % Angular offset estimation
        delta_theta_est(fk,n) =
real(acos((real(d_est_1(fk,n))*real(d_palo(fk,n))+imag(d_est_1(fk,n))*im
ag(d_palo(fk,n)))/(abs(d_est_1(fk,n))*abs(d_palo(fk,n)))));

        a_part(n) = a_part(n)+delta_theta_est(fk,n)/(2*pi*fk*T_prima/T);
% Summatory of the expression
    end

    % Doppler rate estimation
        a_est(n) = a_part(n)/Nsubcarriers;

    for fk = 1:Nsubcarriers

        % Phase shift estimation
        theta_est(fk,n) = theta_est(fk,n-1)+a_est(n)*2*pi*fk*T_prima/T;

        % Zero estimation of symbols
        d_est(fk,n) = gamma_est(fk,n-1)*conj(C_est(fk,n-1))*y(fk,n)
*exp(-1i*theta_est(fk,n));

        % Final decision before channel estimation

        d(fk,n) = decision_QPSK(d_est(fk,n));

        % Channel estimation

        C_est(fk,n) = lambda*C_est(fk,n-1)+(1-lambda)*y(fk,n)
*exp(-1i*theta_est(fk,n))*conj(d(fk,n));

        gamma_est(fk,n) = inv(sigma_z^2+C_est(fk,n)'*C_est(fk,n));

        % MMSE Estimate
        d_est_0(fk,n) =  gamma_est(fk,n)*conj(C_est(fk,n))*y(fk,n)
*exp(-1i*theta_est(fk,n));

        decision_d(fk,n) = decision_QPSK(d_est_0(fk,n));
    end

    serial_symbols = [serial_symbols(1,:) d_est_0(:,n).'];

end

doppler_rate_estimate = a_est;
final_symbols = decision_d;
final_bits = demod_QPSK(serial_symbols);
final_estimation = d_est_0;

end
```

- *mse_time.m*

```
function [ mse_t ] = mse_time( data_symbols, decided_symbols )
% mse_time
%

m = length(data_symbols(1,:));   % Number of OFDM blocks
k = length(data_symbols(:,1));   % Number of subcarriers
sum_mse_t(1:m) = 0;
for n = 1:m

    for fk = 1:k
        error(fk,n) = abs(data_symbols(fk,n)-decided_symbols(fk,n))^2;
        sum_mse_t(n) = sum_mse_t(n) + error(fk,n);
    end
    mse_t(n) = 1/k*sum_mse_t(n);
end

end
```

- *mse_time.m*

```
function [ mse_f ] = mse_frequency( data_symbols, decided_symbols )
%mse_frequency

m = length(data_symbols(1,:));   % Number of OFDM blocks
k = length(data_symbols(:,1));   % Number of subcarriers
sum_mse_f(1:k) = 0;
for fk = 1:k

    for n = 1:m
        error(fk,n) = abs(data_symbols(fk,n)-decided_symbols(fk,n))^2;
        sum_mse_f(fk) = sum_mse_f(fk) + error(fk,n);
    end
    mse_f(fk) = 1/k*sum_mse_f(fk);
end

end
```