École Polytechnique Fédérale de Lausanne

Wireless receiver
OFDM Project

Professor:

Andreas Burg

Candidate:

Cardinale Alessandro 369411
Grassano Beatrice 370780

ACADEMIC YEAR 2023-2024

1 Introduction

The Orthogonal Frequency Division Multiplexing (OFDM) is a multicarrier modulation technique exploited widely in digital communication. The basic idea behind OFDM is that of maximizing the use of the available bandwidth while avoiding intersymbol interference and multipaths effects, in order to guarantee reliability in the transmission and an efficient use of the resources.

The principle behind OFDM is that of sending multiple symbols simultaneously by dividing the available bandwidth into smaller sub-channels, each one of them centered at a different carrier frequency, with a spacing that must be a multiple of the symbol rate to ensure orthogonality of the subcarriers in frequency domain. By partitioning the channel, it is possible to send multiple signal in parallel while, thanks to the orthogonality condition, Nyquist criterion is satisfied, guaranteeing thus an ISI free transmission.

Another advantage of using OFDM modulation resides in the fact that both channel estimation and channel equalization are simplified. Indeed, the subcarriers are narrow-banded signals, which means that during a transmission the channel will remain constant both in magnitude and in phase, allowing for a straightforward channel analysis. This characteristics of OFDM highlights how transmission is resilient to interference and multipath fading, making it suitable for use in environments with a high level of noise.

In the following discussion, a more detailed implementation of an OFDM acoustic transmission system is presented together with a more in depth analysis of the various ways in which the channel can be tracked. Also the dependence of the BER, a key parameter while evaluating the performance of a communication system, with some design dependent variables is studied.

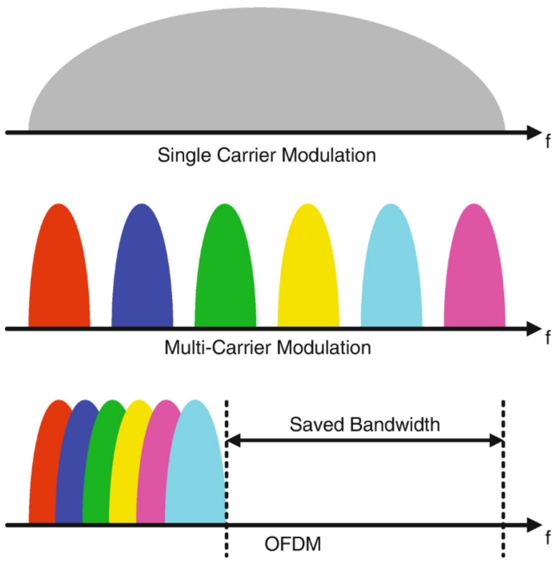


Figure 1: Modulation format comparison

2 Transmitter

The first step for building an OFDM system is to implement the transmitter. The block diagram in Figure 2 is followed while building the Matlab code.

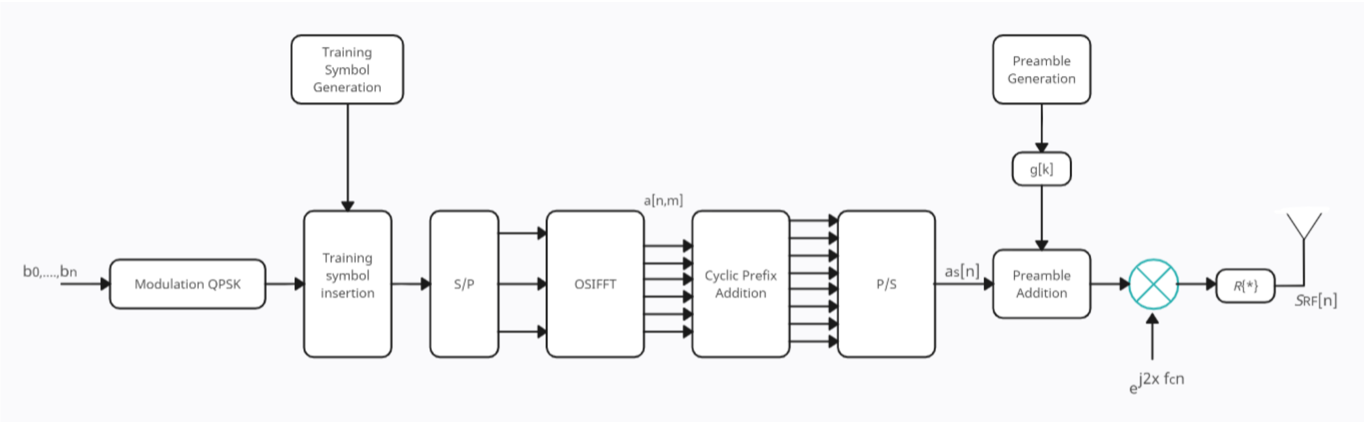


Figure 2: Transmitter block diagram

2.1 Modulation

In this implementation the first step is that of randomly generating the bits representing the data to be sent. Subsequently, the bits are QPSK modulated. In QPSK Modulation the bits are collected in group of two and are labelled following Gray coding.

2.2 Training insertion

After modulation, the training symbols are generated. The training symbols are sequences of bits which are known both at the transmitter and at the receiver which are employed during the channel estimation and equalization to compensate for the distortion effects introduced by the real channel.

It is important to point out that training symbols are BPSK modulated. In order to minimize the average estimation error, it is advisable to have the magnitude of each training to be equal to one another. Then, it is also crucial to have as a training a pseudo random sequence so that it does not produce a high energy peak in time domain. Indeed, the peak to average power ratio is of paramount importance when assessing the robustness and the reliability of a real system.

In Sections 4 and 5, two different type of training insertion, namely block training and comb training, are extensively analyzed.

2.3 Serial to Parallel Conversion

After that, the signal must be converted into a parallel representation to mimic the parallel nature of OFDM. In Matlab, this is done by converting the signal from a vector into a matrix in which each column is an OFDM symbol. In this representation, symbols in the same column are modulated at different subcarrier frequencies but are sent at the same time instant. Instead, symbols in the same row are modulated at the same frequency but at different time instants.

2.4 Oversampled Inverse Fourier Transfrom

Afterwards, each OFDM symbol is transformed in time domain by exploiting the Inverse Discrete Fourier Transform. In this system, the IFFT algorithm is employed thanks to its efficiency. On top of that, an oversampled version of the IFFT was provided to account for the unavoidable oversampling introduced at the system's receiver side.

2.5 Cyclic prefix addition

As a next step, the cyclic prefix is added. This procedure consists in taking the last part of each OFDM symbol and prepose it at the beginning of each frame. This operation guarantees a guard band between each OFDM symbols, but it primarily shields the useful data from ISI interference caused by the channel. Indeed, assuming the channel has a finite length L, smaller than the length N of each symbol sent, its effects will impact only the first L-1 symbols (recall the convolution rules). So by adding a cyclic prefix which is at least L-1 long, the useful data will not be affected by Intersymbol Interference. It is clear now how a wise choice for the cyclic prefix is crucial, an aspect which will be further discussed in Section 7.

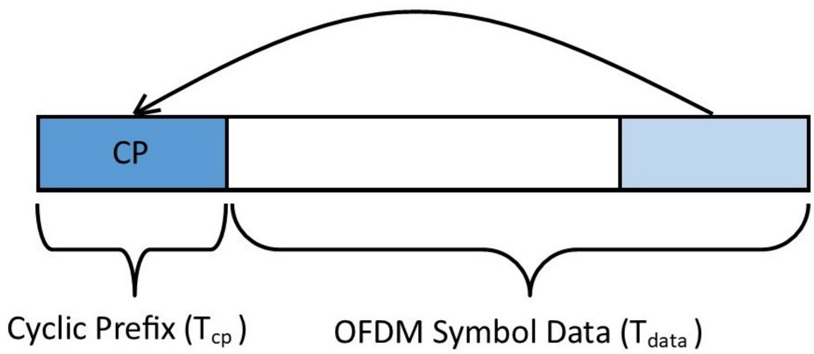


Figure 3: OFDM symbol plus CP structure

2.6 Preamble generation and addition

Subsequently, in order to detect the beginning of the actual data frame, it is necessary to generate and add the preamble, a known data sequence both at the transmitter and at the receiver. In particular, the preamble is constructed using a LPSR to guarantee pseudo randomicity of the sequence and then it is BPSK modulated. Subsequently, the preamble is upsampled and pulse shaped by a root raised cosine filter.

Eventually, both the preamble and the OFDM symbols, are normalized with respect to their average power. The transmitted signal is now constructed.

2.7 Upconversion

The last step on the transmitter side before finally sending the signal into the channel consists in making it real valued and to shift it at higher frequencies, namely at the carrier frequency f_c, which is equal to 8kHz.

Figure 4 depicts the spectrum of the transmitted signal. As expected, the signal sent is real as it is symmetrical with respect to the y-axis and the two bands are centered around the chosen carrier frequency of 8kHz.

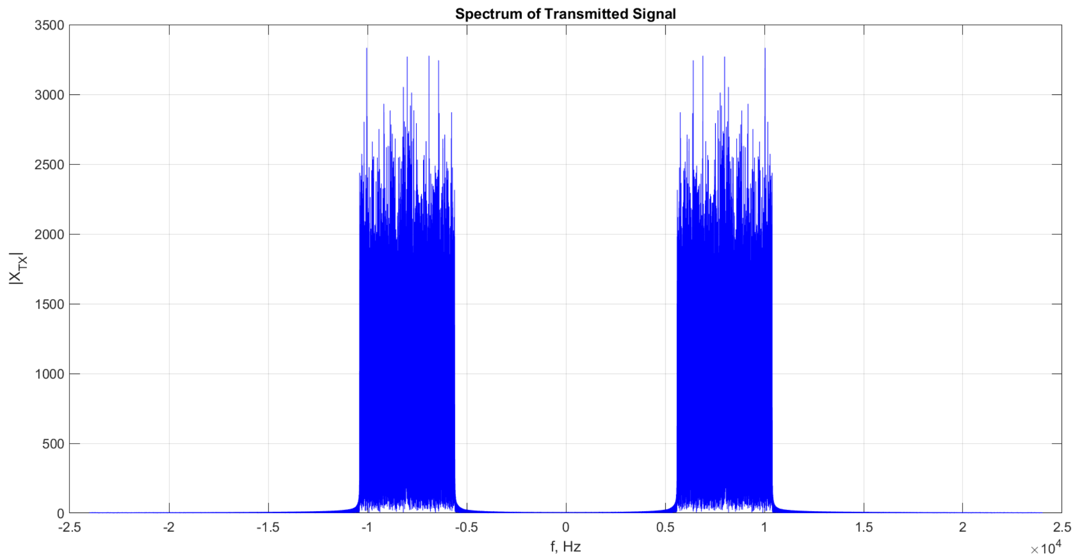


Figure 4: Transmitter spectrum after RF up conversion

3 Receiver

The block diagram shown in Figure 5 shows the receiver of our system.

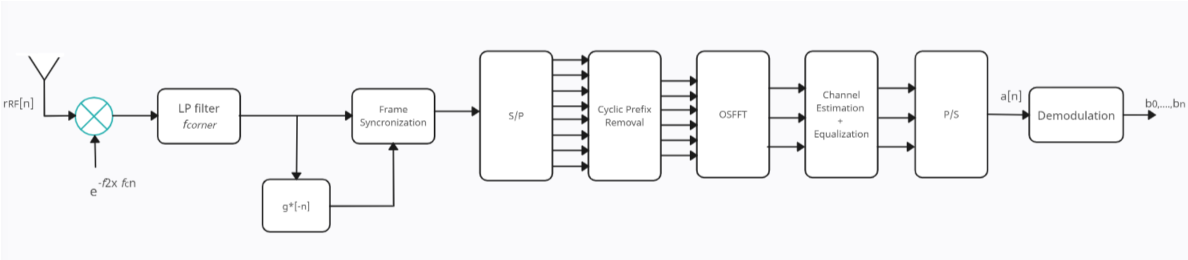


Figure 5: Receiver block diagram

3.1 Down conversion

The first step to be performed is the RF down conversion: with this operation the signal is brought back to baseband by multiplying the received signal by $e^{-j2\pi fcn}$ and it is ready to go through an ADC (not implemented here), to then be processed digitally.

3.2 Low pass filter

For OFDM modulation there is no matched filtering (also, it makes no sense since the channel itself acts as a filter), therefore a simple low pass filter is enough to remove higher frequency noise components. This low pass filter is IIR and is selected to have a corner frequency that is 10% more than the baseband bandwidth of the received signal.

3.3 Frame synchronization

After the filtering one needs to detect the beginning of the OFDM frame. To do so, the preamble is exploited. It is worth noticing that since the preamble is an RRC pulse shaped signal, then to properly detect it we need to pass it through a matched filter. Then the usual frame synchronization based on the correlation between the received signal and the known preamble is implemented.

3.4 CP removal

Once the beginning of the frame is detected, one must remove the Cyclic Prefix as it does not carry useful information. At this point, the serial signal can be converted into a parallel representation of itself. In Matlab, this simply means passing from a vector representation to a matrix representation. The same convention used at the transmitter is followed: each column represents an OFDM symbol and each element of the column represents a subcarrier of the symbol.

3.5 FFT and Downsampling

To perform the OFDM demodulation, the received signal must be converted in frequency domain. Moreover, a downsampling is performed (because symbols at the transmitter have been over-sampled during the modulation phase).

3.6 Channel estimation and equalization

Channel estimation and equalization is the most crucial task the receiver has to perform. In this phase, the system needs to estimate the channel using the training sequence and then use this estimation to recover the transmitted symbols. According to the type of training inserted (Block or Comb), two different estimations must be performed.

More on Block and Comb training will be said in Sections 4 and 5. Despite the theoretical background of the two training structures, practically, in Matlab in both cases the channel estimation was performed by dividing the received training by the known training symbols.

$$\hat{H} = \frac{Y_l}{T_l} \tag{1}$$

In the Block case, the division is between two vectors (that represent an OFDM training symbol), while in Comb case the division is between two complex numbers that represent the BPSK training symbols.
Then, to compensate for both the magnitude and phase distortion introduced by the channel, the following expression is applied

$$\hat{a} = \frac{Y_l}{|\hat{H}|} \cdot e^{-jarg\{\hat{H}\}} \tag{2}$$

In addition to that, in this system also continuous phase tracking of the channel is implemented. For this analysis, Viterbi algorithm is exploited, and it will be shown how effective it is under slow varying channels, that is when phase offset between subsequent carriers is small.

3.7 Training removal

After the equalization phase, the training symbols must be removed from the frames as they are not carrying any useful data. Afterward, parallel to series conversion is performed, where the received signal is converted again to a series representation (a vector in Matlab).

3.8 Demapper

As a last step on the receiver side, the signal is firstly normalized and then is QPSK demapped, to recover the bits. At this point, it is possible to compare the transmitted and received bits in terms of bit error rate to understand the impact of the channel and to study the performance of the system.

4 Block Training: Structure and Channel Estimation

Figure 6 shows the structure of an OFDM frame when block training is applied. In this scenario, all the subcarriers of the OFDM system are trained and thus, periodically, a known training symbol spanning through all the subcarriers is sent.
Here the channel estimation is performed using the ML criterion, represented by the following equation

$$\hat{H} = argmax|Y_l - H_l T_l|^2 \tag{3}$$

with $\hat{H}$ being the estimated channel, Y_l the received training symbol and T_l the known training sequence. The training symbols are used to estimate the channel FIR filter, that will then be used to recover the transmitted data symbols subsequent to the training.
Simple as it is, this kind of training can be used with a decent level of accuracy when the block channel approximation is valid. This consists in assuming that the channel will remain constant for a certain amount of time before taking another realization.

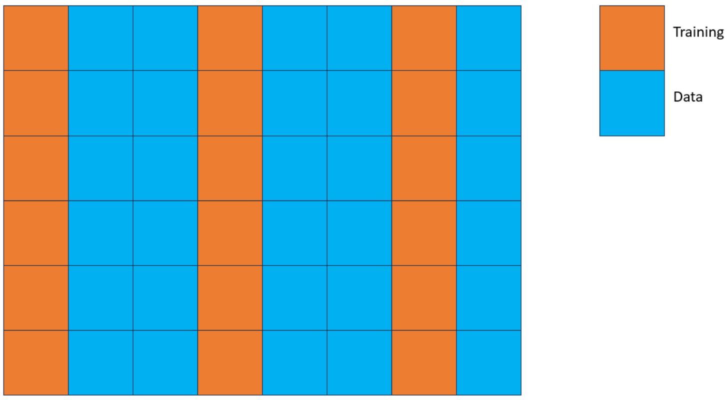


Figure 6: Block Training Structure

5 Comb Training: Structure and Channel Estimation

In Figure 7 one can see the structure of (a possible) Comb type training. Comb training relies on the assumption that the channel is different from zero only in certain taps, thus it may be useless to train for all the subcarrier frequencies. A wiser way to estimate the channel would be training just certain tones, by interleaving known training symbols in between data inside a single OFDM symbol. In the code developed, the comb training allows a user defined value of data symbols in between every training sample. However, for code simplicity, the first tone is always trained, regardless of the training rate chosen, as Figure 7 shows.

As one could easily deduce, this training type allows to insert fewer training symbols inside our frame, thus increasing the data transmitted. However, the real strength of this training type is that it is highly performative in situation of fast changing channels. In this scenario indeed, block training may likely fail, simply because the channel is not block anymore. Comb training, on the other hand, is based on a totally different assumption and it allows to train the channel more frequently, but with a better channel efficiency (in terms of training symbols vs data symbols).

Comb training requires also a different channel estimation method, based on a least square estimation. In formulae, the channel impulse response can be estimated with the following equation:

$$\hat{h} = (F_{NxL}^H T^H \cdot T F_{NxL})^{-1} \cdot F_{NxL}^H T^H \cdot Y \tag{4}$$

where F_{NxL} is a matrix made up of the L columns of the Fourier matrix F_{NxN} corresponding to the tones used for training, T is a diagonal matrix having in its main diagonal the known training sequence and Y is the received OFDM symbol. Now, one can find $\hat{H}$, the estimated frequency response of the channel, by simply doing the Fourier transform of $\hat{h}$. Mathematically:

$$\hat{H} = \mathscr{F}\{\hat{h}\} \tag{5}$$

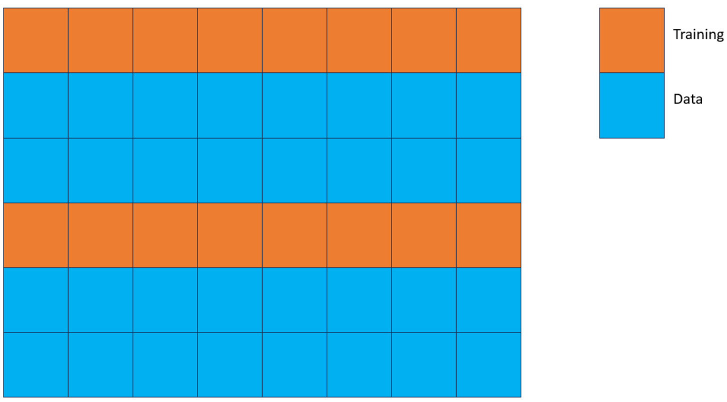


Figure 7: Comb Training Structure

6 Channel Analysis

In this section the real channel and its effect are studied in depth. Figure 8 depicts the channel impulse response when 20 OFDM data symbols plus one training are sent with 1024 subcarriers with a spacing of $f_{spacing} = 5Hz$. The impulse response of the channel is obtained by performing the inverse Fourier Transform of the estimated channel $\hat{H}$. The channel impulse response is exploited to understand where and on which extent the real channel impacts the transmitted signal. From the plot, it is evident that most of the energy of the channel is concentrated in the first $3ms$, showing that the channel mainly influences the signal in the indicated time span. To be precise, the delay was found by taking the 90% decrease in amplitude of the main peak.

From this evaluation, it is also possible to determine the minimum length of the cyclic prefix to avoid having intersymbol interference on the received signal. Being the sampling frequency equal to $f_s = 48kHz$, the minimum number of taps was found to be 144 and it was determined by counting the number of samples in the delay spread interval. Being able to analyze this plot is crucial to understand which length of the cyclic prefix to pick both to counteract ISI effects as explained above, but also to try and save in terms of efficiency (see Section 7), as it is noticeable how, after a long time, the channel almost goes to zero and it has almost no effect on the data symbols.

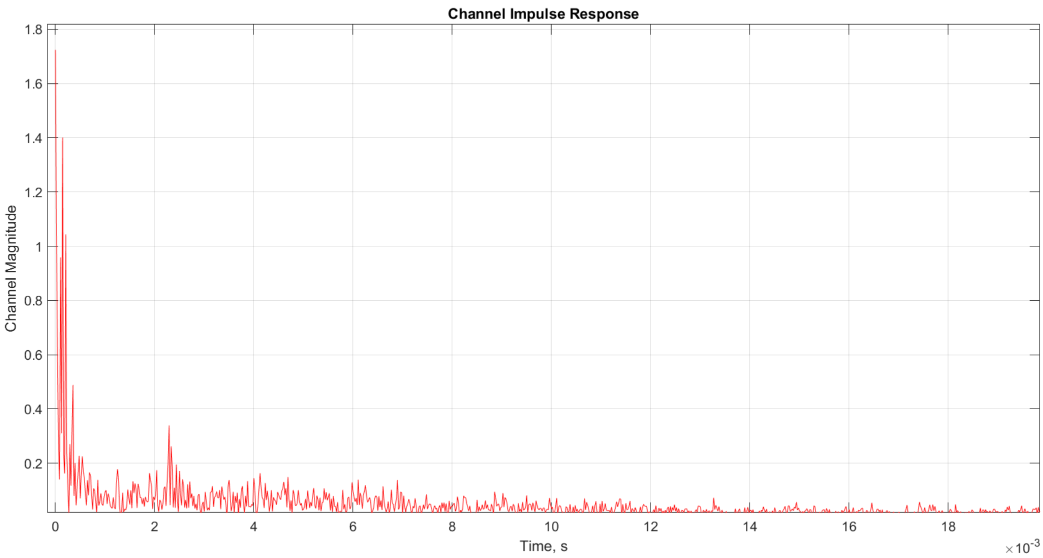


Figure 8: Channel impulse response

Secondly, both the magnitude and the phase of the channel are plotted as a function of frequency. It is possible to notice that the channel is not constant for the whole span, highlighting the advantage of multicarrier modulation format. Indeed, each symbol is sent in different subcarriers, each at a different frequency and with a limited bandwidth, allowing to assume the channel as almost constant and simplify channel estimation.

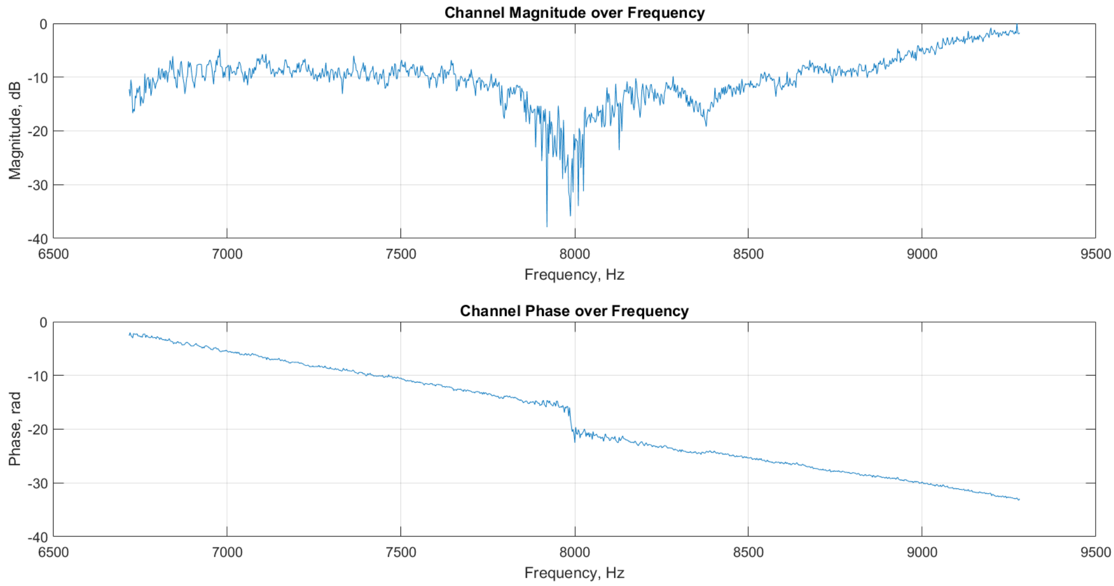


Figure 9: Channel magnitude and phase evolution in frequency domain

Finally, Figure 7 depicts the progression of the channel magnitude and phase as a function of time. Each line in different colors represents the channel behavior evaluated at various frequencies, stressing again how different can be the behavior among the subcarriers and emphasizing the need for a robust system under various conditions.

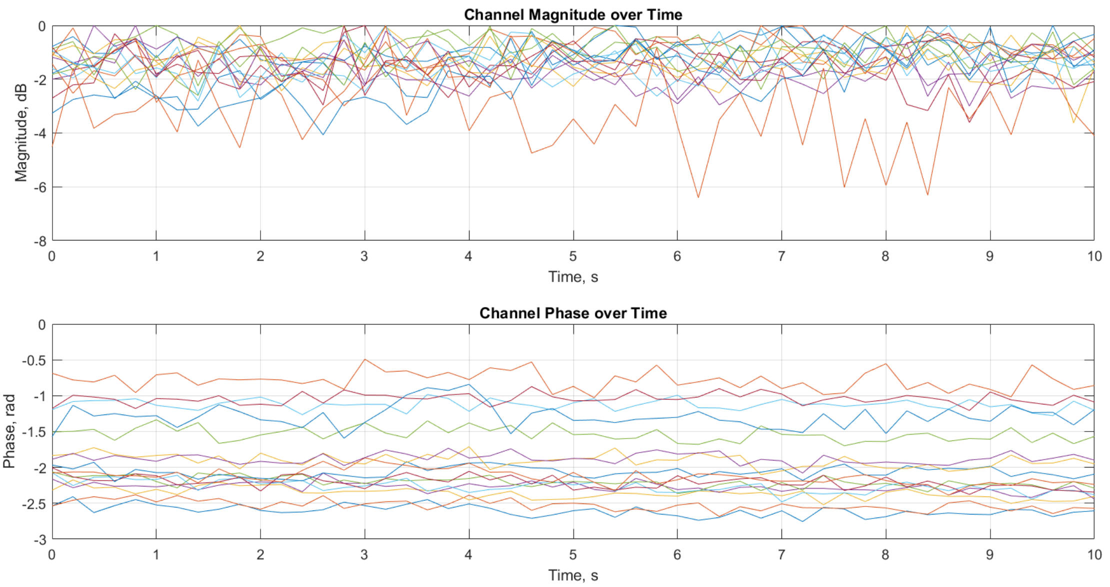
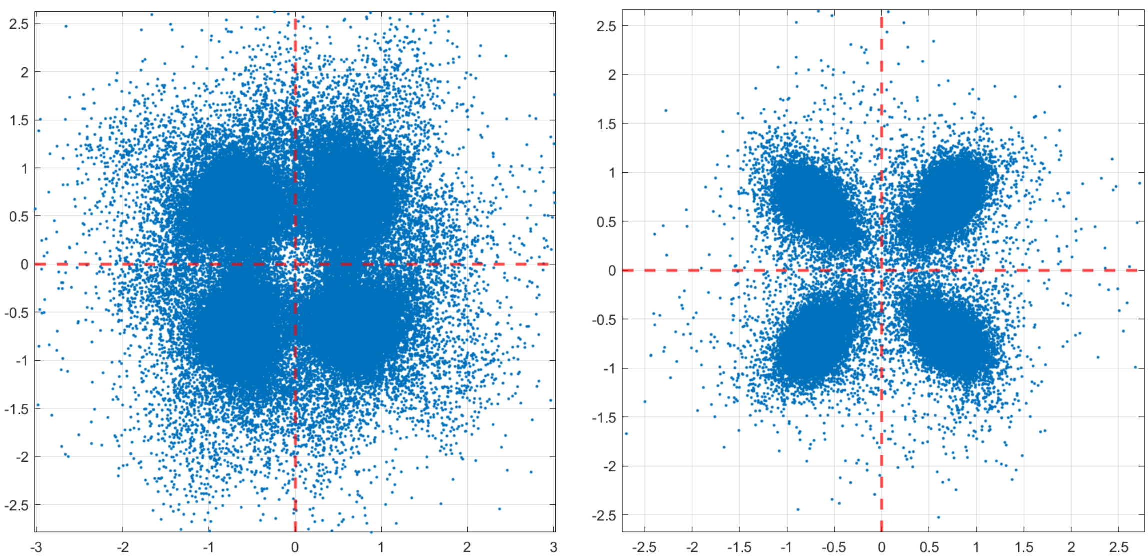


Figure 10: Channel evolution magnitude and phase in time for different frequencies

7 Results discussion

In this section, comparisons between the above-mentioned type of training (Comb and Block) and channel estimation (with and without phase tracking) are studied together with some further explorations.

At the beginning, the effect of continuous phase tracking is explored. The set-up for the measurements is a static microphone (receiver) and a static speaker (transmitter); 500 symbols, excluding the training symbol at the beginning of the frame, are sent. The constellations in Figure 11 shows the result. When continuous phase tracking is not performed, the constellation is rotated with respect to the expected position of the symbols and it is widely spread. Instead, when Viterbi algorithm is implemented, the constellation is more compact and centered around $\pm\frac{1}{\sqrt{2}} \pm j \cdot \frac{1}{\sqrt{2}}$, These differences are explained by the fact that when no phase tracking is applied, the channel estimation is performed based only on one training, the one sent at the beginning, and phase correction is applied based on this estimated value. The channel is although subjected to phase variations, as discussed in section 6, causing errors in the estimation as the channel characteristic are not updated.



(a) *Block training without Viterbi*

(b) *Block training with Viterbi*

Figure 11: Constellation plots with and without Viterbi phase tracking, 500 symbols sent

Figure 12 compares the BER as a function of number of data symbol sent with and without Viterbi algorithm. It is possible to see that when the number of data symbols is limited (up to 80), the performance of block training with and without phase tracking are comparable. When increasing the amount of symbols sent, both BER grow but when phase tracking is performed the system is more robust and the BER is lower. For high data symbols, the BER curve with Viterbi seems to stabilize. This trend can be explained by the fact that when few symbols are sent, the channel has no time to change its phase and so continuous phase tracking is superfluous. Instead, when many symbols are transmitted, the phase of the channel slowly varies so that it can be properly followed by Viterbi algorithm, reducing the error in the channel estimation phase causing a decrease in the final BER.

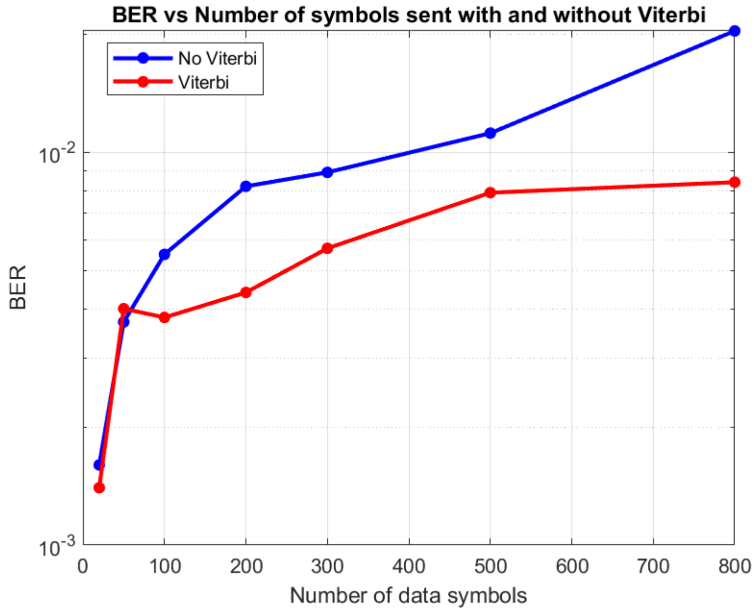
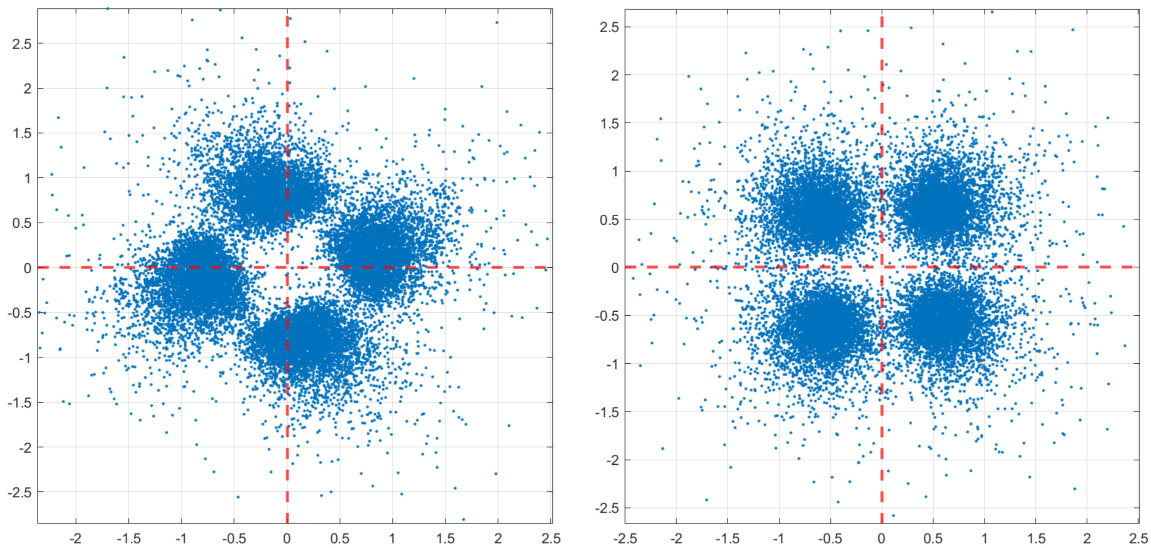


Figure 12: BER comparison with and without continuous phase tracking

Now, 500 OFDM data symbols plus training are sent but the speaker is set in motion towards the microphone in such a way to cause a fast change in the channel characteristics. Constellation a in Figure 13 highlights how, when the phase of the channel varies fast, Viterbi algorithm fails, even with a high (Block) training rate. The phase is not tracked anymore and the constellation points are rotated with respect to the expected position, causing a dramatic increase in the BER. When comb training is applied, the receiver is able to estimate and correctly account for the channel variation, even when they are abrupt. The constellation points are in the correct position, consequently the BER is low. These plots stress the paramount importance of employing comb training, as it makes the system extremely resilient.



(a) *Block training and phase tracking*

(b) *COmb training*

Figure 13: Constellation plots with block training with phase tracking and comb training, moving speaker, 500 symbols sent

At this point, the impact of the length of the cyclic prefix is studied. Figure 14 shows the evolution of the BER as a function of the cyclic prefix ratio, that is how long is the cyclic prefix compared to the total length of the data symbol sent. The graph shows a decreasing trend, where at the beginning it is particularly steep until it almost stabilizes.

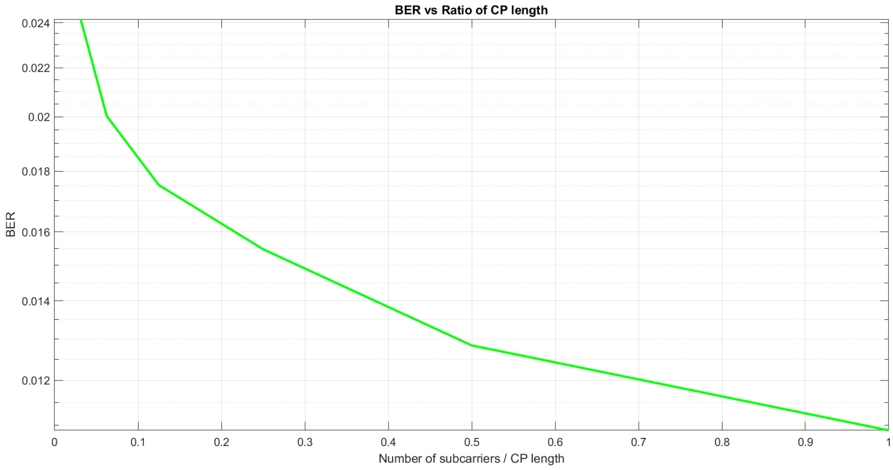


Figure 14: BER vs Cyclic prefix ratio

Figure 15 studies the efficiency as a function of the cyclic prefix length. The efficiency is defined as

$$\eta = \frac{N}{N + Ng} \tag{6}$$

with N the number of subcarriers carrying actual data and NG the length of the cyclic prefix. Obviously, when the latter increases the efficiency is diminished. Indeed, the longer the cyclic prefix the less the actual data is sent in one OFDM symbol.

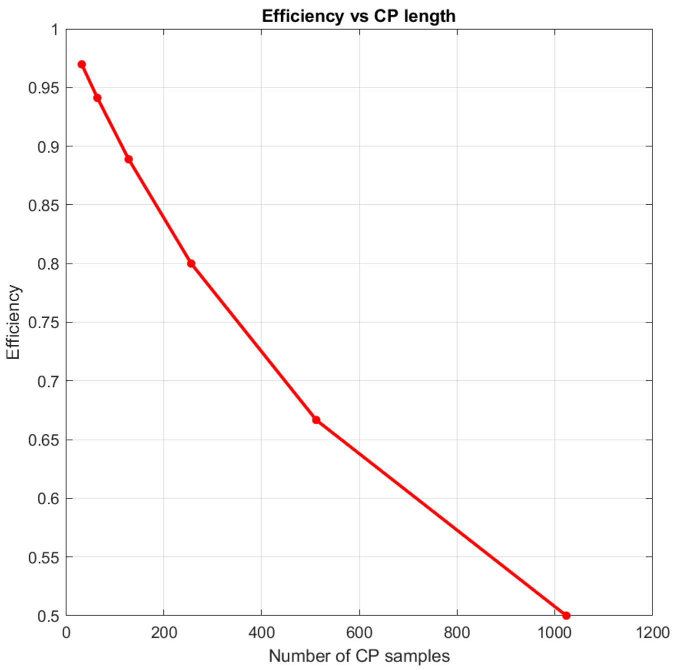


Figure 15: Efficiency vs Cyclic prefix length

Combining the observation made on the above graphs, it is possible to conclude that there is a trade off between a low BER and a high efficiency. In general, it is important to understand on which extent the channel has an impact on the data symbol in order to choose a long enough cyclic prefix to preserve data integrity. Making the cyclic prefix too long would mostly result in a small improvement in the BER (recall the curve stabilization) while causing an impactful degradation in the efficiency, making this choice often not advisable.

Another way to improve the efficiency, apart from shortening the cyclic prefix, could be that of increasing the number N subcarriers send. To do that, the spacing $f_{spacing}$ between each subcarrier must be decreased and thus making the system more sensitive to frequency offset, making ISI more likely to occur. Again there is a trade off between two important figures of merit: efficiency and ISI resilience.

At this stage, the impact of training is studied in terms of BER. Figure 16 illustrates the BER in relation to the number of training inserted per symbol (the number of actual data symbol sent was fixed to 50). The direct consequence of a more frequent training is a decrease in the BER. Indeed, the higher the number of trainings, the better is the channel estimation as the system is able to detect in a finer way channel variations and can precisely compensate for them. Despite that, it is important to remember that training symbols do not carry useful information, so a more frequent training results in longer transmitted signals for an equal amount of data symbols.

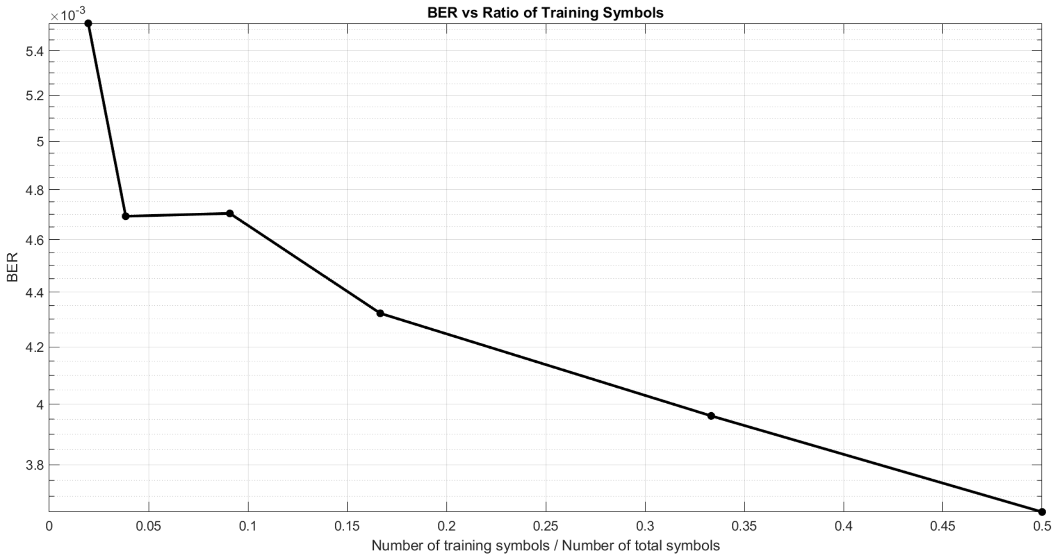


Figure 16: BER vs Number of training

8 Conclusion

In this report several characteristics of OFDM transmission were studied. In particular, comparisons between different type of training methods and channel estimation were made. From this, it was deduced that Block training with phase training yields to robust results in terms of the BER when the channel phase slowly varies over time. Instead, comb training is the best solution when a system has to operate over highly noisy and fast changing channels.
It is worth noticing that just a simple, yet effective, version of the Comb training has been implemented here. More exotic kinds could be explored in order to have more insights, but the solution proposed still allows to grasp the advantages and disadvantages of different training schemes.

Also, the study of the cyclic prefix length and the frequency of training highlighted the relevance of these parameters in the design of an optimal OFDM system.

To conclude, the implemented OFDM system showed to be robust against ISI and is adaptable to different channel conditions. The efficiency is high as numerous data symbols can be sent without suffering from distortion. The adaptability, solidity and effectiveness of OFDM modulation explains why it is now worldwide employed for wireless communication.