ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

EE-442 WIRELESS RECEIVERS: ALGORITHMS AND ARCHITECTURES

Final Project:
Acoustic OFDM Transceiver

Authors : Kai-Wen Cheng, Alban Puech

Contents

1 Introduction

1.1 Context

1.1.1 Multi-Path channels

In multipath channels, frequency selective fading can occur because of the different paths that signals can take, which leads to multiple taps with significant time delays. These delays can be longer than the symbol duration, and thus cause intersymbol interference (ISI) as some signal paths arrive much later than others. Large delay spreads in the time domain i.e., long duration between the earliest and the latest path translate into small coherence bandwidth in the frequency domain. The coherence bandwidth is the bandwidth over which we can assume the channel to be the same. If two bandwidths are separated by more than the coherence bandwidth, they essentially experience the channel independently, leading to different types of distortion in each. Coping with frequency selective fading is complicated when working with wideband single carrier systems.

1.1.2 Frequency Division Multiplexing

To deal with multipath channels, we can use a method called Frequency Division Multiplexing (FDM). When using FDM, we reallocate our resources by converting a high-frequency wideband signal into multiple parallel lower-rate data streams of narrower bandwidth. We thus send many signals at the same time, each using a different part of the bandwidth. By making signals longer, the delay spread becomes smaller than the symbol duration. Similarly, by making each carrier occupy less bandwidth, we make sure that their bandwidth is smaller than the coherence bandwidth. This way, the signals experience the same channel distortion over their bandwidth and don't suffer from ISI.

1.1.3 Orthogonal Frequency Division Multiplexing

So as to not get any ISI, we usually use guard bands i.e., ranges of unused frequencies between the bands used by the sub-carriers. This means that the available bandwidth is not fully used. Orthogonal-Frequency-Division-Multiplexing (OFDM) is a type of FDM that avoids inter-symbol interference by using orthogonal sub-carriers instead of large guard bands. OFDM symbols are still separated by a guard interval so that the system can not reach perfect efficiency.

However, if this interval is long enough i.e., longer than the channel impulse response the system becomes free of ISI while achieving a still very high efficiency.

1.2 Project Goals

In this project, we aim to implement an OFDM system for acoustic transmission. We first describe our transmitter implementation, and our receiver implementation, and then go over some of the key results that we have obtained, as well as the main observations that we have been able to make from our experiments.

2 Methodology

In the following section, we describe the key elements of our transmitter and receiver. We also introduce the key theoretical concepts necessary for the implementation of the transceiver.

2.1 Transmitter

Our transmitter is implemented in the `tx.m` file. It is responsible for the following operations, as shown in Fig. 1:

1. QPSK mapping of the data symbols.
2. Training symbol insertion.
3. OFDM modulation.
4. Addition of the Cyclic Prefixes.
5. Preamble generation and addition.
6. Up-conversion.

2.1.1 Mapping

The first function that our transmitter calls is the `mapper` function, which maps our data represented in binary bits to symbols. These symbols are constellation points that depend on the specified modulation scheme.

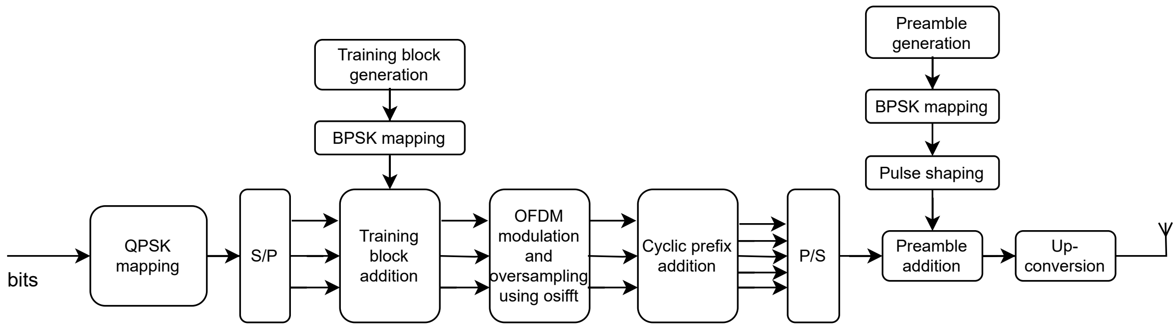


Figure 1: Transmitter Workflow Diagram.

- BPSK (Binary Phase-Shift Keying): This modulation scheme uses 1 bit per symbol encoding. We use it for both the preamble and the training sequences.
- QPSK (Quadrature Phase-Shift Keying): Maps two bits per symbol, with four points evenly spaced on a circle in the constellation diagram. It is applied to the payload data. The QPSK constellation using gray mapping is shown in Fig. 2.

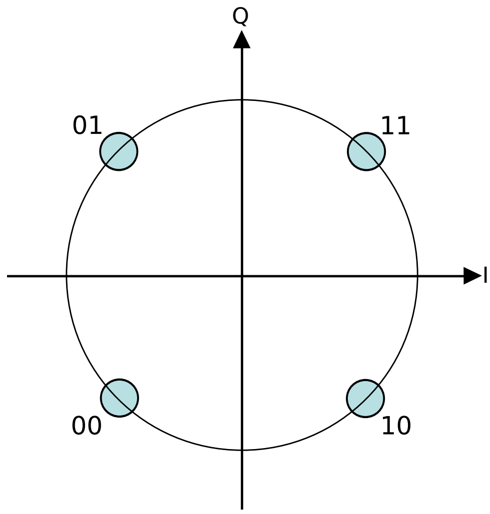


Figure 2: QPSK constellation using gray mapping [1]

To transmit all symbols with the same energy, the constellations were normalized. The average power P of a constellation is given by:

$$P = \frac{1}{M}\sum_{i=1}^{M} |x_i|^2 \tag{1}$$

Where $|x_i|$ represents the power of the i-th constellation point and M is the total number of points. To normalize the constellation points, we divided each constellation point by the square

[1] source: https://fr.wikipedia.org/wiki/Phase-shift_keying

root of the calculated average power P. The normalized constellation point $\tilde{x}_i$ is thus given by:

$$\tilde{x}_i = \frac{x_i}{\sqrt{P}} \tag{2}$$

2.1.2 Training Symbol Insertion

After the mapping of the payload data, we convert the serial stream into parallel streams, and we insert a training symbol between every block of data symbols (which can have variable length) using the `insert_training_blocks` function.
These pseudo-random sequences are known from both the transmitter and the receiver. They are used by the receiver to estimate the channel response and the phase offset. We map them using BPSK mapping.

2.1.3 OFDM Modulation and Cyclic Prefix

OFDM modulation. We transform the symbols by performing an inverse discrete Fourier transform. The Inverse Fast Fourier Transform (IFFT) is performed over the full spectrum given by the output sampling rate, but only the baseband part of the spectrum is modulated. Since our audio transmission system is an all-digital transceiver, we also need to perform oversampling. The `osifft` function computes the inverse Fourier transform of our data and handles the oversampling. In our implementation, the modulation is performed by the `add_cp_and_modulate` function that is also in charge of cyclic prefixes insertion.

Cyclic prefix. In OFDM, the symbols are separated by Cyclic Prefixes (CP) that are obtained by replicating the end of the OFDM symbol that follows the prefix. CPs play different roles. First, they act as a buffer between consecutive OFDM symbols. They act as guard intervals and thus reduce ISI. Denoting by L the length of the discrete channel input response, only the first L 1 samples of the OFDM symbol are affected by the previous one. Hence, by adding a cyclic prefix of sufficient length before the actual samples, we can make sure that they don't get affected by the ISI. Second, the CP ensures that the otherwise linear convolutions of the transmitted signal with the channel input response are transformed into circular ones, which are said to be easier to handle mathematically. This will make channel equalization easier. Finally, the Cyclic Prefixes make the system more robust to timing errors.

The length of the Cyclic Prefix, N_{cp}, is chosen based on the expected maximum channel delay spread and depends on the channel. This value can not be known in advance but can be approximated by running simulations.

2.1.4 Preamble

We use a preamble to allow for timing synchronization. The preamble is a pseudo-random sequence obtained using a Linear Feedback Shift Register (LFSR). It is known by both the receiver and the transmitter and is inserted before the payload. It is BPSK encoded and oversampled. We then use a pulse shape $g[n]$, that is convoluted with the preamble $p_0[n]$:

$$p[n] = p_0[n] * g[n] \tag{3}$$

We choose to work with the root-raised-cosine (RRC) because of its input response properties: it quickly converges to zero and satisfies the Nyquist criterion. We use a roll-off factor equal to $\alpha = 0.22$. The expression of the RRC filter is:

$$g_{\mathrm{RRC}}[n] = \frac{4\alpha}{\pi} \frac{\cos\left(\frac{\pi\alpha}{T}(1+\alpha)n\right) + \frac{1-\alpha}{4\alpha}\sin\left(\frac{\pi\alpha}{T}(1-\frac{1}{\alpha})n\right)}{1-\left(\frac{4\alpha n}{T}\right)^2} \tag{4}$$

The preamble is added to the signal after the signal is converted from parallel to serial format. At the receiver, the preamble is passed through a matched filter $g_{\mathrm{MF}}[n] = g_{\mathrm{RRC}}[-n]$. Since OFDM signals can have a high peak-to-average power ratio (PAPR), we make sure that the preamble and the OFDM symbols have the same energy.

2.1.5 Up-Conversion

The up-conversion of the signal is done by shifting the complex baseband signal to a higher carrier frequency denoted f_c. This is necessary for transmission through a physical channel, as the baseband signal $s(t)$ is initially complex-valued, with its spectrum centered at zero frequency. The resulting RF signal $s_{RF}(t)$ is then given by:

$$s_{RF}(t) = \mathrm{Re}\{s(t)e^{j2\pi f_c t}\} \tag{5}$$

2.2 Receiver

Our receiver is implemented in the `rx.m` file. It is responsible for the following operations, as shown in Fig. 3:

1. Down conversion.
2. OFDM filtering.
3. Frame synchronization.
4. Removal of the Cyclic Prefix.
5. Demodulation and Down-sampling.
6. Channel equalization.
7. Demapping.

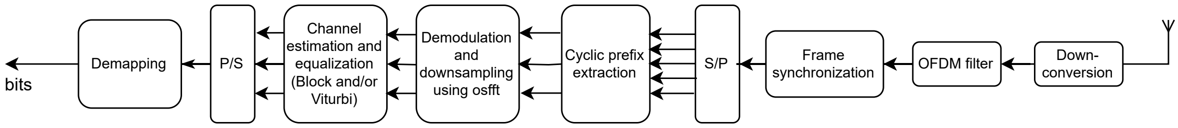


Figure 3: Receiver Workflow Diagram

2.2.1 Down Conversion

The received RF signal, r_{RF}, is first brought down to baseband for processing. We sample the signal at the receiver's sampling frequency, and we multiply it with a complex exponential to shift the spectrum. The down-converted signal, $r_(t)$, is given by:

$$r(t) = r_{\text{RF}}(t)e^{-j2\pi f_c t} \tag{6}$$

where f_c is the carrier frequency.

2.3 Low Pass Filtering

After down-conversion, we filter the signal using an OFDM low pass filter to remove the unwanted component of the spectrum. The filter cut-off point is set slightly above BW_{BB}. The `ofdmlowpass` function implementing the filter was given to us.

2.4 Frame Synchronization

For the receiver to correctly interpret the OFDM symbols, it must synchronize with the symbol frame, starting with a detected preamble.
We detect the preamble using by looking at the results of the auto-correlation function, which correlates the received signal with the known preamble. This is done by the `frame_sync`*function*. The correlator output is expressed as:

$$c[n] = \sum_{i=0}^{N_p-1} p^*[i]r[n+i] \tag{7}$$

where N_p is the length of the preamble, $p[i]$ is the preamble signal, and $r[n]$ is the received signal.
We need to normalize the output with respect to the received signal power to be able to use a fixed threshold for peak detection. Moreover, we use a maximum likelihood test, so that a peak (indicating the start of the frame) is detected if the normalized energy exceeds the threshold γ:

$$\frac{|c[n]|^2}{\sum_{i=0}^{N_p-1} |r[n+i]|^2} > \gamma \tag{8}$$

If the threshold is too low, we get false detections. On the opposite, if we set it too high, we take the risk of missing the preamble.

2.4.1 Removing the Cyclic Prefixes

After frame synchronization, we transform the signal back into parallel streams. We then remove the cyclic prefixes by removing the first OS_FACTOR*N_G samples of each multi-carrier symbol, where OS_FACTOR denotes the oversampling factor.

2.4.2 Demodulating and Downsampling the Signal

We then demodulate the signal using the `osfft` function that applies the Fourier transform while handling the downsampling.

2.4.3 Channel Estimation, Equalization, and De-mapping

At this point, we have not yet taken care of the effect of the channel on the transmitted signal, and we still have not removed the training symbols. We need to compensate for the phase and amplitude variations introduced by the channel. This step is called equalization. Equalization first requires channel estimation. To do this, we use the known training symbols and run maximum likelihood estimation to estimate the channel's effect on each sub-carrier. The estimate $\hat{h}_i$ for sub-carrier i is obtained by:

$$\hat{h}_i = \arg\max_{h_i} |y_i - h_i t_i|^2 \tag{9}$$

where h_i represents the channel's function, y_i the received sub-carrier symbol, and t_i the known training symbol.
We can then perform channel equalization (i.e., correct the channel distortion) as follows:

$$\hat{y}_i = \frac{y_i}{|\hat{h}_i|} e^{-j \arg(\hat{h}_i)} \tag{10}$$

As the channel effect changes over time, we need to continuously estimate the channel. We use two channel tracking algorithms, that are both implemented in the `equalize_channel` function:

- **Block Training:** We use the training symbol sent between each block of data to estimate the channel using 9. The correction of all symbols in the data block following the training symbol is done using the same channel estimate.
- **Viterbi-Viterbi Training:** The initial channel estimate obtained from the last training symbol is further updated before each symbol of the data block using the Viterbi-Viterbi algorithm. That way, the channel estimate is more frequently updated.

We then remove the training symbols and demap the QPSK symbols using the `demap` function.

2.5 Experiment Setup

2.5.1 Transceiver Simulator

The `run_sim` function takes a configuration as input and uses the `tx` and `rx` functions to transmit and receive a random sequence of bits using an audio speaker and a microphone. We have also implemented a bypass channel, that simulates a lossless channel by using the vector created by the transmitter at the receiver, and an AWGN bypass channel, that simulates an Additive White Gaussian Noise channel with random phase noise.

2.5.2 Configuration Files and Experiment Scripts

We set the different parameter values in the configuration files, which are then given as parameters to the `run_sim` function. Each experiment (that is described in section 3) has its script that repeatedly calls the `run_sim` function, changing a configuration parameter value, and storing the results in an array between each call. These results can then be used for plots.

3 Experimental Results

We conducted five main experiments to demonstrate the impact of specific parameters in a wireless transmission system. They are the following:

- Impact of the CP length on the efficiency and the Bit Error Rate
- Pilot Interval v.s Bit Error Rate
- Carrier Frequency Spacing v.s Bit Error Rate
- Carrier Frequency v.s Bit Error Rate
- Channel Condition and its impact on delay spread.

3.1 Impact of the CP length on the Efficiency and the Bit Error Rate

The length of the Cyclic Prefix plays an important role for our system not to experience ISI. As we have seen, it has to be at least larger than the delay spread. However, the efficiency of our system decreases as the CP length increases. Indeed, the system efficiency is expressed as:

$$\eta = \frac{N}{N + N_G} \tag{11}$$

where N is the number of sub-carriers and N_G is the cyclic prefix length. Thus, we want our CP length to be large enough to avoid ISI, but to be as small as possible to keep our efficiency as high as possible.

Using the suggested CP length, $N_G = N/2$, we obtain an efficiency equal to $\frac{N}{N+\frac{N}{2}} = \frac{2}{3}$, and a bit error rate (BER) equal to $1.00 \cdot 10^{-4}$.

To estimate the optimal CP length, we can estimate the delay spread by running a simulation, and by computing the inverse Fourier transform of the channel so as to obtain the channel input response, as shown in Fig. 4. With a power delay $\delta = 0.5$ms and a sampling frequency $f_s = 48$kHz, we obtain an estimate of the optimal CP length $\hat{N}_G = \delta \cdot f_s^{-1} = 24$. The new efficiency of the system is now $\frac{N}{N+\hat{N}_G} = \frac{256}{256+24} \approx 0.91$, which represents a 36% increase compared to the previous efficiency.

We can then run simulations to check that the new CP length $\hat{N}_G$ is enough to avoid most of the ISI. Looking at our results shown in Fig. 5, this is indeed the case, since we obtained a BER close to the lowest one that we obtained when working with a CP length range of [1,120]. Indeed, we obtained a BER of $1.00 \cdot 10^{-3}$, while the lower BER is $1.00 \cdot 10^{-4}$, obtained for slightly higher values than our estimate $\hat{N}_G$ e.g., $N_G = 32$. Our experimental result thus perfectly aligns with the theory.

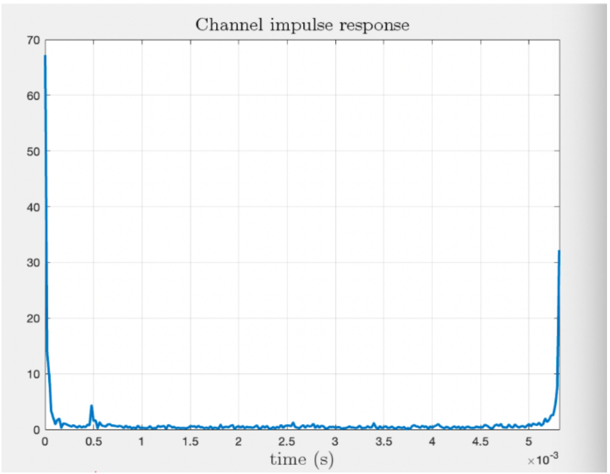


Figure 4: Channel input response. The delay spread shown in the figure is approximately equal to 0.5ms.

3.2 Pilot Interval v.s Bit Error Rate

In this experiment, we aim to discover the relationship between pilot interval and the effect it has on BER. The pilot interval is the number of data symbols sent between each pilot, that are composed of a single training symbol. As mentioned before, the trade-off between pilot interval, BER, and efficiency is an important design decision in wireless communication systems. With smaller pilot training intervals i.e., fewer data symbols between training symbols we

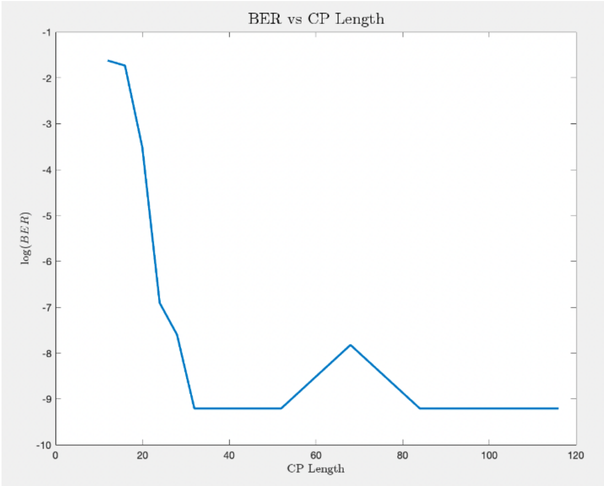
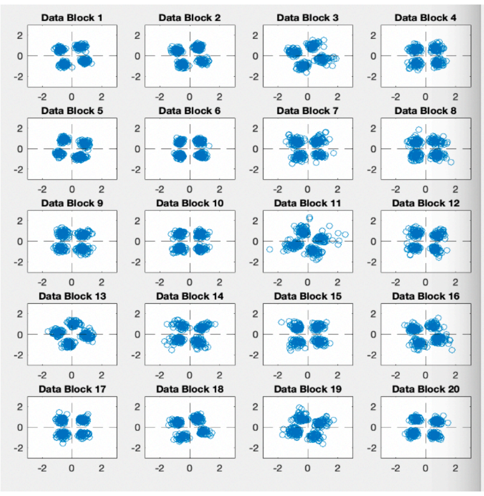


Figure 5: BER as a function of the CP length N_G.

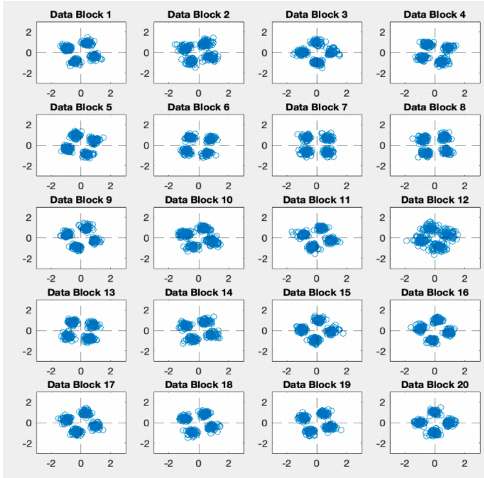
expect better BER (due to the increase in phase correction frequency) while losing transmission efficiency since we are constantly transmitting signals that do not relate to the data we aim to transmit.

To measure the trade-off, we designed our experiment to transmit 20 data symbols with different pilot intervals ranging from 1 to 20. We only use block-type estimation. In the case of a pilot interval of 1 data symbol, the transmitted signals are phase-corrected for every data symbol using the block-type estimation. On the other hand, when working with a pilot interval of 20 data symbols, we are only correcting the phase offset at the start of our transmission (since our signal only contained 20 multi-carrier symbols). The result we obtained is described in the following figures. In Fig. 6, we can observe how the phase change impacted wireless transmission under different pilot intervals. In the case where the pilot interval is set to 20, we observe huge phase shifts i.e., the QPSK symbols are more rotated. Such phase impact can lead to a high probability of misrecognition of the QPSK symbols. The qualitative result can also be observed in Fig. 7. From the figure, we see an increase in the log bit error rate when increasing the pilot interval. This matches our intuition because as we increase the pilot interval, we are re-correcting our phase less often, which indeed leads to worse BER results (high BER).

When using the Viterbi-Viterbi algorithm, we are correcting our initial channel estimate obtained from the training symbol at every data symbol, independently of the pilot interval set. This mitigates the impact of increasing the pilot interval since the channel estimate is still continuously updated. However, increasing the pilot interval means that we get fewer initial phase estimations, so large channel changes become difficult to track.



(a) Transmission with a Pilot Interval of 1 (every data symbol is phase corrected using the estimate from the training symbol that preceded it)



(b) Transmission with a Pilot Interval of 20 (the training symbol is used to estimate the phase offset of the whole data block and to correct the 20 following symbols of the data block)

Figure 6: Transmitted QPSK Data under different pilot interval

3.3 Carrier Frequency Spacing v.s Bit Error Rate

In this experiment, we discuss the effect of carrier frequency spacing on the bit error rate. Carrier frequency spacing in OFDM transmission is defined as $\frac{1}{T_{sampling}}$. The Carriers are lined up in such a way that they are orthogonal in the frequency domain. However, due to the non-idealities of the channel, frequency offset can lead to the loss of orthogonality in the frequency domain. This phenomenon is called inter-carrier interference (ICI). ICI worsens the BER. By increasing the frequency spacing, we allocate more frequency bandwidth to each carrier. When more frequency bandwidth is given, inter-carrier interference is less prominent. This is due to a lower overlapping in the frequency domain when influenced by the same amount of frequency offset. However, increasing frequency spacing means that the transmitted signal will take on a wider bandwidth.

In our experiment, the default spacing was set to 5 Hz. With 256 carriers, our transmission requires a 1.28kHz bandwidth signal. With a 1Hz increase in each frequency spacing, our bandwidth widens drastically. In real conditions, due to the increasing popularity of wireless

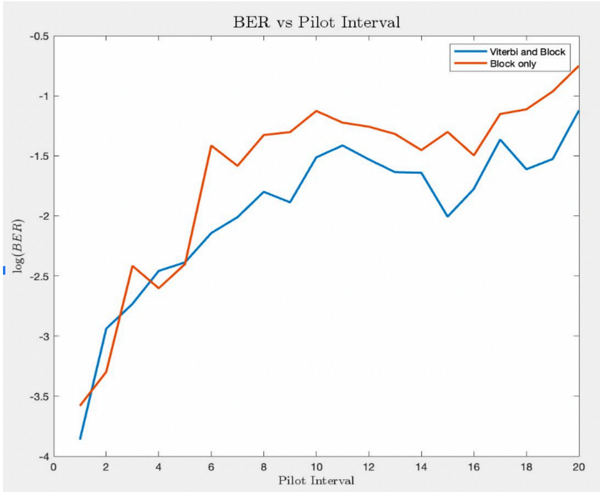


Figure 7: BER as a function of Pilot Interval

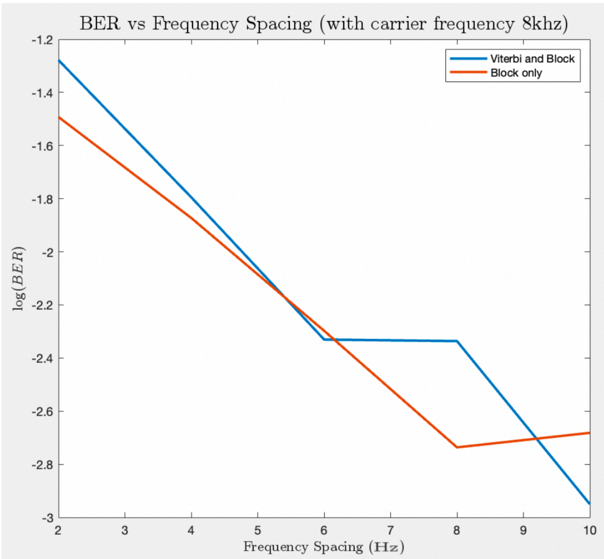


Figure 8: BER v.s Carrier Spacing

transmission, radio signal bandwidth is expensive and limited. Therefore, it is very important to observe the trade-off between BER and bandwidth to prevent the waste of bandwidth resources. We tested our transmission with a varying range of carrier frequency spacing. The result is summarized in Fig. 8. We can see that a 2 Hz and a 10 Hz frequency spacing has very different BER. The BER of 2 Hz spacing is around -1.2 (log scale) while 10 Hz spacing is around -2.8 (log scale). This is equivalent to 5 5-times reduction in BER. This is rather a linear relationship in the log BER. System designers will now have to address the trade-off between BER and the linear decrease in bandwidth efficiency.

3.4 Carrier Frequency v.s Bit Error Rate

Carrier Frequency is also another parameter that can be experimented using our OFDM transmission system. As mentioned in previous parts and lectures, the need for higher carrier frequency transmission comes from the need to shrink antenna size as well as to improve the propagation characteristics for the given transmission environment. In our experiment, since we are dealing with audio OFDM transmission, we are conducting transmission at a range of 20 Hz to 20 kHz. Lower carrier frequencies have a longer propagation distance but are more susceptible to background interferences (such as other pieces of electronic equipment that produce lower-frequency hummings). However, higher frequency signals have limited traveling distance but are more robust to environmental interference.

In this experiment, we aim to look into the effects of transmission carrier frequency as well as their negative effects on BER. From Fig. 9, we can see that starting from 800Hz and above, the BER does not change significantly. However, signals that are transmitted below this frequency suffer a huge impact on BER. Initially, we believe that the deterioration of BER is due to low-frequency environmental noise. However, after we looked into the signal transmission in the frequency domain, we observed a fundamental issue that led to the increase in our BER. In Fig. 10, we show the FFT of our transmitted signal. At 800Hz, we see that our signal FFT is not overlapped. On the other hand, at 200Hz, the signal FFT is highly overlapped. Thus, we believe that instead of environmental noise, it is due to a fundamental design limit. A signal with 256 carriers and a carrier spacing of 5Hz implies a need for 1.28kHz bandwidth. Signals centered at 200Hz mean that we will need around 600Hz of frequency band on both sides. At 200Hz carrier frequency, around 400Hz of bandwidth's signal is overlapped. Therefore, around one-third of the signal is affected.

3.5 Channel Condition and Power Delay Profile

In this section, we aim to discuss the effect of channel conditions through the demonstration of channel delay profile and the channel FFT. We know that signals traveling through different environments will have varying effects. These effects can be characterized by the channel delay profile. From our channel delay profile, we can understand how often and how much of our signals are received during a given time interval. In this section, we showcased three different channel environments: outdoor, indoor (medium-spaced), and indoor (densely spaced) indoor.

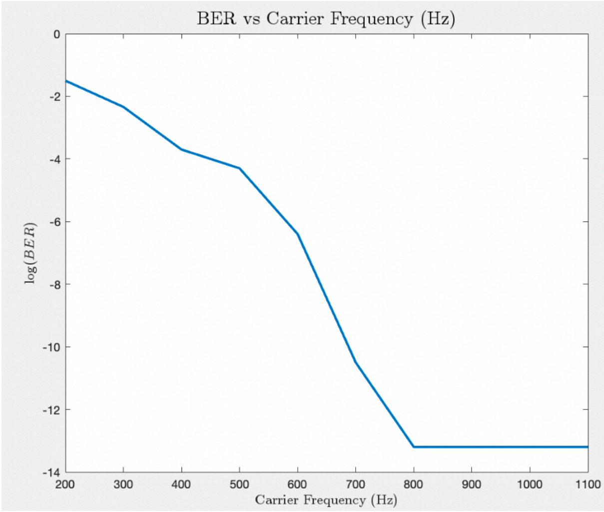
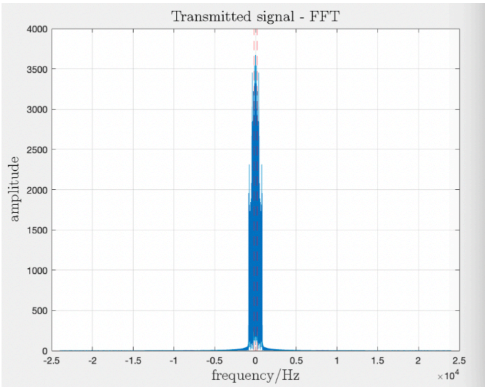
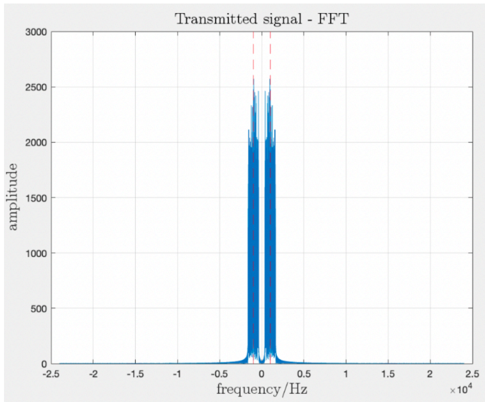


Figure 9: BER v.s Carrier Frequency



(a) Transmission signal FFT at 200Hz



(b) Transmission signal FFT at 800Hz

Figure 10: Transmission FFT at different carrier frequency

From Fig. 11, we can see how the channel delay differs. In an outdoor environment, we observe a delayed-echoed signal at 0.5ms. In an indoor environment, we observe multiple echoed signals. These are important parameters for system designers to know because if a signal is expected to be transmitted in all three channels, one must pick a CP length that will be able to accommodate all these channels. We discuss how to pick an adequate CP length according to channel delay in 3.1.

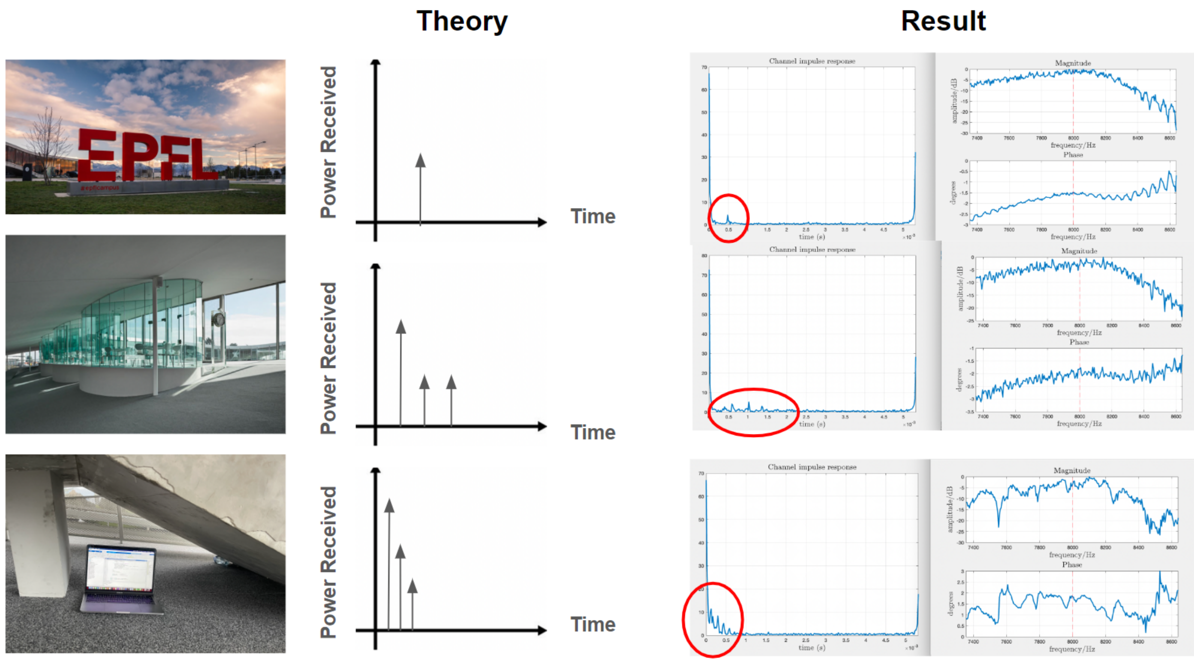


Figure 11: Various Channel Condition's Delay Profiles

4 Considerations for OFDM System Design

We have conducted experiments on OFDM system design parameters to discover their relationship with BER as well as the figure of merit in a typical wireless transmission system. Our results are to take into account when designing such systems, and can be summarized in the following five points:

- Choice of CP Length is a trade-off of accuracy and efficiency.
- CP length design must include consideration of all channel conditions to ensure robust transmission.
- Increasing frequency spacing is a trade-off between system robustness and bandwidth efficiency.
- Carrier frequency does not generally affect BER unless at a very low frequency.
- Frequencies of channel estimation is a trade-off between transmission efficiency and BER.

To briefly discuss them again for clarity, the choice of CP length is a trade-off of accuracy and efficiency because increasing the CP length yields a lower BER while decreasing the efficiency of the system. From the default CP length to our optimized CP length, we were able to increase the efficiency by 36%, while keeping the BER small.

Secondly, we concluded that CP length design also highly depends on the channel delay profile. In the case where we expect our signals to travel in different delay profiles, we must carefully measure the delay profile of each and pick a CP length that will ensure low ISI in all channel conditions.

Thirdly, we observed that increasing frequency spacing increases the robustness of the system since frequency offset-induced inter-carrier interference is less prominent with larger spacing. However, we also see an increase in signal bandwidth when increasing frequency spacing. In real-life scenarios, the scarcity of bandwidth implies a larger cost for a larger bandwidth. This again demonstrates the trade-off in such a wireless communication system.

Fourthly, by observing how our signal behaves at different carrier frequencies, we observe that only at a very low frequency (under 800Hz) will there be a worsening of BER. This is due to the overlapping of signals in the frequency domain. However, if avoided, signals should not suffer a dramatic decrease in BER at other carrier frequencies.

Lastly, we also found that pilot interval is essential to an accurate transmission in real channels. With a smaller pilot interval, we can estimate our phase offset more often, which improves BER. However, this directly leads to us allocating transmissions to a larger number of training symbols where no data transmission is done, which reduces how many bits we can send during the same time interval. The trade-off between optimal phase estimation and BER is yet another essential design trade-off.

5 Conclusion

In this project, we discussed the transmission flow of OFDM transmission through detailing the implementation of our OFDM transceiver in MATLAB. We found that many aspects influence the design decision of an OFDM transmission system. We demonstrated some parameters that system designers can tune in order to meet their given specifications in terms of bandwidth, BER, as well as efficiency. We also showed the power delay profile in different channels to emphasize the importance of CP length under different conditions. Looking forward, we believe that there are more parameters in this current project that can be experimented upon. If given more time, we would like to also observe the effect of more carriers, different modulation orders, different modulation functions, etc.