Digital Communications - AY 2020/2021
MATLAB implementation of an OFDM system

Amedeo Giuliani - 2005797

May 31, 2021

1 Introduction

The *Orthogonal Frequency Division Multiplexing* (OFDM) approach is quite useful when we are facing a wideband, or equivalently, a frequency selective, channel, that is, the bandwidth of the user signal is much larger than the coherence bandwidth of the channel. This means that the channel will vary a lot in frequency within the user frequency range, thus degrading the performances of the communication system. And here's why we apply OFDM: it can turn a wideband channel into a set of narrowband channels, in which the frequency response does not change too much. This result can be achieved through a series of operations, shown in Fig. 1. First of all, the N symbols to transmit

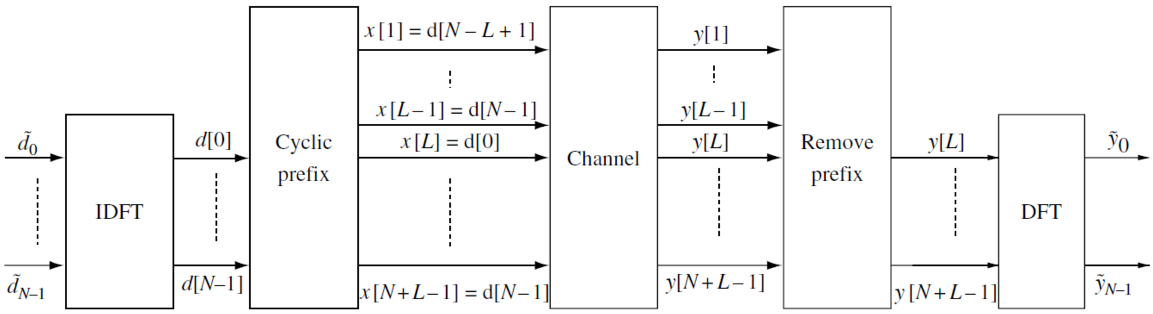


Figure 1: OFDM scheme. [1]

are generated in frequency, so an IDFT is performed to go back in the time domain. Then, we add a *Cyclic Prefix* (CP), long at least $C \geq L-1$, where L is the length of the channel impulse response, which consist of the repetition of the last symbols to transmit. One may wonder why we need to add this CP. The first reason is that in this way we ensure that no *Inter-Symbol Interference*(ISI), also called *Inter-Block Interference* (IBI), arises, as we are waiting for the channel impulse response to be exhausted before transmitting user data. For the second reason, if we analyze mathematically what we transmit

$$\text{after adding CP} \quad x[m] = \sum_{n=0}^{N-1} \tilde{d}_n e^{j2\pi \frac{n(m-C)}{N}}, \quad m = 0, ..., N+C-1 \tag{1}$$

and what we receive (neglecting noise)

$$\text{after receiving and discarding CP} \quad y[m+C] = h_\ell * x[m+C] = \sum_{l=0}^{L-1} h_\ell x[m+C-l] = \tag{2}$$

$$= \sum_{l=0}^{L-1} h_\ell \sum_{n=0}^{N-1} \tilde{d}_n e^{j2\pi \frac{n(m-l)}{N}} = \sum_{n=0}^{N-1} e^{j2\pi m \frac{n}{N}} \tilde{d}_n \sum_{l=0}^{L-1} h_\ell e^{-j2\pi l \frac{n}{N}} = IDFT\left\{\tilde{d}_n \tilde{h}_n\right\} \implies \tag{3}$$

$$\implies \tilde{y}_n = DFT\left\{IDFT\left\{\tilde{d}_n \tilde{h}_n\right\}\right\} = \tilde{d}_n \tilde{h}_n, \quad n = 0, ..., N-1 \tag{4}$$

clearly, from the last equation, we see that adding the CP has also the effect of canceling *Inter-Carrier Interference* (ICI). In other words, if we design the system correctly, we can derive an equivalent model for OFDM, which is a set of N parallel narrowband AWGN channels, as illustrated in Fig. 2. This is also the reason why we can still talk about capacity, even though the channel is frequency selective and changing in time. In Chapter 3 we will also prove the last relationship between input and output by simulation.

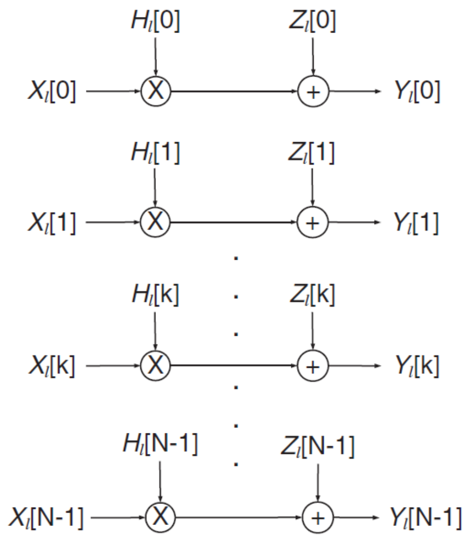


Figure 2: OFDM equivalent scheme. [2]

2 Step 1 and 2

We will study the OFDM system in a "static" scenario, that is, with a frequency selective channel which does not changes in time. Hence, we drop the time index in all our discussion. The channel is modeled with a real impulse response h_ℓ with $L = 5$ taps, which is shown in Fig. 3a. The number of subcarriers is chosen such that the loss due to the presence of the CP is less or equal than 10%. In mathematical terms, this means:

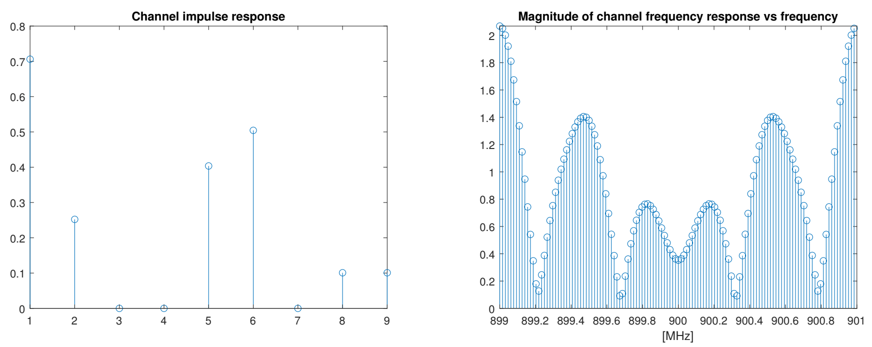
$$\frac{C}{N+C} \leq \frac{1}{10} \iff N \geq 9C \iff N \geq 2^{\log_2(9C)} \implies N = 2^{\lceil \log_2(9C) \rceil} \tag{5}$$

where we have taken the ceiling of the exponent in order to pick N as the first power of 2 that satisfy the inequality. Now we can perform the N-DFT of the channel impulse response, which returns the (complex) channel frequency response, whose magnitude is shown in Fig. 3b. Notice that h_ℓ has been normalized as

$$h_\ell \to \frac{h_\ell}{\|h_\ell\|} \tag{6}$$

in order to have $h_\ell \leq 1 \forall l$ and to have the power of the channel

$$\sum_{\ell=0}^{L-1} |h_\ell|^2 = 1 \tag{7}$$



(a) Channel impulse response h_ℓ.

(b) Channel amplitude response $|\tilde{h}_n|$.

Figure 3

3 Step 3

We start with a simple simulation in which we transmit only one OFDM block, i.e., a random vector of zeroes and ones long N, which is the number of subcarriers, for each SNR value in the interval $[0, 30]$ dB. Therefore, an N-IDFT is performed and the CP is added and finally the symbols are sent over the channel. Please note that, since MATLAB computes the IDFT as

$$x[m] = \frac{1}{N} \sum_{n=1}^{N} \tilde{x}_n e^{j2\pi(m-1)(n-1)/N}, \quad 1 \leq m \leq N \tag{8}$$

we have to multiply this by $\sqrt{N}$ to preserve the input power. In fact:

$$x[m] \to \sqrt{N}x[m] \implies \sum_m |x[m]|^2 = \sum_m \left|\sqrt{N}x[m]\right|^2 = N \sum_m |x[m]|^2 = N\frac{1}{N} \sum_n |\tilde{x}_n| = \sum_n |\tilde{x}_n|^2 \tag{9}$$

where the second last equation is due to Parseval's theorem for DFT.
At the receiver, we simply do the inverse computations of the transmitter: the CP is removed and the N-DFT is computed. Again, since MATLAB computes the DFT as

$$\tilde{x}_n = \sum_{m=1}^{N} x[m] e^{-j2\pi(m-1)(n-1)/N}, \quad 1 \leq n \leq N \tag{10}$$

we must multiply this by $\frac{1}{\sqrt{N}}$ to preserve the received power. In fact:

$$\tilde{x}_n \to \frac{1}{\sqrt{N}}\tilde{x}_n \implies \sum_n |\tilde{x}_n|^2 = \sum_n \left|\frac{1}{\sqrt{N}}\tilde{x}_n\right|^2 = \frac{1}{N} \sum_n |\tilde{x}_n|^2 = \frac{1}{N} N \sum_m |x[m]|^2 = \sum_m |x[m]|^2 \tag{11}$$

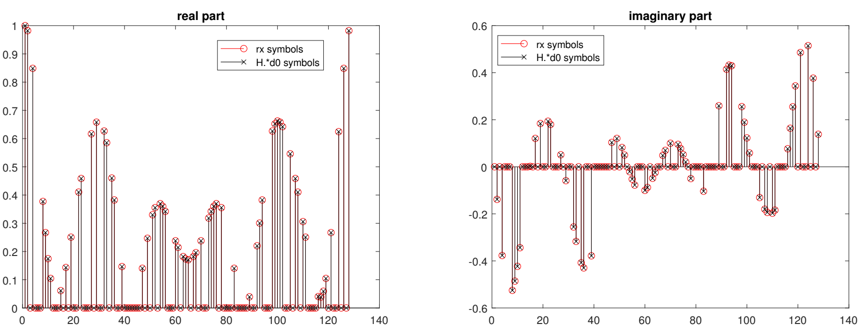
where the second last equation is again due to Parseval's theorem for DFT.
We want now to prove the following equation:

$$\tilde{y}_n = \tilde{h}_n \tilde{d}_n + \tilde{w}_n, \quad n \in [0, N-1] \tag{12}$$

where $\tilde{y}, \tilde{h}, \tilde{d}$ and $\tilde{w}$ are the N-DFTs of the received signal, the channel, the transmitted signal and the noise, respectively. If we run the simulation in absence of noise and we take the real and imaginary part of the received signal and the element-wise product between $\tilde{h}$ and $\tilde{d}$, we can clearly see from Fig. 4a and Fig. 4b that they perfectly overlap, that is

$$\tilde{y}_n = \tilde{h}_n \tilde{d}_n, \quad n \in [0, N-1] \tag{13}$$

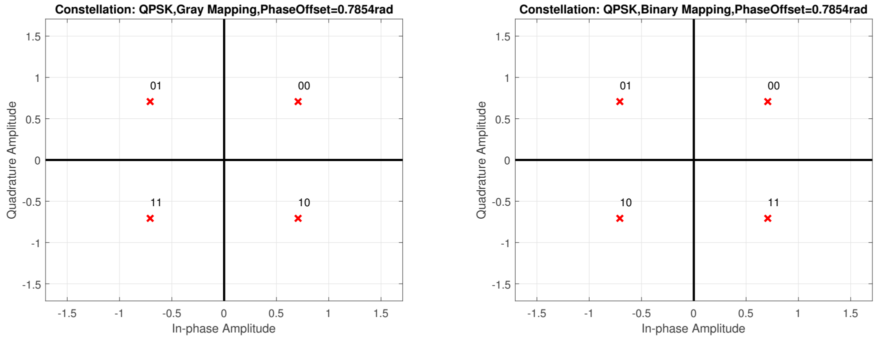
If we now add noise, we will end up with a situation described by Eq. 12.



(a) Real part of equation 13.

(b) Imaginary part of equation 13.

Figure 4



(a) (b)

Figure 5

4 Step 4

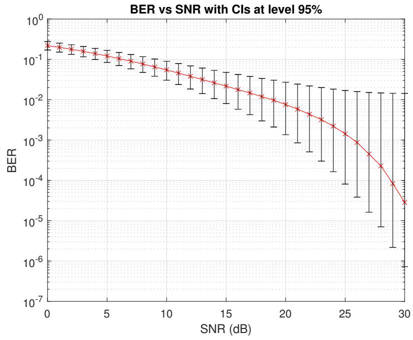
We now want to analyze the *Bit Error Rate* (BER) and the capacity of this communication system, both of them versus the *Signal to Noise Ratio* (SNR). Hence, we need to extend the simple system developed in Ch. 3 by introducing additive white Gaussian noise. In addition, we assume that the transmitted symbols belongs to a QPSK constellation, which does not change the input power as the symbols belong to the unitary circle, with Gray symbol mapping to maximize the BER. In fact, as we can see from Fig. 5a, between near symbols only one bit changes. Thus, if the receiver erroneously decode a symbol, only one bit will be wrong, regardless of the symbol. Instead, considering a QPSK with binary symbol mapping (Fig. 5b), we can have up to two wrong bits, i.e., when the receiver confuses the second symbol with the third one, or the first one with the fourth one, leading to an increase in the BER.
Since the OFDM technique converts a wideband channel into N parallel AWGN subchannels, we can compute the capacity as the sum of the capacity of each subchannel:

$$C_{\text{OFDM}} = \frac{W}{N+C}\sum_{n=1}^{N} log_2(1+\frac{\tilde{h}_n P_n}{N_0}) \quad , \quad n \in [0, N-1] \tag{14}$$

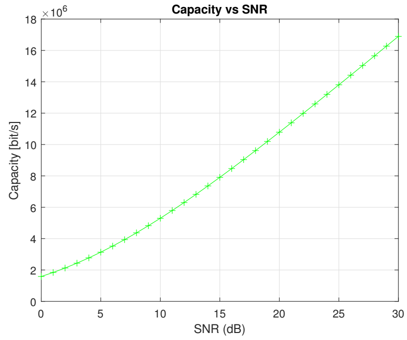
where $\frac{1}{N+C}$ is the penalty due to the transmission of the CP, W is the allocated bandwidth, P_n is the allocated power to each subcarrier, which is the same for each of them as we assume the transmitter does not have *Channel State Information at Transmitter* (CSIT) and N_0 is the noise power.
Since the channel is known at the receiver, it can perform a coherent detection, and in particular it implements a zero-forcing receiver after the N-IDFT operation, that is:

$$\tilde{y}_n^{\text{ZF}} = \frac{\tilde{y}_n}{\tilde{h}_n} \quad , \quad n \in [0, N-1] \tag{15}$$

where $\tilde{y}, \tilde{h}$ are the N-DFTs of the received symbols and the channel, respectively. Now the symbols can be mapped back to the generated bits through a QPSK demodulation and the receiver will compute the BER by checking bit per bit if the transmitted sequence is different from the received sequence: if so, then it is an error event and it is counted. If we have a sufficiently large amount of errors and transmitted bits, then the BER will be the ratio between the errors and the total transmitted bits. Thus, we will need to let the simulation run for enough time to have a consistent result. To do so, for each value of the SNR, we transmit not only one OFDM block, but several of them so that we can perform an average to smooth out the BER. In addition, we can repeat this entire experiment independently for some times in order to have even more accurate results.
In Fig. 6a the Confidence Intervals (CIs) are computed with the MATLAB built-in function `berconfint`. We notice that the CIs become wider with increasing SNR. This is due to the fact that the width of a CI is inversely proportional to the number of samples with which the statistic is computed: the larger the SNR gets, the less the receiver is making mistakes in decoding the received symbols, i.e., the number of errors will get close to zero, making the estimate of the BER less precise.



(a) BER vs SNR.



(b) Capacity vs SNR.

5 Step 5

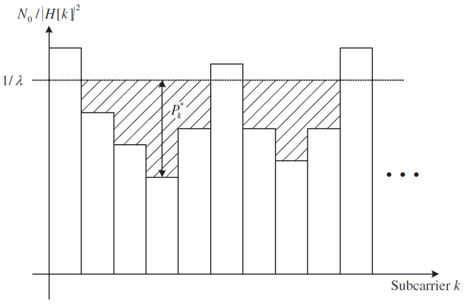
Ultimately, we assume that the transmitter can perfectly track the channel, which means that it has full CSIT. Thus, it can apply the waterfilling algorithm to allocate power to the subcarriers in function of the channel gain of each of them. In particular, we want to solve the following problem:

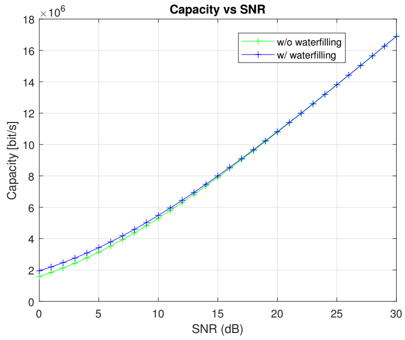
$$\max_{P_1,\dots,P_N} C_{OFDM} = \max_{P_1,\dots,P_N} \frac{W}{N+C} \sum_{n=1}^{N} \log_2 \left(1 + \frac{|\tilde{h}_n|^2 P_n}{N_0}\right) \\ \text{subject to the constraint } \sum_{n=1}^{N} P_n = \bar{P} \tag{16}$$

where $\bar{P}$ is the total maximum power. By solving this optimization problem with Algorithm 1, we end up with the optimal power allocation, that is, an array of non-zero values only at the indices representing the subcarriers that satisfy the condition $\frac{N_0}{|\tilde{h}_n|^2} \leq \frac{1}{\lambda}$. In other words, the waterfilling algorithm allocates power only to the subcarriers with a sufficiently good channel, as depicted in Fig. 7a. If we apply this algorithm to our system, by looking only at the BER, we cannot see any difference. This is due to the fact that, even though we are allocating different power on different subcarries, we use the same modulation for each of them and we are computing the statistic of the entire system: arranging power per subcarrier differently does not matter at a system level view. Instead, if we look to the capacity, in Fig. 7b, we will see an increase with respect to the case without CSIT, especially in the low SNR regime. On the other hand, in high SNR regime the two curves will overlap as deciding to allocate equal power to every subcarrier is asymptotically optimal when the channel is almost always good.

Algorithm 1: Waterfilling

sort channel gains in descending order;
n = N;
while $n \leq 1$ **do**
  $\lambda = \frac{n}{\bar{P} + \sum_{k=1}^{n} \frac{N_0}{|\tilde{h}_k|^2}}$;
  P $= \left(\frac{1}{\lambda} - \frac{N_0}{|\tilde{h}_k|^2}\right)$, $\forall k \in [1, n]$;
  if *last element of array P is non negative* **then**
    P1 = P;
    break;
  end
end
allocate P^* vector of zeroes long N;
fill P^* with the values in P1 only at the indexes (subcarriers) that received some power;
return P^*

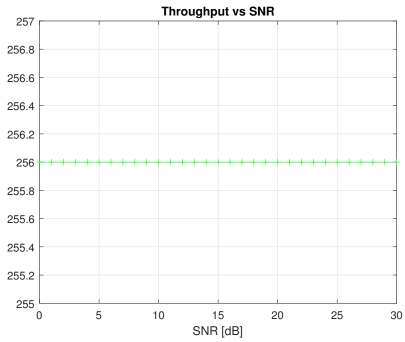


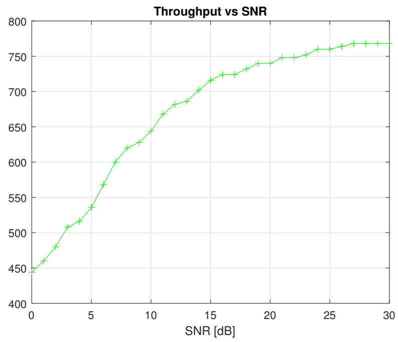


(a) Optimal power allocation with the waterfilling algorithm. [2]

(b) Capacity comparison with and without waterfilling.

Figure 7





(a) Throughput with QPSK and N = 128.

(b) Throughput with different modulations per subcarrier and N = 128.

Figure 8

However, the capacity is the maximum rate achievable by the system. If we compute the actual throughput, with or without waterfilling, we obtain a constant line throughout the whole SNR range, as shown in Fig. 8a. This is because we always transmit the same fixed amount of bits for each subcarrier, no matter what. Hence, to improve the situation, we need to transmit more bits on the subcarriers that have a good channel. To accomplish this goal, we use the strategy in Algorithm 2, the result of which is depicted in Fig. 9 with $SNR = 0$ dB and in Fig. 10 with $SNR = 35$ dB. In

Algorithm 2: Modulation assignment

$P_{max} = max(P^*)$;
$\forall k$ subcarriers having $P_k^* \geq P_{max} - \frac{P_{max}}{4}$ a 64-QAM is used;
$\forall k$ subcarriers having $P_{max} - \frac{P_{max}}{2.1} \leq P_k^* < P_{max} - \frac{P_{max}}{4}$ a 16-QAM is used;
$\forall k$ subcarriers having $P_{max} - \frac{P_{max}}{0} \leq P_k^* < P_{max} - \frac{P_{max}}{2.1}$ a 4-QAM is used;

Fig. 8b we have the resulting throughput, which is strictly better throughout the whole SNR range with respect to the scenario without CSIT, and thus, waterfilling. Of course, this does not come for free: by using high order modulations, the BER will be affected negatively (Fig. 11) as we are using more dense constellations, which are less robust to noise and thus the probability to decode the received symbols erroneously gets higher.

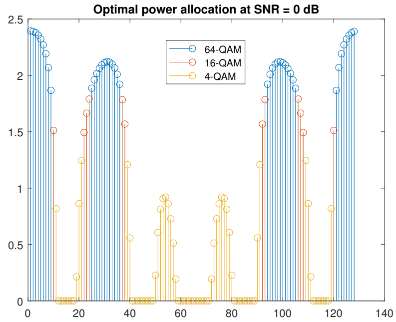


Figure 9

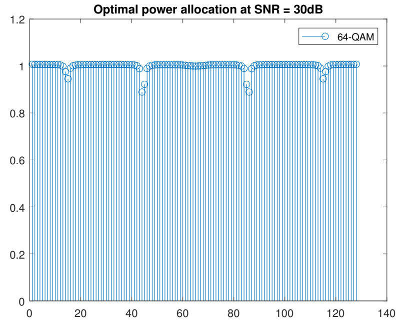


Figure 10

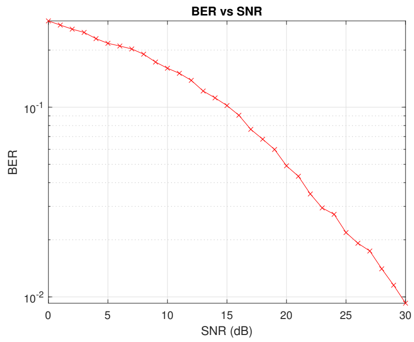


Figure 11: BER curve with waterfilling and different modulations per subcarrier.

References

[1] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[2] Yong Soo Cho, Jaekwon Kim, Won Y Yang, and Chung G Kang. *MIMO-OFDM wireless communications with MATLAB*. John Wiley & Sons, 2010.