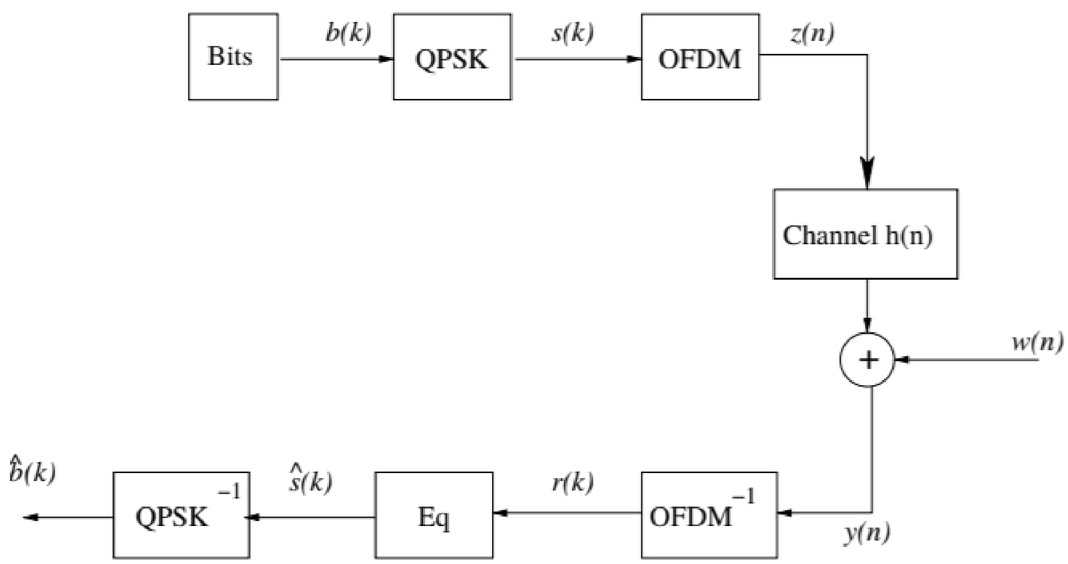
SSY130 - Project 1A
Baseband Communication

Diego Jauregui   Lucas Rath   Sondre Wiersdalen   Mohammad Shaikh

November 28, 2018

student id: 19930425
secret key: Jumpluff



1 Part A: Baseband Communication

Question 1) Default case: ideal channel, no noise added, $N_{cp} = 0$, know channel

(a) The ideal channel consists of a Kronecker delta function $h = \delta_0(n)$, which in frequency domain results in $H(\omega) = 1$. Therefore, the channel does not modify the signal $z(n)$ and for the cases where we do not have any noise, $z(n) = y(n)$.

The equalization will then convolute the distorted signal $r(k)$, removing the effect of the channel. However, since $H(\omega) = 1$, there is no effect to remove and therefore $\hat{s}(k) = r(k)/H(k) = r(k)$. As a result, the transmitted symbols are exactly equal the received symbols.

Furthermore, the errors related to the message transmitted are EVM: 2e-16, BER: 0, confirming that there were no errors bigger than machine precision and the transmitted symbols are correct.

(b) The cyclic prefix is essential for the proper operation of out communication scheme to work reliably because it acts as a guard region preventing interference between two sequence of OFDM symbols that might occur during the channel convolution.

In order to avoid completely this inter-symbol interference the cyclic prefix must be at least of the same size of the discrete impulse response of the channel $N_{cp} \geq N_h - 1$, where N_h is equal to the number of samples needed to the channel input response reach a "stationary" behaviour with value very close to zero.

In this way, the interference caused by the channel convolution of the last block will be entirely absorbed by the prefix samples, that will be discarded later by the receiver [Oltean and Naforniţă, 2003].

Another benefit of using cyclic prefix is that using this technique, we get a convoluted signal $y(n)$ that is exactly the same to the case when the transmitter was turned on at n=-∞. In this way, we can use the equalization procedure for periodic signals, which is faster since it makes operations using a DFT of length N samples instead of N+N_h-1.

(c) Both channels h_2 and h_3 consist of a Kronecker delta function $h = \delta_0(n)$ scaled by a factor. In other words, the channel convolution does not change the "dynamics" of the signal but only shifts the phase and/or scale the module. In frequency domain, those channel impulse responses can be described as:

$$\begin{aligned} h_2 &= 0.5 \cdot \delta_0(n) &&\Rightarrow H_2(w) = 0.5 \\ h_3 &= exp(0.5i) \cdot \delta_0(n) &&\Rightarrow H_3(w) = exp(0.5i) = 1\ \underline{/0.5} \end{aligned} \tag{1}$$

Therefore, α is equal 0.5 for h_2 and exp(0.5i) for h_3. We show then in figure 1 the pre-equalization symbol constellation diagram for both channels:

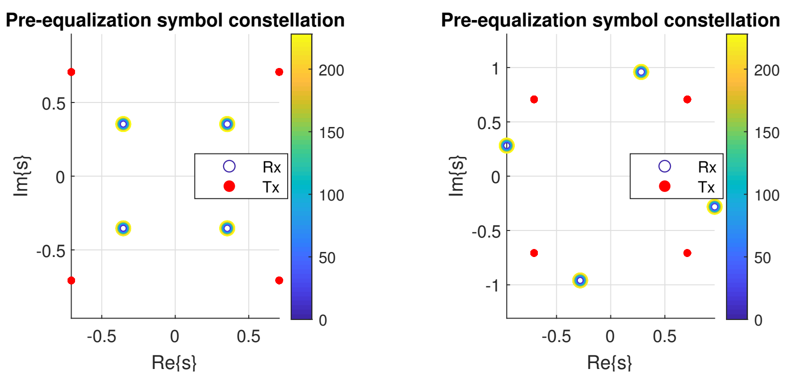


Figure 1: *Pre-constellation for channel $h_2(n)$ on the left and $h_3(n)$ on the right side.*

The post-constellation graphs are not displayed here but the transmitted and received signal matched perfectly for both cases, so that BER=0 and EVM=3e-16. As expected, the h_2 channel scaled the module of the symbols by 0.5, while the h_3 channel shifted the phase by 0.5 rad ≈ 28 deg.

(d) We now simulate the effect of a synchronization error when receiving the message. Introducing an delay of 1 time step we get errors EVM:1 and BER:0.5. Curiously, the received message is correct in the beginning and in the end but completely messed in the middle. To investigate this effect, first we recall that this introduced synchronization error is equal to an unexpected delay of the signal. In a frequency perspective:

$$x(n-k) \quad \rightarrow \quad X(w) \cdot exp(-j \cdot w/w_s \cdot 2\pi \cdot k) \tag{2}$$

where k is the delay in time step units. As can be noticed, this delay introduced a distortion in the frequency domain, shifting the phase of $X(w)$ in $-(w/w_s \cdot 2\pi \cdot k)$ rad. Notably, this will have very little effect when $(w/w_s \cdot 2\pi \cdot k) = 0$. Therefore, as $w \rightarrow 0$ or $w \rightarrow w_s$ this phase distortion will tend to zero.

In figure 2 we can confirm exactly this effect of phase shifting in the received symbols $\hat{s}(k)$. Using synch. delay $= 2$, we see that the phase was shifted $2\pi * 2$ rad in total between $w = 0$ and $w = w_s$. In this case, the symbols situated in $(w/w_s \cdot 2\pi \cdot k) = 0$ were not affected by the shift distortion, which corresponds to $w \rightarrow 0$, $w \rightarrow w_s$ and $w \rightarrow w_s/2$, i.e. the beginning, the end and the middle of the message respectively.

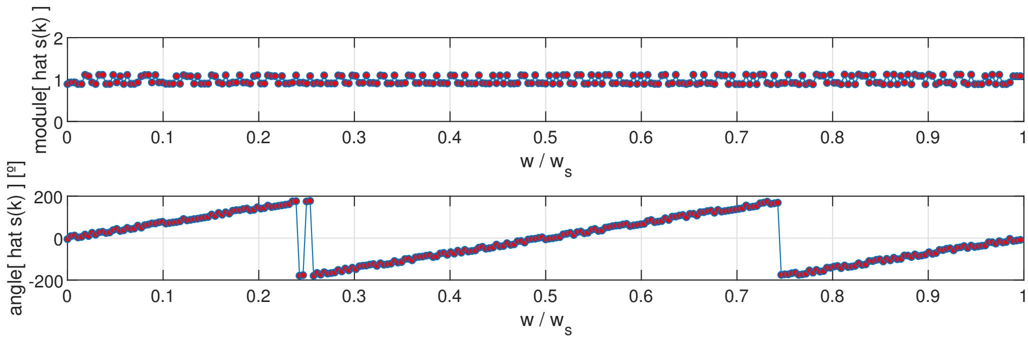


Figure 2: *Magnitude and phase of the received symbols $\hat{s}(k)$ for synch. delay = 2*

(e) Now we investigate the effect of adding noise when transmitting the message by changing the effective signal to noise ratio (SNR).

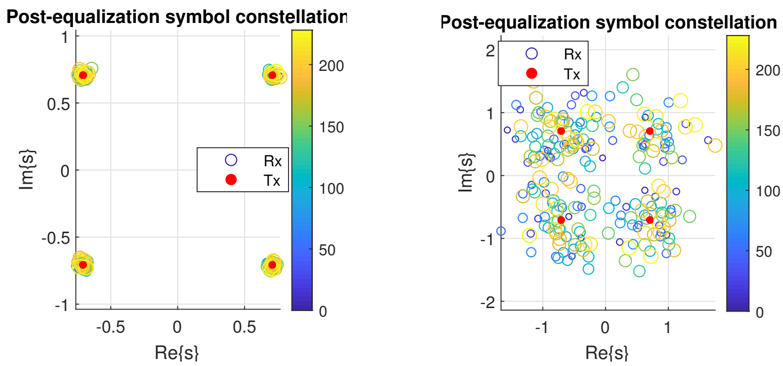


Figure 3: *Post-constellation for channel $h_1(n)$ with snr=30dB on the left and snr=5dB on the right side. Tx and Rx are the transmitted and received symbols respectively.*

As can be seen in figure 3 the more noise we add the more corrupted are the symbols. Clearly, one symbol is labeled correctly if after equalization it appears in its correct quadrant in the complex Cartesian plot. In other words, if the noise affects the received symbols but they still remain in their right quadrant, then the final message will not be corrupted. For snr=30dB the introduced noise was not enough to corrupt the message, so it was transmitted without miss-classifications. On the other hand, for snr=5dB we added so much noise that some symbols moved to the wrong quadrant, resulting in the following communication (EVM:0.5, BER:0.02):

Transmitted: `'Alice: Would you tell me, please, which way I ought to go'`
Received: `'Alice: Wo}ld you ]d,l me, please, whic_ way I\$ouoht to go'`

Question 2) channel model to h_4 (the low-pass system) and set the cyclic prefix to 60.

(a) Since we do not have noise in the system, we know that $r(k) = H(k) \cdot s(k)$. For this reason, the vector $r(k)$ will be a simply element-wise multiplication of $H(k)$ and $s(k)$. It follows then that $r(1) = s(1) \cdot H(1) = \sqrt{1/2}(-1+i) \cdot 5 = -3.5355 + 3.5355i$, so that $r(1)$ can be found in the constellation diagram.

We can confirm that relation by checking that symbols at the beginning and at the end of the message present almost no phase deviation since $\angle H(w \to 0) \approx 0$ and module amplification close to 5, since $|H(w \to 0)| \approx 5$.

(b) When the system is in this setup, the value of EVM=1e-15 and BER=0. We notice that for all $N_{cp} \geq 59$ EVM remains approximately zero. On the other hand, if we choose $N_{cp} < 59$ we get some considerable error EVM that starts to increase as we keep reducing N_{cp}. However, even setting $N_{cp} = 0$ is not enough to get miss-classification of symbols, so BER=0.

We conclude then that the magic number is $N_{cp} \geq 59$, which is exactly the size of the channel impulse response minus one, such that $N_{cp} \geq N_h - 1$. This size of cyclic prefix is the minimum size so that the finite transmitted signal $z(n)$ looks like being periodic.

It is also important to notice that if we are transmitting two consecutive OFDM blocks, the cyclic prefix area will also absorb the convolution interference of the channel h between the blocks.

(c) The value of N_{cp} is of high importance when using the channel $h_4^{'}$. If we use $N_{cp} \geq N_h - 1 = 59$ we get zero EVM and BER. As we keep reducing the value of N_{cp} bellow 59, we starts to get considerable EVM error, until the point we start to get bit errors BER.

Since we are doing equalization on only N samples instead of N+M-1, we need the cyclic prefix to make the finite transmitted signal looks like being periodic. That is why we get errors when we do not use cyclic prefix using this channel.

Question 3) Realistic case: ideal channel, no noise added, $N_{cp} = 0$, unknown channel

(a) In the unknown-channel scenario the effect of the nonzero sync error is less destructive for the message than in the know-channel scenario. This phenomena happens because the receiver uses the pilot frame to estimate the channel impulse response during the transmission of the the data frame. In addition, the pilot also captures this effect of synchronization delay in time domain. Therefore, during the equalization of the unknown channel, we remove the effect of the channel convolution as well as the effect of synchronization delay. The time delay captured by $\hat{H}_1(k)$ can be confirmed in figure 4, that is the same phenomena from question 1d).

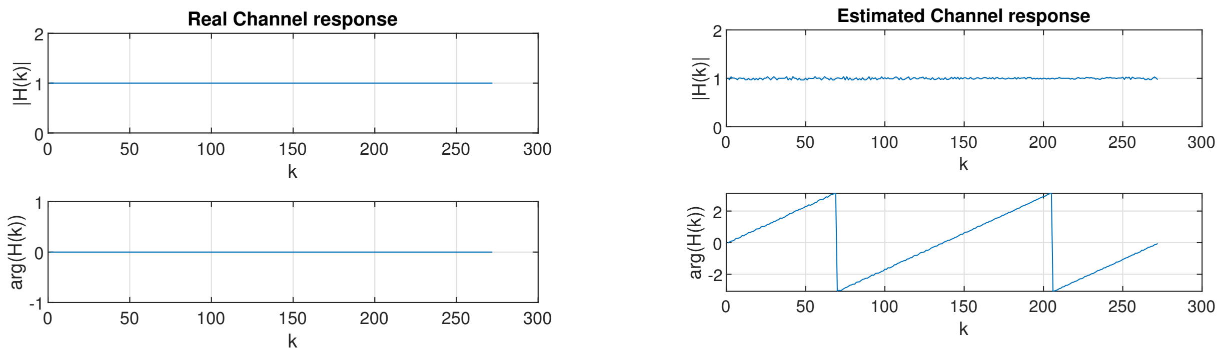


Figure 4: *Frequency response of $H_1(k)$ on the left and predicted $\hat{H}_1(k)$ on the right for synch. delay = 2*

(b) We need $N_{cp} \geq 59$ to obtain EVM=10^{-14}. This value is the same value we obtain in the know-channel scenario (question 2c). We obtain the same value for the N_{cp} in both scenarios because this is the minimum size that will avoid inter-symbol interference between the pilot and the data and also the minimum size that makes both frames look like being periodic signals.

After running some simulations, we noticed that this setup is more sensitive to noise level than the know-channel scenario. This happens because the system identification of the channel will also capture the noise introduced in the system:

$$\hat{H}(k) = \frac{r_p(k)}{s_p(k)} = \frac{s_p(k) * H(k) + w(k)}{s_p(k)} = H(k) + w(k)/s_p(k) \tag{3}$$

The equalization will then transmit this noise again to the final symbol, increasing then the error.

$$\hat{s}(k) = \frac{r(k)}{\hat{H}(k)} = \frac{s(k) * H(k) + w(k)}{H(k) + \underbrace{w(k)/s_p(k)}_{\substack{\text{Extra term}\\\text{due to sys.id.}}}} \tag{4}$$

To confirm this we can see in figure 5 the prediction of the channel corrupted by the introduced noise $w(k)$:

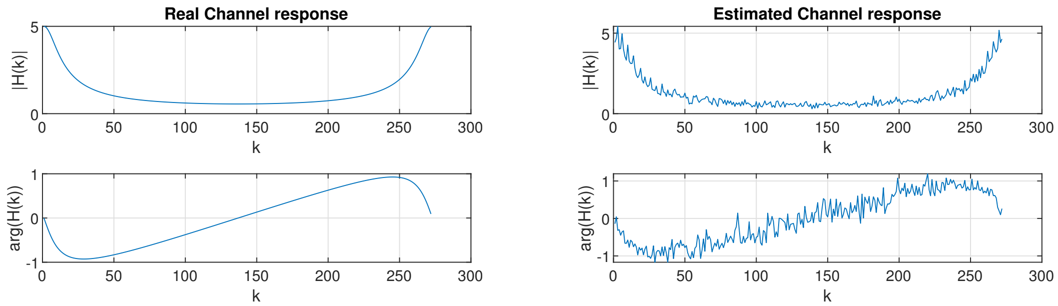


Figure 5: *Frequency response of the real channel h_4 on the left and the estimated channel on the right*

(c) Observing the results we confirm that the cause of getting some bit errors for the multi-path channel h_5 is because its impulse response has some periodic frequencies where $||H_5(w)|| = 0$. For this reason, the convolution $r(k) = H(k) * z(k)$ will then lose information in these mentioned frequencies. In addition to that, the equalization procedure will not be able to recover the data with zero magnitude and therefore $\hat{s}(k) = r(k)/H(k) = 0/0$, leading to bit errors for characters that correspond to those sub-channel frequencies.

The critical difference between h_5 and h'_5 is that the last one does not present any magnitude of zero in the frequency domain. In this way, the equalizer can remove the effect of h'_5 and recover all the information, avoiding completely the bit errors.

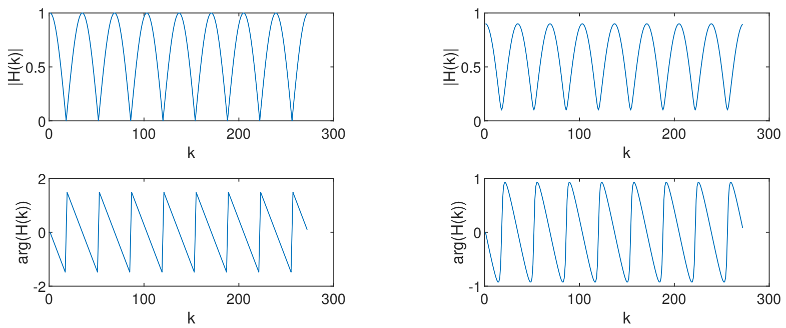


Figure 6: *Frequency response of $H_5(k)$ on the left and $H'_5(k)$ on the right*

(d) After playing around with the simulation, we focus in the BER as indicator of robustness against disturbance. In figure 7 are shown three analysis varying N_{cp}, SNR and sync-err. All the experiments were made using the unknown channel scenario, because it represents a more realistic simulation.

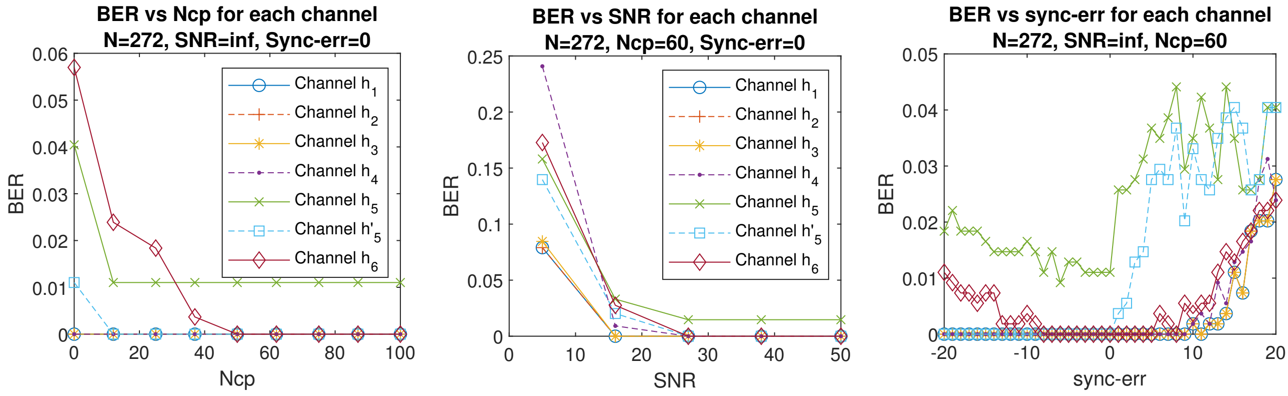


Figure 7: *BER vs* $\{N_{cp}$, *SNR, sync-err*$\}$ *plots for unknown channel scenario*

ISI - If we choose the value of N_{cp} wisely according to the length of the channel impulse response, we should not get any communication error in the system. However, even if we decide for a N_{cp} not big enough, the bit error BER will be relatively small in comparison to other communication problems, like noise of sync-error.

Noise (SNR) - For all the channels, the noise effect starts to be critical when SNR ≤ 30. As we reduce SNR, the bit error BER starts to increase exponentially. This shows how sensible is our model to disturbances. A possible way to avoid such sensibility might be through introduction of filters right after the receiver gets the message in order to filter the noise.

Synchronization error (early/late frame sync) - In these simulations, we have noticed that, most of the channels are quite sensible to synchronization error but it starts to really corrupt the message when $|\text{sync-error}| \geq 10$, for most of the channels.

We conclude then, that our communication scheme is not much robust against noise and synchronization error, since highly noisy environments or long synchronization delays can even cause more than 20% of bit errors in the received message.

We can also make a last remark about comparing the known and the unknown channel scenario. The noise sensitivity increases when the channel is estimated compared to when the channel is known. But on the other hand, we are more robust against synchronization errors when the channel is estimated. Hence, there is a trade off between different desirable quantities where we cannot get all of them together.

References

[Oltean and Naforniţă, 2003] Oltean, M. and Naforniţă, M. (2003). The cyclic prefix length influence on ofdm-transmission ber. *tom*, 48:62.