ELEC-H-401: Modulation and Coding

Students:

Fjonja KEÇI

Jialun KOU

Bilal MOUSA FARES

Giovanny V. RICO SORIA GALVARRO

Professor:

François HORLIN

Table of contents

1. Introduction

We are presenting with this report the simulation of the DVB-S2 communication chain. Various important parameters are shown in graphs to explain their impact. The project was divided in three parts which are presented and explained below.

2. Optimal communication chain over ideal channel

2.1 Simulation

Below is shown the block diagram of the communication system.

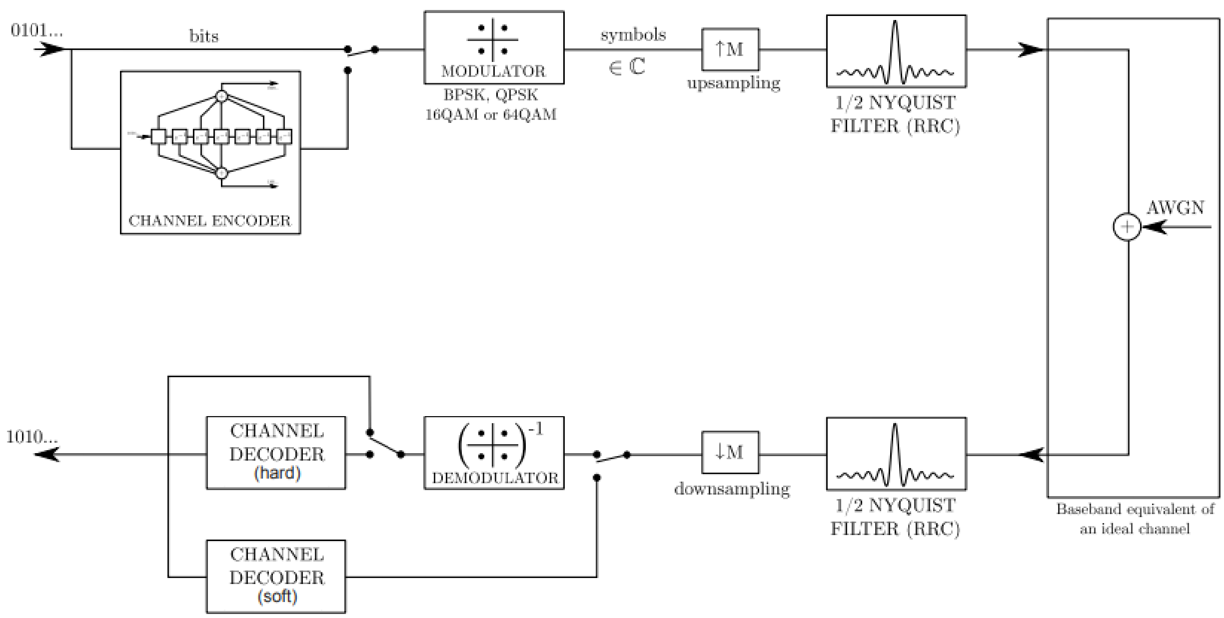


Figure 1. Block diagram of the communication system.

This schematic is implemented in the ideal channel. The mapping/demapping were provided to us.

2.1.1 Halfroot Nyquist Filter

The first step to do is implementing and simulating the mapping. After the mapping, the symbols will be convolved with a root raised cosine filter. This filter is used to limit the bandwidth of the channel. Figure 2, below, shows the PSD of the signal at the transmitter. The symbol stream below is shown after filtering, with the assumed cutoff frequency of 1 MHz and roll off factor equal to 0.3.

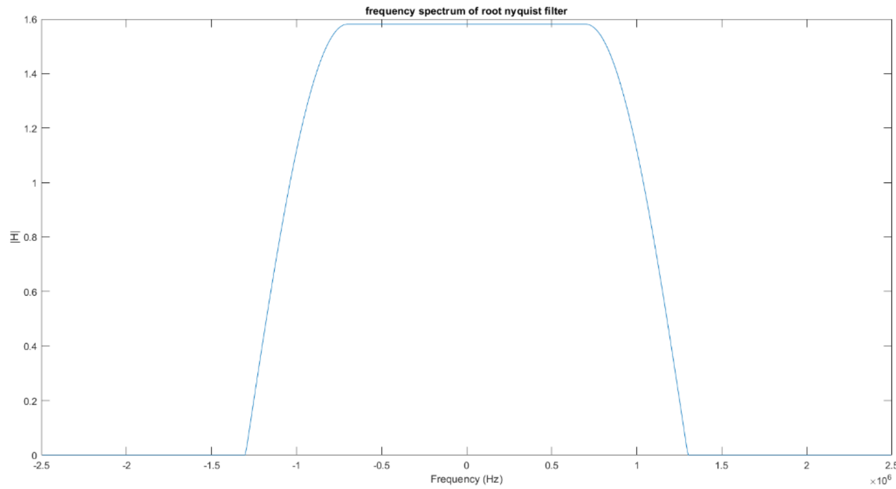


Figure 2. The PSD of the signal at the transmitter.

A filter which is matched to the one in the transmitter is applied to the receiver side as well. This filter, like the other, is a halfroot Nyquist filter that makes the signal-to-noise (SNR) ratio being maximized at the output. Both of the Nyquist halfroot filters together form a Nyquist filter that will cause a Dirac when sampled at a maximum. This is illustrated at Figure 3 below. This will cancel out the interference, meaning there will be no overlapping between the successive symbols.

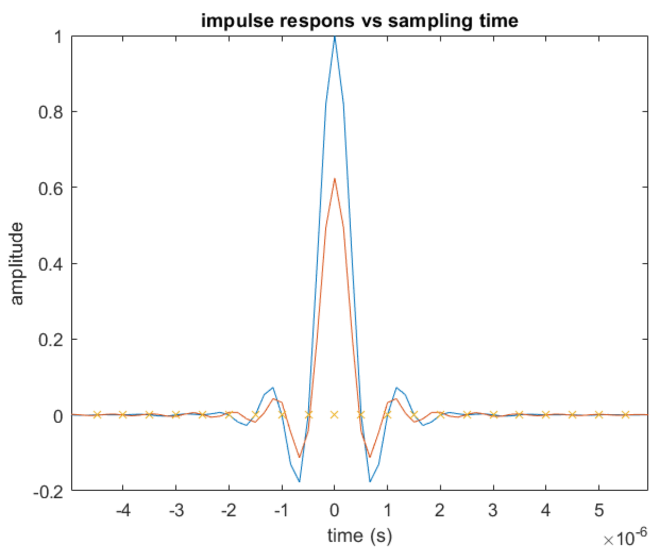


Figure 3. Plot of Dirac pulse caused by the Nyquist halfroot filter.

2.1.2 Noise Addition

The performance of the channel is currently limited by the Additive White Gaussian Noise (AWGN). We will use the baseband equivalent model of the noise since the simulation is performed at baseband. The channel was simulated for various signal-to-noise ratios and the results are shown below.

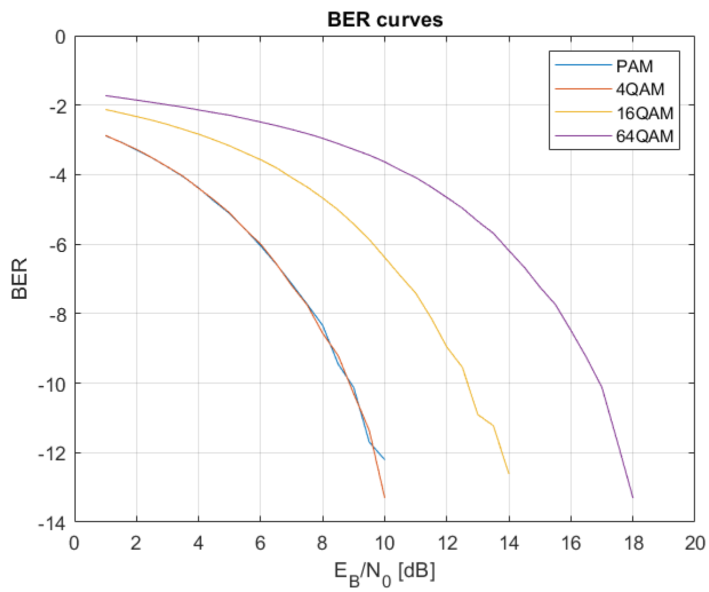


Figure 4. BER as a function of the signal-to-noise ratio for the PAM, 4QAM, 16QAM, 64QAM

They confirm the theoretical part of our course. As we can see when the constellation size is increased, for the same BER, more energy is needed. The plot also shows that PAM and 4QAM are overlapped and almost look like one. This happens because 4-QAM can also be seen as two PAM modulations in quadrature.

2.2 Questions

2.2.1 Regarding the simulation:

• *It is proposed to use the baseband equivalent model of the AWGN channel. Would it be possible to work with a bandpass implementation of the system?*

A bandpass implementation would need a really high sampling frequency, meanwhile the baseband equivalent model allows the implementation regardless of the carrier frequency.

Advantages:1.- To reduce significantly the necessary signal sampling rate (and therefore reduces the computational complexity); 2.- To develop modulation/demodulation techniques independently of the carrier frequency (and therefore supports system flexibility).

• *How do you choose the sample rate in Matlab?*

The sampling rate should be at least doubled the amount of the symbol frequency.

• *How do you make sure you simulate the desired Eb/N0 ratio?*

We use the SNR desired from 1 to 20 dB, and we calculate the signal power that is in symbols through the trapez() function (approximation of the integral). The power of the bandpass signal is the power of the complex envelope halved. We apply that to signal power and we get the No with the formula: N0 = Eb/(10.^(SNR/10)).

• *How do you choose the number of transmitted data packets and their length?*

It will depend on the desired BER range. The rule is that for a BER of 10^{-n}, we would need to send more than 10^{n} bits, since only then we would accurately confirm how many bits are badly received. Furthermore, we also need to make sure that the amount of bits fit the amount of desired symbols, if we want n symbols with x bits per symbol, we would need to send n*x bits, if the ratio #bits/#symbol is not an integer, then there would not be enough bits to make the last symbol.

2.2.2 Regarding the communication system:

• *Determine the supported (uncoded) bit rate as a function of the physical bandwidth.*

We know that the bit rate is given by the formula $R = \frac{\log_2 M}{T}$ where M represents the number of symbols and T represents the symbols duration. The Nyquist filtering gives us the relationship between the duration of the symbols T and the bandwidth ($\Delta f = \frac{2}{T}$). Therefore, we have the formula that shows the relationship between the bandwidth and the bit rate: $R = \frac{\log_2(M)\Delta f}{2}$

• *Explain the trade-off communication capacity/reliability achieved by varying the constellation size.*

The increase of the constellation size (the number of bits transmitted per symbol M) allows us to increase the bit rate. However, the BER increases because of the decrease of the minimum Euclidian distance between symbols (constellation size decrease).

• *Why do we choose the halfroot Nyquist filter to shape the complex symbols?*

This way we can limit the bandwidth, which is more efficient. The matched filter at the receiver maximizes the SNR of the output signal. The overall filter satisfies the Nyquist ISI criterion (both half root filter in transmitter and receiver), which cancels out the Inter Symbol Interference.

• *How do we implement the optimal demodulator? Give the optimisation criterion.*

The received signal is passed through a bank of K filters, that compute the projection of r(t) onto basis functions of modulation scheme. $h_i=s_i$ (-t) i=1,…,K. This is attained by using the matched filter. The received signal is matched filtered and sampled at the symbol rate. Matched filters can maximize the SNR at the symbol rate.

• *How do we implement the optimal detector? Give the optimization criterion.*

We have two possible ways of implementing the optimal detector. The first one is the Maximum a posteriori or simply MAP, which maximizes the probability of a correct decision, or equivalently minimizes the probability of making an error (bit error rate)

$$\tilde{\underline{s}}_m^{MAP} = \max_{\underline{s}_m} p(\underline{s}_m | \underline{r})$$

The MAP detector is equivalent to the ML detector when the M transmitted signals are equally probable $p(\underline{s}_m) = 1/M$. The ML criterion reduces to find the signal s_m that is closest in distance to the received vector $\underline{r}$ (minimum Euclidian distance).

$$\tilde{\underline{s}}_m^{ML} = \max_{\underline{s}_m} p(\underline{r} | \underline{s}_m)$$

3. Low-density parity check code

3.1 Simulation

In section 2 we skipped the channel encoder, so in this is section we are going to show what happens after we implement it as a low density parity check (LPDC) code.

We compare the performance of hard and soft decoding, which we implemented with a rate of ½.

3.1.1 Hard Decoding

According to the Tanner graph we implement the hard decoding. The graph below in figure 5 shows the BER curves.

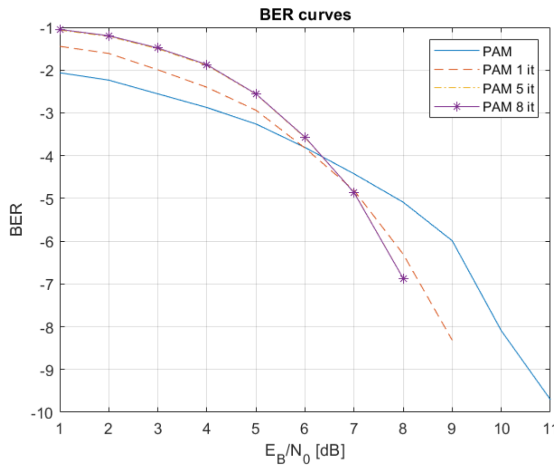


Figure 5. Effect of Hard decoding on BER curves

As the figure above shows, we can see the performance is worsened for a small SNR. As we know from theory, this happens because there are errors added, as a result of a noise, which is in fact trying to fix them, but does the opposite. The performance in the high SNR region is improved after we raise the number of iterations, while low SNR BER is worse when the number of iterations is increased.

3.2 Soft Decoding

The next step was implementing the soft decoding to our channel. As shown in the figure below we can see an improvement to the BER. This proves that implementing the soft decoder does help reducing the bit error rate. This happens because the hard decoder while trying to fix the errors, in fact, increases them in low SNR, meanwhile the soft decoder, which uses the probabilities, decreases them. We can say that the soft decoder is more reliable and shows better performance.

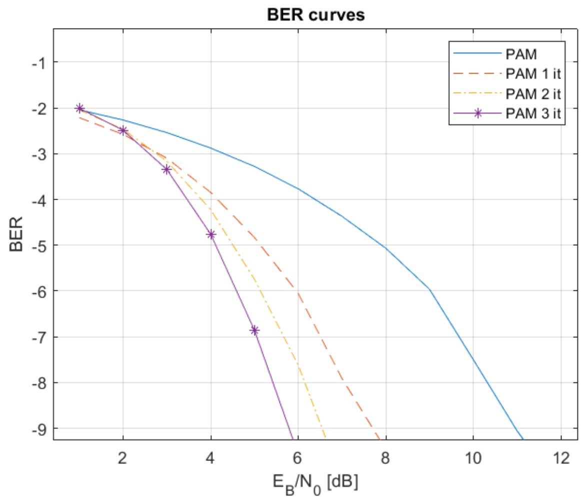


Figure 6. Effect of the Soft decoding on the BER curve.

3.3 Questions

3.3.1 Questions regarding simulations

- *When building the new BER curves, do you consider the uncoded or coded bit energy on the x-axis?*

We consider the coded one.

- *How do you limit the number of decoder iterations?*

For the hard decoder the limit is the maximum number of iterations or when the syndrome is zero, meanwhile for the soft decoder it is either the maximum number of iterations or if it is detected that there are no errors in the code.

- *Why is it much simpler to implement the soft decoder for BPSK or QPSK than for 16-QAM or 64-QAM?*

In case of BPSK or QPSK to compute Euclidian distance we only take into consideration the real and the imaginary part, meanwhile for 16-QAM or 64-QAM it should be computed when coding.

3.3.2 Regarding the communication system:

- *Demonstrate analytically that the parity check matrix is easily deduced from the generator matrix when the code is systematic.*

When the code is systematic, the Generator matrix has the form: $\underline{\underline{G}} = \left[\underline{\underline{P}} \,\middle|\, \underline{\underline{I}}_K \right]$ where P is the parity array portion with size K x (N – K) and I the identity matrix with size K. In case of a systematic code also (modulo-2 addition or subtraction are equivalent) the H matrix is given by: $\underline{\underline{H}} = \left[\underline{\underline{I}}_{N-K} \mid \underline{\underline{P}}^T \right]$ of size (N – K) x N such that the rows are orthogonal to the generator matrix: $(\underline{\underline{G}} \cdot \underline{\underline{H}}^T = \underline{\underline{0}})$ so from this equation and the other ones, replacing here we have:

$$\underline{\underline{P}}\,\underline{\underline{I}}_K(\underline{\underline{I}}_{N-K}P^T)^T = \underline{\underline{P}}\,\underline{\underline{I}} \oplus \underline{\underline{I}}\,\underline{\underline{P}} = 0$$

This is a modulo-2 addition, as we have systematic code, what shows that P can be obtained from G as it matches the condition.

- *Explain why we can apply linear combinations on the rows of the parity check matrix to produce an equivalent systematic code.*

The rows of the parity check matrix are orthogonal to the rows of the generator matrix. Basically the rows of the matrix H form a subspace complementary to the one of the code. So the linear combinations of the base vectors that form the columns of parity check matrix H will span the same subspace, and thus produces a code that will be equivalent.

- *Why is it especially important to have a sparse parity check matrix (even more important than having a sparse generator matrix)?*

The sparse parity check matrix has a small number of "1"s, therefore there are not many connections between the check nodes and the variable nodes, which reduces the complexity. Having a sparse parity check matrix is important because it reduces the computation time of the decoding.

- *Explain why the check nodes only use the information received from the other variable nodes when they reply to a variable node.*

The information should be stochastically independent for both of the decoders (hard / soft), which is why the information used is the one not related to the current node. The check node assumes the other connections with variable nodes are correct in order to "fix" the current node connected.

4. Time and frequency synchronization

4.1 Simulation

In this section we have simulated the synchronization errors, frequency shift and the carrier phase. We will show how the algorithm and codes reduce and correct these errors.

4.1.1 Synchronization errors

The figure below shows the effects of the implementation of the CFO. The phase shift which is linearly increasing in this case, will cause a rotation, which will lead to creating a circle. Figure 7 shows the noiseless channel and the figure 8 is where we added the noise.

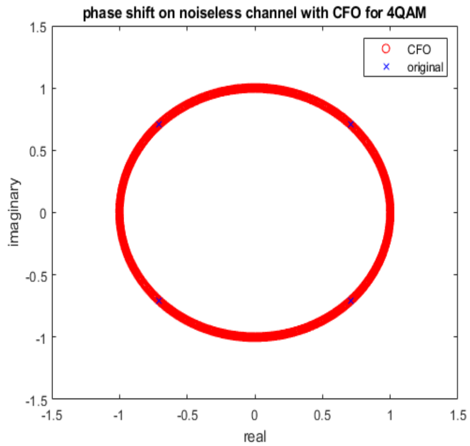


Figure 7. Phase shift on noiseless channel with CFO.

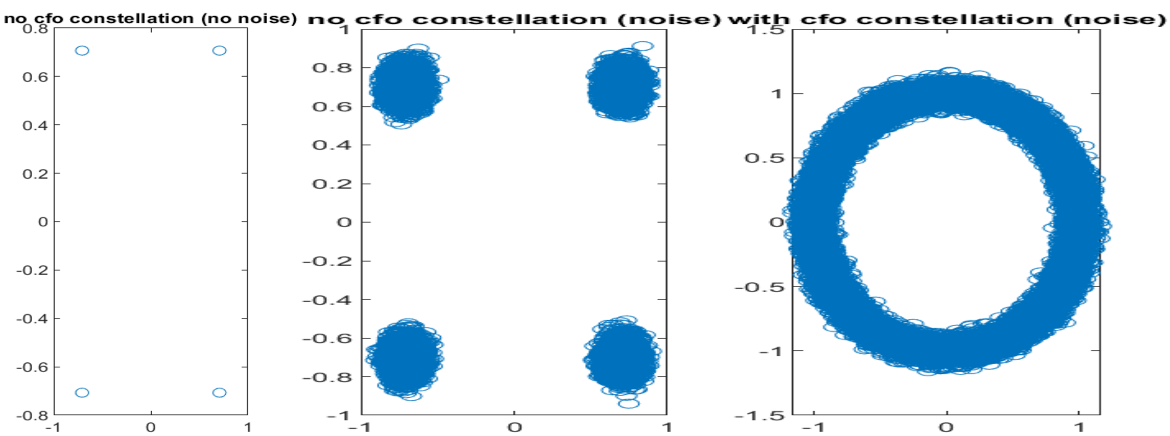


Figure 8. The channel after adding noise.

The figure 9, below, shows the impact of the time shift we simulated. We can see that for values equal or lower than 0.02T there is nearly no impact, but when the time shift rises above 0.02T then the error becomes more significant.

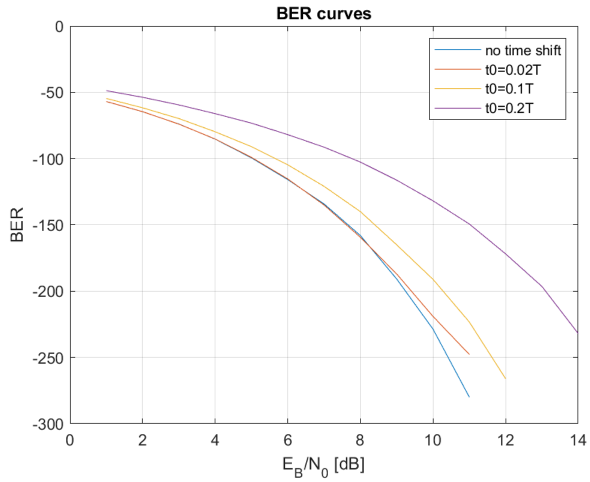


Figure 9. The effect of the time shift on the BER curve.

Figure 10 shows the impact on the BER after implementing the CFO. Lower values of the E_B/N_0 do not disturb the curve much, but higher values prevent a correct communication as the figures clearly show.

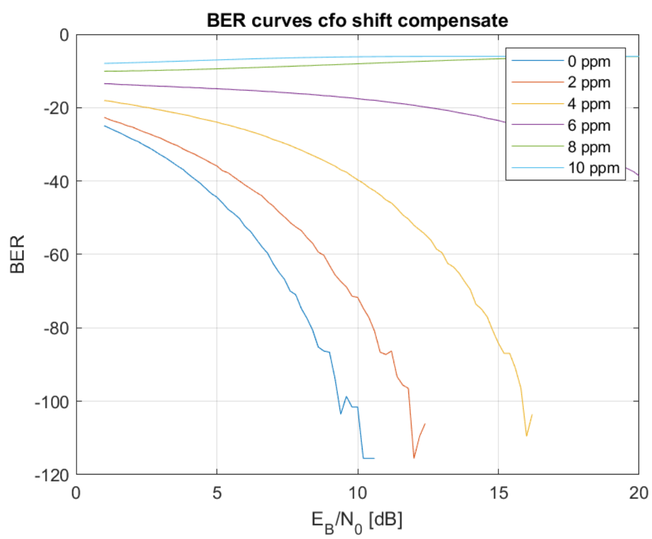


Figure 10. The effect of CFO in the BER curve.

4.1.2 Correction algorithms

The first unwanted effect is the shifting of the sampling time. This will get corrected by the Gardner algorithm. Figure 11 shows this correction.

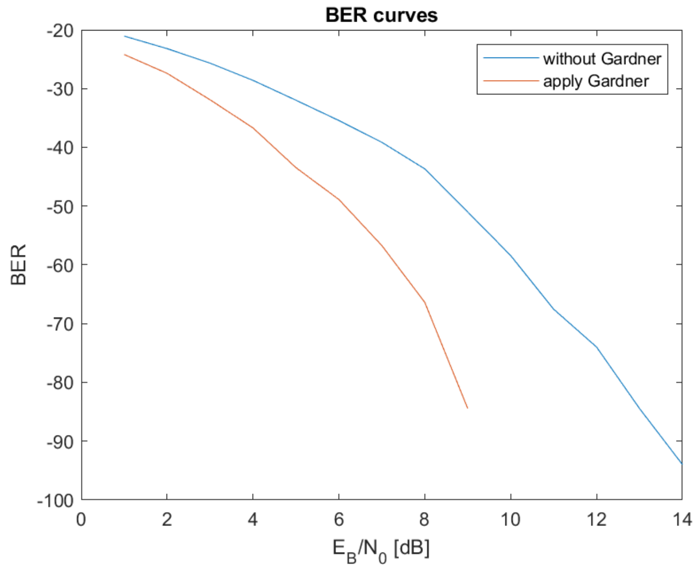


Figure 11. The effect of Gardner algorithm in sampling time shift.

An important parameter of the Gardner algorithm is the factor κ (kappa). We can see the impact of this parameter on figure 12. This figure shows clearly that increasing the k can certainly allow a faster convergence but it also increases the variance which means a less stable estimate.

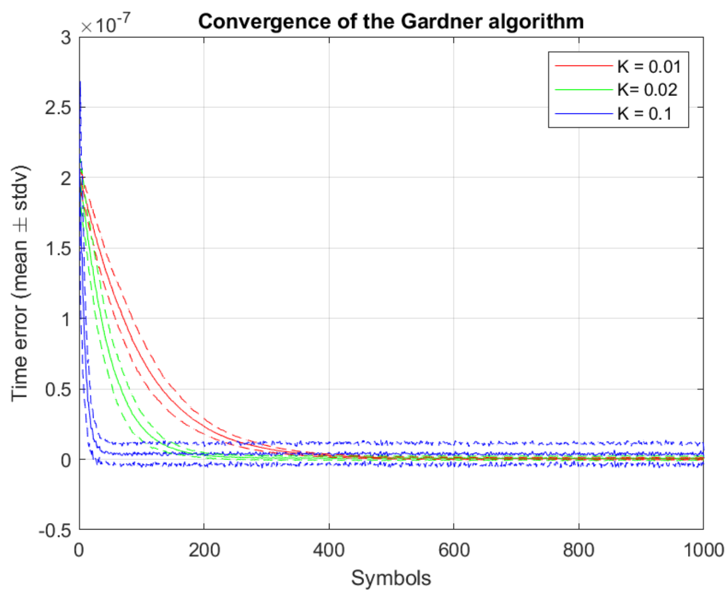


Figure 12. Impact of k in the convergence of the Gardner algorithm.

The last algorithm is the frame and time acquisition ones, in the Figure 13 are shown below. The time and frequency errors variance, when we keep N constant and we increase K (N being the length of the pilot and K the cross correlation averaging window) we notice that we improve at the beginning but then increasing K won't improve anymore, the error is still equal. Keeping K constant and increasing N, it improves the curves in a different way.

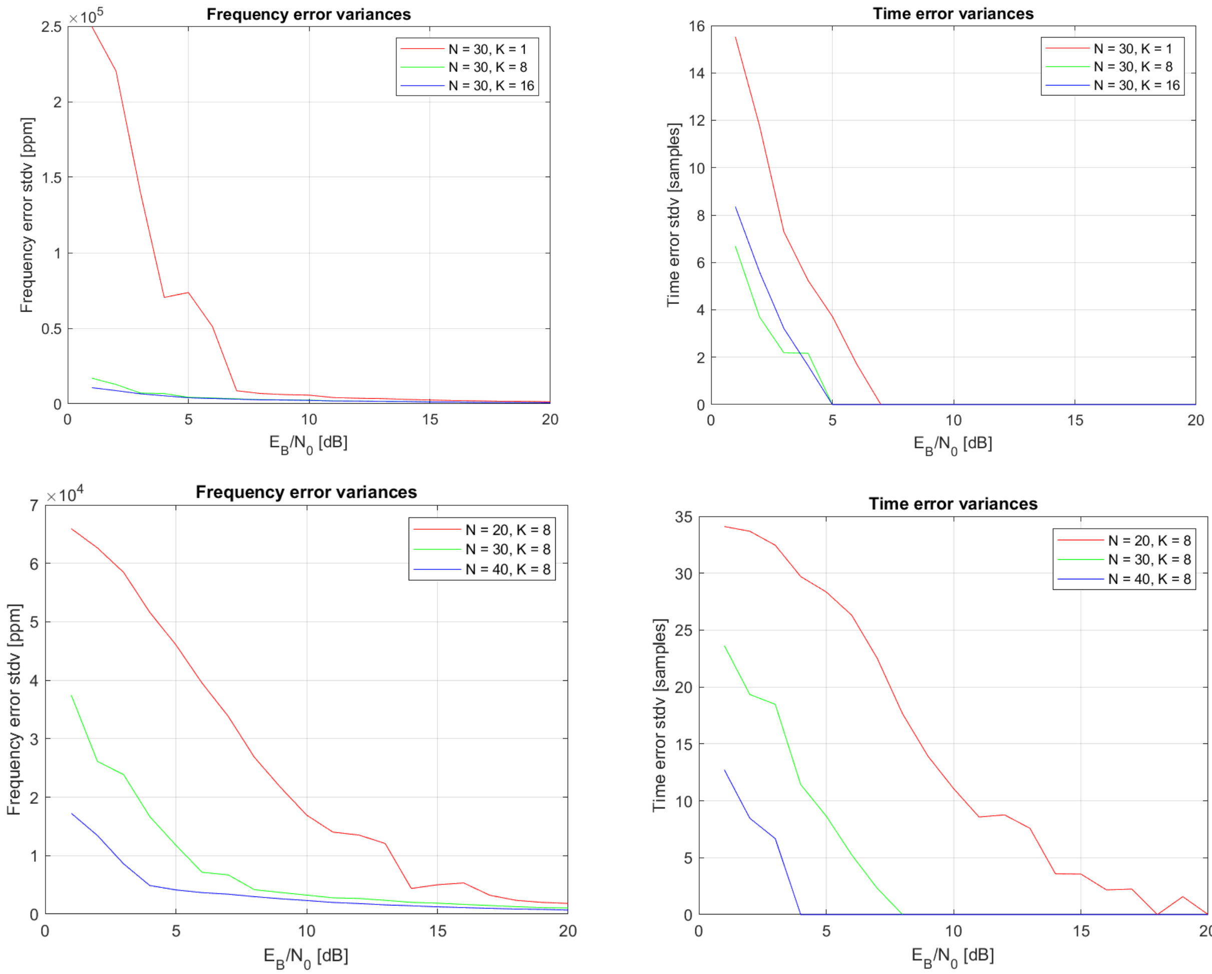


Figure 13. Impact of K and N in the Frame and Frequency Acquisition algorithms

One more time, we can see the robustness of the algorithm against the CFO, as it doesn't change the shape of the curve respect to the one without the CFO (Fig. 14).

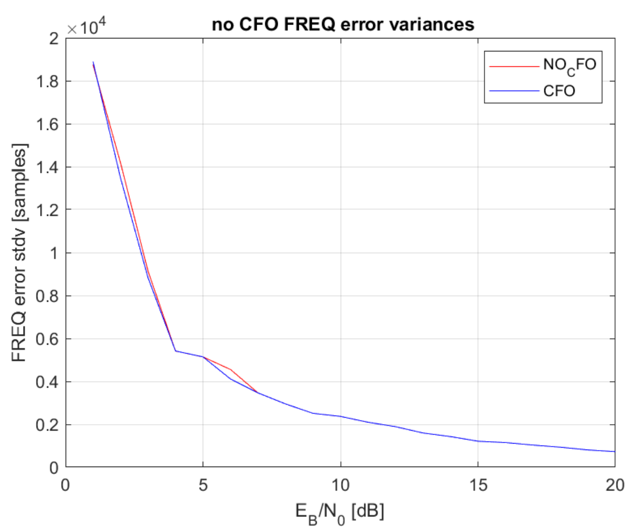


Figure 14. CFO effect on the algorithm.

4.2 Questions

4.2.1 Questions regarding simulations

- *Derive analytically the baseband model of the channel including the synchronization errors.*

$$y_1(t) = (x_1(t)\cos(\omega_0 t) + x_2(t)\sin(\omega_0 t))\cos(\omega_0 t + \Delta\omega t + \Phi_0)$$

$$y_1(t) = x_1(t)\cos(\Delta\omega t + \Phi_0) + x_2(t)\sin(\Delta\omega t + \Phi_0)$$

$$y_2(t) = (x_1(t)\cos(\omega_0 t) + x_2(t)\sin(\omega_0 t))\sin(\omega_0 t + \Delta\omega t + \Phi_0)$$

$$y_2(t) = x_1(t)\sin(\Delta\omega t + \Phi_0) + x_2(t)\cos(\Delta\omega t + \Phi_0)$$

Applied the Low Pass Filter (LPF) to the lasts y_1 and y_2. Now adding the results, considering y_2 is multiply by the imaginary unit j, we obtain:

$$y_1(t) + jy_2(t) = x_1(t)\,(\cos(\ldots) + jsin(\ldots)) + x_2(t)\,(-\sin(\ldots) + jcos(\ldots))$$

$$y_1(t) + jy_2(t) = [x_1(t) + jx_2(t)]e^{j(\Delta\omega t + \Phi_0)}$$

- *How do you separate the impact of the carrier phase drift and ISI due to the CFO in your simulation?*

We implemented a function at the output of the matched filter. The function is represented by this formula: exp(-1i.*(dw.*t+phi)). This manually compensates the phase drift, afterwards we can observe the ISI.

- *How do you simulate the sampling time shift in practice?*

We upsample the signal. Then we sample a couple of shifts from the original sample index when downsampling the signal. If the upsampling ratio is 100, we can move all symbols for example 10 samples to the right, this would correspond with a sampling shift of 0.1T.

- *How do you select the simulated Eb/N0 ratio?*

To support the data communication, so that the impact on the performance is not too high, the error on the sampling time instant should be no more than 2%. Our Eb/N0 needs to be moderately high in order for the algorithms to converge, since a noiseless simulation does not exist, we decided to go with a SNR of at least 5 dB, this would ensure that our algorithms will sufficiently correct the CFO and time shift errors.

- *How do you select the lengths of the pilot and data sequences?*

To ensure a correct phase interpolation between two consecutive pilots we select the length of the data sequence. We need to limit our length in order to keep the bitrate from decreasing too much.

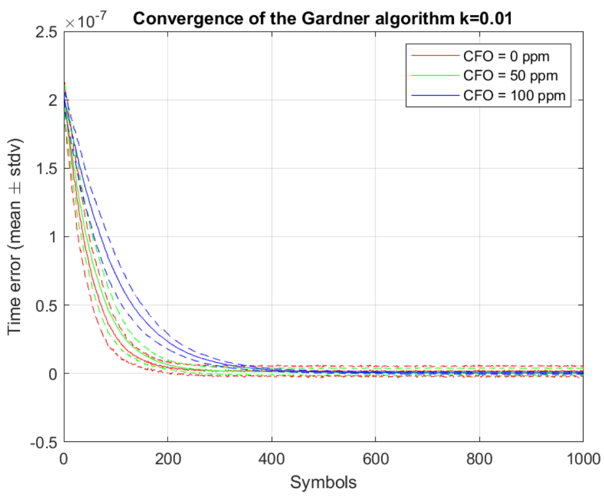
Meanwhile the length of the pilot should be long enough so that we can get a good estimate of the CFO and ToA.

4.2.2 Regarding the communication system:

- *In which order are the synchronization effects estimated and compensated. Why?*

Since the Gardner algorithm is robust to CFO, it is the first to be executed to correct the time shift, afterwards the remaining CFO and ToA are corrected using frame acquisition.

- *Explain intuitively how the error is computed in the Gardner algorithm. Why is the Gardner algorithm robust to CFO?*



The estimation of the time shift is obtained by the estimation of a previous one weighted by a certain factor. This factor will depend on the sample in the middle of two time steps. The magnitude of the correction is given by the magnitude of this sample and the direction of the correction is given by the sign of the middle sample. If there is a zero crossing, the middle sample will be near to zero and the correction will be low. The robustness of Gardner to CFO is displayed in the Figure. The Gardner is applied to 50 and 100 ppm CFO and the curve converges in all cases, the only difference is the lower conversion speed for higher CFO. This is because the algorithm is local, the CFO is therefore not relevant for the calculation of the error since we only look at 3 sequential samples.

- *Explain intuitively why the differential cross-correlator is better suited than the usual cross-correlator? Isn't interesting to start the summation at $k = 0$ (no time shift)?*

Because the usual cross correlator depends on a complex term in the maximum likelihood criterion, where we have to compute all the possible shifted by CFO values of all the possible preambles, what derivates a 2D problem that is not easy or practical to implement. We do not need to start with k=0 since then we do not have any information besides the power of the window.

- *Are the frame and frequency acquisition algorithms optimal? If yes, give the optimization criterion.*

No, these algorithms aren't optimal since the CFO and ToA are corrected sequentially and not at the same time. However, if we have CFO values that are low enough, which is the case in most realistic cases, we can correct the synchronization errors with reasonable accuracy.