Brussels Faculty of Engineering

Modulation & Coding

ELEC–H401

Design & Simulation of a DVB–S2 Communication Chain

Authors:
Cédric Hannotier
Badr-Ali Mouaden
Mahtieu Petitjean

May 7, 2017

Contents

1 Introduction

This report presents the simulation of a DVB-S2 communication chain. The most important design parameters were varied in order to illustrate their impact on the quality of the communication via the bit error rate (BER).

This project was divided in three steps. Each of these steps is presented, then various graphs illustrate the important concepts and results. To conclude each section, various questions regarding the simulation and the communication chain itself are answered.

2 Optimal communication chain over the ideal channel

2.1 Simulation

The implementation scheme is shown in Figure 1.

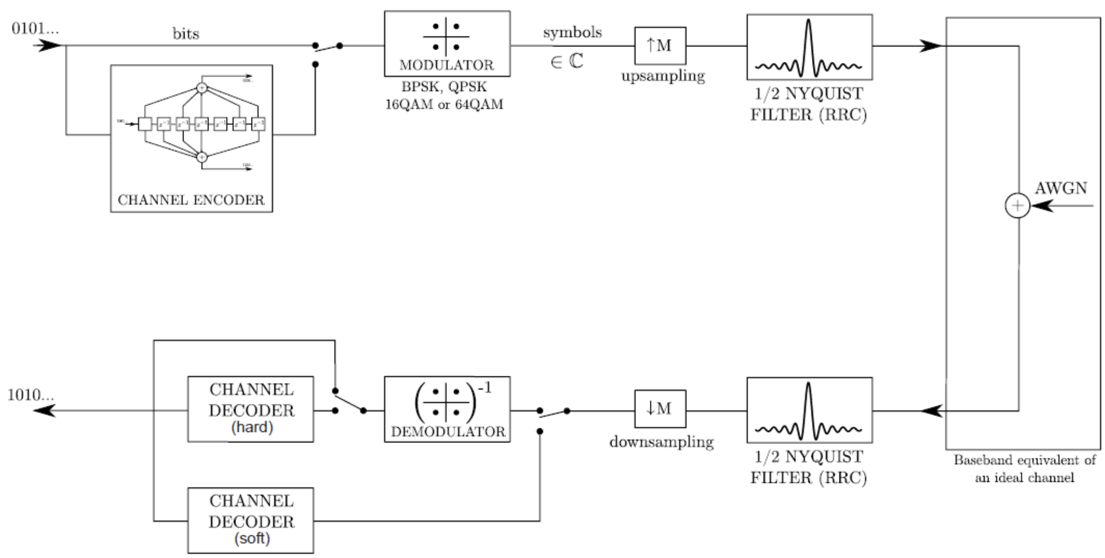

Figure 1: Block scheme of the ideal channel

The various components of the communication chain are detailed hereafter. The channel encoder is not yet implemented, and the symbols mapping/demapping were provided.

2.1.1 Halfroot Nyquist filter

After the mapping, the complex symbols are convolved with a raised-cosine filter. It allows to limit the communication bandwidth, as illustrated in Figure 2 where the PSD of the symbol stream are shown before and after filtering, with a rolloff factor β of 0.3. The symbol frequency was fixed to 2 MHz, which translates to a cutoff frequency of 1 MHz.

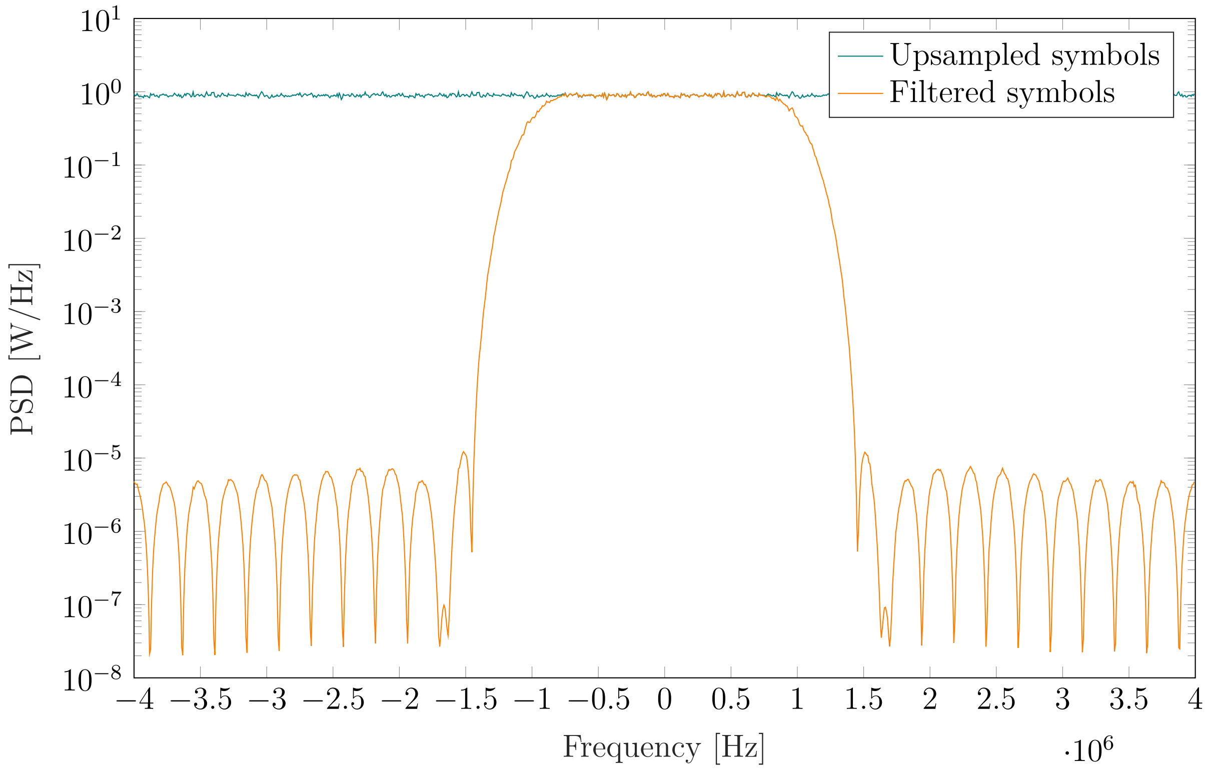

Figure 2: PSD of the symbol stream at the transmitter before and after filtering

At the receiver side, the stream is convolved again with the matched filter, forming a Nyquist filter and maximizing the SNR. The Nyquist filter allows to cancel the ISI in the sense that it reduces to a Dirac pulse when sampled at the symbol frequency, as shown in Figure 3. Hence, no symbol of the symbol stream overlaps with the current received symbol.

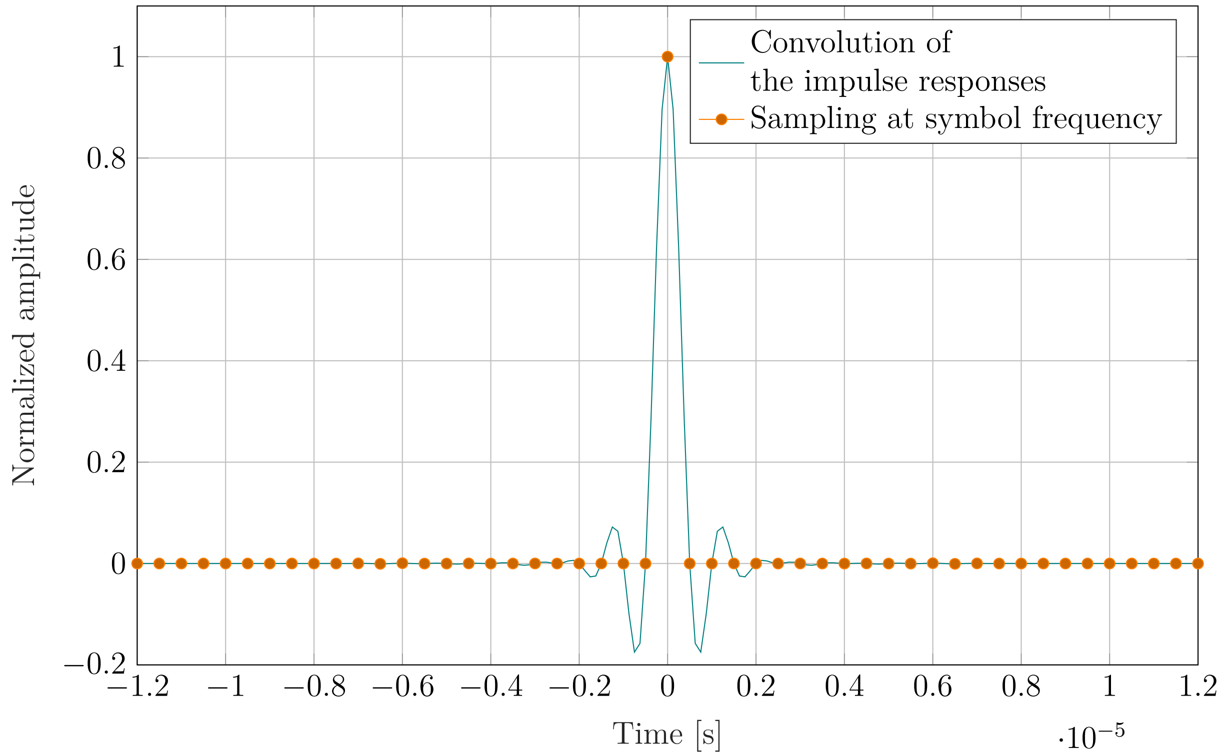

Figure 3: Illustration of the cancellation of ISI by the Nyquist halfroot filter

2.1.2 Noise addition

The channel is corrupted by Additive White Gaussian Noise (AWGN), which is simulated using its baseband equivalent representation. The channel was simulated for various SNR (defined with respect to the bit energy) and various modulation schemes, and the results are plotted in Figure 4. The simulations confirm the theoretical results.

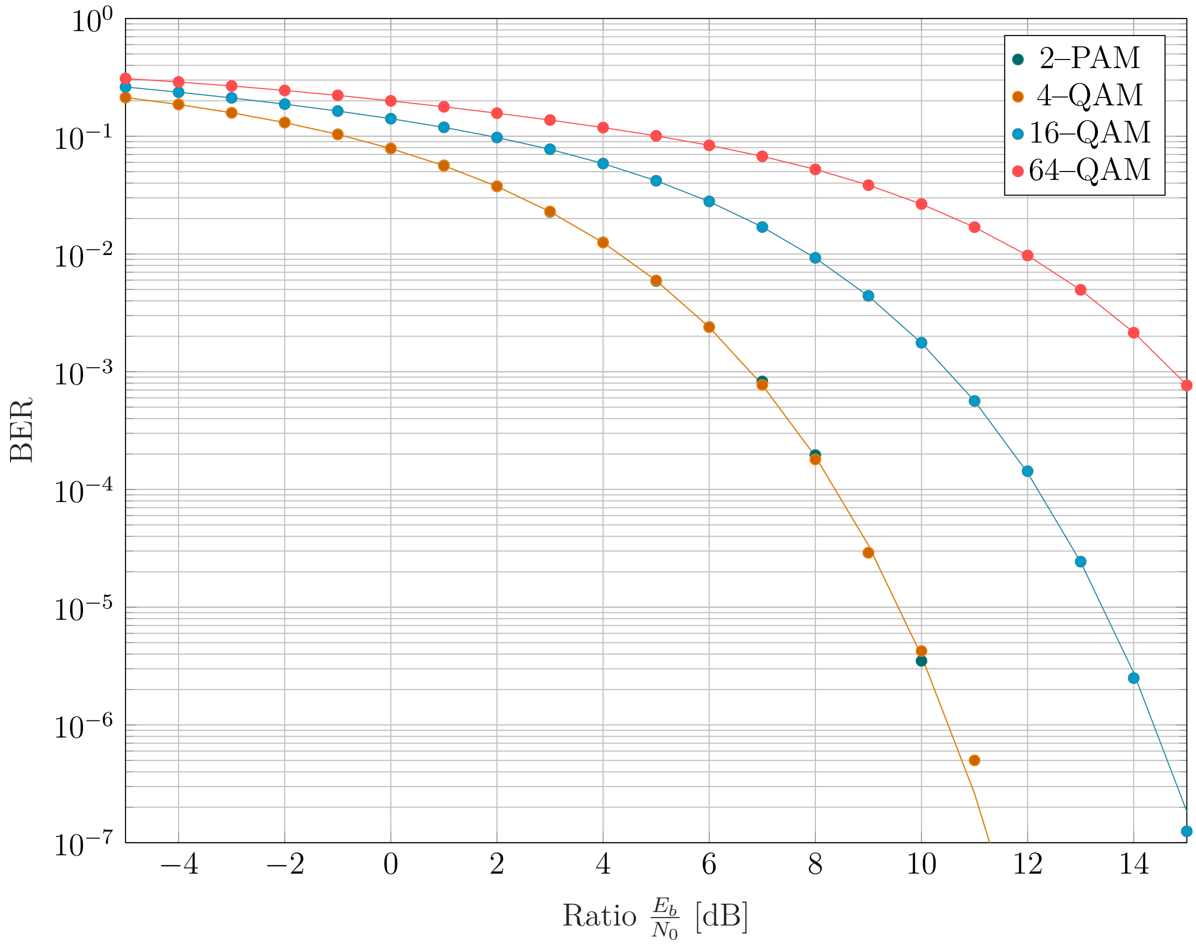

Figure 4: Bit error rate as a function of the SNR for various modulations

As expected, more energy is required to have the same BER when the constellation size is increased. It is worth noting that the worst BER value that could be achieved is 0.5, corresponding to a random choice of the binary values. Moreover, the BER curves for 2-PAM and 4-QAM are superposed because a 4-QAM modulation can be seen as two 2-PAM modulations in quadrature.

2.2 Questions

2.2.1 Questions regarding the simulation

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

The baseband model allows to implement the simulations regardless of the carrier frequency. If the system were simulated in bandpass, the sampling frequency needed would be extremely high and unrealistic.

How do you choose the sample rate in Matlab ?

The sampling rate should be at least twice as high as the symbol frequency.

How do you make sure you simulated the desired E_b/N_0 ratio ?

First the energy of a bit is evaluated, then the noise power is computed in order to obtain the desired E_b/N_0 ratio. The energy of the bit E_b is given by

$$E_b = \frac{T_{samp}}{2n_{bit}} \sum_n |s[n]|^2 \quad (1)$$

Then N_0 can be deduced and the noise is added to the symbols using the baseband representation :

$$r[k] = \sqrt{N_0 f_{samp}} \{\mathcal{N}(0,1) + j\mathcal{N}(0,1)\} \quad (2)$$

The square root is present because of the baseband representation of the noise. Indeed, the computed power is equally distributed between the real and imaginary parts of the noise.

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

A rule of thumb is to send 10^{n+2} bits of data in order to reliably observe of BER of 10^{-n} as it corresponds to 100 detected errors. Moreover, the total number of bits should be a multiple of the number of bits per symbol used in the modulation scheme. Figure 4 was realised by sending around 10^12 bits of data.

2.2.2 Questions regarding the communication system

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

The bit rate R is given by $R = \frac{log_2(M)}{T}$ where M is the number of symbols and T is the symbol duration. The Nyquist filtering yields a relationship between the symbol duration and the bandwidth, $\Delta f = 2/T$ (this can be seen on Figure 2 where T = 0.5 µs and $\Delta f = 2$ MHz). This gives:

$$R = \frac{log_2(M)\ \Delta f}{2} \quad (3)$$

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

As detailed in the previous question, increasing the constellation size (the number of bits transmitted per symbol M) allows to increase the bit rate. However, the minimum euclidean distance between the symbols in the constellation size decreases, which causes the error probability to increase. As a result, to achieve a similar BER, more energy is required when the constellation size increases but the bit rate also increases.

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

The halfroot Nyquist filter allows to limit the communication bandwidth (it is more spectrally efficient than the plain rectangular window) and allows to cancel ISI.

How do we implement the optimal demodulator? Give the optimization criterion.

The optimal demodulator is implemented as a bank of K filters matched to the basis functions of the modulation scheme:

$$h_i(t) := s_i(-t) \qquad i = 1, ..., K$$

The outputs to such filters are the autocorrelations between the input signal and the basis functions. The filters are mathematically equivalent to a bank of K correlators. They have the property to maximize the SNR at the output, which will be equal to $\frac{2\mathcal{E}}{N_0}$.

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

There exist two possible optimization criteria for the detector:

- The maximum a posteriori (MAP) criterion, which minimizes the error probability by exploiting the knowledge of the a priori probabilities $p(\underline{s}_m)$ and the conditional probabilities $p(\underline{r}|\underline{s}_m)$, where $\underline{r}$ is the observation.

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

- The maximum likelihood criterion is equivalent to the MAP criterion when all the M symbols are equally probable, ie $p(\underline{s}_m) = 1/M$.

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

The ML criterion determines what symbol is the most likely to have produced the received signal by selecting the symbol which has the lowest euclidean distance to the received symbol.

3 Low-density parity check code

3.1 Simulation

The channel encoder that was skipped in section 2 is now implemented as a low-density parity check code (LDPC) encoder. Soft and hard decoding have been implemented with a code rate of 1/2 and their performance is compared next.

The channel coding adds structured redundancy to the information sent by the transmitter to be able to detect and correct errors at the receiver side. LDPC has the property that both the generator and check matrices are sparse, allowing a lower complexity of the implementation.

3.1.1 Hard decoding

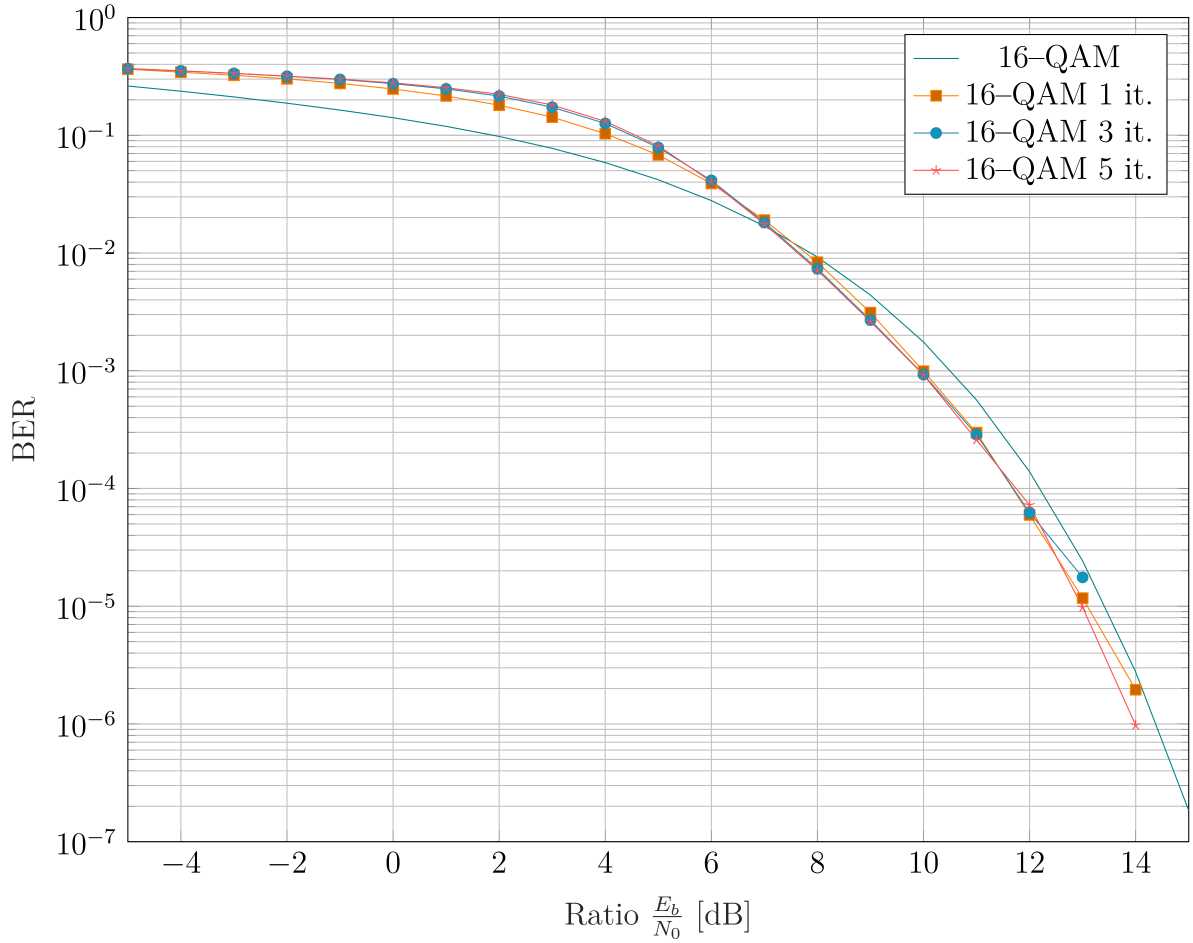

Figure 5: Hard LDPC

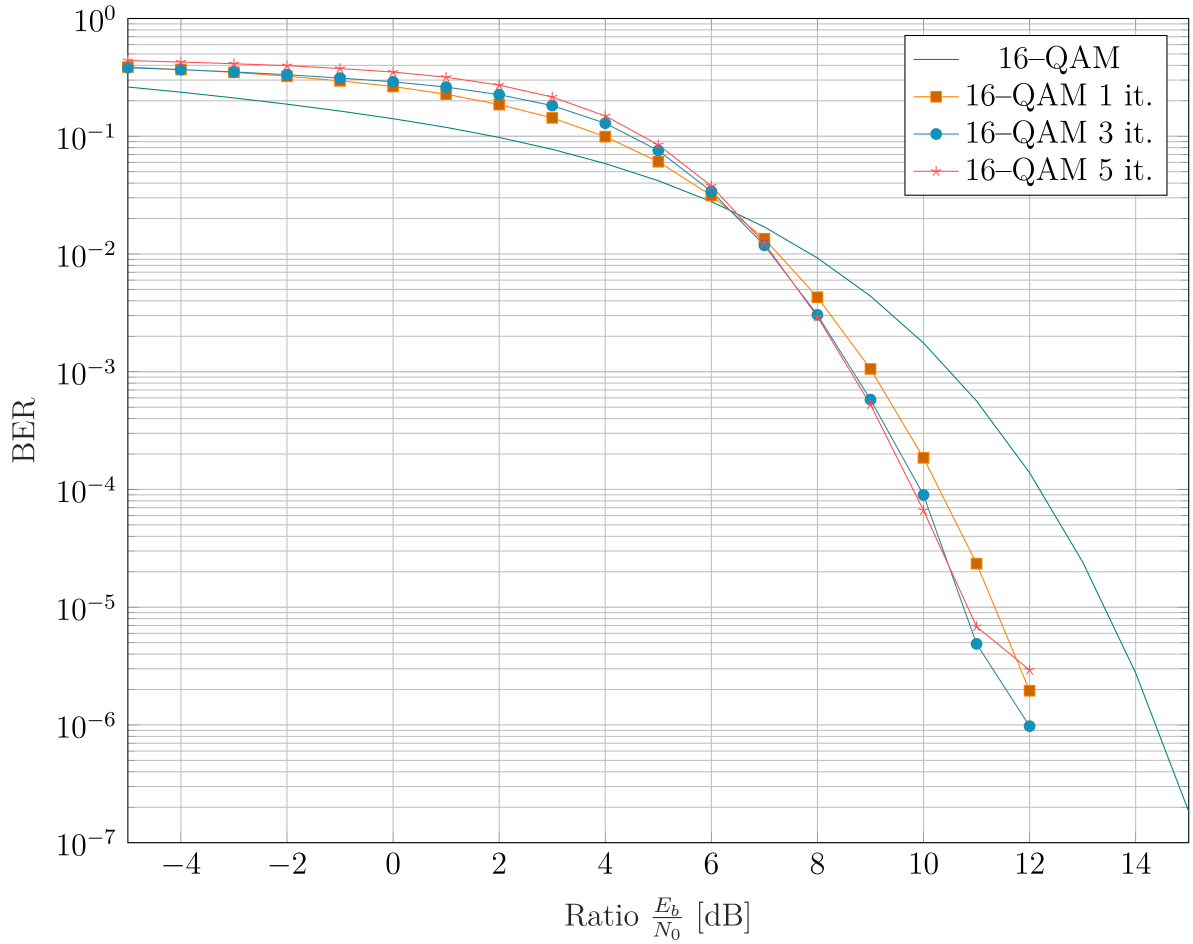

Figure 6: Hard LDPCbis

3.1.2 Soft decoding

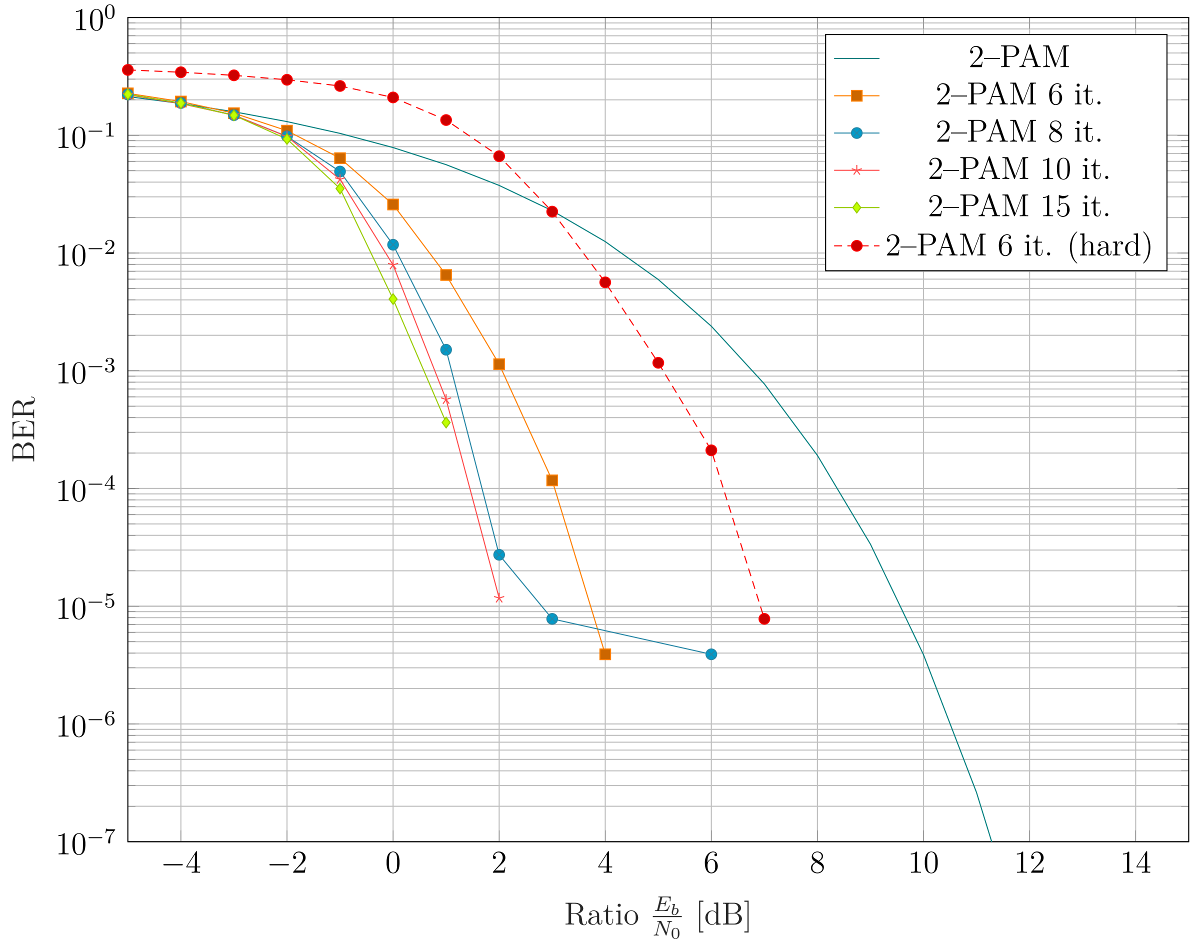

Figure 7: Soft LDPC

3.2 Questions

3.2.1 Questions regarding the simulation

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

How do you limit the number of decoder iterations?

For the hard decoding, the iterations stop either when the maximum number of iterations is reached or when an iteration does not update the coded block anymore.

For the soft decoding,

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

In the case of the 16-QAM or 64-QAM, the euclidean distance should be computed when decoding. This is not the case for BPSK and QPSK, where the decision can be made only by looking at the real and imaginary part of the received symbols.

3.2.2 Questions regarding the communication system

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

A code is systematic when the mapping is such that part of the code vector coincides with the message vector. In that case, the generator matrix has the form $\underline{\underline{G}} = \left[\underline{\underline{P}}|\underline{\underline{I}}\right]$, with $\underline{\underline{P}}$ the parity array and $\underline{\underline{I}}$ the identity matrix of the corresponding size. The parity check matrix should be created so that its rows are orthogonal to the **rows (?)** of the generator matrix:

$$\underline{\underline{G}} \cdot \underline{\underline{H}}^T = \underline{\underline{0}}$$

The solution to this equation is given by $\underline{\underline{H}} = \left[\underline{\underline{I}}|\underline{\underline{P}}^T\right]$. Indeed:

$$\underline{\underline{G}} \cdot \underline{\underline{H}}^T = \left[\underline{\underline{P}}|\underline{\underline{I}}\right] \cdot \left[\underline{\underline{I}}|\underline{\underline{P}}^T\right]^T = \underline{\underline{P}} \oplus \underline{\underline{P}} = 0$$

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 the basis vectors of a subspace which is perpendicular to the subspace spanned by the generator matrix. A linear combination of these basis vectors will still be a basis for the same subspace and the resulting code will be equivalent.

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

A sparse parity check matrix allows to significantly reduce the complexity of the decoder. Indeed, element H_{ij} being equal to 1 results in a logical connection in the Tanner graph between the check-node c_i and the variable-node v_j. Hence, reducing the number of 1's in $\underline{\underline{H}}$ reduces the number of exchanged messages and the number of computations.

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

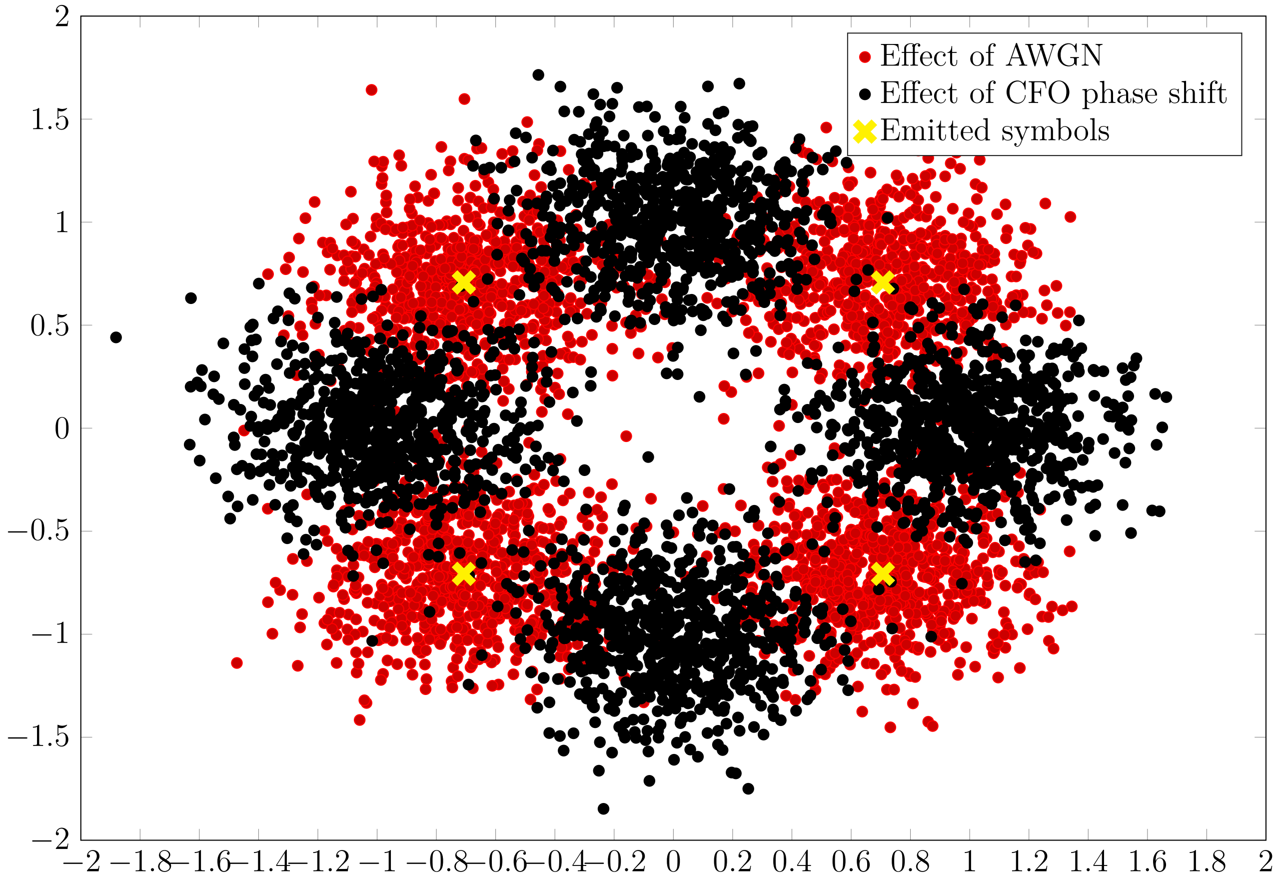

Figure 8: Illustration of the effect of the CFO phase shift on a 4-QAM constellation for $\frac{E_b}{N_0} = 6\text{dB}$ and $\phi_0 = \pi/4$

4 Time and frequency synchronisation

4.1 Simulation

4.2 Questions

4.2.1 Questions regarding the simulation

Derive analytically the baseband model of the channel including the synchronisation errors.

Assuming that the oscillators from the emitter and the receiver differ from each other by a pulsation $\Delta\omega$ and a phase shift ϕ, the signal at the receiver can be written as a function of the complex envelope of the emitted signal $e_s(t)$ (neglecting the noise):

$$r(t) = \Re\{e_s(t)\}\cos(\omega_c t) + \Im\{e_s(t)\}\sin(\omega_c t)$$

The complex envelope must be computed to go to the baseband model.

$$\begin{aligned} e_r(t) &= r(t)\cos\left[(\omega_c + \Delta\omega)t\right] + r(t)\sin\left[(\omega_c + \Delta\omega)t\right] \\ &= \cos\left[(\omega_c + \Delta\omega)t\right] \times \left[\Re\{e_s(t)\}\cos(\omega_c t) + \Im\{e_s(t)\}\sin(\omega_c t)\right] \\ &+ \sin\left[(\omega_c + \Delta\omega)t\right] \times \left[\Im\{e_s(t)\}\cos(\omega_c t) + \Im\{e_s(t)\}\sin(\omega_c t)\right] \end{aligned}$$

Using the Simpsons rules and lowpass filtering out the high frequency components, one can find

$$\begin{aligned} e_r(t) = \Re\{e_s(t)\}\cos(\Delta\omega_c t + \phi) - j\Im\{e_s(t)\}\sin(\Delta\omega_c t + \phi) \\ \Rightarrow e_r(t) = e_s(t)\cdot e^{j\Delta\omega t + \phi} \end{aligned}$$

Hence, the synchronisation errors can be modelled by a multiplication of the baseband model by $e^{j\Delta\omega t + \phi}$.

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

The linearly increasing phase shift is manually corrected after the convolution with the halfroot filter.

How do you simulate the sampling time shift in practice?

How do you select the simulated E_b/N_0 ratio?

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

4.2.2 Questions regarding the communication system

In which order are the synchronisation effects estimated and compensated? Why?

The Gardner's algorithm being robust to CFO, it is first used to correct the sampling time shift. Then the CFO is handled.

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

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)?

Are the frame and frequency acquisition algorithms optimal? If yes, give the optimisation criterion.