Simulation of a DVB-S communication channel

Marie-Laure Boucheret - Nathalie Thomas

2018-2019

1 Introduction

The aim of this project is to simulate the physical layer of a DVB-S standardized system (see figure 1): QPSK mapping, raised cosine shaping and concatenated forward error correcting codes. The transmission channel will be an additive white gaussian noise (AWGN) channel (forward link). An optimal receiver will be implemented.

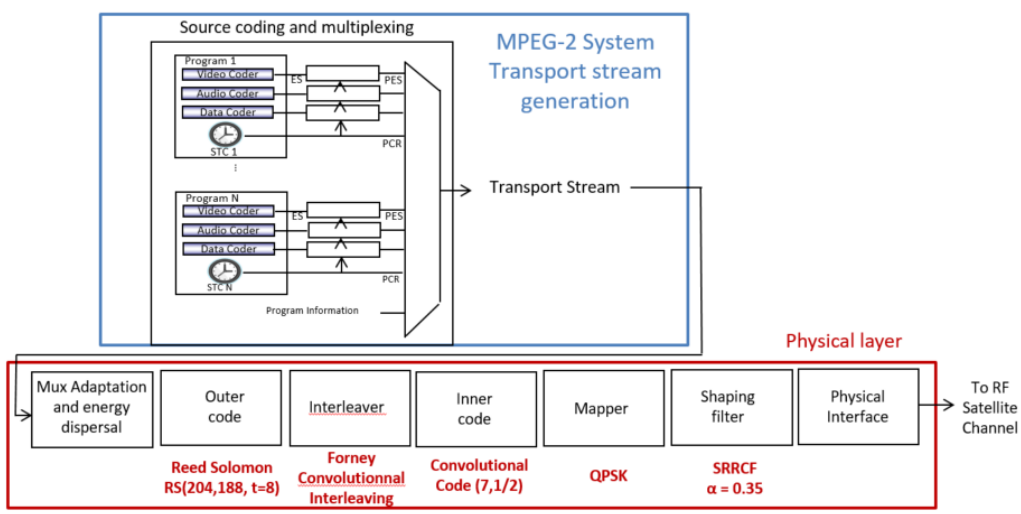

FIGURE 1 – DVB-S physical layer

First of all, we will begin by implementing some baseband transmission channels without channel coding. Then we will define the equivalent low pass (baseband) channel associated to the DVB-S transmission and implement it with channel coding.

2 Baseband channels

2.1 Elements of a baseband channel

The baseband channels to implement will be composed of the following elements:

2.1.1 Generation of the binary information to transmit

You will first have to generate a series of equally likely and independent bits 0 and 1 representing the binary information to be transmitted. This binary information can be generated using the *randi.m* Matlab function.

2.1.2 Baseband modulation

The baseband modulator includes 3 elements:

— The Mapping, to obtain symbols a_k from the bits to be transmitted.

— The oversampling process, to generate a series of Delta functions spaced from T_s and weighted by the symbols a_k, T_s being the symbol duration. The symbol period T_s will be composed of N_s samples spaced from T_e: $T_s = N_s T_e$, T_e denoting the sampling period. The oversampling factor N_s (number of generated samples per symbol) has to be defined so as to generate a digital signal fulfilling Shannon condition.

— The shaping filter, in order to obtain the baseband signal. We will only consider Finite Impulse Response Filters in this laboratory. You can use Matlab *filter.m* function to implement the filtering.

2.1.3 AWGN Transmission Channel

An AWGN transmission channel will be considered. It consists in adding a white Gaussian noise to the modulated signal. This noise will have to be generated using the *randn.m* Matlab function. Several noise powers will have to be tested in order to plot the obtained BER as a function of the signal to noise ratio per bit, $\frac{E_b}{N_0}$, at the receiver input. The noise power, σ_n^2, is given below as a function of the desired $\frac{E_b}{N_0}$ (see in the annex for the demonstration):

$$\sigma_n^2 = \frac{\sigma_a^2 \sum_{k=1}^{N} |h(k)|^2}{2log_2(M)\frac{E_b}{N_0}}$$

where σ_a^2 represents the symbols variance, $h(k)$ the shaping filter digital impulse response of order N and M the modulation order. A noise of power σ_n^2 can be generated on Matlab using σ_n*randn(1, *noise length*). The noise length has to be the same as the signal length.

2.1.4 Baseband demodulation and BER computation

The baseband demodulator includes the following elements:

- A receiver filter, chosen in order to be able to respect Nyquist criterion and matched filtering.
- A sampling process at times $t_0 + mT_s$. An eyediagram will have to be plotted in order to choose the optimal time sampling instants.
- A decision block, here a threshold detector, to decide for the received symbols.
- The demapping, in order to compare the received bits with the transmitted ones and compute the transmission bit error rate (BER).

2.2 Baseband channels to implement

You will have to implement two baseband channels : one to transmit a NRZ signal, the over one to transmit a square root raised cosine (SRRC) signal.

2.2.1 2-NRZ signal

Binary symbols $a_k \in \{\pm 1\}$ can be considered and a rectangular shaping filter will have to be used. You can take $N_s = 5$ samples per symbol. Observe the transmitted signal, as well as its power spectral density. Matlab *pwelch.m* function can be used to plot the power spectral density, or you can implement it using what you did in the signal processing lab.

An optimal receiver will have to be implemented. First of all, check wether the BER is zero without noise. Then, plot the obtained BER as a function of E_b/N_0 in decibels[1]. $(E_b/N_0)_{dB}$ tested values will be taken between 0 and 6

1. be careful the BERs will have to be plotted in log scale and you will have to make accurate measurements (see in appendix)

dB. You will have to compare, on the same figure, the simulated BER with the theoretical one to validate your simulated transmission channel.

The choices you made at the receiver side will have to be explained in your report. Obtained plots for the BERs (simulated and theoretical) will have to be included in your report, as well as the theoretical BER expression.

2.2.2 SRRC signal

Binary symbols $a_k \in \{\pm 1\}$ can be considered and a square root raised cosine shaping filter of roll off 0.35 will have to be used. Its impulse response $h(n)$ can be obtained using matlab *rcosfir()* or matlab *rcosdesign()* functions. What is the minimum necessary value for N_s in that case ? You have to justify your answer. Observe the transmitted signal, as well as its power spectral density.

An optimal receiver will have to be implemented. Like previously, check wether the BER is zero without noise. Be careful here with the filters delays (see in appendix) ! ! Then, plot the obtained BER as a function of E_b/N_0 in decibels. $(E_b/N_0)_{dB}$ tested values will be taken between 0 and 6 dB. You will have to compare, on the same figure, the simulated BER with the theoretical one to validate your simulated transmission channel.

The choices you made at the receiver side will have to be explained in your report. Obtained plots for the BERs (simulated and theoretical) will have to be included in your report, as well as the theoretical BER expression.

3 DVB-S transmission

3.1 DVB-S equivalent low pass channel: modulation

The DVB-S transmission transmits a carrier modulated signal. In order to reduce the processing time, the simulation will be done using the equivalent low pass (baseband) channel. Figure 2 represents the carrier modulated transmission channel, while figure 3 represents the equivalent low pass channel to be implemented.

$$x_e(t) = I(t) + jQ(t) = \sum_k d_k h(t - kT_s)$$

represents the complex envelop (complex baseband signal) associated to $x(t)$, the signal transmitted on f_p:

$$x(t) = Re\left[x_e(t)e^{j2\pi f_p t}\right]$$

Complex symbols $d_k = a_k + jb_k$ must be in accordance with the DVB-S standard constellation (see figure 4). We also define a complex envelop, $n_e(t)$, associated to the noise $n(t)$ introduced by the channel : $n_e(t) = n_I(t) + jn_Q(t)$ with $S_{n_I}(f) = S_{n_Q}(f) = N_0$ if $S_n(f) = \frac{N_0}{2} \forall f$.

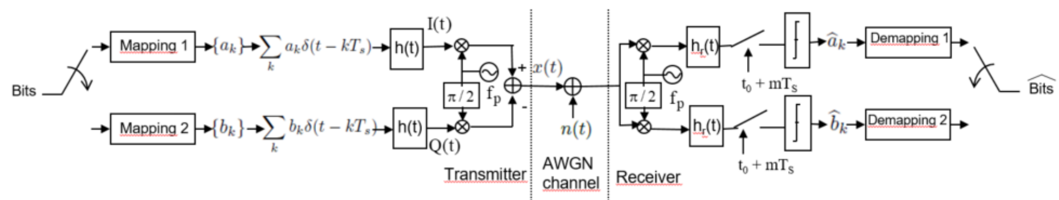

FIGURE 2 – QPSK transmission on a carrier frequency

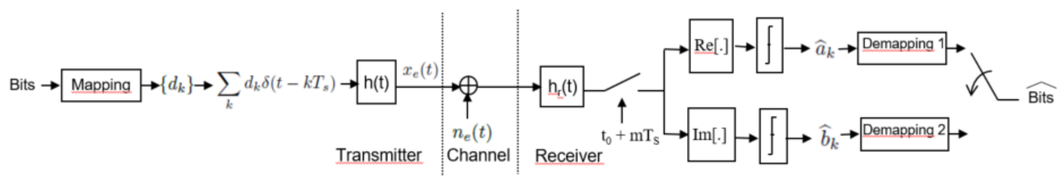

FIGURE 3 – QPSK low pass equivalent channel

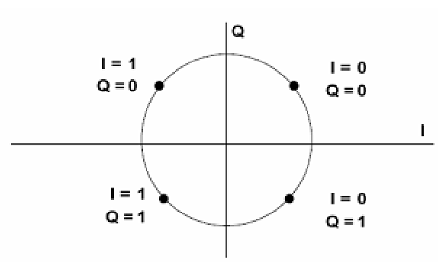

FIGURE 4 – DVB-S QPSK constellation

The DVB-S shaping filter is a squared root raised cosine filter with a roll off factor equal to 0.35. An optimal receiver will have to be implemented. Check wether the BER is zero without noise. Then, plot the obtained BER as a function of E_b/N_0 in decibels and compare it to the theoretical one. $(E_b/N_0)_{dB}$ tested values will be taken between 0 and 6 dB. To simulate the channel for a given value of E_b/N_0, the noise variance to be applied on the In-phase, $n_I(t)$, and the Quadrature, $n_Q(t)$, components of $n_e(t) = n_I(t) + jn_Q(t)$ is given by (see appendix) :

$$\sigma_{n_I}^2 = \sigma_{n_Q}^2 = \frac{\sum_{k=1}^{N} |h(k)|^2 \sigma_d^2}{2log_2(M)\frac{E_b}{N_0}}, \quad (1)$$

where σ_d^2 denotes the variance of symbols d_k, $h(k)$ the shaping filter digital impulse response of order N and M the modulation order.

The choices you made at the receiver side will have to be explained in your report. Obtained plots for the BERs (simulated and theoretical) will have to be included in your report, as well as the theoretical BER expression.

3.2 DVB-S equivalent low pass channel with Forward Error Correction

DVB-S standard defines a coding scheme composed of two concatenated codes: a RS(204,188) followed by a convolutional code (7,1/2) which can be punctured to obtain several coding rates (1/2, 2/3, 3/4, 5/6 et 7/8). To improve the performance, an interleaver is introduced between the Reed Solomon and the convolutional code.

3.2.1 Convolutional code

Using *poly2trellis()* and *convenc()* Matlab functions, first add to the previous channel a convolutional code (7, 1/2), with generator polynomials $g_1 = 171_{oct}$ and $g_2 = 133_{oct}$ (defined in the DVB-S standard, see figure 5) [2].

Decoding will be done using Viterbi algorithm (use *vitdec()* matlab function). Two modes are possible : hard decision (decisions are made before using Viterbi algorithm) and soft decision (decisions are made in the decoding process). For E_b/N_0 varying from -4 to 6 dB, compare the obtained BERs, with and without convolutional coding, considering hard and soft decisions.

Use then the puncturing matrix $P = [1101]$ in order to obtain a coding rate of 2/3. In order to use the same decoder whatever is the coding rate, punctured

2. To understand the behavior of the Matlab functions, you can begin by implementing a simpler code, such as a (3, 1/2) with generator polynomials $g_1 = 5_{oct}$ et $g_2 = 7_{oct}$

Table 2: Punctured code definition

Original code			Code rates									
			1/2		2/3		3/4		5/6		7/8	
K	G_1 (X)	G_2 (Y)	P	dfree	P	dfree	P	dfree	P	dfree	P	dfree
7	171_{OCT}	133_{OCT}	X: 1 Y: 1 $I=X_1$ $Q=Y_1$	10	X: 1 0 Y: 1 1 $I=X_1\ Y_2\ Y_3$ $Q=Y_1\ X_3\ Y_4$	6	X: 1 0 1 Y: 1 1 0 $I=X_1\ Y_2$ $Q=Y_1\ X_3$	5	X: 1 0 1 0 1 Y: 1 1 0 1 0 $I=X_1\ Y_2\ Y_4$ $Q=Y_1\ X_3\ X_5$	4	X: 1 0 0 0 1 0 1 Y: 1 1 1 1 0 1 0 $I=X_1\ Y_2\ Y_4\ Y_6$ $Q=Y_1\ Y_3\ X_5\ X_7$	3

NOTE: 1 = transmitted bit
0 = non transmitted bit

FIGURE 5 – DVB-S convolutional code.

symbols will be reintroduced as zeros before using the soft decision Viterbi decoding algorithm. It is possible to give the puncturing matrix as a parameter in the Matlab *convenc()* and *vitdec()* functions.

For E_b/N_0 varying from -4 to 6 dB, compare obtained BERs with and without puncturing.

3.2.2 Reed Solomon

To add the RS(204,188) to the previous channel, use first *comm.RSEncoder.m* and then *step.m*. The decoding will be done using *comm.RSDecoder*. Compare obtained BERs, with only the convolutional code and with the concatenated scheme.

3.2.3 Interleaving

To improve the performance of the coding process, DVB-S standard defines an interleaver to be inserted between the Reed Solomon and the convolutional code. This interleaver is a convolutionnal interleaver : see figure 6.

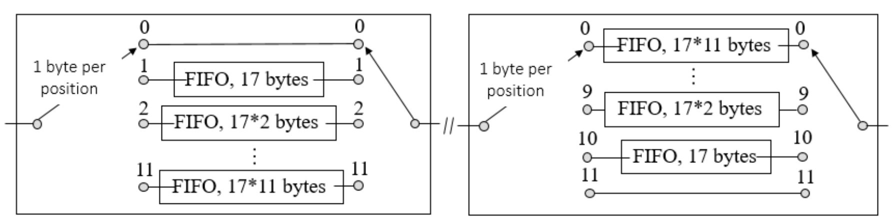

FIGURE 6 – DVB-S interleaver.

To implement it, you can use *convintrlv.m* matlab function (*convdeintrlv.m* for desinterleaving). Compare obtained BERs, with and without interleaving.

4 Appendix

4.1 Noise power to be introduced in the simulations

4.1.1 Baseband transmission channels (PAM modulations)

The noise is introduced in the frequency band F_e, with a PSD $N_0/2$. The noise variance is then given by :

$$\sigma_n^2 = \frac{N_0}{2} F_e = \frac{E_s}{2\frac{E_s}{N_0}} F_e = \frac{P_r T_s}{2\frac{E_s}{N_0}} F_e,$$

where E_s represents the symbol energy at the receiver input, T_s the symbol duration and P_r the power of the received signal.
For an AWGN propagation channel and equally likely, independent and zero-mean symbols, P_r is given by :

$$P_r = \frac{\sigma_a^2}{T_s} \int_R |H(f)|^2 \, df = \frac{\sigma_a^2}{T_s} \int_R |h(t)|^2 \rightarrow \frac{\sigma_a^2}{T_s} \sum_{k=1}^{N} T_e \left| h(kT_e) \right|^2,$$

where σ_a^2 is the symbol variance, T_e the sampling period and $h(kT_e)$ the k^{th} coefficient of the shaping filter impulse response whose order is N.
It comes :

$$\sigma_n^2 = \frac{\sigma_a^2 \sum_{k=1}^{N} |h(k)|^2}{2 \log_2(M) \frac{E_b}{N_0}},$$

M being the modulation order.

4.1.2 Carrier modulated transmission channel

The noise is introduced in the frequency band F_e, with a PSD $N_0/2$. The noise variance is then given by :

$$\sigma_n^2 = \frac{N_0}{2} F_e = \frac{E_s}{2\frac{E_s}{N_0}} F_e = \frac{P_r T_s}{2\frac{E_s}{N_0}} F_e,$$

where E_s represents the symbol energy at the receiver input, T_s the symbol duration and P_r the power of the received signal.
For an AWGN propagation channel and equally likely, independent and zero-

mean symbols, P_r is given by :

$$P_r = \frac{P_{r_e}}{2} = \frac{\sigma_d^2}{2T_s}\int_R |H(f)|^2 \, df = \frac{\sigma_d^2}{2T_s}\int_R |h(t)|^2 \rightarrow \frac{\sigma_d^2}{2T_s}\sum_{k=1}^{N} T_e \, |h(kT_e)|^2 \, ,$$

P_{r_e} representing the power of the complex envelop associated to the received signal, T_e the sampling period and $h(kT_e)$ the k^{th} coefficient of the shaping filter impulse response whose order is N.

This leads to :

$$\sigma_n^2 = \frac{\sigma_d^2 \sum_{k=1}^{N} |h(k)|^2}{4\log_2(M)\frac{E_b}{N_0}},$$

M being the modulation order.

4.1.3 Low pass equivalent transmission channel

The noise, added to the complex envelop $x_e(t)$ associated to the transmitted signal $x(t)$, is complex : $n_e(t) = n_I(t) + jn_Q(t)$ (see figure 3). Its power spectral density $S_{n_e}(f) = 2N_0$ in F_e bandwidth, giving N_0 for $S_{n_I}(f)$ and N_0 for $S_{n_Q}(f)$. The variances for $n_I(t)$ and $n_Q(t)$ are given by :

$$\sigma_{n_I}^2 = \sigma_{n_Q}^2 = N_0 F_e = \frac{E_s}{\frac{E_s}{N_0}} F_e = \frac{P_r T_s}{\frac{E_s}{N_0}} F_e,$$

where E_s represents the symbol energy at the receiver input, T_s the symbol duration and P_r the power of the received signal.

For an AWGN propagation channel and equally likely, independent and zero-mean symbols, P_r is given by :

$$P_r = \frac{P_{r_e}}{2} = \frac{\sigma_d^2}{2T_s}\int_R |H(f)|^2 \, df = \frac{\sigma_d^2}{2T_s}\int_R |h(t)|^2 \rightarrow \frac{\sigma_d^2}{2T_s}\sum_{k=1}^{N} T_e \, |h(kT_e)|^2 \, ,$$

where P_{r_e} represents the power of the complex envelop associated to the received signal, T_e the sampling period and $h(kT_e)$ the k^{th} coefficient of the shaping filter impulse response whose order is N.

This leads to :

$$\sigma_{n_I}^2 = \sigma_{n_Q}^2 = \frac{\sigma_d^2 \sum_{k=1}^{N} |h(k)|^2}{2\log_2(M)\frac{E_b}{N_0}},$$

M being the modulation order.

4.1.4 Precision on the BER measurements

The BER can be modeled as a sum of random variables X_k taking their values in the set $\{0,1\}$ with probabilities $P[X_k = 0] = 1 - p$ (no error) and $P[X_k = 1] = p$ (error) :

$$BER = \frac{1}{N}\sum_{k=1}^{N} X_k.$$

The relative square error on the BER is given by :

$$\epsilon^2 = \frac{\sigma_{BER}^2}{m_{BER}^2},$$

where m_{BER} and σ_{BER}^2 respectively represent the average and the variance on the BER estimation.
The precision on the BER measurement will be given by ϵ. We can write :

$$m_{BER} = \frac{1}{N}\sum_{k=1}^{N} E[X_k] = \frac{1}{N} N(1 \times p + 0 \times (1-p)) = p$$

and

$$\sigma_{BER}^2 = E\left[\left(\frac{1}{N}\sum_{k=1}^{N} X_k\right)^2\right] - p^2 = \frac{1}{N^2}\sum_{k=1}^{N}\sum_{i=1}^{N} E[X_k X_i] - p^2$$

— if $k = i$ (N cases) then $E\left[X_k^2\right] = 1^2 \times p + 0^2 \times (1-p) = p$
— if $k \neq i$ ($N^2 - N$ cases) then $E[X_k X_i] = E[X_k]\,E[X_i] = p^2$

So :

$$\sigma_{BER}^2 = \frac{1}{N^2}\left\{Np + \left(N^2 - N\right)p^2\right\} - p^2 = \frac{p(1-p)}{N}$$

Error variance tends to 0 when N increases and the relative square error on the BER can be written as :

$$\epsilon^2 = \frac{\sigma_{BER}^2}{m_{BER}^2} = \frac{1-p}{Np} \simeq \frac{1}{Np} \; for\; p << 1$$

leading to :

— the number of binary elements, N, to generate in order to obtain a precision ϵ on the BER measurement, when the value is *a priori* known. For example, if we want to measure a BER of 10^{-2} with a precision of 10%, we need to generate $N = \frac{1}{10^{-2} \times (10^{-1})^2} = 10^4$ bits.
— the number of simulations to run if the value of the BER to measure is

not *a priori* known. We will have to run simulations till we observe $1/\epsilon^2$ errors in order to obtain a measure with a given precision ϵ. For example, if we want to measure the BER with a precision $\epsilon = 10\%$, we will have to count the errors till we obtain $1/\epsilon^2 = 10^2$ errors before considering the BER measurement as sufficiently accurate.

4.2 About the delays due to the filters in the transmission channel

4.2.1 NRZ channel

Let's take, first, the example of a transmission channel with rectangular filters of T_s duration ($T_s = (2N+1)T_e$) at the transmitter and at the receiver and an AWGN transmission channel. Figure 7 plots the filters impulse responses for $2N+1 = 5$ coefficients, as well as their convolution product, representing here the whole transmission channel impulse response. From these plots, it is possible to deduct that the Nyquist criterion can be met if we sample at t_0+mT_s, with $t_0 = 0$. However figure 7 plots impulse responses representing non causal filters. In order to obtain causal filters, able to be implemented, we have to shift the impulse responses $h(t)$ and $h_r(t)$ of NT_e. The whole transmission channel impulse response $(h(t)*h_r(t))$ then indicates that we have to sample at t_0+mT_s, with $t_0 = 2NT_e = 4T_e$: see figure 8. Indeed, the need to shift $h(t)$ and $h_r(t)$ introduces a delay equal to $2NT_e = 4T_e$: see figure 9. The response to the first point (vector before the first filter) is the $(2N+1)^{th}$ point after the two filters. Figure 10 shows a part of the obtained digital signal after the two filters to observe the absence of interference at time instants t_0+mT_s, with $t_0 = 2NT_e = 4T_e$.

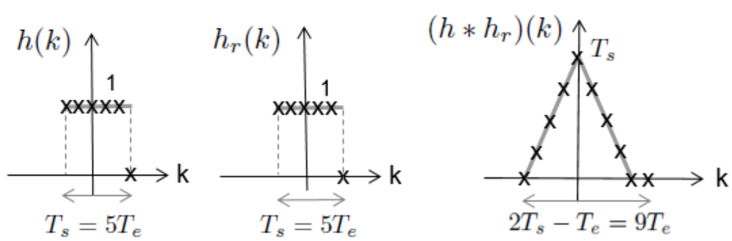

FIGURE 7 – NRZ transmision channel : transmitter and receiver filters impulse responses, whole transmission channel impulse response - Non causal filters

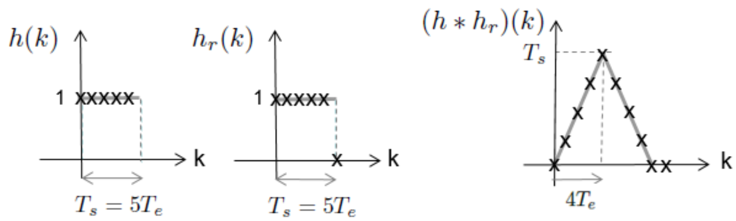

FIGURE 8 – NRZ transmision channel : transmitter and receiver filters impulse responses, whole transmission channel impulse response - Causal filters

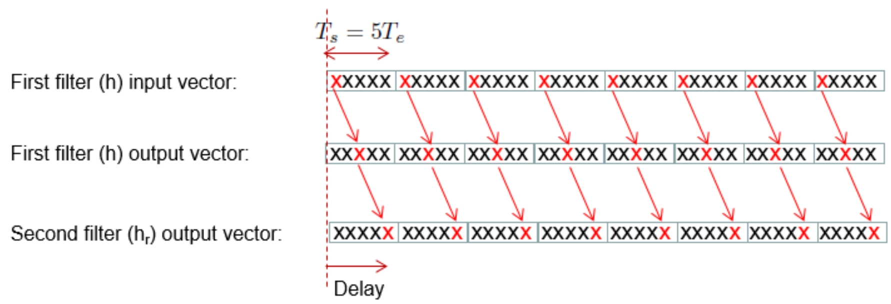

FIGURE 9 – NRZ transmission channel : illustration of the delay due to filters causality.

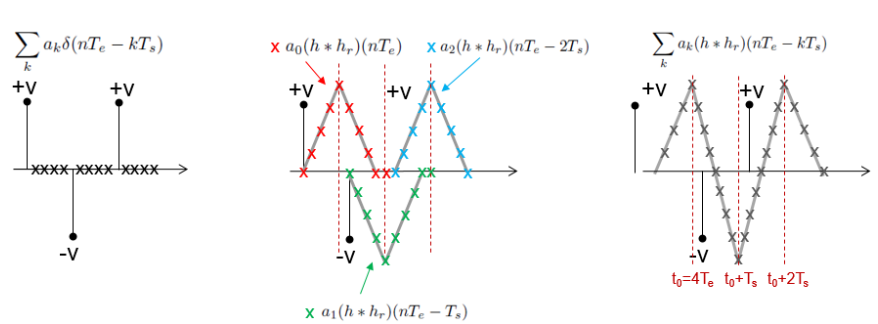

FIGURE 10 – NRZ transmision channel : part of the obtained digital signal after the two causal filters.

4.2.2 RCF channel

Let's take now, the example of a transmission channel with square root raised cosine filters at the transmitter and at the receiver (same roll off) and an AWGN transmission channel. Figure 11 plots the filters impulse responses, as well as their convolution product, representing here the whole transmission channel impulse response. From these plots, it is possible to deduct that the Nyquist criterion can be met if we sample at $t_0 + mT_s$, with $t_0 = 0$. However figure 11 plots impulse responses representing non causal filters. In order to obtain causal filters, able to be implemented, we have to shift the impulse response of $2T_s$, the whole transmission channel impulse response $(h(t) * h_r(t))$ then indicates that we have to sample at $t_0 + mT_s$, with $t_0 = 4T_s + T_e$: see figure 8. Indeed, the need to shift $h(t)$ and $h_r(t)$ introduces a delay equal to $2 \times 2T_s$, or $4T_s$: see figure 13. Figure 13 also shows that, in that case (introduced delay $> T_s$) a part of the information will be lost. However, it is possible to avoid this using some zero padding before filtering : see figure 14.

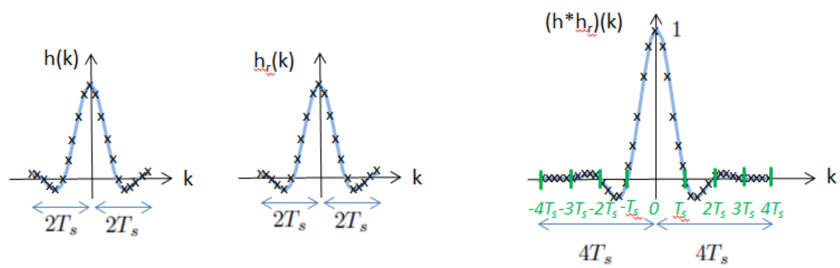

FIGURE 11 – Raised cosine transmission channel : transmitter and receiver filters impulse responses, whole transmission channel impulse response - Non causal filters

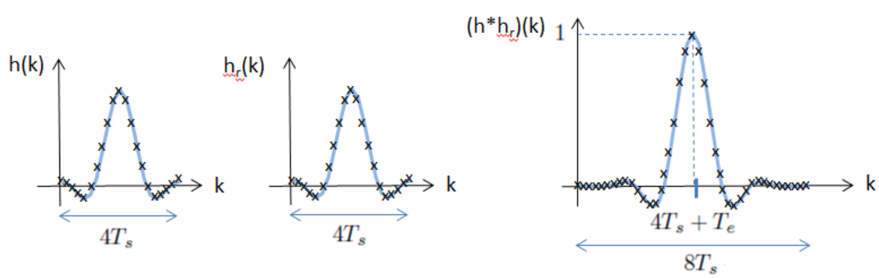

FIGURE 12 – Raised cosine transmission channel : transmitter and receiver filters impulse responses, whole transmission channel impulse response - Causal filters

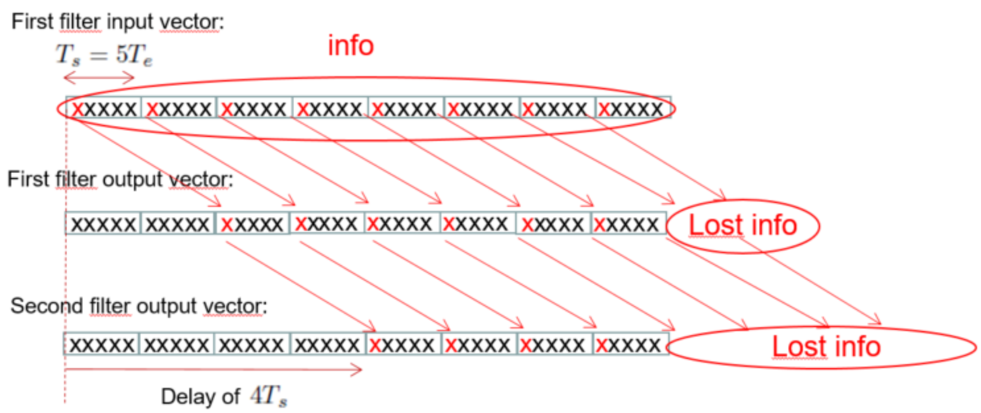

FIGURE 13 – Raised cosine transmission channel : illustration of the delay due to filters causality and of the resulting information loss.

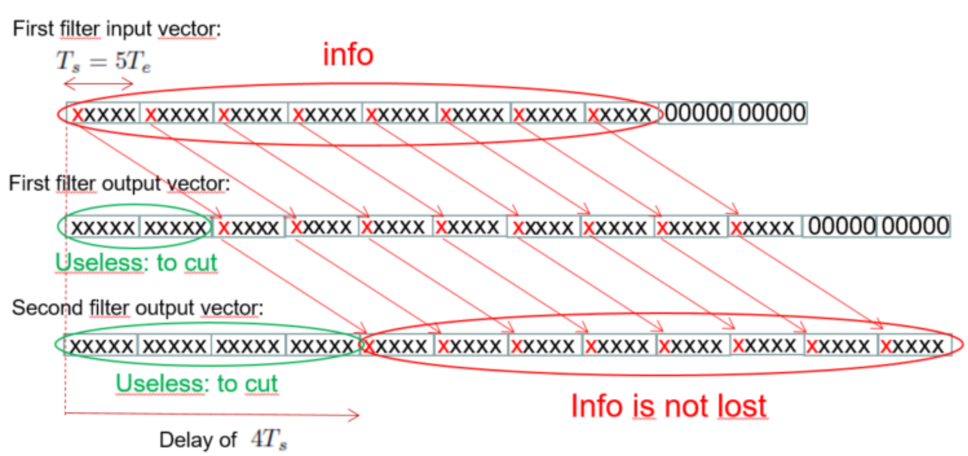

FIGURE 14 – Raised cosine transmission channel : illustration of the delay due to filters causality and of the way to avoid an information loss.

5 Material for the evaluation

Codes and reports will have to be sent to nathalie.thomas@enseeiht.fr.

5.1 Report

The report will have to include a table of contents, an introduction presenting the objectives, a conclusion summarizing the main results and used references.

Figures will have to be readable, with titles, labels on the axis and legends if several curves are plotted on the same figure.

Equations will have to be included using an equation editor.

Be careful : All the explanations/justifications will have to use arguments coming from the course or other documents (books, internet) appearing in your references. Be precise and clear in your explanations/justifications, with the good technical vocabulary.

5.2 Codes

Be careful :
BERs will have to be plotted in log-scale and measurements have to be meaningful (see in appendix).
Avoid, as far as possible, the loops with Matlab. If you can't think to initialize your variables.

Codes will have to contain clear and enough comments :
"enough" : at least one by action in the transmission channel (for example : bits generation, mapping ...). Each action can take several lines and comments can/must, of course, be added to explain what is done.
"clear" : a good vocabulary to introduce the classical elements of the transmission channel (for example : mapping instead of +1/-1 generation).

The names of the provided files (.m) have to be significative.