DVB-S2 Satellite Communication – MATLAB Simulation

Modulation and Coding Project

April 2016 Asfour A. Omar, Liu Yu, Mroueh Michael

# Table of Contents

[Table of Contents 1](#_Toc447399195)

[Introduction 2](#_Toc447399196)

[The Digital Video Broadcasting-Satellite 2](#_Toc447399197)

[Purpose of the project 2](#_Toc447399198)

[Part 1 – Optimal communication chain over the ideal channel 3](#_Toc447399199)

[Hypothesis and assumptions 3](#_Toc447399200)

[Implementation 3](#_Toc447399201)

[Question regarding the simulation 7](#_Toc447399202)

[Questions regarding the communication system 7](#_Toc447399203)

[Part 2 – Time and frequency synchronisation 9](#_Toc447399204)

[Hypothesis and assumptions 9](#_Toc447399205)

[Implementation 9](#_Toc447399206)

[Question regarding the simulation 9](#_Toc447399207)

[Questions regarding the communication system 9](#_Toc447399208)

[Part 3 – Low-density parity check (LDPC) 10](#_Toc447399209)

[Hypothesis and assumptions 10](#_Toc447399210)

[Implementation 10](#_Toc447399211)

[Question regarding the simulation 10](#_Toc447399212)

[Questions regarding the communication system 10](#_Toc447399213)

[Conclusion 11](#_Toc447399214)

[TODO 12](#_Toc447399215)

# Introduction

## The Digital Video Broadcasting-Satellite

The Digital Video Broadcasting-Satellite (**DVB-S**) is a well-spread set of norms defining the communication for the broadcasting of digital television. This standard, used in large number of countries, is based on Quaternary Phase Shift Keying (**QPSK**) modulation and convolutional Forward Error Correction (**FEC**). A second generation of DVB standard, namely the **DVB-S2**, enhances various points its predecessor as well as bringing new features, such as the support of Internet access.

The two main features added compared to its predecessor are:

* The best performance: high order modulation formats combined with Low-Density Parity Check (**LDPC**) codes allowing to make the probability of lost information as small as desired;
* The flexibility: Adaptive Coding and Modulation (**ACM**) is applied, which, as its name suggests, involves adapting correcting codes and modulation used depending on the signal quality and thus lead to an optimizing bandwidth utilization.

It is commonly recognized that thanks to these improvements, the DVB-S2 provides an increase of approximately 30% performance compared to its predecessor for the same bandwidth and emitted signal power.

## Purpose of the project

The objective of this project is to design and simulate the DVB-S2 communication chain. The communication channel will be modelled as an ideal channel only corrupted by Additive White Gaussian Noise (**AWGN**). The project will be implemented step by step by aiming for successive communication chain models increasingly comprehensive and realistic. Thereby this division of work allows to divide the design and simulation of the DVB-S2 communication channel into three important parts:

* The simulation of the **optimal communication chain over the ideal channel;**
* The simulation of the time/frequency **synchronisation** algorithms;
* The simulation of the **LDPC** channel encoder and decoder.

# Part 1 – Optimal communication chain over the ideal channel

## Description of the communication system

The first objective of the project is then to simulate the optimal communication chain over an ideal channel, which is represented in a block-diagram below (Figure 1). Before going into the details of the simulation, a first step would be the description of the main stages, their operation as well as their usefulness.

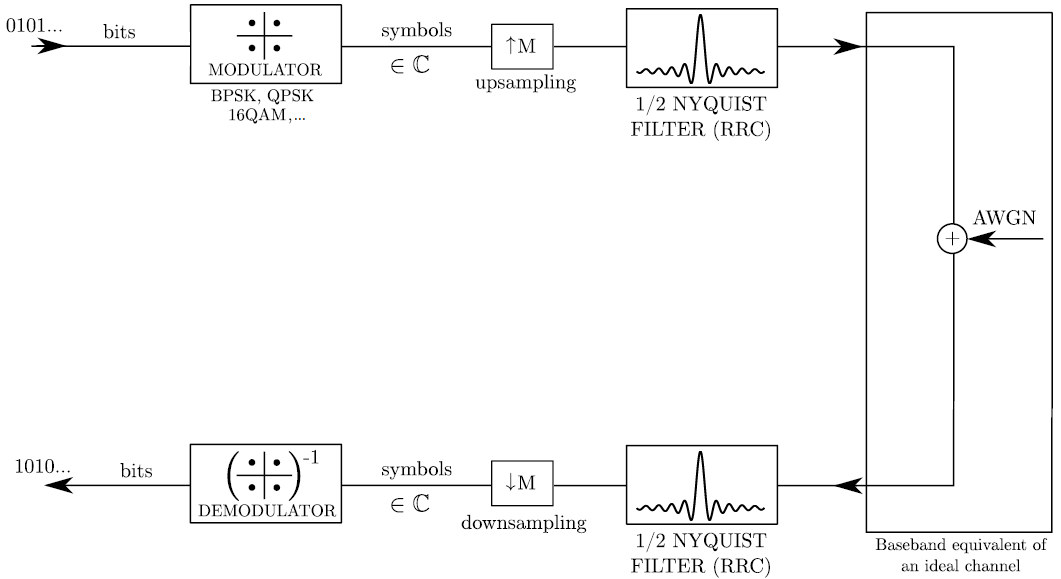


Figure 1 - Block diagram of the communication system

**1st Step – Generating a random stream of bits**

* The first step is naturally the creation of the data that is intended to be transmitted to the receiver. This data is generated in a random way, using a function directly provided by MATLAB.
* **tx\_bin** = randi([0 1],tx\_len,1); % Generate a random stream of bits

**2° Mapping**

* The stream (**tx\_bin**) is transformed into a sequence of complex symbols (**tx\_symb**) in order to improve the spectral efficiency of the system.
* In our simulation, the supported mappings are PAM and QAM
* In both PAM and QAM, there is an important trade-off to be made between the BER and the energy of the signal
  + If the distance between the symbols is increased and/or if the energy of the pulse is increased: the BER will be improved, at the cost of the signal energy which will increase
* [**tx\_symb**] = mapping(tx\_bin,Modu.Nbps,Modu.mod); % Apply the mapping
* An example of 16QAM (4bits) of our MATLAB simulation is depicted on the Figure 2 below.

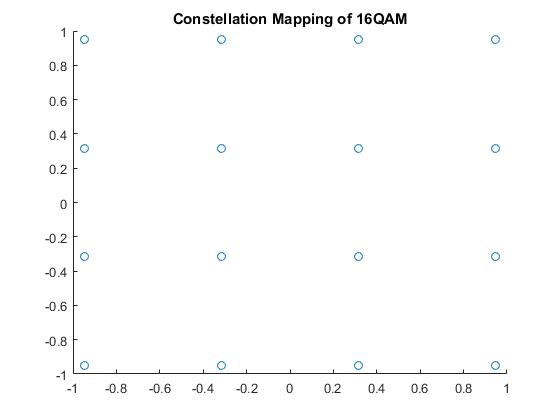


Figure 2 - Constellation Mapping of 16QAM

**3° Upsampling**

* The upsampling (???) consists of the emulation of a higher rate sampling by inserting M-1 zeros between input samples. The embedded function **upsample** take as inputs the points of the original sequence of symbols (**tx\_symb**), and produces a sequence (**tx\_usymb**) similar to the one we would have obtained by sampling the signal at a higher rate. This upsampling step is needed in order to satisfy the ISI Nyquist Criterion with the RRC filters (see below).
* An important distinction must be made between
  + The number of bits in each symbol (which is fixed by the **modulation**)
  + The number of symbols in each data packet (fixed by the **upsampling**)
* **tx\_usymb** = upsample(tx\_symb,Modu.M); % Upsample by M = 4

**4° ½ Nyquist Filter**

* When symbols are transmitted over a channel, there’s a risk that they would be spread in the time domain. This aliasing could cause **Inter-Symbol Interference** (ISI) between consecutive symbols: the ith symbol could have some repercussion on the (i+1)th one, which would make the BER worse.
* By using a **Root-Raised-Cosine filter** (RRC) in both the transmitter and the receiver, we manage to perform some matched filtering, which then allows to reduce the ISI. In order to avoid any ISI, the **Nyquist ISI criterion** must be verified and RRC filters allow to easily satisfy this criterion.
* To be realized, the filters must be implemented in both the transmitter and the receiver, which then form a **Nyquist Filter** which reduces to a Dirac Pulse when it is sampled at the symbol rate from its maximum.
* **h1** = rcosdesign(RRCF.beta,span,sps,'sqrt'); % Design the filter
* **tx\_af** = conv(tx\_usymb,h1); % Apply the filter
* tx\_af\_trunc = tx\_af((RRCF.Ntaps+1)/2:length(tx\_af)-(RRCF.Ntaps+1)/2+1); % Truncates the useless part due to the convolution
* tx\_af\_R\_trunc = real(tx\_af\_trunc); % Takes the real part of it
* The Figure 3 below shows two Root Raised Cosine Filters, respectively obtained through the provided MATLAB Toolbox and a SelfDefine one.

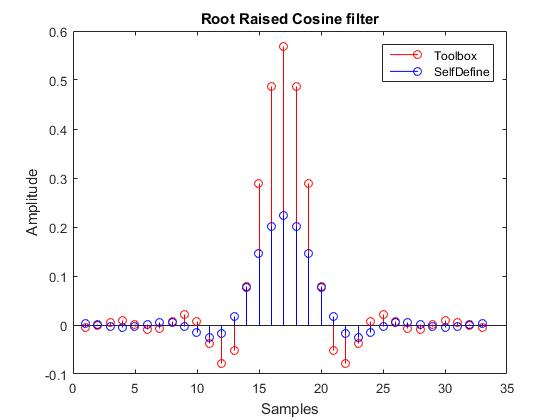


Figure 3 - Root Raised Cosine Filter

**5° Propagation in the Ideal Channel and adding of AWGN**

* Even by considering and Ideal Channel, a first attempt to model slightly realistic device is to implement some Additive White Gaussian Noise (AWGN) that will corrupt the transmitted signal.
* noise = sqrt(NoisePower/2)\*(randn(1,length(noise)) +1i\*randn(1,length(noise))); % Create the noise
* **rx\_usymb** = tx\_af\_trunc+noise'; % Add the noise to the signal
* Baseband equivalent !!!

**6° ½ Nyquist Filter**

* The utility and operation of this complementary filter have already been explained above.

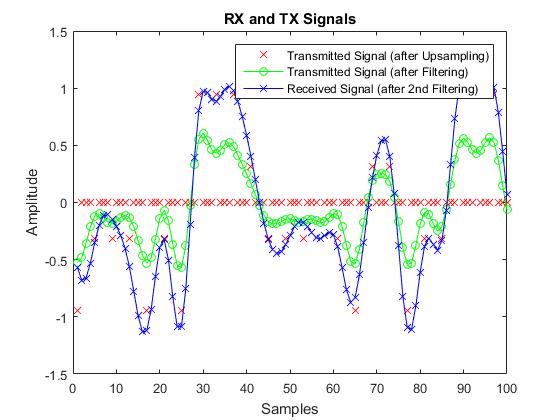


Figure 4 - RX and TX Signals

* The Figure 4 above allows to highlight the effect of the RRC Filters on both the transmitted and received signals: as expected, this method allows the RX sequence to map quite accurately the TX one, and then to avoid almost any ISI. The slight differences are due to the AWGN present in the ideal propagation channel.

**7° Downsampling**

* In contrast to the upsampling step at the transmitter, the Downsampling will allow to get rid of the irrelevant zeros-samples in order to only keep the useful information.
* **rx\_symb** = downsample(rx\_af\_trunc,Modu.M); % Downsample by M = 4

**8° Demapping**

* In contrast to the mapping step at the transmitter, the Demapping will allow to transform the received sequence of complex symbols (**rx\_symb**)into the stream (**rx\_bin**) in order to improve the spectral efficiency of the system.
* [**rx\_bin**] = demapping(rx\_symb,Modu.Nbps,Modu.mod); % Demapping

## Implementation

As it could be expected, the steps 6, 7 and 8 (at the receiver) mirror steps 2, 3 and 4 (at the transmitter). The following Figure 5 shows the Constellation Demapping of 16QAM, and it’s quite interesting to directly compare it with the Figure 2 (Mapping Constellation). As expected, it can be seen at the first glance that no, or almost no, bit will be misinterpreted after the Demapping operation. It is possible to deduce from these points the exact value of the BER of our Communication Channel: after several successive simulations, the **BER** seems always oscillating between 0.0006 and 0.0014.

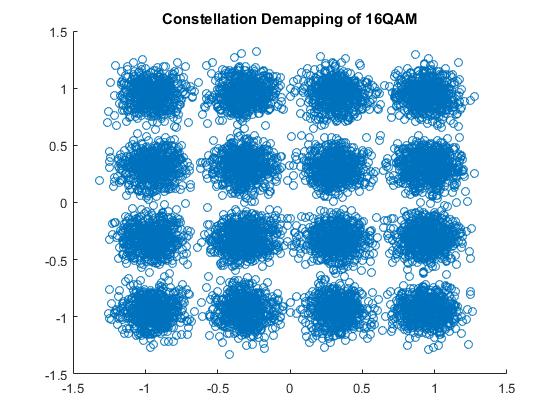


Figure 5 - Constellation Demapping of 16QAM

## Question 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?***
  + Our optimal Ccmmunication chain is completely simulated in the Baseband equivalent model. A Bandpass simulation, even if theoretically possible, would require a huge amount of computation power in order to realize all the Signal Processing. Indeed, the require computation power would be proportional to the Sampling Frequency which is required to be, at least, twice as big as the higher frequency component of the signal in order to avoid aliasing. However, a Baseband simulation allows to highlight the same interesting properties and mechanisms as the practical Bandpass communication channel.
* ***How do you choose the sample rate in MATLAB?***
  + ***TODO: From here I still have to complete (I’ll finish it after the IP lab tomorrow)… If you want to do it, go ahead ☺ I am on it ☺***
  + The factor M determines the sampling frequency in MATLAB.??
  + Sample rate (fsample) is chosen based on the symbol rate(fsymb). According to Nyquist sampling theory, which indicates that the frequency of sampling should be at least 2 times larger than the symbol rate.
  + In discrete Fourier Transform, the frequency domain, the frequency spectrum is aliased by the frequency fsample.
* ***How do you make sure you simulate the desire ratio?***
  + To ensure simulating the right Eb/N0, we first give the definition of the symbol here.
  + Eb: bit Energy for the signal. Eb=signal energy/encoded bits. Here we have to pay attention that the encoded bits are the bits we used to transfer the signal, rather than the symbols after the modulation.

[?] Signal Energy = integration(|signal|^2)/Fsampling [?]

* + N0 noise PSD(power spectrum density), stands for the noise at 1Hz bandwidth.
* ***How do you choose the number of transmitted data packets and their length?***
  + ***I remembered I have read somewhere that the number of data packets and length should be selected based on the BER of theory, will check that later. Or anyone any idea what this is about?***

## Questions regarding the communication system

* ***Determine the supported (uncoded) bit rate as a function of the physical bandwidth.***
  + TODO
* ***Explain the trade-off communication capacity/reliability achieved by varying the constellation size.***
  + TODO
* ***Why do we choose the halfroot Nyquist filter to shape the complex symbols?***
  + TODO
* ***How do we implement the optimal demodulator? Give the optimisation criterion.***
  + TODO
* ***How do we implement the optimal detector? Give the optimisation criterion.***
  + TODO
  + In our case, all the symbols (codewords) are equally probable, so the MAP criterion reduces to the ML one. If the symbols had different probability to be sent, we could easily implement a real MAP criterion to enhance the selection

# Part 2 – Time and frequency synchronisation

## Hypothesis and assumptions

TODO

## Implementation

TODO

## Question regarding the simulation

* ***Derive analytically the baseband modelof the channel including the synchronisation errors.***
  + TODO
* ***How do you separate the impact of the carrier phase drift and ISI due to the CFO in your simulation?***
  + TODO
* ***How do you simulate the sampling time shift in practice?***
  + TODO
* ***How do you select the simulated ratio?***
  + TODO
* ***How do you select the lengths of the pilot and data sequences?***
  + TODO

## Questions regarding the communication system

* ***In which order are the synchronisation effects estimated and compensated? Why?***
  + TODO
* ***Explain intuitively how the error is computed in the Gardner algorithm. Why is the Gardner algorithm robust to CFO?***
  + TODO
* ***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)?***
  + TODO
* ***Are the frame and frequency acquisition algorithms optimal? If yes, give the optimisation criterion.***
  + TODO

# Part 3 – Low-density parity check (LDPC)

## Hypothesis and assumptions

TODO

## Implementation

TODO

## Question regarding the simulation

* ***When building the new BER curves, do you consider the uncoded or coded bit energy on the x-axis?***
  + TODO
* ***How do you limit the number of decoder iterations?***
  + TODO
* ***Why is it much simpler to implement the soft decoder for BPSK or QPSK than for 16-QAM or 64-QAM?***
  + TODO

## Questions regarding the communication system

* ***Demonstrate analytically that the parity check matrix is easily deduced from the generator matrix when the code is systematic.***
  + TODO
* ***Explain why we can apply linear combinations on the rows of the parity check matrix to produce an equivalent systematic code.***
  + TODO
* ***Why is it especially important to have a sparse parity check matrix (even more important than having a sparse generator matrix)?***
  + TODO
* ***Explain why the check nodes only use the information received from the other variable nodes when they reply to a variable node.***
  + TODO

# Conclusion

TODO

# TODO

I list here the different points on which I think we still should work

* Completely finish the report
* Completely comment and clean the code
* Implementing a Circular-16QAM, in order to compare its efficiency (minimal BER for a given energy) with our actual Rectangular-16QAM ?
* Add a graph: BER in function of the Eb/N0 for different Modulation Scheme ?
* Also add a word about the Es/N0 (Energy per symbol to noise PSD) ?