

# Simulation and Performance Evaluation of Digital Communication Systems

## Abstract

The purpose of the project is the simulation and the performance evaluation of different models of transmission systems for radio relay link, focusing on the optimization of the main design parameters. The analysis is taken at physical layer by using a modular approach in order to inspect the properties and the mathematical model of each component. The results of the study are expressed in terms of energy required to satisfy a given quality of the link by considering different impairments in system configurations.

Master of Science Academic Project - 01NVTOT *Computer Aided Design of Communication Systems* - Professor: **Marina Mondin**

## Contents

<b>1 Introduction</b>	<b>1</b>	<b>6.3 Cross-Polarization Co-Channel Interference</b>	<b>14</b>
<b>2 Theoretical Foundations</b>	<b>2</b>	In-Band Co-Channel Interference - Interleaved Co-Channel Interference	
2.1 Simulation Theorem	2	<b>7 Optimization Stage</b>	<b>16</b>
2.2 Monte Carlo Simulations	2	7.1 Cost Function	16
2.3 Base Band Signal	3	7.2 Optimization Implementation	16
<b>3 Single Carrier System</b>	<b>3</b>	<b>8 Results</b>	<b>18</b>
<b>4 Single Carrier System Implementation</b>	<b>3</b>	8.1 Optimum System Parameters	18
4.1 Simulation Parameters	3	Considerations about $\beta_{in}$ - Considerations about $\rho$ - Considerations about $\alpha$	
4.2 Random Binary Sequence Generator	3	8.2 Analysis of Losses	19
4.3 QPSK/OQPSK Modulator	4	<b>9 Bibliographical References</b>	<b>24</b>
QPSK Modulation - Offset QPSK		<b>10 Appendix: MATLAB® code</b>	<b>24</b>
4.4 Upsampling Stage	4		
4.5 Shaping Filtering	5		
Root Raised Cosine - Power Normalization			
4.6 Amplification Stage: NLA	7		
4.7 Channel Model: AWGN/Two-Ray Path	8		
AWGN Channel - Two-Ray Channel			
4.8 Receiving Filtering	10		
4.9 Transient Removal	10		
4.10 Downsampling Stage	11		
4.11 Demodulation Stage	11		
4.12 Performance Evaluation	11		
Spectral Estimation - Semi-Analytic BER Estimator - Bit Error Rate Counter			
<b>5 Multi Carrier System</b>	<b>14</b>		
<b>6 Multi Carrier System Implementation</b>	<b>14</b>		
6.1 Simulation Parameters	14		
6.2 Adjacent Channels	14		

## 1. Introduction

Radio relay link is a common approach in communication systems, it is largely adopted for many applications like satellite transmission, airborne communication system, cellular network and many others. It is intrinsically affected by several issues mainly due to the physical nature of the propagation medium.

The simulation follows a bottom-up approach: starting from a simpler system asset, it proceeds with the implementation of additional features which better model the true nature of the real system. It is developed following the natural work-flow of conventional communication systems: each component is mathematically modelled with MATLAB® by paying attention to the computational complexity that the implementation may involve.

The main task of a first basic model is to simulate the transmission of a bit sequence modulated over an I/Q scheme. The resulting symbol sequence is upsampled and filtered by a Root Raised Cosine shaping filter, then after the power normalization, it is non-linearly amplified (NLA) and transmitted over AWGN or Two-Ray channel. Once the signal is received, it is

filtered by the reciprocal Root Raised Cosine filter, downsampled trying to reconstruct the original symbol constellation. From the received set of points, the effect of AWGN component over signal samples is evaluated semi-analytically in order to perform an estimation of the BER.

A further configuration of the model supplies the implementation of a Multiple Carrier System composed by Adjacent Channels, Co-Channel interferers from Cross-Polarization signals, and minor details about frequency spacing and time synchronization.

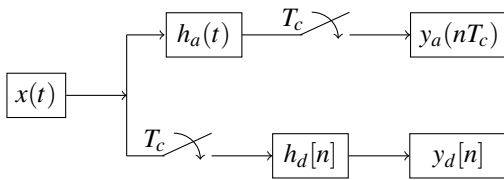
The result of the performance evaluation consists in the computation of the Cost in terms of Energy per bit to achieve a target BER. Once the Non-Linear Amplifier is implemented in the system, the Cost computation is re-considered by taking into account the distance of the Working Point from the Saturation one. The Cost value obtained in this way is then used to optimize system parameters adopted for a final performance characterization. Some additional features have been implemented in order to exploit an exhaustive analysis of the problem: the Demodulation Stage which allows to retrieve the received bit sequence, the Error Counter which evaluates empirically the BER. Some extra tools have been developed to make simpler the whole performance evaluation and optimization.

## 2. Theoretical Foundations

In this section, some theoretical principles are presented in order to describe the approach to the problem.

### 2.1 Simulation Theorem

The fundamental task of digital simulation of systems working in analog domain, is to build a discrete model that working on digital input generates a digital output which is as likely as possible to the discretized version of the analog output.



**Figure 1.** Key Idea behind Simulation Theorem

$$y_a(t) = x_a(t) * h_a(t) = \int_{-\infty}^{+\infty} x_a(\tau) \cdot h_a(t - \tau) d\tau \quad (1)$$

$$y_d[n] = x_d[n] * h_d[n] = \sum_{k=-\infty}^{+\infty} x_d[k] \cdot h_d[n - k]$$

A graphical representation of this, can be seen in Figure 1. Given a Sampling Time  $T_c$ , every discrete sequence  $x[n]$  can be considered as the sampled version of a continuous signal  $x(t)$  at time  $nT_c$ , represented as  $x[n] = x(nT_c)$ .

A good discrete model has to verify the condition:

$$y_a(nT_c) = y_d[n] \quad (2)$$

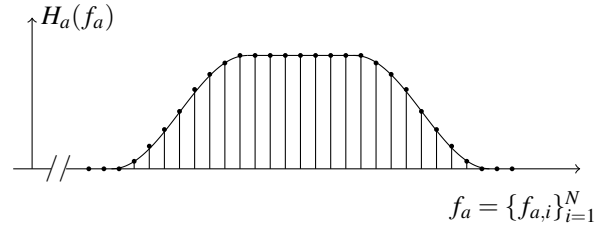
In [1] it is shown that the condition (2) is obtained if:

$$H_a(e^{j2\pi f_a}) = H_d(e^{j2\pi f_a T_c}) \quad (3)$$

That holds under the condition  $f_d = f_a T_c$ .

In this simulation, the technique adopted to characterize the filtering operations in discrete domain is the so called Frequency Sampling Method.

It consists of sampling the analog frequency response (obtained from analytical expression or from measurements) of the desired filter and extracting its digital impulse response by applying IDFT.



**Figure 2.** Spectral Sampling Method

In Figure 2 we can see the  $N$  samples of an analog frequency response at frequency  $f_{a,i} = i/NT_c$ .

$$h_d[n] = \sum_{i=0}^{N-1} H_a(f_{a,i}) e^{j2\pi f_{a,i} n T_c} \quad (4)$$

Discrete time impulse response is obtained by applying IDFT from Equation (4).

By doing this, every filter impulse response implemented in this project aims to satisfy the condition (3).

### 2.2 Monte Carlo Simulations

In order to obtain a reliable performance estimation from the analysis of the digital communication system of interest, Monte Carlo Simulation is performed.

A large number of i.i.d. random binary sequences are generated, so that the overall analysis, exploiting all the possible event space, allows to evaluate reliable performance indices as referred in [2].

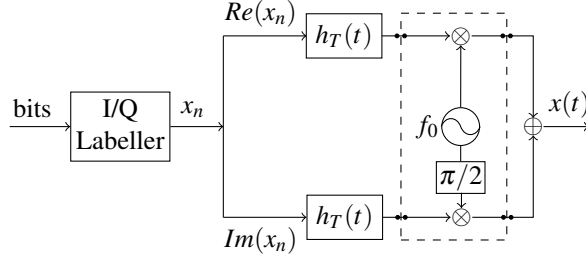
Exploiting the Ergodicity property of WSS random processes [3], for which  $\langle x(t) \rangle = \mathbb{E}[x(t)]$ , the average results are executed by averaging long sequences instead of averaging multiple times over realizations of shorter sequences.

### 2.3 Base Band Signal

Instead of working around a given carrier frequency  $f_0$ , the system is supposed to work with signals in base band as shown in Figure 3, taking off the dashed frequency modulator.

In real systems, the signal is shifted around  $f_0$  before it is transmitted over the channel. Once it is received, it is shifted back in base band before being processed in digital domain, where the frequency shift is not considered.

Systems working in digital domain process samples of waveforms, operating in base-band generally guarantees a lower



**Figure 3.** Digital I/Q Modulator

sampling frequency, according to Nyquist Sampling Theorem for narrowband signals.

### 3. Single Carrier System

In order to give a simpler description of the whole simulation model which is composed by multiple parallel channels, the main work-flow of simulator is previously analyzed for a single data stream.

The single carrier transmission system is represented by the block schemes in Figure 4. It is composed by a random bit generator which emits a binary uniformly distributed random sequence that is modulated by QPSK/OQPSK modulator in order to produce a symbol sequence, where each symbol carries  $N_b$  bits. The so obtained sequence is then upsampled to obtain a sufficient number of samples for the following pulse shaping stage which is performed with Root Raised Cosine Filter. In order to get a unitary average signal power, Power Normalization is performed. NLA amplifies the shaped symbols before they are transmitted over the Channel. According to the adopted model, it is possible to implement AWGN or Two-Ray path channel. At the receiver side a complementary Shaping Filter performs a LPF operation. Finally, the received signal is down-sampled to reconstruct the correct number of samples to be demodulated from its constellation scheme.

The adoption of Shaping Filters introduces transient to the shaped sequence. The evaluation and the removal of this portion of signal is mandatory to reconstruct the transmitted sample sequence. Even if this is not reported by the block scheme, it is provided by the simulator.

The approach to the simulation is *Switch-Based*. By following the modular configuration of the scheme we introduce a flag variable for each impairment of the system, in order to activate/deactivate specific simulation blocks. We have three specific switches respectively for QPSK/OQPSK, NLA and AWGN/Two-Ray channel. The multi-carrier system will take in account further features.

### 4. Single Carrier System Implementation

In this section, a complete simulation implementation of the Single Carrier System is provided. Some minor basic and

theoretical details are omitted in order to focus the attention on specific implementation tasks. Main important theoretical results are instead remarked in order to justify adopted technical solution.

#### 4.1 Simulation Parameters

In order to build a suitable simulator, several parameters are instantiated at the beginning of transmission simulation routine. These parameters are in general given from specifications.

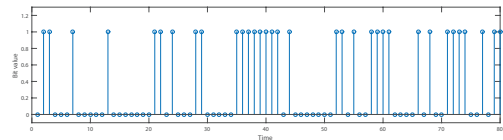
##### Simulation Parameters

Rb	[18, 36 Mbps] <i>Bit rate</i>
Nb	[2,4] <i>Bits per Symbol</i>
Rs = Rb/Nb	[Mbauds] <i>Symbols per Second, Baud-rate</i>
Ts = 1/Rs	[ $\mu$ s] <i>Symbol interval (Time of Symbol)</i>
Tb = 1/Rb	[ $\mu$ s] <i>Bit interval (Time of bit)</i>
Ns	<i>Samples per symbol</i>
Tc = Ts/Ns	[ $\mu$ s] <i>Sampling interval</i>
v	[25] <i>Number of symbols for window</i>
N = v · Ns	<i>Number of samples Window</i>
Tw = N · Tc	<i>Window time duration</i>
Nwin	[100] <i>Number of windows</i>
NTotBit = Nwin · N	<i>Number of bits to simulate</i>
NTotSym = NTotBit/Nb;	<i>Number of symbols to simulate</i>
winr = window(·, N)	<i>Welch Periodogram window</i>

The length of the simulated sequence  $N_{TotBit}$ , as will be seen in Sections 4.12.1, 4.12.2, 4.12.3 is set by taking into account the frequency resolution and variance of the Spectral Estimator, Semi-Analytic and BER Counter Estimators' variance. All of these constraints do not affect the simulation performances, but they induce a longer or shorter simulation time duration by increasing or reducing the length of the generated random sequence.

#### 4.2 Random Binary Sequence Generator

The random data are generated with a single MATLAB instruction which allows to obtain a desired-length sequence of random bits, by rounding random samples of normal distribution. The `rand()` function is used as reported in (Appendix.1). It is possible to verify the effective randomness of the sequence and to use some better random generators, but this is not the purpose of this simulator. In Figure 5 is shown a small portion of a realization of a generated bit sequence.



**Figure 5.** Random Generated Data Sequence

The characteristics of the random generator and the overall length of the simulated sequence has to be sufficiently large in order to provide an exhaustive set of occurrences of generated symbols.

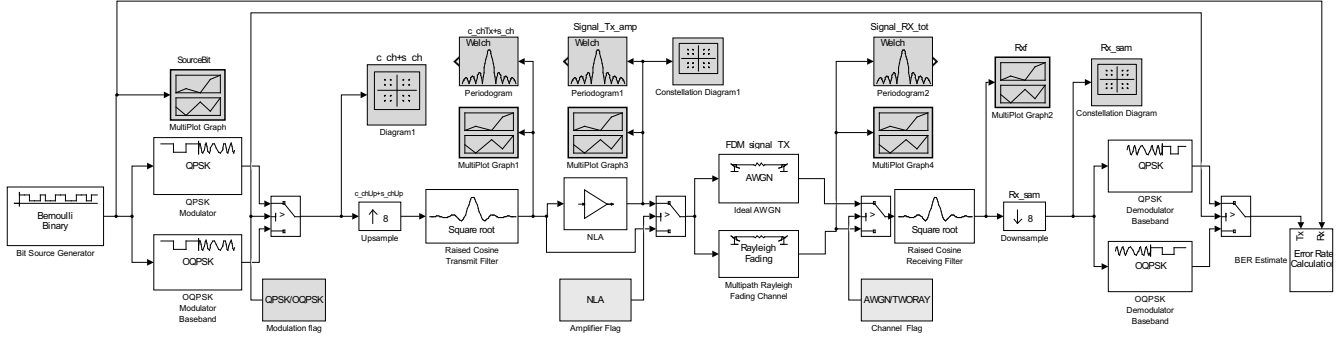


Figure 4. Single Carrier Simulation System

### 4.3 QPSK/OQPSK Modulator

For the simulation both QPSK and OQPSK are implemented.

#### 4.3.1 QPSK Modulation

The modulation schemes (4-QAM, 16-QAM) are obtained using a given function called `QPSK_mod.m` (Appendix.2). The function takes as input the generated bit sequence, the number of total symbols to be generated and the modulation efficiency coefficient. It is possible to use this same function to modulate a group of parallel channel specifying how many are as second parameter.

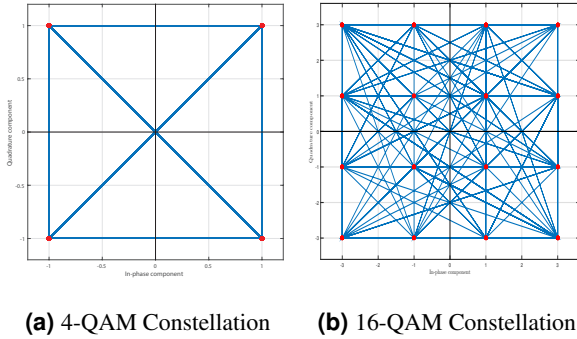


Figure 6. Symbols and Energy transitions over the two modulation schemes

The outputs of the function are respectively the quadrature (Q) and the in-phase (I) components of the modulated signal that is represented as the complex envelope of the original one. As shown in Figure 6, the points are exactly mapped to the respective I and Q bit coordinates.

As previously said in Section 2.3, it is assumed to work with baseband signals, so any sinusoidal modulation and demodulation is represented or simulated.

#### 4.3.2 Offset QPSK

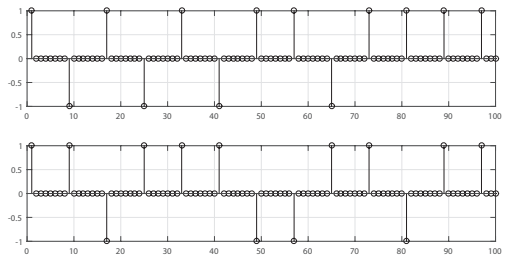
The Offset QPSK (OQPSK), is a technique used to avoid zero-crossing transitions over signal's scattering diagram, meaning that it has phase transitions of  $\pi$ , which lead to higher values of Peak-to-Average Power Ratio (PAPR), that is discussed in 4.6. It consists of shifting the Quadrature component of half of the

symbol time, hence the digital sequence of  $N_s/2$ . The implementation is made by first using the function `QPSK_mod.m` and then applying the shift (Appendix.6).

### 4.4 Upsampling Stage

This operation is required to obtain a properly shaped digital sequence from the symbol sequence. By the simulation point of view, analog waveforms are never considered because they only come out in the real system after the analog conversion on the RF front-end, as reported in Section 2.3.

Since this is a digital simulation program, the so called digital waveform is considered practically as a sampled version of analog waveforms even if quantization task is neglected.

Figure 7.  $N_s = 8$ , Upsampled Data Sequence

This operation is performed by running the function called `upsampling(.)`, (Appendix.3) on the outputs of the modulation block and applying zero padding of  $N_s - 1$  samples for each real sample of the original sequence. The result is reported in Figure 7.

After the Upsampling Stage, zero-crossings appear in the scattering diagram, due to the combination of zero-samples of In-phase and Quadrature components of the signal, but this is not actually significant.

### 4.5 Shaping Filtering

The Shaping Filter is one of the key elements of the transmission system since it characterizes the spectral occupancy of the single symbol over the channel and so the dynamics of the whole signal.



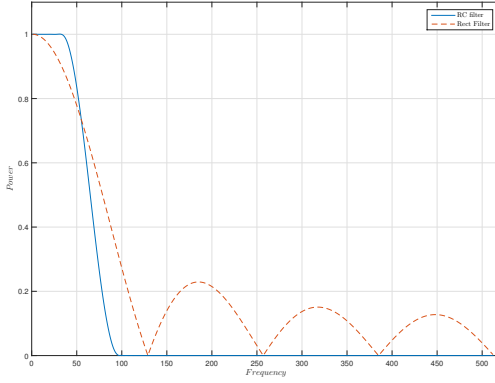
According to Section 2.1, the Frequency Sampling Method is implemented in order to extract discrete filter impulse response from samples of its analog frequency response.

Once a specific Shaping Filter is chosen, it influences the performances of the whole system.

In general, over the frequency axis, it acts as a good low pass filter and it is characterized by few amount of power located in its frequency side-lobes, whereas over the time axis, it has a lot of oscillations and its support is generally wide.

By setting its proper parameters it allows to find a trade-off between the signal's bandwidth occupancy and its behaviour over time.

In general the shaped pulsed sequence needs to be filtered at the receiver in order to reconstruct the original sample sequence, the re-filtering operation is usually done with a matched filter.



**Figure 8.** Rectangular Filter vs Raised Cosine Filter Spectrum

In Figure 8 is reported the comparison between RC filter and Rectangular bandwidth occupancy. It is possible to observe that RC filter can reach better performances, since its bandwidth occupancy is narrower it induces less ICI. This makes it more suitable to real applications where more than a single frequency slot has to be allocated over the available bandwidth.

Since this project aims to analyze a multi-carrier system, it is useful to choose RC shaping filter, even if it is less effective in single-channel systems due to the induction of ISI.

#### 4.5.1 Root Raised Cosine

For the purpose of this project, a Root Raised Cosine (RRC) shaping filter is implemented.

In order to use a Raised Cosine shaping filter for the overall filtering operation, it is required to split it in the two relative TX and RX filters.

This is made by means of a splitting factor  $\alpha$  at the transmitter,

hence it needs to be complementary at the receiver  $(1-\alpha)$ .

$$\begin{aligned} R(f) &= H_T(f) \cdot H_R(f) \\ H_T(f) &= R(f)^\alpha \\ H_R(f) &= R(f)^{1-\alpha} \end{aligned} \quad (5)$$

By splitting RC filter in two filters, the purpose is to get a smoother signal in transmission and a narrower bandwidth occupancy at the receiver, so that it intercepts less noise.

The general analytic formulation for the RC is the following:

$$R(f) = \begin{cases} T, & |f| \leq \frac{1-\rho}{2T} \\ \frac{T}{2} \left[ 1 + \cos \left( \frac{\pi T}{\rho} \left[ |f| - \frac{1-\rho}{2T} \right] \right) \right], & |f| \leq \frac{1+\rho}{2T} \\ 0, & \text{elsewhere} \end{cases} \quad (6)$$

The parameters of the shaping filter are  $\alpha, \rho$ , and  $M$ :

- **Roll-off factor ( $\rho$ ):** it characterizes the bending of the fall region in frequency domain, hence the bandwidth occupancy  $B_X = (1 + \rho)/2T_s$ .

In time, low values of  $\rho$  induce a smoother behaviour more similar to a  $\text{sinc}(\cdot)$  function, whereas large values of  $\rho$  induce a sharper time evolution.

In Figure 9 is possible to see the effect of  $\rho$  on the reduction of bandwidth occupancy of the main lobe with an concentration of power in lower frequency region.

Higher values of  $\rho$  let the immunity to erroneous sampling instants increase, since the eye diagram gets wider [4].

- **Splitting Factor ( $\alpha$ ):** it corresponds to the exponent of the RRC formula (5). When  $\alpha = 0.5$  the filter is said to be a Square Root Raised Cosine.

In time, low values of  $\alpha$  induce a lower peak and higher oscillations with respect to high values.

In Figure 10 is shown how splitting factor impacts on frequency domain. It does not affect the bandwidth occupancy of the filter, but it acts on the distribution of the spectral components over it.

- **$M$ :** is a implementation parameter which defines the additional support of the filter over time axes. It describes the number of side-lobes that exceed the main  $N_s$  typical support of the shaping filter.

$$N_{taps} = 2(M + 1) \cdot N_s + 1 \quad (7)$$

Where  $N_{taps}$  is the number of samples per symbol after the transmitting filter, whose output signal is the convolution of the upsampled symbol sequence by the transmitter's RRC pulse shape.

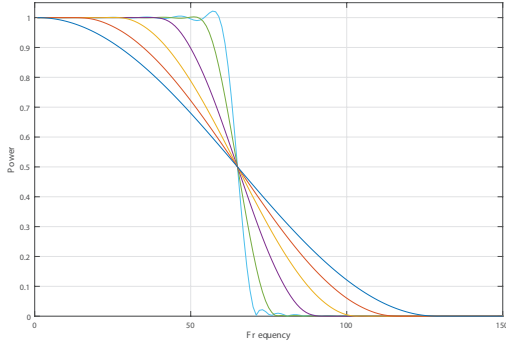
The shaping filters are defined by using the appropriate function `RaisedCosineFIR.m` (Appendix.4) which returns the coefficients of the digital filter itself. Since it is a FIR filter its impulse response is defined by the  $B_i$  coefficients.

The function takes as input parameters the symbol rate  $R_s$ , the number of samples per symbol  $N_b$ ,  $M$ ,  $\rho$ ,  $\alpha$ . The length of the output sequence of coefficients is odd because the shape must be centered over the symbol sample and the phase of the signal must be linear [5].

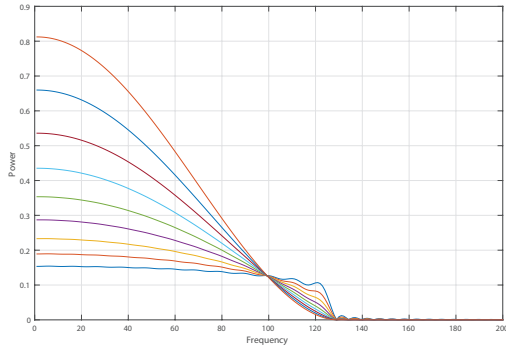
After the definition of the filters, the upsampled sequence is filtered by using the `filter()` command (Appendix.5).

The filtering operation performed by the simulation environment provides as output a shaped sequence which still has the support of the upsampled sequence.

Since the shaping filter may have a wide time duration, and given that convolution operation exceeds upsampled sequence's support, some of the symbols are not shaped.



**Figure 9.** RRC Frequency Response  $\rho = [0.1:0.9]$ ,  $\alpha = 1$ ;



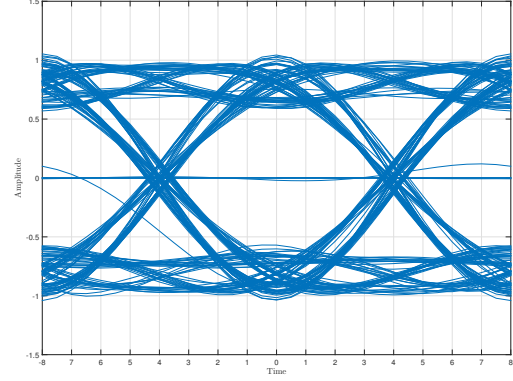
**Figure 10.** RRC Frequency response  $\alpha = [0.1:0.9]$ ,  $\rho = 1$ ;

In Figure 11 is reported an eye-diagram showing the effect of pulse shaping on the modulated sequence.

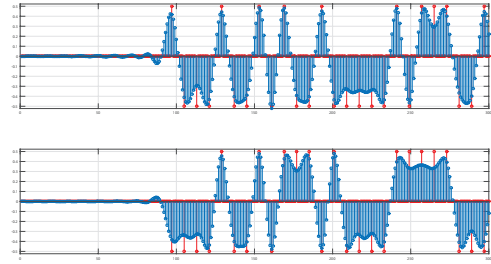
In Figure 12 it is shown that the shape of the pulses is not preserved because the side-lobes in the time evolution of each symbol affect the samples of close symbols. It is visible that the the upsampled signal is in reality modified due to the dynamics of the raised cosine. This induces a new constellation shape made by some disperse point around original ones.

#### 4.5.2 Power Normalization

Since NLA model works with normalized output power a Power Normalization needs to be performed to exploit correctly the amplifier model. The main purpose is to send to the



**Figure 11.** Eyediagram after TX Pulse Shaping.



**Figure 12.** Original I/Q symbols over Shaped-Sequence

Non Linear Amplifier a maximum average power equal to 1 because in NLA digital models the analysis is designed with unitary input and saturation powers.

So, the average power is computed from:

$$P_0 = \frac{1}{N_{sam}} \sum_{n=1}^{N_{sam}} |x(nT_c)|^2 = \frac{1}{N_{sam}} \sum_{n=1}^{N_{sam}} x_I^2(nT_c) + x_Q^2(nT_c) \quad (8)$$

Where  $x_I(nT_c)$  and  $x_Q(nT_c)$  represent in-phase and quadrature samples of complex envelope and  $N_{sam} = N_{TotBit} \cdot N_s$ .

The signal samples are scaled by  $P_0$ , so that the final mean power of the overall signal is unitary (Appendix.7).

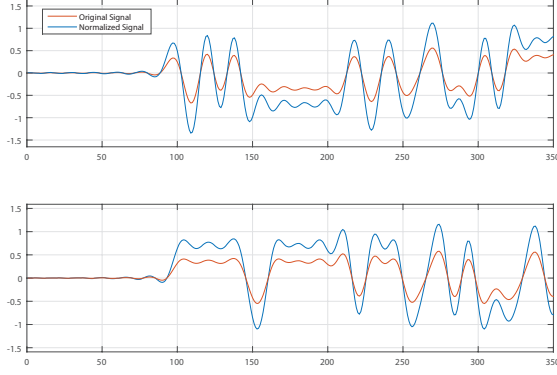
As the Figure 13 shows, the mean value of the signal is now equal to one, so the original signal is practically defined in order to have better property in terms of power/energy.

#### 4.6 Amplification Stage: NLA

Due to RF propagation law referred in Section 4.7, the signal has to be amplified before the transmission at the physical layer, in order to ensure sufficient power to reach the transmitter at a given distance.

Since the signal attenuates while propagating, its integrity may be compromised, so Amplifiers provide additional power to the signal.

Working at ideal conditions, they should produce a linear amplification consisting in supplying an output power linearly proportional to the input one. However real devices can not behave like this and they usually introduce amplitude and phase



**Figure 13.** Effects of Power Normalization on I/Q modulated sequence

distortions which depend on their proper I/O characteristic and parameters. Since the amplifiers are memoryless devices, the effects on Amplitude and Phase are only dependent on input power at current time.

The observable effect is an alteration of the signal due to the combination non-linearity and to the PAPR of the input signal.

$$PAPR = \frac{\arg\max_{x(nT_c)} (|x(nT_c)|^2)}{P_0} \quad (9)$$

This is a key index of variation of the instantaneous power with respect to the average one, namely signal dynamics.

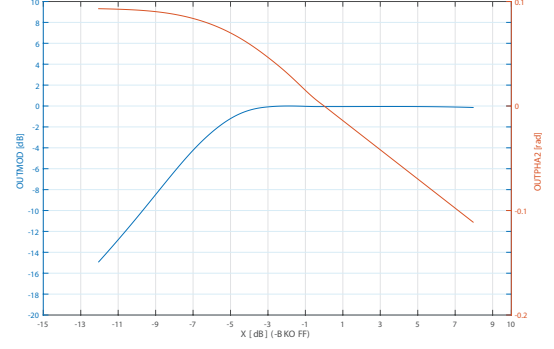
In order to limit undesired effects of NLA the aim is to reduce it, hence to let PAPR tend to 1.

The total amount of power they can provide as output is finite, so they can not exceed a saturation threshold called  $P_{out,sat}$ . Each device is characterized by a specific saturation zone, in which every input amplitude is mapped to values which are close to the saturation threshold. In general, saturation point is the best working condition in terms of power utilization. Unfortunately, as much as the amplifier works close to the saturated regime, it introduces more and more distortion: when the output power is constant for every different input power, the dynamics of the signal is significantly altered if not destructed.

It is possible to set the parameter known as Back-off in  $\beta_{in}$ , so that the amplifier works far from saturation point as much as this value grows.

The drawback of having a high value of  $\beta_{in}$ , is that the power of the signal is enhanced less than it could, so that the NLA may not reach the value of saturation (and relative good utilization of available power), then there is an amount of power which is not used and is actually lost.

A measure of the power that is lost when the NLA is working with  $\beta_{in} > 0$  is the Back-off out  $\beta_{out}$  which is directly computed as difference from  $P_{out,sat}$  and the output power corresponding to  $\beta_{in}$  input one.



**Figure 14.** NLA Characteristic NLA\_flag=2,  $\beta_{in}=0$

In Figure 14 is reported the AM-AM/AM-PM curve of the implemented amplifier model coming from ESA satellite transmission system [6].

Two main effects are induced by the NLA that operates on the power of the signal considering its complex envelope.

- **Bandwidth Enlargement:** Input signals with wider dynamics, as said before, reduce the enhancement of side-lobes in power spectral density.

Dynamics mainly depends on the adopted  $h_T(t)$  shaping filter time evolution and on the chosen modulation scheme and on the previously discussed PAPR.

In single-carrier context, it does not represent a significant issue but in a multi-carrier approach it may be destructive for the integrity of the whole transmission.

The power spectrum of the amplifier is given from the following formula:

$$G(\rho_x) = A\rho_x \cdot \frac{1}{1 + B\rho_x^2} \quad (10)$$

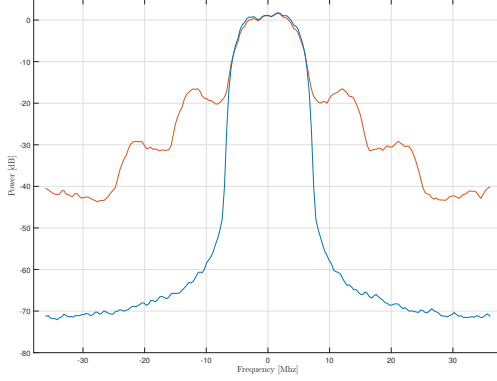
Where  $A$  and  $B$  are parameters of NLAs model. The last term of the equation can be described with Taylor expansion as follows:

$$\begin{aligned} G(\rho_x) &= A\rho_x (1 + \alpha_1 \cdot B \cdot \rho_x + \alpha_2 \cdot B^2 \cdot \rho_x^2 + \dots) \\ &= A (\rho_x + \beta_1 \cdot \rho_x^3 + \beta_2 \cdot \rho_x^5 + \dots) \end{aligned} \quad (11)$$

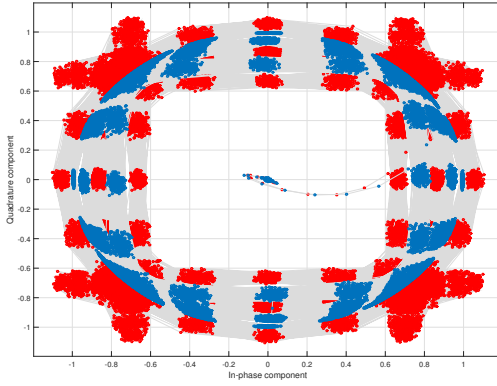
It is easy to observe that the overall power spectrum can be approximated as the linear combination of the three main components which are respectively: the main lobe in band  $B_A$ , a lower amplitude spectral portion with bandwidth  $3B_A$  and a still lower amplitude portion with wider bandwidth  $5B_A$ .

- **Time Distortion:** Is the natural effect on waveform induced by amplitude and phase distortion in Non-Linear Behaviour.

In this project is implemented the model of a Solid State Non-Linear Amplifier. In Figure 15 is shown the power spectrum



**Figure 15.** Effects of Power Amplification on Signal original Power Spectral Density,  $\beta_{in} = 3.5$  dB



**Figure 16.** Effects of Power Amplification on Signal original Constellation,  $\beta_{in} = 3.5$  dB

of the same signal pre and after the amplifier. The signal is visibly distorted after Amplifier due to its previously described effects.

In the scattering Diagram reported in Figure 16 it is possible to observe a great distortion due to the saturation of the amplifier when  $\beta_{in}$  assumes low values.

Constellation Points with higher amount of energy are mapped at the same level of energy, distributed around a circle whose radius is defined by the  $P_{out,sat}$ .

In order to Simulate NL Amplification (Appendix.8), it is defined an opportune set of parameters.

#### NLA Parameters

NLA_flag	Selects the kind of NLA to use
P_out_sat	Saturation Power of the NLA

Since the simulation provides more than an NLA model, the flag NLA\_flag is used to identify the one to be used.

### 4.7 Channel Model: AWGN/Two-Ray Path

The channel model represents in a compact way the alteration of the signal received from the RF front-end with respect to

the transmitted one.

Channel effects on real systems are the following:

- **Attenuation:** According to Free-Space propagation law, the signal itself is affected by attenuation:

$$A_{fs} = \left( \frac{\lambda}{4\pi l} \right)^2 = \left( \frac{c}{4\pi l f} \right)^2 \quad (12)$$

- **Time Delay:** A transmitted signal propagating in Line Of Sight (LOS), is received a delay  $D = l/c$ , where  $l$  is the direct path distance between TX and RX.

When the signal propagates in LOS, these two effects are transparent to the analysis performed in this project, so they are not taken into account for the simulation.

Both the time evolution  $h_C(t)$  and the relative frequency response  $H_C(f)$  of the channel need to be modeled.

A true channel has to take into account that the propagation involves more than a single ray path and is referred to multipath environment. The issues involved in this model require to compensate the effect of the channel over the signal, equalization and the implementation of more complex achievements to the simple model. The most common approach is trying to invert the frequency response of the channel, but this will not be object of inspection.

Depending on channel conditions, propagation environment and the bandwidth occupation of the signal, the channel may induce deviation of the attenuation on signal's spectral components.

This effect is known as Fading and may be considered of two kinds: Slow Fading and Fast Fading.

Slow fading, unlike Fast Fading, refers to the case where the channel frequency response is almost flat over the signal bandwidth.

In order to implement the channel behaviour, different models may be considered. In this project are used the AWGN Channel and the Two-Ray Path ones. Both are treated as linear filters. As for shaping filters, it is supposed to work under Simulation Theorem condition (2.1).

#### 4.7.1 AWGN Channel

The simplest channel model that one may think of is the Additive White Gaussian Noise. Additive because it is supposed to add linearly noise samples to the digital waveform of the signal; White because its Power Spectral Density is supposed to be flat over the frequency axis and equal to  $N_0/2$ ; Gaussian because the noise samples are supposed to be random distributed with a zero-mean Gaussian distribution. Since the signal samples of I/Q modulation schemes are supposed to be complex, the noise samples are modelled as Complex Gaussian r.v. with zero mean and variance  $N_0$ .

In this case, since the frequency response of the channel is constant, the coefficient of the equivalent FIR filter would be a single delta in the origin.

$$y(t) = x(t) * \delta(t) \quad (13)$$

Once the effect of delay and attenuation is neglected, the signal received is supposed to be equal to the transmitted one (Appendix.9).

As will be reported in the Section 4.12.2, the evaluation of BER as a function of the ratio  $E_b/N_0$ , will be computed semi-analytically, so the generation of noise samples will not be performed, whereas it will be done for the empirical computation of bits in error that will be reported in 4.12.3.

#### 4.7.2 Two-Ray Channel

A simplified version of the multipath channel is adopted: the channel is modelled as Two-Ray path, where the main signal and a single reflected version reaches the receiving antenna with a certain delay, as shown in Figure 17.

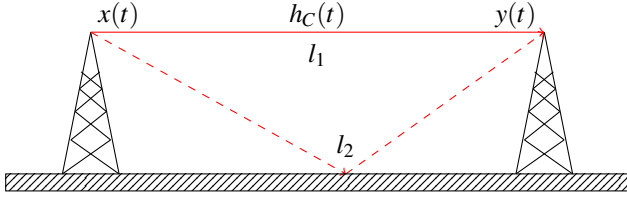


Figure 17. Two-Ray Path Model

The attenuation experienced by a signal propagating along a Two-Ray path would be:

$$A_{lr} = A_{fs} \cdot |\alpha_c|^2 \quad (14)$$

Where  $\alpha_c = |\alpha_c|e^{j\phi_c}$  is a complex coefficient (its conjugate form is  $\alpha_c^* = |\alpha_c|e^{-j\phi_c}$ ), that keeps into account the extra attenuation due to Two-Ray path propagation.

Two-ray model has a quite simple representation, that can be analyzed almost easily both in time and frequency domain. Its mathematical model is obtained designing a specific FIR filter with two delta pulses. As previously assumed, it is supposed that the effect in terms of delay and attenuation of the ray propagating along the shortest path may be neglected. It is considered only the differential attenuation and delay in time between the two paths.

The theoretical idea is to represent our received signal as follows:

$$y(t) = x(t) + \alpha_c \cdot x(t - \Delta D) \quad (15)$$

where  $\Delta D = D_2 - D_1 = (l_2 - l_1)/c$  is the difference in terms of time delay between the two ray paths.

The channel impulse response may be formulated as:

$$y(t) = x(t) * [\delta(t) + \alpha_c \cdot \delta(t - \Delta D)] = x(t) * h_C(t) \quad (16)$$

$$h_C(t) = \delta(t) + \alpha_c \cdot \delta(t - \Delta D) \quad (17)$$

$$H_C(f) = 1 + \alpha_c \cdot (e^{-j2\pi f \Delta D}) \quad (18)$$

Channel's frequency response is [1]:

$$|H_C(f)|^2 = 1 + 2|\alpha_c| \cos(2\pi f \Delta D - \phi_c) + |\alpha_c|^2 \quad (19)$$

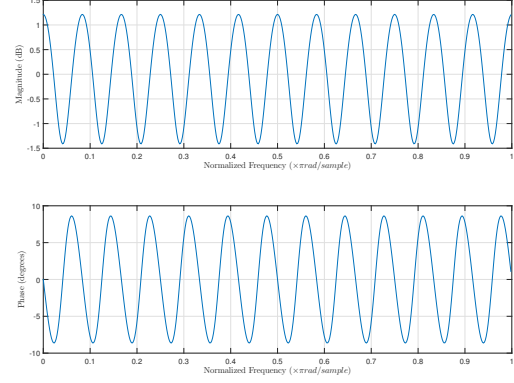


Figure 18. Power Spectral Density and Phase variation of the channel

The Channel PSD modelled as Two-Ray Path Propagation is reported in Figure 18.

In Figure 19 is shown the overall signal PSD after the channel. It reports a zoomed version of Figure 15, showing sinusoidal oscillations that appear quite small, but induce a power variation up to 3 dB so are responsible of considerable distortion for the received signal.

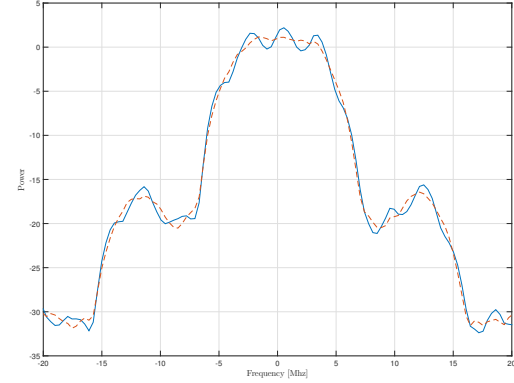
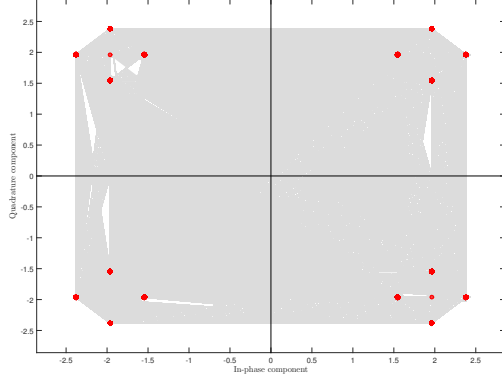


Figure 19. Spectral differences pre and after channel filtering

Multipath typically induces the Fast Fading effect. The fluctuations in the distribution of channel's power spectrum are very dense. In the Two-Ray model, they are sinusoidal with period  $B_c = 1/\Delta D$ , where  $B_c$  is the so called Coherent Bandwidth, and determines the spectral slot in which, a signal of bandwidth  $B \ll B_c$  experiences an almost flat frequency response [7]. It is simulated a Delay  $\Delta D = 3T_s = 0.33 \mu s$ , which, in a real environment would be a space difference between the two paths of about  $l_2 - l_1 \simeq 100m$ .

In Figure 20 is shown the effect of the superposition of the original signal and its delayed copy due to Two-Ray Channel. The plot is obtained by switching off any additional source of distortion (such as NLA).





**Figure 20.** Effects of Two-Ray Propagation Channel with any impairment.

In order to Simulate Two-Ray Channel model (Appendix.8), is defined an opportune set of parameters:

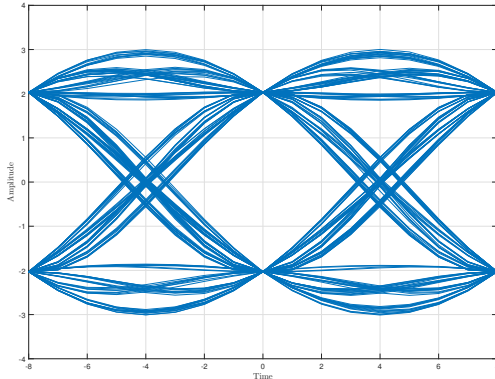
**Two-Ray Channel Parameters**

Coch_Att	=	$\alpha_c$	[dB] Extra Attenuation on the second ray path
Phase_Shift	=	$\phi_c$	[rad] Extra Phase Shift on the second ray path
Delay_Spread	=	$\Delta D$	[ $\mu$ s] Delay Spread of two ray path channel
ch_A			Channel FIR A coefficient
ch_B			Channel FIR B coefficient

The implementation (Appendix.10), as previously said, is done by modelling the channel as FIR filter, then filtering the signal by using the `filter()` instruction from the Expression (16).

#### 4.8 Receiving Filtering

After the channel, the signal is hence distorted, at this step it is filtered by reciprocal RRC filter (with splitting factor  $1-\alpha$ ). The properties of RRC shaping filters were already discussed in Section 4.5.1, so it can be useful to report here the overall effect of the combination of TX and RX filtering operations.



**Figure 21.** Eyediagram of Equivalent RC Filter

The equivalent double filtering operation corresponds to the

filtering of the signal by the convolution of the two filters. It is important to underline the presence of a transient due to the long time support of the filters and the property of convolution.

Differently from the transmitted signal, the received signal obtained after the receiver's filtering satisfies Nyquist's first criterion for ISI. The product of the two Raised Cosine pulses with complementary splitting factor, relies a final pulse shape whose samples at time instants multiple of  $T_c$  that corresponds to the pulse center does not change over time [8].

The Nyquist NO-ISI condition can be expressed as:

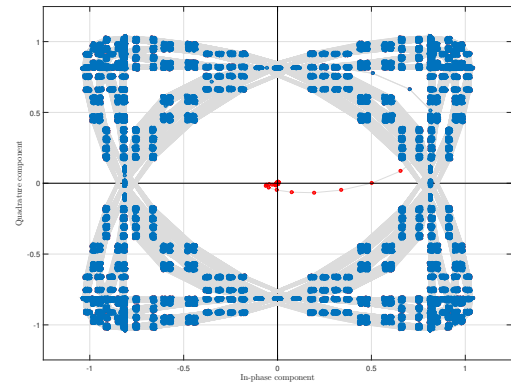
$$r(nT_c) = \begin{cases} 1, & \text{if } n = 0; \\ 0, & \text{if } n \neq 0; \end{cases} \quad (20)$$

where  $r(t) = h_T(t) * h_R(t)$ , the convolution of the two pulse shapes has to be constant each time instant at the center of each sample, while it has to be zero at every integer multiple of the sampling time. In Figure 21 is reported the Raised Cosine eyediagram, showing that for each sampling instant the value assumed by the shaping filter is constant. The amplitude is not equal to one, as reported in Section 4.5.2, the simulation performs power normalization, so that the amplitude of the transmitting signal is not preserved.

This was simulated by constructing the reciprocal receiver raised cosine filter, then using it to `filter()` the received signal (Appendix.11).

#### 4.9 Transient Removal

As already introduced, the shaping filter stage introduces a transient, consisting of additional samples that are not part of the symbol sequence labelled on the original constellation. They are represented by red points in Figure 22. Such plot reports the I/Q pairs of all the received sample sequence, the actual received symbols will be obtained in the following stages.



**Figure 22.** Transient Removal shown over the RX Constellation.

The support of the equivalent filter is  $2 \cdot N_{taps}$ , so that after the first  $N_{taps}$  samples, the process reaches steady state.

The length of such a transient is evaluated as follows:

$$N_{tran,sym} = \frac{N_{taps} - 1}{N_s} \quad (21)$$

$$N_{tran} = N_{tran,sym} \cdot N_s + 1$$

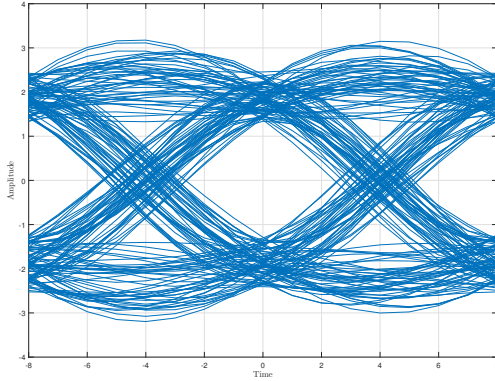
Where  $N_{tran,sym}$  is the number of symbols affected by transient and  $N_{tran}$  is the corresponding number of samples which compose it.

It is needed to remove  $N_{tran}$  samples from the received sequence. In order to perform a consistent demodulation, it is important to remove also the first symbol samples  $N_s$ . If this task is not performed correctly, previously described extra symbols should be identified in the constellation. The implementation was easily made by truncating signal samples sequence (Appendix.12).

As anticipated in Section 4.5.1, the filtering operation fits the output of the convolution in the support of the input sequence. Since it introduces transient, the symbols at the end of the sequence exceed the support hence they are not shaped, hence not transmitted.

#### 4.10 Downsampling Stage

This section is properly divided taking into account the selection of OQPSK or classical QPSK approach, is important to consider the exact sample shift adopted in the first context in order to realign correctly the sequence.



**Figure 23.** Eyediagram of the received signal, showing optimal sampling instant with NLA, Two-Ray Path.

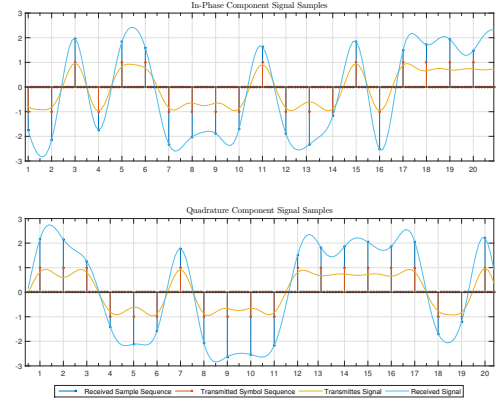
It is easy to see from the Eyediagram such as the one reported in Figure 23 that the optimum sampling instant is  $N_s$  in both cases, but in the case of OQPSK the Quadrature component needs to be shifted back by  $N_s/2$ .

This is exactly what the simulation does, before the received signal is downsampled by taking a sample every  $N_s$ , (Appendix.13).

#### 4.11 Demodulation Stage

Functions for I/Q demodulation have been implemented for 4-QAM and 16-QAM modulation schemes in order to reconstruct the original bit sequence and experiment the presence

of binary errors. Demodulated bits are less than modulated ones because of shaping filtering issues already discussed in Section 4.9.



**Figure 24.** Match between Symbols and Signals TX and RX for 4-QAM with Offset.

In Figure 24 is reported the time evolution of a 4-QAM modulated signal and Offset QPSK and the sample sequence at the receiver with respect to the transmitted one.

The simulation implements the computation of threshold values that define Voronoi Decision regions by using a simple algorithm based on the folding of I-Q plane until the mean values of received points match to a single one (Appendix.14). Once the Voronoi Decision regions are defined, the decision is performed, based on the position of the points in the constellation either in 4-QAM (Appendix.15) and in 16-QAM (Appendix.16) modulation schemes.

#### 4.12 Performance Evaluation

In this section is reported a set of tools which are used to appreciate system performances. As reported in Section 2.2, all of the performance evaluation apply the technique of Monte Carlo simulations. Most of the performance indices are provided by estimations which, instead of considering a statistic average of results, run over long sequences of data in order to provide outputs which are meaningful on the average.

##### 4.12.1 Spectral Estimation

The spectrum behaviour of the base-band signals obtained by modulating a random binary sequence can not be obtained by Fourier Transform because it has a non-deterministic time evolution.

Power Spectral Density (PSD) Estimation is hence performed by an average sliding window estimator: Welch's Periodogram [9]. The property that is exploited is the following:

$$S_X(f) = \mathcal{F}\{R_X(\tau)\} = \mathcal{F}\{\mathbb{E}[x(\tau) * x(t + \tau)]\} \quad (22)$$

By considering  $N_{win}$  portions of signal with duration  $T_w$  called window, it is possible to estimate its PSD, from:

$$S_{X_{T_w}}^{(i)}(f) = \frac{|X_{T_w}^{(i)}(f)|^2}{T_w} \quad (23)$$

This is applied to each portion of the signal, by shifting the window and averaging the overall result.

$$\hat{S}(f) = \frac{1}{N_{win}} \sum_{i=1}^{N_{win}} S_{X_{Tw}}^{(i)}(f) \quad (24)$$

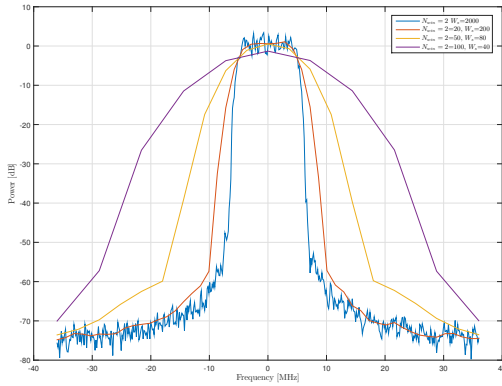
The Welch's Periodogram is affected by bias due to the windowing operation. From Signal Theory, windowing in time means convolving the PSD of the signal by the PSD of the window. Since PSD of the window is not impulsive over the frequency axis, but usually has side-lobes, the true value of the signal PSD is altered.

$$\hat{S}_X(f) = S_X(f) * W_{Tw}(f) \quad (25)$$

By choosing an opportune window  $W_{Tw}(f)$ , the effect of side-lobes may be mitigated [5]. The estimator has a variance:

$$\sigma_{\hat{S}}^2 \propto \frac{1}{N_{win}} \quad (26)$$

Working in discrete domain, since  $T_w = NT_c$ , the spectral resolution of the periodogram is  $1/T_w = 1/NT_c$ .



**Figure 25.**  $\hat{S}_X(f)$  over samples sequence of length  $N_{TotSym} \cdot N_s = 4000$  with  $N_{TotBit} = 1000$ , Blackman window.

As shown in Figure 25, in order to have sufficiently low variance and good spectral resolution, a large number of wide windows is needed. Once the bit sequence length  $N_{TotBit}$  is fixed, there is a trade-off between two properties.

One thing that could be done is to use overlapping between windows, so that the variance decreases. Since the length of the simulation can be set, this is not necessary.

The drawback of having a long sequence is that the simulation lasts for a long time.

In this simulation the bit sequence length is  $N_{TotBit} = N \cdot N_{win}$ , chosen in such a way that the Estimator's properties of spectral resolution and variance can be properly set.

The simulator executes the PSD Estimation by applying the Welch's periodogram `pwelch()`, with Blackman shaped window, then defines a digital frequency axis over the set  $f_d \in [-1/(2T_c), 1/(2T_c)]$ , then shifting the digital spectrum with the command `fftshift()`, (Appendix.17).

#### 4.12.2 Semi-Analytic BER Estimator

The  $P_b(e)$  Estimator provides an estimate of bit error probability working on received signal samples versus  $\eta = E_b/N_0$ . In a real system, received signal samples are affected by complex noise samples distributed as  $\mathcal{N}_c(0, \sigma_n^2)$ .

In this simulation, in order to obtain  $P_b(e)$  curves, instead of generating additional complex noise samples with variance proportional to  $N_0$  for each value of  $E_b/N_0$ , such probability is evaluated semi-analytically by using noiseless received signal samples. The main idea behind the estimator is to compute an estimation for each value of  $\sigma_n^2$ , see [1].

$$\hat{P}_s(e|\sigma_n^2) = \frac{1}{N_{TotSym}} \sum_{i=1}^{N_{TotSym}} P_s(e|S_i, \sigma_n^2) \quad (27)$$

**4-QAM Error Probability Estimator** Since the simulation implements Gray Labelling and considering 4-QAM constellation as a couple of 2-PAM constellations in quadrature, Voronoi Decision Rule constituent constellations can be used. From this, we get that:

$$P_b(e|S_i, \sigma_n^2) = \frac{1}{4} \operatorname{erfc} \left( \frac{|S_{I,i}|}{\sqrt{2\sigma_{I,n}^2}} \right) + \frac{1}{4} \operatorname{erfc} \left( \frac{|S_{Q,i}|}{\sqrt{2\sigma_{Q,n}^2}} \right) \quad (28)$$

The Energy per bit  $E_b = P_x T_b$ , where  $P_x$  is the average power of the signal before it is filtered at the receiver.

The noise power, hence its variance is amplified by the filtering operation at the receiver side:  $\sigma_n^2 = N_0 \cdot \mathcal{E}(h_R)$ .

Where  $\mathcal{E}(h_R)$  is the energy of the filter.

From these considerations, we have that:

$$\sigma_{I,n}^2 = \sigma_{Q,n}^2 = \frac{P_x \cdot T_b \cdot \mathcal{E}(h_R)}{2 \cdot (E_b/N_0)} \quad (29)$$

That is a compact way of obtaining an estimation of  $P_b(e)$  as a function of  $E_b/N_0$ .

The simulator runs exactly the reported set of computations in the function `BERComputation_4QPSK()`, (Appendix.18).

#### 16-QAM Error Probability Estimator

As reported in Section 4.11, Voronoi Decision Regions are defined by evaluating the values of threshold  $S_\tau$  from function `threshold16QAM()` (Appendix.14). Such values are used to evaluate the semi-analytic  $P_b(e)$  by reducing I/Q samples to their respective Real and Imaginary coordinates so that they belong to a couple of 4-PAM constellations in quadrature. In order to evaluate the performance of the constituent constellations, their absolute value are taken, so that they are wrapped into symbol sets of two points. Then, according to the position of each point with

respect to the previously defined threshold, probability of error for  $S_{I,i} = \text{Re}\{S_i\}$  is evaluated as follows:

$$P_s(e|S_{I,i}, \sigma_n^2) = \begin{cases} \frac{1}{2} \text{erfc} \left( \frac{|S_{I,i} - S_\tau|}{\sqrt{2\sigma_{I,n}^2}} \right) & \text{if } |S_{I,i}| > S_\tau \\ \frac{1}{2} \text{erfc} \left( \frac{|S_{I,i} - S_\tau|}{\sqrt{2\sigma_{I,n}^2}} \right) + \frac{1}{2} \text{erfc} \left( \frac{|S_{I,i}|}{\sqrt{2\sigma_{I,n}^2}} \right) & \text{else} \end{cases} \quad (30)$$

The same is done for  $S_{Q,i} = \text{Im}\{S_i\}$ . In the end it is evaluated the intersection of the two error probability events:

$$\begin{aligned} P_b(e|S_i, \sigma_n^2) &= P_s(e|(S_{I,i}, S_{Q,i}), \sigma_n^2) \\ &= P_s(e|S_{I,i}, \sigma_n^2) + P_s(e|S_{Q,i}, \sigma_n^2) \\ &\quad - P_s(e|S_{I,i}, \sigma_n^2) \cdot P_s(e|S_{Q,i}, \sigma_n^2) \end{aligned} \quad (31)$$

Where  $\sigma_n^2$  has the same value as previously expressed for 4-QAM. The simulator runs exactly the reported set of computations in the function `BERComputation_16QAM(.)`, (Appendix.19).

The semi-analytic estimators just modelled are unbiased:

$$\mathbb{E}\{\hat{P}_b(e|\sigma_n^2)\} = \frac{1}{N_{TotSym}} \sum_{i=1}^{N_{TotSym}} \mathbb{E}\{\hat{P}_b(e|S_i, \sigma_n^2)\} = P_b(e|\sigma_n^2) \quad (32)$$

Considering  $p = \mathbb{E}\{\hat{P}_b(e|\sigma_n^2)\}$  and  $\Delta p = (p_{max} - p_{min})$ , the variance of the estimator is:

$$\sigma_p^2 = \frac{p \cdot \Delta p}{N_{TotSym}} \quad (33)$$

The normalized variance is hence:

$$\alpha_p = \frac{\sigma_p^2}{p^2} = \frac{p \cdot \Delta p}{N_{TotSym} \cdot p^2} \quad (34)$$

As general convention, the results are reported with estimations having  $\alpha_p < 0.01$ .

This leads to set simulations where  $N_{TotSym} > 100 \cdot (1 - p)/p$ .

#### 4.12.3 Bit Error Rate Counter

In order to inspect the consistency of Semi-Analytic  $P_b(e)$  Estimations, the simulation implements an empirical error counting technique which evaluates the number of errors when  $E_b/N_0$  varies. As for the real system model, since the noise is additive, complex Gaussian noise samples are added to the received sequence.

The noise samples generation is done by considering that the Noise Power over the whole bandwidth  $B = [-1/2T_c, 1/2T_c]$  (that can be inspected at sampling time  $T_c$ ), is equal to its variance:  $\sigma_n^2 = 2N_0/T_c$ , see [1].

$$\sigma_{I,n}^2 = \sigma_{Q,n}^2 = \frac{P_x \cdot T_b}{2 \cdot T_c \cdot (E_b/N_0)} \quad (35)$$

The received bit sequence is obtained by the demodulation of noisy samples. Since at this stage the signal transient has been removed, the demodulated bit sequence is shorter than the original one. To remedy to this issue, the two sequences have to be aligned by removing the equivalent number of bits to the original one.

As reported in Section 4.9 the transient removal operation always involves the first symbol, the equivalent number of bits of misalignment is the number of bits per symbol  $N_b$ .

Once the bit sequence at the receiver is computed, the number of bits in error is obtained by summing bit per bit the two sequences modulo 2. The number of ones occurring in the sum will be the number of bits in error. The ratio between the number of bits in error and the total number of useful bits received is the so called Bit Error Rate.

Since, for quite large values of  $E_b/N_0$  the probability of error is almost low, measuring low probabilities requires very long sequences of bits to be generated. Since large simulations take a long time and given that counting a few number of errors with low probability introduces high variance in the BER counting estimation, the simulation uses a value of threshold to stop when the number of counted errors with respect to the length of the generated sequence is too low.

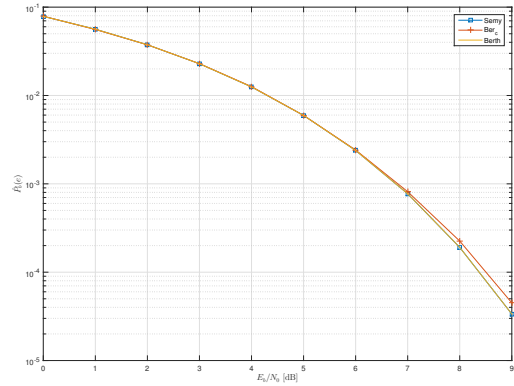


Figure 26. Theoretical  $P_b(e)$  and  $\hat{P}_b(e)$  Estimators.

In Figure 26 is shown the performance of both the 4-QAM Semi-Analytic and the BER Counter Estimators with respect to the theoretical one.

Both the Estimators behave good when compared to the theoretical curve. As expected, for higher values of  $E_b/N_0$ , the Estimation obtained by Error Counting does not match exactly to the theoretical one.

The variables which are used to store the parameters relative to performances are the following:

Both in case of 4-QAM modulation (Appendix.20) and in the case of 16-QAM (Appendix.21), the simulator implements the Bit Error Rate Counting procedure by generating complex noise with variance given by (35), by multiplying it by unitary complex gaussian random samples.



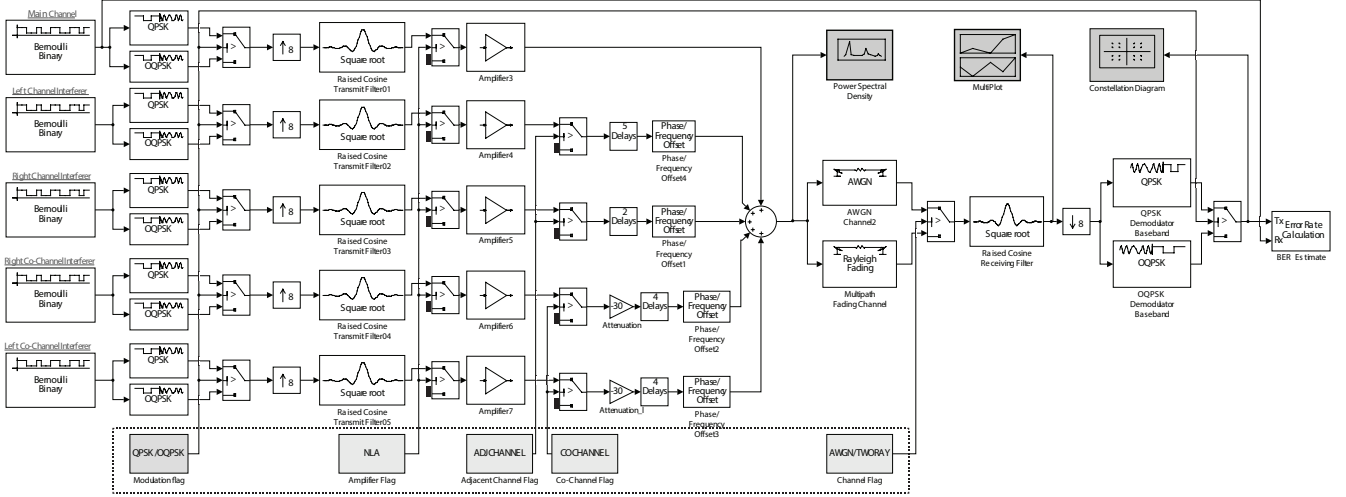


Figure 27. Multi Carrier System block diagram

### Performance Parameters

Alpha	Vector containing values of $\alpha$
Rho	Vector containing values of $\rho$
Bkoff	[dB] Vector containing values of $\beta_{in}$
BER_Target	Target BER the system has to achieve
F	[dB] Cost Function Values
F_linear	Linear Values of Cost Function
EbNo	[dB] $E_b/N_0$ Values for which BER is computed
EbNo_linear	Linear Values of $E_b/N_0$
Min_Counted_Errors	Minimum number of errors to count
Optimize_active	Running optimization or Loading workspace

## 5. Multi Carrier System

The real system analyzed is the one shown in Figure 27, where a set of independent channels, described previously as Single Carrier Systems are combined together to simulate a frequency division transmitting system where three frequency slots are used by three independent transmitters. Since the utilization of cross-polarization channels is frequently adopted to increment the capacity of the channel in radio-relay links, the simulation takes into account additional residual interference induced by co-channels. The purpose of the analysis is still focused on the central frequency slot, where the useful channel is located even if the coexistence of multiple signals requires a better setting of operational parameters in order to reach given target performances.

## 6. Multi Carrier System Implementation

The implementation of complex multi carrier system is realized starting from the basic single channel model by adding specific features to additional signal given by specification. The following section provides a description of the effects of interest induced by blocks previously described in Section 4.

### 6.1 Simulation Parameters

In order to simulate the behaviour of our multi carrier communication systems, a new set of parameters modelling the available bandwidth utilization have been adopted.

#### Adjacent Co-Channel Parameters

Df_adj	= $\Delta f$	[13 MHz] Frequency shift for side channels
Df_coch	= $\Delta f/2$	[7.5 MHz] Frequency shift for Co-channels
Coch_att	= $A_{coch}$	[30 dB] Attenuation of Co-channel
Delay_r	= $d_r$	[2 samples] delay for right Adjacent Channel
Delay_l	= $d_l$	[5 samples] delay for left Adjacent Channel
Delay_coch	= $d_{coch}$	[1 samples] delay for Co-Channels
Adjacent_active		Activates / Deactivates Adjacent channels
Cochannel_active		Activates / Deactivates Co-channels

### 6.2 Adjacent Channels

It is assumed to observe the closest lateral frequency slots where two other transmitter are sending data. Side frequency slots, also referred as Adjacent channels, are equally spaced around the central frequency  $f_0$  of the useful channel of  $\pm\Delta f$  as reported in Figure 28.

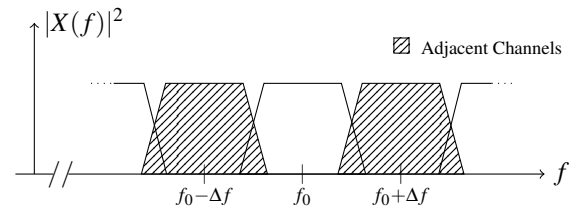


Figure 28. Adjacent Channels

In order to simulate Adjacent Channels, new independent random sequences of bits are generated. The amplified signals for the transmission are obtained in the same way as for the Single Carrier system, after power normalization. Even if the adjacent channels' signals come from the same device,



their time synchronization is not ensured. This is modelled by adding a relative time delay to each signal (Appendix.22). Since the system works in digital domain, the time delays are expressed in terms of signal samples. The outcoming signals are then shifted in frequency as follows:

$$\begin{aligned} x_r(t) &= x_{r,0}(t - d_r Tc) \cdot e^{j2\pi\Delta f t} \\ x_l(t) &= x_{l,0}(t - d_l Tc) \cdot e^{-j2\pi\Delta f t} \end{aligned} \quad (36)$$

where  $t$  is properly defined by scaling it on the previously introduced sampling time  $Tc$ . The implementation executes exactly the same operations (Appendix.24).

### 6.3 Cross-Polarization Co-Channel Interference

Cross-polarization co-channels usually do not affect respective reciprocal channels, but since the correct projection of the orthogonal polarization cannot be performed perfectly, due to polarization imperfect polarization filtering, some power from their spectrum is added to reciprocal channel.

#### 6.3.1 In-Band Co-Channel Interference

Two Co-channel interferers are implemented by generating i.i.d. random sequences, mapping them into waveforms, adding time delays, attenuating of  $A_{coch}$ , then shifting their spectral components by  $\pm\Delta f/2$  as reported in Figure 29. Since it is not supposed that the single components of the channel are synchronous, the filtered sequences are delayed (Appendix.23).

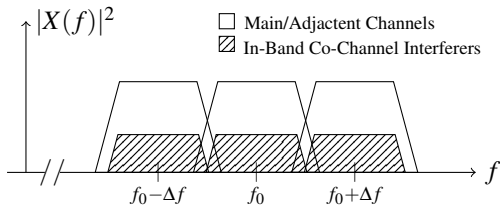


Figure 29. In-Band Co-Channels

The expression of the Co-Channel interferers is the following:

$$\begin{aligned} x_{r,coch}(t) &= x_{r,coch,0}(t - d_{coch} Tc) \cdot A_{coch} \\ x_{l,coch}(t) &= x_{l,coch,0}(t - d_{coch} Tc) \cdot A_{coch} \end{aligned} \quad (37)$$

As reported in Figure 31a, by excluding NLA, in order not to have any additional distortion, when the Co-Channel interferences are centred in the band of the useful signal, sub-constellations appear around the points of the main constellation. This due to the addition of attenuated samples of interference signals.

#### 6.3.2 Interleaved Co-Channel Interference

In order to limit this effect, an Interleaved Structure is usually adopted: Co-channels are so shifted in frequency in order to have a misalignment of their lobes respect to the cross-polarization channels' main lobes, as reported in Figure 30.

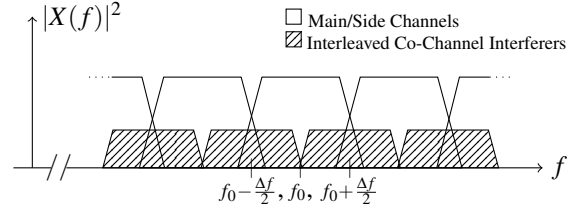


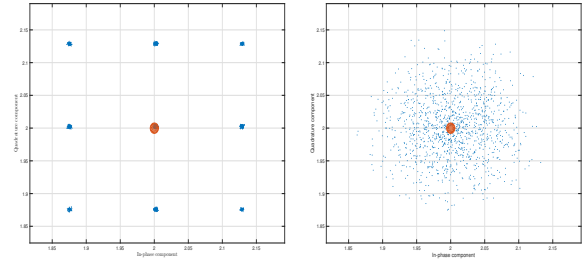
Figure 30. Interleaved Co-Channels

In this case, the expression of the Co-Channel interferers is the following:

$$\begin{aligned} x_{r,coch}(t) &= x_{r,coch,0}(t - d_{coch} Tc) \cdot A_{coch} \cdot e^{j\pi\Delta f t} \\ x_{l,coch}(t) &= x_{l,coch,0}(t - d_{coch} Tc) \cdot A_{coch} \cdot e^{-j\pi\Delta f t} \end{aligned} \quad (38)$$

The implementation employs this Co-Channel model to consider the impairment (Appendix.25).

As reported in Figure 31b, due to time misalignment and frequency shift, the in-band portion of Co-channels' spectrum causes an additional distortion caused by a not exactly matched filtering.



(a) Non-Delayed In-Band. (b) Delayed Interleaved.

Figure 31. Effects of Co-Channel Interference.

In real systems, the channel behaves as an adder for the different signals, with its own impulse and frequency responses.

In the simulation the two operations are made separately: All the signals are summed together, the sum is filtered with the impulse response of the channel.

In Figure 32 is reported the effect of both Adjacent and Co-Channels over the main signal. The effect of adjacent band slots occupied by signals with the same amplitude of our channel, if  $\Delta f$  is not large enough, is very strong over our useful signal's PSD. The effect of co-channels, since they are attenuated very much, is not a big issue for our useful channel.

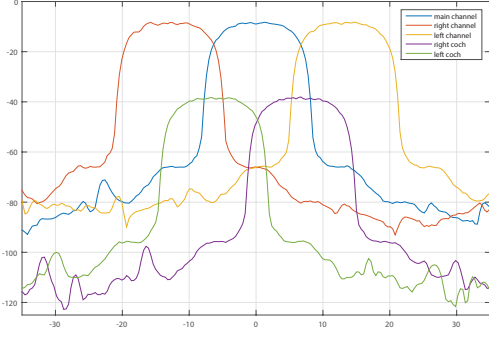


Figure 32. Overall Power Spectral Density

## 7. Optimization Stage

This is the main target that the overall simulation has to achieve.

The goal is to set the parameters used by the communication system in a convenient way, so that the all works in optimal conditions. In order to do this, let us consider the following parameters of interest:

- $\rho$  roll-off factor of the root-raised cosine shaping filter;
- $\alpha$  splitting factor of the transmitter's shaping filter (the receiver one has factor  $1 - \alpha$ );
- $\beta_{in}$  back-off factor of the Non Linear Amplifier.

By setting different values of such parameters, the system has different performances.

The aim of the optimization is to set them in such a way that they minimize the required  $E_b/N_0$  to reach a given target  $BER_t$ .

If the system has signal amplification, they have to be set in order to minimize what is defined as Cost Function.

### 7.1 Cost Function

As previously described, the NLA provides linear power amplification up to a given threshold of saturation called  $P_{out,sat}$ . When the optimization is performed over a system supplied by NLA, it aims minimize the value assumed by the Cost Function:

$$F(\alpha, \rho, \beta_{in}) = \frac{E_b}{N_0} \Big|_{dB} + \beta_{out} \quad (39)$$

The value  $F$  is a measure of the amount of energy to be spent to achieve a given  $BER_t$ , by taking into account the amount of power that is wasted by amplifying the signal in near-saturated regime.

The value of Cost Function at target BER  $F$  is computed easily by performing Cost Function Evaluation by using Semi-Analytical BER Computation as reported in Section 4.12.2, where, instead of using values of  $E_b/N_0$  are used the values of  $F$ , instead of using  $P_x$  are used the values of  $P_{out,sat}$ .

This is implemented by opportune functions both in 4-QAM (Appendix.26) and in 16-QAM (Appendix.27).

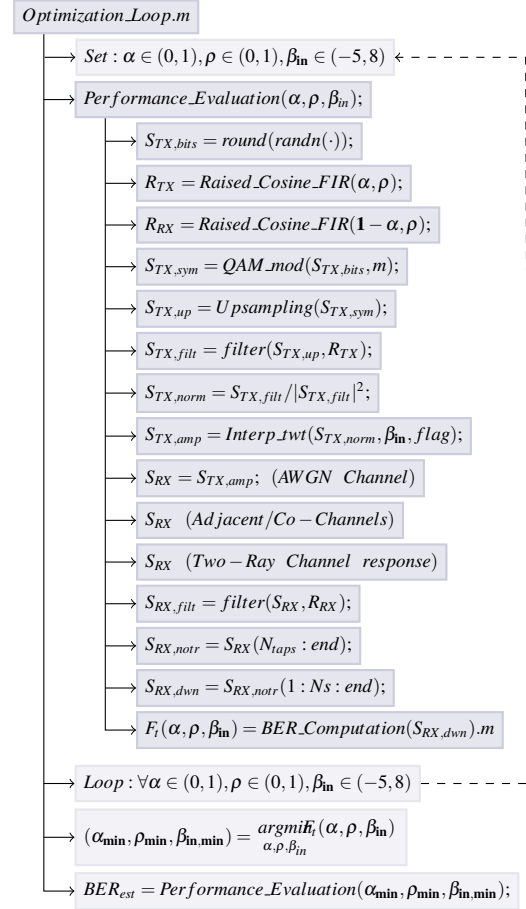


Figure 33. Scheme block for the OptimizationLoop function.

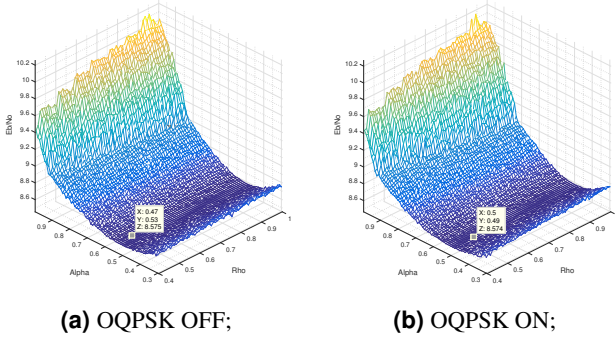
### 7.2 Optimization Implementation

The optimization of the radio relay link design is implemented in Matlab by building an appropriate routine, whose main structure is shown in Figure 33.

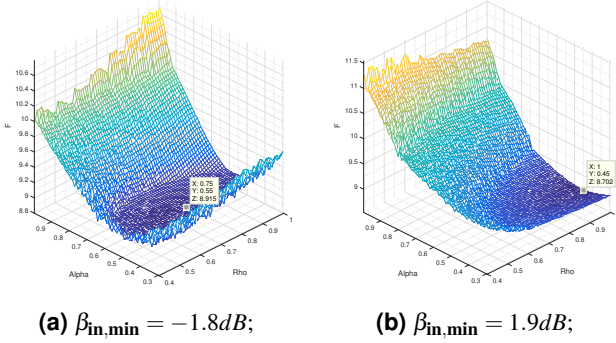
The variables which are used to set the system impairments are the following:

Impairments Flags	
OQPSK_active	Activates/Deactivates Offset QPSK
NLA_active	Activates/Deactivates NL Amplifier
AdjacentCoch_active	Activates/Deactivates Adjacent and Co-channels
TwoRay_active	Activates/Deactivates Two-Ray Ground Channel
Cost_active	Sets $E_b/N_0$ or Cost Function evaluation
Plots_active	Activates/Deactivates Plots
Error_counter_active	Activates/Deactivates Error Counter

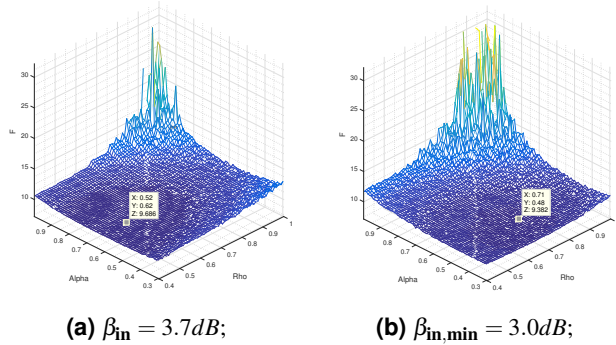
The purpose of the function *OptimizationLoop.m* is to run many times the routine *Performance\_Evaluation\_function.m* which evaluates the values assumed by the  $E_b/N_0$  or by the Cost Function when the system parameters  $\alpha, \rho, \beta$  vary in pre-defined ranges and save them into a 3D matrix.



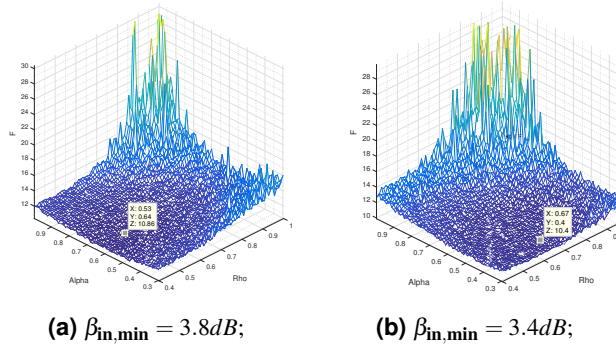
**Figure 34.** 4-QAM Optimization without impairments.



**Figure 35.** 4-QAM: Optimization surface with NLA



**Figure 36.** 4-QAM: NLA, Adjacent Channels, Co-channels.



**Figure 37.** 4-QAM: NLA, Adjacent, Co-channels, Two-Ray.

The routine *Performance\_Evaluation\_function.m* runs a complete system simulation, by using flags which activates or de-

activates the impairments that are supposed to be tested, hence with different elementary or advanced system configurations.

At the end of the Loop, the function gives back:

#### Output of Optimization Loop

$\alpha_{min}$	Optimal value of $\alpha$
$\rho_{min}$	Optimal value of $\rho$
$\beta_{in,min}$	Optimal value of $\beta_{in}$
$F_{min}$	[dB] Cost Function Optimal Value
$(E_b/N_0)_t$	[dB] $E_b/N_0$ at Target BER with optimal parameters
$BER_{est}$	Semi-Analytical BER at optimal parameters
$BER_{count}$	Counted BER at optimal parameters

Optimal output values are finally used to evaluate the performance of the system by setting different sets of impairments.

At the end of the whole analysis, losses due to each single impairment are evaluated in terms of difference between  $(E_b/N_0)_t$  with respect to the theoretical one.

The behaviour of Cost Function surface when  $\beta_{in} = \beta_{in,min}$ , by letting  $\alpha$  and  $\rho$  vary for 4-QAM modulation scheme is reported in the following Figures: 34, 35, 36, 37 by considering different sets of impairments.

For each of them, (a) reports results where Offset QPSK is not active and (b) where it is active. As a general trend, the values assumed by the Cost Function is lower after optimization. As reported in 4.3.2, adding Offset leads to lower values of PAPR, which means a reduction on the dynamics of the signal.

## 8. Results

The results of the performance evaluation and optimization stages are obtained by considering  $BER_{Target}=10^{-4}$  given from specifications.

The system configurations taken in exam are the following:

- 4-QAM QPSK/OQPSK modulations with NLA, Adjacent, Co-Channels;
- 4-QAM QPSK/OQPSK modulations with NLA, Adjacent, Co-Channels, Two-Ray Channel;
- 16-QAM QPSK/OQPSK modulations with NLA, Adjacent, Co-Channels.

The results are reported respectively in Table 1, 2, 3.

### 8.1 Optimum System Parameters

The simulation and performance evaluation of Cost Function is performed to find optimal values of  $\alpha$ ,  $\rho$ ,  $\beta_{in}$ .

As a general observation, when many impairments are involved, it is difficult to inspect the effect of each single optimization parameter, because the value assumed by  $F_{min}$  is the result of a non linear combination of all of them.

This can be experienced by looking at Figure 49 where is reported an example of the behaviour of  $F(\alpha, \rho, \beta_{in,min})$ .

4-QAM		
	QPSK	OQPSK
$F$ [dB]	9.69	9.38
$\alpha$	0.62	0.48
$\rho$	0.52	0.71
$\beta_{in}$ [dB]	3.70	3.00

**Table 1.** Optimal values with NLA, Adjacent Channels, Co-channels.

4-QAM		
	QPSK	OQPSK
$F$ [dB]	10.86	10.40
$\alpha$	0.64	0.40
$\rho$	0.53	0.67
$\beta_{in}$ [dB]	3.80	3.40

**Table 2.** Optimal values with NLA, Adjacent Channels, Co-channels, Two-Ray Path Channel.

16-QAM		
	QPSK	OQPSK
$F$ [dB]	16.19	15.94
$\alpha$	0.55	0.41
$\rho$	0.53	0.56
$\beta_{in}$ [dB]	7.90	7.60

**Table 3.** Optimal values with NLA, Adjacent Channels, Co-channels.

### 8.1.1 Considerations about $\beta_{in}$

A specific remark may be done on the values of minimum cost related to the effects of NLA. It depends on  $\beta_{out}$ , which is determined by  $\beta_{in,min}$ , selected according to the signal dynamics as referred in Section 4.6.

By considering the case of multi-level constellation formats, such as 16-QAM, the modulated signal has a wider dynamic with respect to the single-level ones.

This implies a higher value of  $\beta_{in}$  in order to cover a wider linear zone of NLA I/O characteristic curve that leads to a higher value of  $F_{min}$ .

For low values of  $\beta_{in}$ , as expected from Section 4.6 the amplifier works close to saturation zone, so it has a good power efficiency, but this implies higher signal distortion.

### 8.1.2 Considerations about $\rho$

Referring to what is reported in Section 4.5, the value of  $\rho$  has a direct impact over the bandwidth occupancy  $B = (1 + \rho)/2T_s$ .

When  $\rho$  is high, the frequency response of the shaping filter is wider and it induces a higher inter-carrier interference. Besides this, smaller bandwidth occupancy means less Noise Power at the receiver, due to the integration of Noise PSD over a narrower support.

By the case in which  $\rho$  is low, oscillations in signal's impulse response are so high that they induce enlargement of instantaneous power leading to ISI and possible distortion especially

after the amplification stage.

By considering that  $\rho$  affects signal dynamics, it is possible to observe its effect on PAPR. As reported in Figure 50, when Offset QPSK is active, the curve of PAPR is monotonic decreasing, whereas when Offset QPSK is not active, it has a change in its slope and assumes a value of minimum.

### 8.1.3 Considerations about $\alpha$

Without considering any impairment, both in the case where  $\alpha$  is high, and where it is low, the filtered signal has a higher PAPR with respect to the case where it has medium value, as it can be seen in Figure 50.

After the computation of the Optimal values  $\alpha_{min}, \rho_{min}, \beta_{in,min}$ , the performances of the system are evaluated using those parameters.

## 8.2 Analysis of Losses

Using optimal parameters, curves of  $\hat{P}_b(e)$  are finally evaluated Semi-Analytically. They are interpolated in order to obtain  $(E_b/N_0)_t$  when working at target BER.

In Figures 38, 39, are reported the curves for 4-QAM modulation schemes, working with parameters in Table 1. Respectively, they are relative to the the case of Offset QPSK active or inactive.

In Figures 40, 41, it is considered Two-Ray Path Channel model, with optimal parameters from Table 2.

In Figures 42, 43 are reported the curves for 16-QAM modulation schemes when the optimal parameters are the ones in Table 3.

The impairments losses can be evaluated as the difference between each  $(E_b/N_0)_t$  and the theoretical one.

The received constellations of signal samples altered by additive noise with zero mean and variance discussed in Section 4.12.3, and relative to the discussed configurations are reported in Figures 44, 45, 46, 47, 48.

They provide an experimental view of the behaviour of the system keeping fixed the value of  $(E_b/N_0)_t$  computed when all the impairments are involved.

By removing each of them, it is possible to appreciate, for each constellation, the BER decreases as a function of minimum distance of scattering diagram points.

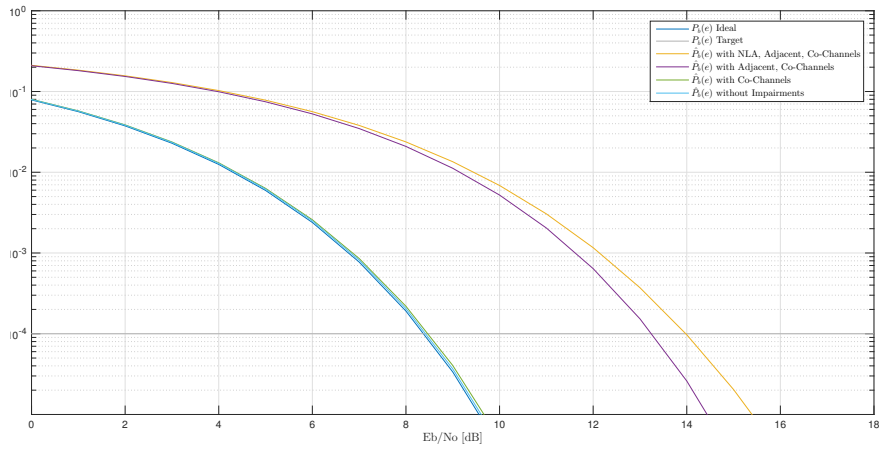
The most evident reason for losses ( $\simeq 5$  dB) is the presence of Adjacent Channels since they work with the same power of the useful signal and induce interference in its band.

In general, this channel model has to take into account ISI effect induced by Two-Ray propagation. This may be either constructive or destructive, but, on the average, system performances are worse [8]. Some minor losses are induced by Two-Ray channel model ( $\simeq 1$  dB), such that in Figure 18, it possible to notice that their impact is not that significant.

Since the optimization stage suggests to choose a value of  $\beta_{in}$  such that NLA works in almost linear zone, its effect in terms of loss is generally reduced.

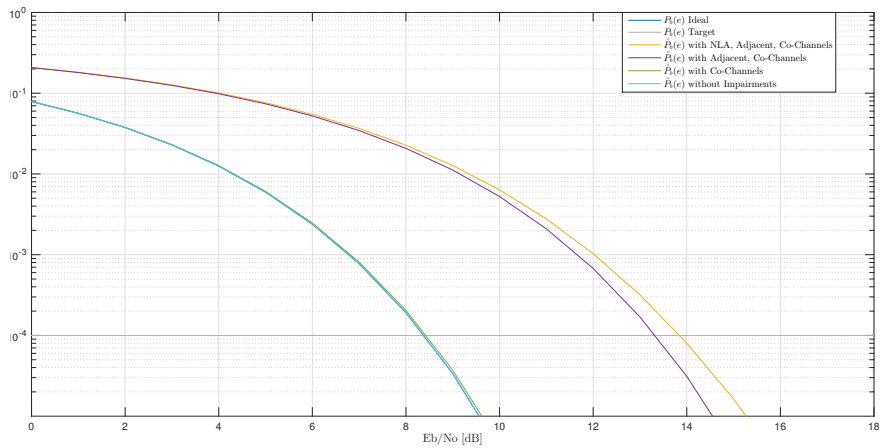
The impact of co-channel interferers is almost weak because of the high attenuation and due to the adoption of Interleaved Structure.





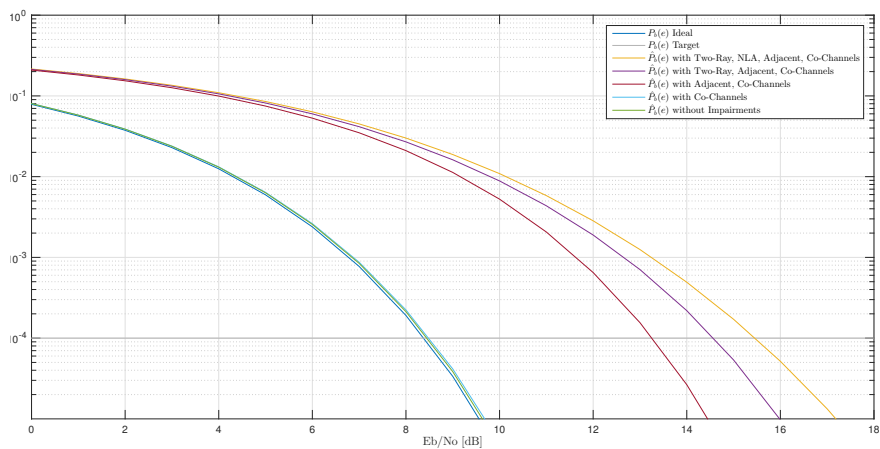
**Figure 38.** 4-QAM WITHOUT OFFSET Loss analysis: NLA, Adjacent, Co-Channels.

System Impairments		$(E_b/N_0)_t$
NLA Adjacent Cochannels		13.98 [dB]
Adjacent Cochannels		13.42 [dB]
Cochannels		8.67 [dB]
Model		8.62 [dB]
Ideal		8.38 [dB]
System Impairments		Losses
NLA Adjacent Cochannels		5.60 [dB]
Adjacent Cochannels		5.03 [dB]
Cochannels		0.28 [dB]
Model Loss		0.24 [dB]



**Figure 39.** 4-QAM WITH OFFSET Loss analysis: NLA, Adjacent, Co-Channels.

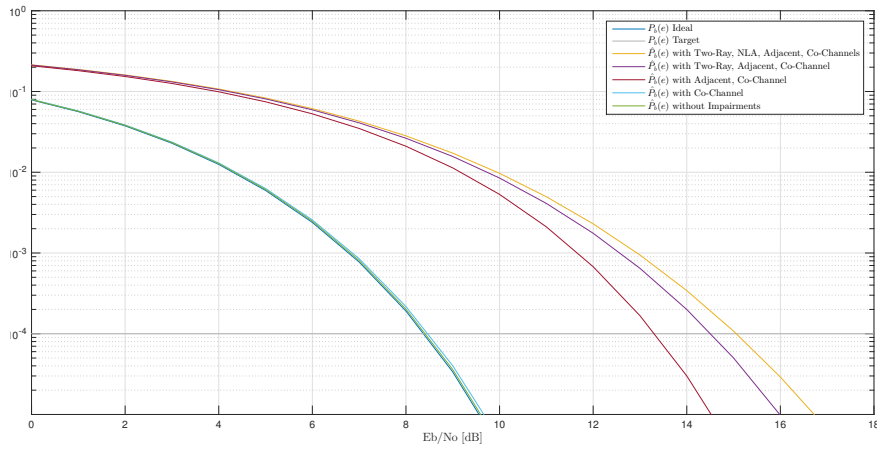
System Impairments		$(E_b/N_0)_t$
NLA Adjacent Cochannels		13.90 [dB]
Adjacent Cochannels		13.49 [dB]
Cochannels		8.62 [dB]
Model		8.58 [dB]
Ideal		8.38 [dB]
System Impairments		Losses
NLA Adjacent Cochannels		5.52 [dB]
Adjacent Cochannels		5.11 [dB]
Cochannels		0.24 [dB]
Model Loss		0.19 [dB]



**Figure 40.** 4-QAM WITHOUT OFFSET Loss analysis: Two-Ray, NLA, Adjacent, Co-Channels.

System Impairments		$(E_b/N_0)_t$
NLA Two-Ray Adjacent Cochannels		15.64 [dB]
Two-Ray Adjacent CoChannels		14.74 [dB]
Adjacent Cochannels		13.44 [dB]
Cochannels		8.67 [dB]
Model		8.62 [dB]
Ideal		8.38 [dB]
System Impairments		Losses
NLA Two-Ray Adjacent Cochannels		7.25 [dB]
Two-Ray Adjacent Cochannels		6.35 [dB]
Adjacent Cochannels		5.06 [dB]
Cochannels		0.29 [dB]
Model Loss		0.26 [dB]





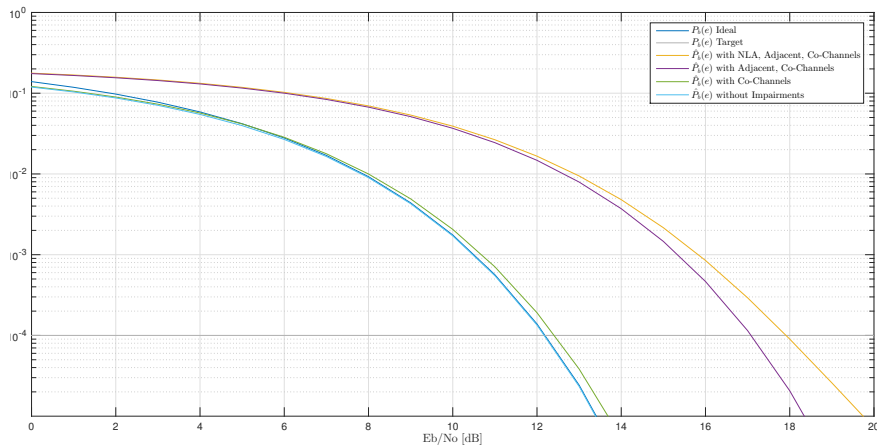
**Figure 41.** 4-QAM WITH OFFSET Loss analysis: Two-Ray, NLA, Adjacent, Co-Channels.

#### System Impairments

NLA TwoRay Adjacent Cochannels	$(E_b/N_0)_t$ 15.07 [dB]
TwoRay Adjacent CoChannels	14.65 [dB]
Adjacent Cochannels	13.48 [dB]
Cochannels	8.62 [dB]
Model	8.58 [dB]
Ideal	8.38 [dB]

#### System Impairments

NLA TwoRay Adjacent Cochannels:	Losses 6.69 [dB]
TwoRay Adjacent CoChannels:	6.27 [dB]
Adjacent Cochannels:	5.09 [dB]
Cochannels:	0.28 [dB]
Model Loss:	0.24 [dB]



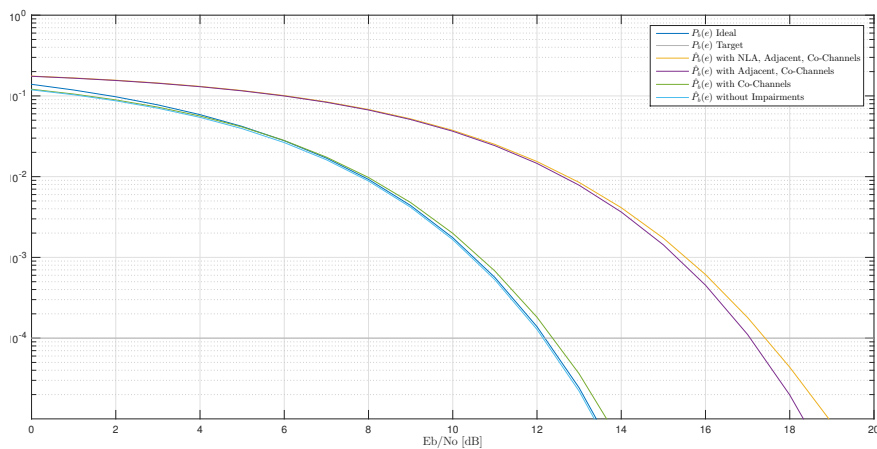
**Figure 42.** 16-QAM WITHOUT OFFSET Loss analysis: NLA, Adjacent, Co-Channels.

#### System Impairments

NLA Adjacent Cochannels	$(E_b/N_0)_t$ 17.96 [dB]
Adjacent Cochannels	17.17 [dB]
Cochannels	12.46 [dB]
Model	12.28 [dB]
Ideal	12.19 [dB]

#### System Impairments

NLA Adjacent Cochannels	Losses 5.78 [dB]
Adjacent Cochannels	4.99 [dB]
Cochannels	0.27 [dB]
Model Loss	0.09 [dB]



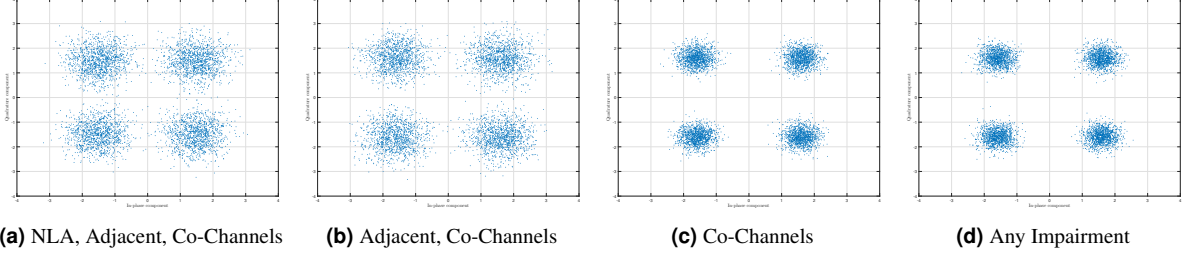
**Figure 43.** 16-QAM WITH OFFSET Loss analysis: Two-Ray, NLA, Adjacent, Co-Channels.

#### System Impairments

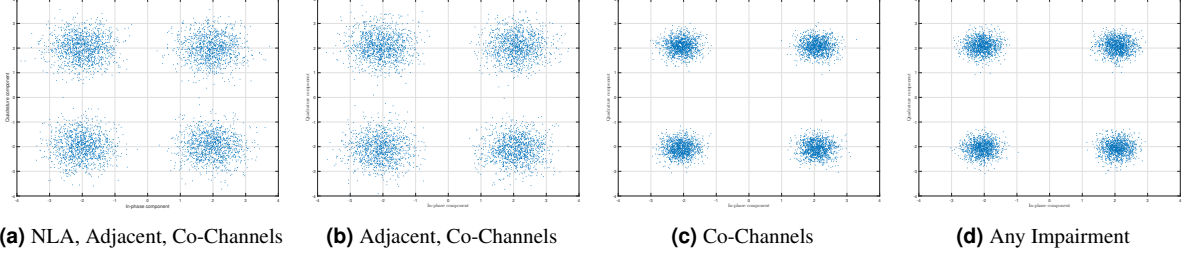
NLA Adjacent Cochannels	$(E_b/N_0)_t$ 17.59 [dB]
Adjacent Cochannels	17.12 [dB]
Cochannels	12.56 [dB]
Model	12.26 [dB]
Ideal	12.19 [dB]

#### System Impairments

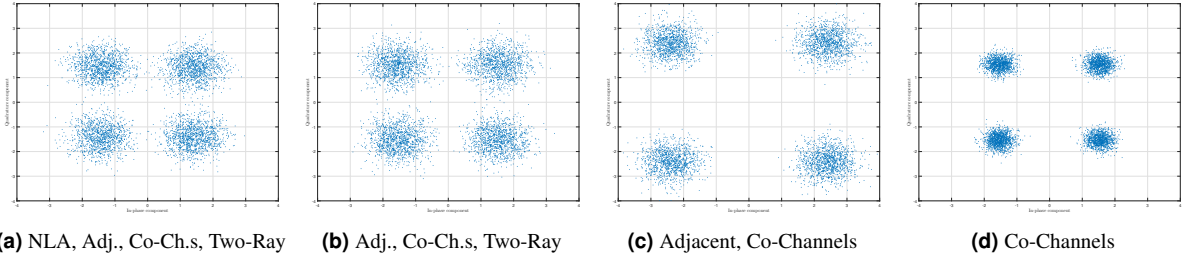
NLA Adjacent Cochannels Loss:	Losses 5.40 [dB]
Adjacent Cochannels:	4.93 [dB]
Cochannels:	0.38 [dB]
Model Loss:	0.08 [dB]



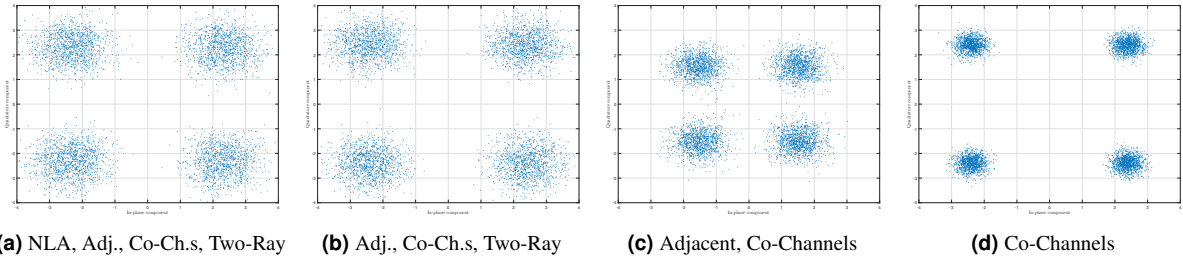
**Figure 44.** 4-QAM Scattering Diagram at the Receiver with  $(E_b/N_0)_t = 13.98$  dB, optimal values from Table 1, column 1.



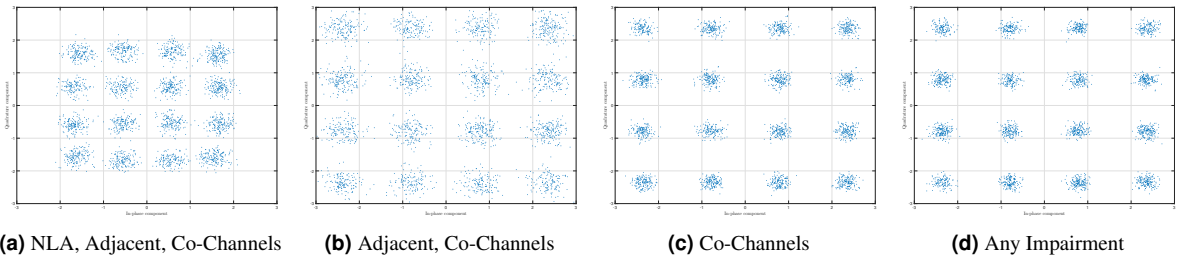
**Figure 45.** 4-OQAM Scattering Diagram at the Receiver with  $(E_b/N_0)_t = 13.90$  dB, optimal values from Table 1, column 2.



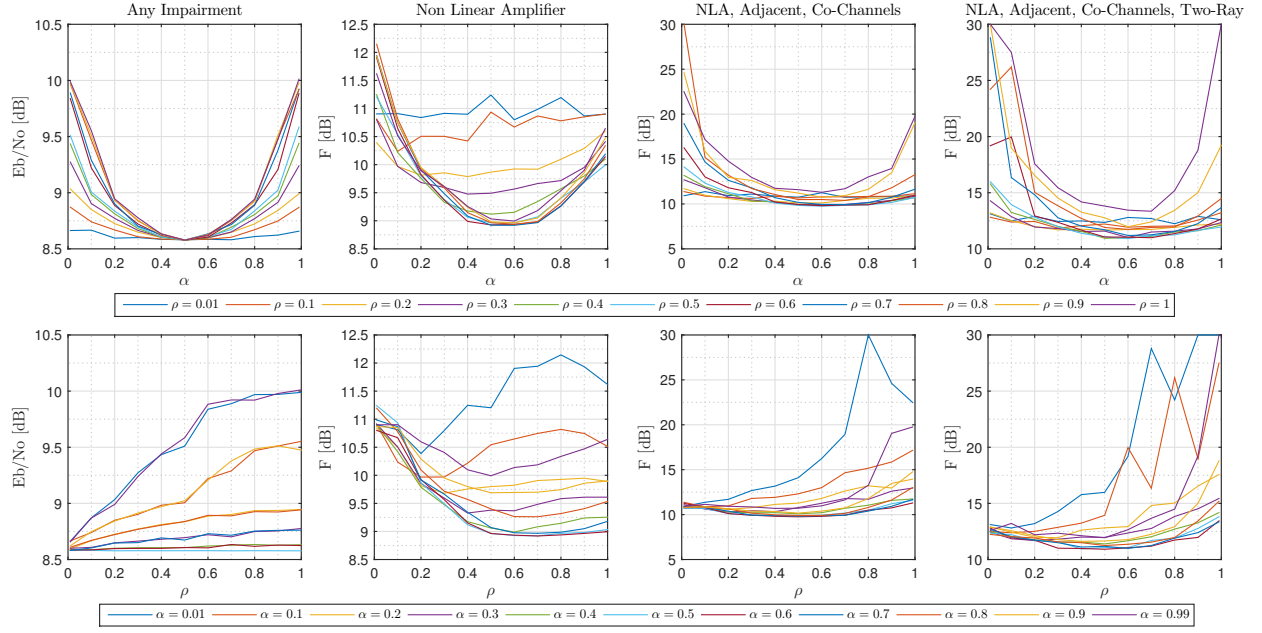
**Figure 46.** 4-QAM Scattering Diagram at the Receiver with  $(E_b/N_0)_t = 15.64$  dB, optimal values from Table 2, column 1.



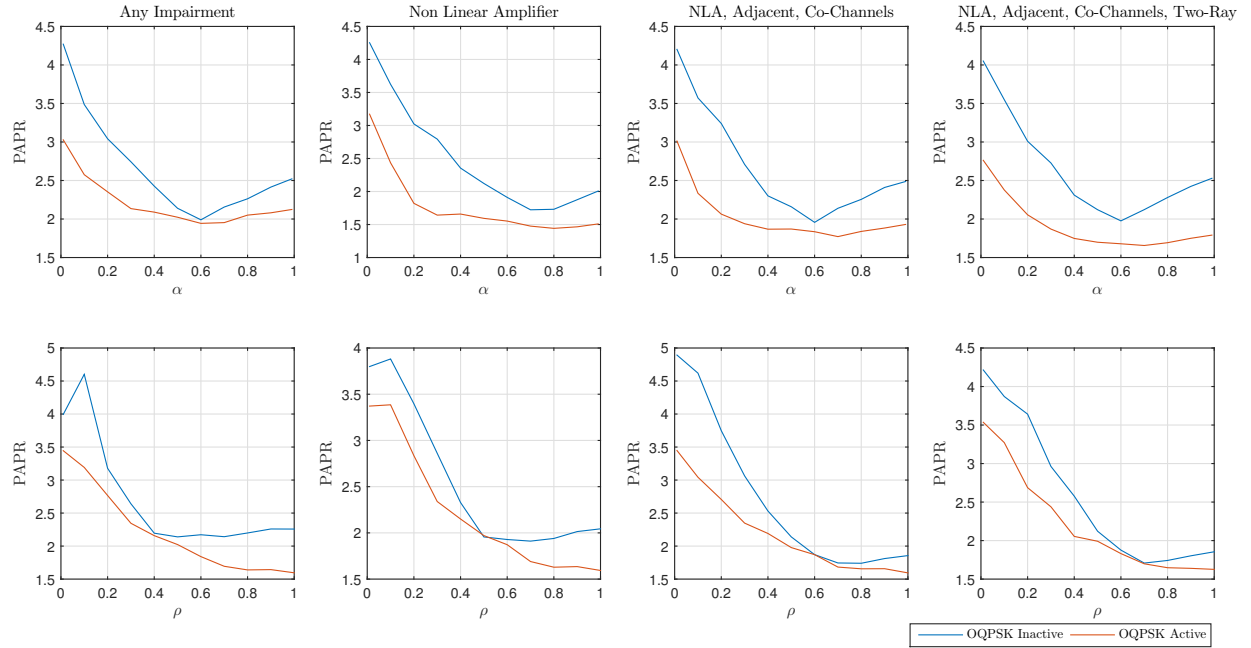
**Figure 47.** 4-OQAM Scattering Diagram at the Receiver with  $(E_b/N_0)_t = 15.07$  dB, optimal values from Table 2, column 2.



**Figure 48.** 16-QAM Scattering Diagram at the Receiver with  $(E_b/N_0)_t = 17.96$  dB, optimal values from Table 3, column 1.



**Figure 49.** Sections of  $F(\alpha, \rho, \beta_{in,min})$  with respect to  $\alpha, \rho$  for 4-QAM when OQPSK is not active.



**Figure 50.** Sections of PAPR curves with respect to  $\alpha, \rho$  for 4-QAM.

## 9. Bibliographical References

- [1] Marina Mondin. Simulation of communication systems, 2014/15.
- [2] Ernesto Conte. *Lezioni di Teoria dei Segnali*. Liguori Editore, 1996.
- [3] Fabio Neri Letizia Lo Presti. *Introduzione ai processi casuali*. CLUT, 1992.
- [4] Ezio Biglieri Sergio Benedetto. *Principles of Digital Transmission With Wireless Applications*. Kluwer Academic / Plenum Publishers, 1999.
- [5] Massimiliano Laddomada Marina Mondin. Elaborazione numerica dei segnali, 2007.
- [6] Galileo Tech. Galileo solid state power amplifier (sspa). Technical report, ESA, 2001.
- [7] Giorgio Taricco. Wireless and optical communications, 2013/14.
- [8] Roberto Garelli. Communication systems, 2014/15.
- [9] Lorenzo Galleani. Statistical signal processing and multimedia, 2013/14.

## 10. Appendix: MATLAB® code

**Listing 1.** Random Sequence Generation Instruction

```
SourceBit = round(rand(1,NTotBit));
```

**Listing 2.**  $2^{N_b}$ -QAM Modulation Function

```
% qpskmod.m
% Function to perform QPSK-QAM mapping
% Programmed by H.Harada
%
function [iout,qout]=qpskmod(paradata,para,nd,m1)
% ***** variables *****
% paradata : input data (para-by-nd matrix)
% iout :output Ich data
% qout :output Qch data
% para : Number of parallell channels
% nd : Number of data
% m1 : Number of modulation levels
% (QPSK ->2 16QAM -> 4)
% *****
m2=m1./2;
paradata2=paradata.*2-1;
count2=0;
for jj=1:nd
    isi = zeros(para,1);
    isq = zeros(para,1);
    for ii = 1 : m2
        isi = isi + 2.^( m2 - ii ) .* paradata2((1:para),ii+count2);
        isq = isq + 2.^( m2 - ii ) .* paradata2((1:para),m2+ii+count2);
    end
    iout((1:para),jj)=isi;
    qout((1:para),jj)=isq;
    count2=count2+m1;
end
%***** end of file *****
```

```
[c_ch, s_ch] = qpskmod(SourceBit,1,NTotSym,Nb);
```

**Listing 3.** Upsampling Function

```
function [c_chUp,s_chUp] = upsampling(c_ch,s_ch,Ns)
c_chUp=zeros(1,Ns*length(c_ch));
s_chUp=zeros(1,Ns*length(s_ch));

c_chUp([1:Ns:end])=c_ch;
s_chUp([1:Ns:end])=s_ch;
end
```

```
[c_chUp,s_chUp] = upsampling(c_ch, s_ch,Ns);
```

Listing 4. Raised Cosine Function

```

function [B,A,t] = RaisedCosineFIR(Rs,Ns,M,Rho,Alpha,Display)
% Purpose.
%   Raised Cosine FIR filter design, by Frequency Sampling method.
% Syntax.
%   [B,A,t] = RaisedCosineFIR(Rs,Ns,M,Rho,Alpha,Display)
% Input Parameters.
%   Rs   : [Real Number] Symbol Rate [Symbols/sec]
%   Ns   : [Integer Number] Number of samples per symbol interval.
%   M    : [Integer Number] The impulse response is truncated over 2*(M+1)
%           symbol intervals, centered around the maximum.
%   Rho  : [Real Number] Rolloff factor.
%   Alpha : [Real Number] Power to which the transfer function is raised.
%   Display : [String] The display mode. The allowed values are:
%             - 'Yes': The impulse response and the transfer function are
%                   plotted.
%             - 'No' : No graphical output are generated.
% Output Parameters.
%   B : [Row Vector 2*M*Ns+1x1] The numerator of the FIR filter.
%   A : [Integer Number] The denominator of the FIR filter (i.e. A=1)
%   t : [Row Vector 2*M*Ns+1x1] The time instant at which the samples of the
%       filter impulse response are taken.
% Toolbox.
%   Array Signal Processing.
% Keywords.
%   Array Signal Processing, Communication, Raised Cosine, FIR Filter.
% Bibliographic References.
% Author.
%   SELLONE Fabrizio - sellone@polito.it
% Version.
%   1.0 - Saturday 10th March 2001 - 11.00 AM

T0 = 1/(Rs*Ns);
T = 2*(M+1)/Rs;
f0 = 1/T;
Bw = 1/T0;
NFFT = 2*(M+1)*Ns;

t = 0:T0:(T-T0);
f = (-Bw/2):f0:(Bw/2-f0);

Ind1 = find(abs(f) <= (((1-Rho)*Rs)/2));
Ind2 = find((abs(f) <= (((1+Rho)*Rs)/2)) & (((1-Rho)*Rs)/2) <= abs(f)));

H = zeros(1,NFFT);
H(1,Ind1) = 1/Rs/T0;
H(1,Ind2) = (1-sin((pi/(Rs*Rho))*(abs(f(Ind2))-Rs/2)))/(2*Rs)/T0;
H = fftshift(H).^Alpha;

B = real(fftshift(iff(H)));
B = [B B(1,1)];
A = 1;
t = 0:T0:T;

if (strcmp(Display,'Yes') == 1)

    figure
    plot(t,B,'o-')
    title('Filter Impulse Response');
    xlabel('Time [s]');
    grid on;
    zoom on;

    N = 4096;
    fDisp = linspace(-Bw/2,Bw/2-Bw/N,N);
    figure;
    plot(fDisp,10*log10(abs(fftshift(fft(B,N)))));
    hold on;
    AX = axis;
    Ind1 = find(abs(fDisp) <= (((1-Rho)*Rs)/2));
    Ind2 = find((abs(fDisp) <= (((1+Rho)*Rs)/2)) & (((1-Rho)*Rs)/2) <= abs(fDisp)));
    R = 10^(AX(1,3)/(Alpha*10))*ones(1,N);
    R(1,Ind1) = 1/Rs;
    R(1,Ind2) = (1-sin((pi/(Rs*Rho))*(abs(fDisp(Ind2))-Rs/2)))/(2*Rs);
    R = R.^Alpha;
    plot(fDisp,10*log10(abs(R)),'r');
    title('Filter Transfer Function [dB]');
    xlabel('Frequency [Hz]');
    legend('Practical','Theoretical');
    grid on;
    zoom on;

end;

```

```

[Bt,At,~] = RaisedCosineFIR(Rs,Ns,M,Rho,Alpha,'No');
[Br,Ar,~] = RaisedCosineFIR(Rs,Ns,M,Rho,Alpha,'No');

```

Listing 5. Transmitter Filtering Operation

```

c_chTx = filter(Bt,At,c_chUp);
s_chTx = filter(Bt,At,s_chUp);

```

Listing 6. Offset QPSK



```
s_chTx=[zeros(1,Ns/2), s_chTx];
c_chTx=[c_chTx, zeros(1,Ns/2)];
```

Listing 7. Power Normalization

```
Pow0 = mean(c_chTx.^2+s_chTx.^2);
c_chTxNorm = c_chTx/sqrt(Pow0);
s_chTxNorm = s_chTx/sqrt(Pow0);
SignalTX = c_chTxNorm + 1i*s_chTxNorm;
```

Listing 8. Amplification Function Implementation

```
function [Z,AM,PM]=interp_twt(X,BKOFF,flag)
% function [Z,AM,PM]=interp_twt(X,BKOFF,flag)
% Normalized Non Linear Amplifiers
% This version has been modified on october 23rd, 2011
%
% Input data
% X [vector] input complex signal
% BKOFF [scalar] amplifier backoff (operational point) in dB
% flag [scalar] choice of the amplifier:
% 1 = TWT (Travelling Wave Tube), RAI model
% 2 = SSPA (Solid State Power Amplifier), ESA model
% 3 = TWT (Travelling Wave Tube), 2nd model
% 4 = SSPA (Solid State Power Amplifier), UOY model
%
% Output data
% Z [vector] output complex signal
% AM [vector] amplitude assigned to X
% PM [vector] extra phase rotation assigned to X
%
% NOTES:
% The maximum output power gain is normalized to 0 dB
% The input mean power must be normalized to 0 dB
% The phase rotation is recovered, so that it is 0 rad at the backoff

switch flag
case 1
    Type = 'TWT - RAI model';
    PinSat = 30.0;
    % A = [];
    % B = [];
    % C = [];
case 2 %SSPA
    Type = 'SSPA - ESA model';
    % PinSat = 30.0;
    PinSat = 30.02616556567280;
    % A = [];
    % B = [];
    % C = [];
case 3 % TWT
    Type = 'TWT M2';
    PinSat = 30;
    % A = [];
    % B = [];
    % C = [];
case 4
    Type = 'SSPA - UOY model';
    PinSat = 30.335389;
    % A = [];
    % B = [];
    % C = [];
    C = C+180/pi;
case 5
    Type = 'SSPA - UOY model';
    PinSat = 30.335389;
    A=ones(1,50);
    B=ones(1,50);
    C=zeros(1,50);
    C = C+180/pi;
end;
N=length(A);
ALEV=PinSat-BKOFF; %PinSat=input power at saturation, ALEV=working point
% Identification of the woking point and the corresponding phase (in radians)
IdxPin=find(A>ALEV);
IdxPin=IdxPin(1);
if isempty(IdXPIn)
    IdXPIn=N;
    PHA= C(IdXPIn);
else if (IdXPIn(1)==1)
    PHA= C(IdXPIn);
else
    PHA=C(IdXPIn-1)+( (ALEV-A(IdXPIn-1)) * (C(IdXPIn)-C(IdXPIn-1)) / (A(IdXPIn)-A(IdXPIn-1)) );
end;
end;
end;
PHARAD=pi*PHA/180;
LA=10.^((A-PinSat)./10);
LB=10.^((B-PinSat)./10);
LC=C*pi/180;
L=10.^(-BKOFF/10);
XARG=L*(abs(X).^2);
XPHA=angle(X);
OUTMOD = interp1(LA,LB,XARG,'spline','extrap');
OUTPHA = interp1(LA,LC,XARG,'spline','extrap');
OUTPHA=OUTPHA+XPHA;
```

```
Z=sqrt(OUTMOD).*exp(1i*(OUTPHA-PHARAD*(ones(size(OUTPHA)))));
AM = OUTMOD;
PM = OUTPHA-PHARAD*(ones(size(OUTPHA)));
%
```

```
SignalTXamp=interp_twt(SignalTX,BKOFF,nla_flag);
```

#### Listing 9. AWGN Channel Model

```
SignalRX=SignalTXamp;
```

#### Listing 10. Two-Ray Channel Model

```
ch_A=1;
ch_B=[1 zeros(1,(DeltaT/Ts)*Ns-1) alpha_c*exp(1i*phi_c)];
SignalRX=filter(ch_B,ch_A,SignalTXamp);
```

#### Listing 11. Receiver Filtering Operation

```
Rxf = filter(Br,Ar,SignalRX);
```

#### Listing 12. Transient Removal

```
RxfNoTrans = Rxf(NTran:end);
```

#### Listing 13. Downsampling Stage

```
if(OQPSK_active)
    OffsetSamI = Ns;
    OffsetSamQ = OffsetSamI+ Ns/2;
else
    OffsetSamI = Ns;
    OffsetSamQ = OffsetSamI;
end
c_chRxNoTrans = real(RxfNoTrans);
s_chRxNoTrans = imag(RxfNoTrans);
c_chRxSam = c_chRxNoTrans(OffsetSamI+1: Ns :end);
s_chRxSam = s_chRxNoTrans(OffsetSamQ+1: Ns :end);

% Vector length control -- to guarantee that the I and Q channels carry the same number of symbols
LL = min([length(c_chRxSam) length(s_chRxSam)]);
c_chRxSam = c_chRxSam(1,1:LL);
s_chRxSam = s_chRxSam(1,1:LL);
RxSam = c_chRxSam + 1i*s_chRxSam;
```

#### Listing 14. 16-QAM Threshold Evaluation

```
function [d, dave]=threshold16QAM(RxSam)
M=16;
Rxabs=abs(real(RxSam))+1i*abs(imag(RxSam));
Eave=mean(abs(RxSam).^2);
dave=sqrt(6*Eave/(M-1));
dmin=dave*0.7;
dmax=dave*1.3;
Nstep=20;
vRx=zeros(1,Nstep);
dvec=linspace(dmin,dmax,Nstep);
for index=1:Nstep;
    dcur=dvec(index);
    Rxfold=rem(real(Rxabs),dcur)+1i*rem(imag(Rxabs),dcur);
    vRx(index)=var(Rxfold);
end
[vmin,imin]=min(vRx);
d=dvec(imin);
%figure
%plot(dvec,vRx);
%grid on
%dx=mean(abs(real(RxSam)));
%dy=mean(abs(imag(RxSam)));
%dave
end
```

#### Listing 15. 4-QAM Demodulation Function

```
function [DemodBits] = FourQpskDemod(RxSamples)
DemodBits=[];
for ii = 1 : length(RxSamples)
    DemodBits=[DemodBits real(RxSamples(ii))>0 imag(RxSamples(ii))>0];
end
end
```

Listing 16. 16-QAM Demodulation Function

```

function [DemodBits] = SixteenQpskDemod(RxSamples)
DemodBits=[];
for ii = 1 : length(RxSamples)
    bi=real(RxSamples(ii));
    bq=imag(RxSamples(ii));
    if(bi > 0 && bi >= b)
        DemodBits=[DemodBits, 1 1];
    elseif(bi > 0 && bi < b)
        DemodBits=[DemodBits, 1 0];
    elseif(bi < 0 && bi > -b)
        DemodBits=[DemodBits, 0 1];
    elseif(bi < 0 && bi <= -b)
        DemodBits=[DemodBits, 0 0];
    end
    if(bq > 0 && bq >= b)
        DemodBits=[DemodBits, 1 1];
    elseif(bq > 0 && bq < b)
        DemodBits=[DemodBits, 1 0];
    elseif(bq < 0 && bq > -b)
        DemodBits=[DemodBits, 0 1];
    elseif(bq < 0 && bq <= -b)
        DemodBits=[DemodBits, 0 0];
    end
end
end

```

Listing 17. Spectral Estimation

```

PSD_X=pwelch(SignalRX,PlotsParam.winr,0,N);
freq=linspace(-0.5/Tc,0.5/Tc,length(PSD_X));
%figure('Name','Spectral Estimation of the signal X');
plot(freq,10*log10(fftshift(PSD_X)));
grid on;

```

Listing 18. 4-QAM Semi-Analytical BER vs  $E_b/N_0$  Estimation Function

```

function [EstimatedBER]=BERComputation_QPSK(Si, EbNoValues, Ps, Tb, Eh)
Sigma2m = (Ps*Tb*Eh/2)./EbNoValues;
EstimatedBER=mean((1/4*erfc((1./sqrt(2*Sigma2m))'*abs(real(Si))))' +...
    (1/4*erfc((1./sqrt(2*Sigma2m))'*abs(imag(Si))))');
end

```

```

Eh=(1/Tc)*sum(Br.^2);
Pow=mean(abs(SignalRX_tot(NTran+1:end)).^2);
BER_Est=BERComputation_4QPSK(RxSam, F_lin, Pow, Tb, Eh);
EbNo_t=interp1(BER_Est,EbNo,BER_target);

```

Listing 19. 16-QAM Semi-Analytical BER vs  $E_b/N_0$  Estimation Function

```

function [EstimatedBER]=BERComputation_16QAM(Si, EbNoValues, Ps, Tb, Eh, bmax, b, Nb)
Sigma2m = (Ps*Tb*Eh/2)./EbNoValues;
NormIsi=0;
NormQsi=0;
ISi = abs(real(Si));
QSi = abs(imag(Si));
sumP=0;
for jj=1:length(Si)
    if(ISi(jj) > bmax)
        NormIsi=ISi(jj)-bmax;
        p1=0.5*erfc(NormIsi./sqrt(2*Sigma2m));
    else
        NormIsi=rem(ISi(jj),b);
        p1= 0.5*erfc(NormIsi./sqrt(2*Sigma2m)) + 0.5*erfc((b-NormIsi)./sqrt(2*Sigma2m));
    end
    if(QSi(jj) > bmax)
        NormQsi=QSi(jj)-bmax;
        p2=0.5*erfc(NormQsi./sqrt(2*Sigma2m));
    else
        NormQsi=rem(QSi(jj),b);
        p2= 0.5*erfc(NormQsi./sqrt(2*Sigma2m)) + 0.5*erfc((b-NormQsi)./sqrt(2*Sigma2m));
    end
    sumP = sumP + p1+p2-p1.*p2;
end
EstimatedBER=1/(length(Si)*Nb)*sumP;
end

```

```

Eh=(1/Tc)*sum(Br.^2);
Pow=mean(abs(SignalRX_tot(NTran+1:end)).^2);
[bmax, b]=threshold16QAM(RxSam);
BER_Est=BERComputation_16QAM(RxSam, EbNo_lin, Pow, Tb, Eh, bmax, b, Nb);
EbNo_t=interp1(BER_Est,EbNo,BER_target);

```

Listing 20. 4-QAM BER Counter

```

Pow=mean(abs(SignalRX_tot(NTran+1:end)).^2);
CountedErrors=100; jj=1;
while jj<=length(EbNo_lin) && CountedErrors>MinCountedErrors;
    NoisePower=Pow*Tb/(Tc*EbNo_lin(jj));
    UnitPowComplexNoiseSamples=randn(1,length(RxSam))+1i*randn(1,length(RxSam));
    ComplexNoiseSamples=(sqrt(NoisePower/2))*UnitPowComplexNoiseSamples;
    NoisyRxSam=RxSam+ComplexNoiseSamples;

    [DemBits]=FourQpskDemod(NoisyRxSam, 1, length(RxSam),Nb);
    OffsetBitSeq=Nb;
    TXBits=SourceBit(OffsetBitSeq+1:OffsetBitSeq+length(DemBits));
    CountedErrors=symerr(TXBits,DemBits);
    BER_counter=[BER_counter, CountedErrors/length(DemBits)];
    jj=jj+1;
end

```

Listing 21. 16-QAM BER Counter

```

Pow=mean(abs(SignalRX_tot(NTran+1:end)).^2);
CountedErrors=100; jj=1;
while jj<=length(EbNo_lin) && CountedErrors>MinCountedErrors;
    NoisePower=Pow*Tb/(Tc*EbNo_lin(jj));
    UnitPowComplexNoiseSamples=randn(1,length(RxSam))+1i*randn(1,length(RxSam));
    ComplexNoiseSamples=(sqrt(NoisePower/2))*UnitPowComplexNoiseSamples;
    NoisyRxSam=RxSam+ComplexNoiseSamples;

    [bmax, b]=threshold16QAM(NoisyRxSam);
    DemBits=SixteenQpskDemod(NoisyRxSam, 1, b, length(NoisyRxSam),Nb);
    OffsetBitSeq=Nb;
    TXBits=SourceBit(OffsetBitSeq+1:OffsetBitSeq+length(DemBits));
    CountedErrors=symerr(TXBits,DemBits);
    BER_counter=[BER_counter, CountedErrors/length(DemBits)];
    jj=jj+1;
end

```

Listing 22. Adjacent Channels Delays

```

c_chTx_r = [zeros(1,Delay_r) c_chTx_r(1:end-Delay_r)];
s_chTx_r = [zeros(1,Delay_r) s_chTx_r(1:end-Delay_r)];
c_chTx_l = [zeros(1,Delay_l) c_chTx_l(1:end-Delay_l)];
s_chTx_l = [zeros(1,Delay_l) s_chTx_l(1:end-Delay_l)];

```

Listing 23. Co-Channels Delays

```

c_chTx_coch_r = [zeros(1,Delay_coch) c_chTx_coch_r(1:end-Delay_coch)];
s_chTx_coch_r = [zeros(1,Delay_coch) s_chTx_coch_r(1:end-Delay_coch)];
c_chTx_coch_l = [zeros(1,Delay_coch) c_chTx_coch_l(1:end-Delay_coch)];
s_chTx_coch_l = [zeros(1,Delay_coch) s_chTx_coch_l(1:end-Delay_coch)];

```

Listing 24. Adjacent Channels Definition

```

NL=length(SignalTX);
m=linspace(0,NL-1,NL);
SignalTXamp_r=SignalTXamp_r.*exp(1i*2*pi*Df*m*Tc);
SignalTXamp_l=SignalTXamp_l.*exp(-1i*2*pi*Df*m*Tc);

```

Listing 25. Co-Channels Definition

```

NL=length(SignalTX);
m=linspace(0,NL-1,NL);
SignalTXamp_coch_r=SignalTXamp_coch_r*10^(-Coch_att/20).*exp(1i*pi*Df*m*Tc);
SignalTXamp_coch_l=SignalTXamp_coch_l*10^(-Coch_att/20).*exp(-1i*pi*Df*m*Tc);

```

Listing 26. 4-QAM Cost Function Evaluation

```

function [EstimatedBER]=BERComputation_CF(Si, F_lin, P_out_sat, Tb, Eh)
Sigma2m = (P_out_sat*Tb*Eh/2)./F_lin;
EstimatedBER=mean((1/4*erfc((1./sqrt(2*Sigma2m))'*abs(real(Si))))' + ...
    (1/4*erfc((1./sqrt(2*Sigma2m))'*abs(imag(Si))))');
end

```

```

Eh=(1/Tc)*sum(Br.^2);
Pow=mean(abs(SignalRX_tot(NTran+1:end)).^2);
BER_Est=BERComputation_4QPSKCostF(RxSam, F_lin, P_out_Sat, Tb, Eh);
F_t=interp1(BER_Est,F,BER_target);

```

### Listing 27. 16-QAM Cost Function Evaluation

```

function [EstimatedBER]=BERComputation_16QAM(Si, F_lin, P_out_sat, Tb, Eh, Threshold, Nb)
    Sigma2m = (P_out_sat*Tb*Eh/2)./F_lin;
    NormIsi=0;
    NormQsi=0;
    ISi = abs(real(Si));
    QSi = abs(imag(Si));
    sumP=0;
    for jj=1:length(Si)
        if (ISi(jj) > Threshold)
            NormIsi=ISi(jj)-Threshold;
            p1=0.5*erfc(NormIsi./sqrt(2*Sigma2m));
        else
            NormIsi=rem(ISi(jj),Threshold);
            p1= 0.5*erfc(NormIsi./sqrt(2*Sigma2m))+ 0.5*erfc((Threshold-NormIsi)./sqrt(2*Sigma2m));
        end
        if (QSi(jj) > Threshold)
            NormQsi=QSi(jj)-Threshold;
            p2=0.5*erfc(NormQsi./sqrt(2*Sigma2m));
        else
            NormQsi=rem(QSi(jj),Threshold);
            p2= 0.5*erfc(NormQsi./sqrt(2*Sigma2m))+ 0.5*erfc((Threshold-NormQsi)./sqrt(2*Sigma2m));
        end
        sumP = sumP + p1+p2-p1.*p2;
    end
    EstimatedBER=1/(length(Si)*Nb)*sumP;
end

```

```

Eh=(1/Tc)*sum(Br.^2);
Pow=mean(abs(SignalRX_tot(NTran+1:end)).^2);
[bmax, b]=threshold16QAM(RxSam);
BER_Est=BERComputation_16QAMCostF(RxSam, F_lin, P_out_Sat, Tb, Eh, bmax, b, Nb);
F_t=interp1(BER_Est,F,BER_target);

```