



**Politecnico  
di Torino**

01TWPBG

## Signal Processing and Wireless Transmission Lab

Wireless Transmission Part

Group 4:  
Zhang Ziteng S289309  
Kang Yali S287831  
Tong Lin S287649  
Cen Jialuo S287781  
Hassan Abeyat S305435

Professor: Guido Montorsi  
Academic year: A.Y. 2022-23  
Date: 16.01.2023



# Contents

<b>Contents</b>	<b>i</b>
<b>1 Simulation of a Basic Digital Transmission System</b>	<b>1</b>
1.1 System without filters . . . . .	1
1.1.1 Block diagram . . . . .	1
1.1.2 BER results . . . . .	3
1.2 System with filters . . . . .	8
1.2.1 Block diagram . . . . .	8
1.2.2 BER results . . . . .	9
1.2.3 Eye diagram, IQ plot, scattering diagram . . . . .	12
1.2.4 Spectrum of the signals . . . . .	19
<b>2 Implementation of a DSP C++ Class</b>	<b>25</b>
2.1 BER example . . . . .	25
2.2 PN_Source . . . . .	26
2.2.1 Algorithm description . . . . .	26
2.2.2 Functionalities . . . . .	28
2.2.3 Source code (header and implementation) . . . . .	28
2.2.4 Validation tests . . . . .	30
<b>3 A Realistic Receiver for a Realistic Wireless Channel</b>	<b>33</b>
3.1 Narrowband channel model results . . . . .	33
3.1.1 The block diagrams of TX and RX . . . . .	33
3.1.2 Description of narrow-band channel model . . . . .	35
3.1.3 Some relevant plots with comments . . . . .	36
3.1.4 Description of frame structure (header, pilots..) . . . . .	38
3.2 Test of the considered receiver block . . . . .	39
3.2.1 Algorithm description . . . . .	39
3.2.2 Description of performed tests and motivations . . . . .	40
3.2.3 Tests results and comments . . . . .	41

<b>4 USRP Results</b>	<b>45</b>
4.1 Closed loop with cable . . . . .	46
4.2 Broadcast wireless channel . . . . .	48
4.3 Frequency Division Duplex wireless channel . . . . .	50
<b>5 GNU Radio Lab Results</b>	<b>53</b>
5.1 Tuning simulation . . . . .	53
5.2 QPSK transmission . . . . .	59
5.3 Real sinusoidal transimission . . . . .	60
<b>6 Appendix</b>	<b>65</b>
6.1 Code . . . . .	65
6.1.1 My_Ber_Meter.h . . . . .	65
6.1.2 My_Ber_Meter.cpp . . . . .	66
6.1.3 PN_Source_Matlab . . . . .	69
6.2 Screen shots and results of GTX . . . . .	70

# 1 | Simulation of a Basic Digital Transmission System

In this chapter, the basic digital transmission systems are discussed, involving the simulated parameters from the given Lab1 and Lab2. There are two parts in this chapter, respectively corresponding with the block diagram without- and with- the SRRC filters.

In section 1.1, a simple ideal digital transmission system without filters is implemented by using TOPCOM++ library and Microsoft Visual C++ (MSVC++) as IDE. The performed measurements are the BER results, demonstrated in section 1.1.2, which have been specifically discussed then.

In section 1.2, the related digital transmission system with the SRRC filters has been discussed. The implemented simulations have used the same method and process as in section 1.1. Excepting for the BER plots, the results have also been displayed in various types of illustrations, such as eye diagrams, IQ plots, scattering parameters, and spectrum plots.

The main steps of the implemented simulations are, firstly, opening the provided solution MSVS++ file of "TOPCOM++USRP.sln" in the folder of "Projects" and "Simple", then building the solution of the code in "main. cpp" and debugging it when there are no mistakes. By changing the parameters in the file "input.txt", the corresponding results can be saved in the set file of "Data/BER.txt". The derived results can be automatically plotted with the given *gnuplot* programs, and the displayed figures can be adjusted by slightly modifying the codes inside the *gnuplot* programs.

## 1.1 System without filters

### 1.1.1 Block diagram

The block diagram of the simplified digital transmission system without the SRRC filters is shown in figure 1.1. The system consists of four main parts, which are the information

## 1 | Simulation of a Basic Digital Transmission System

source, modulator, AWGN channel, and demodulator.

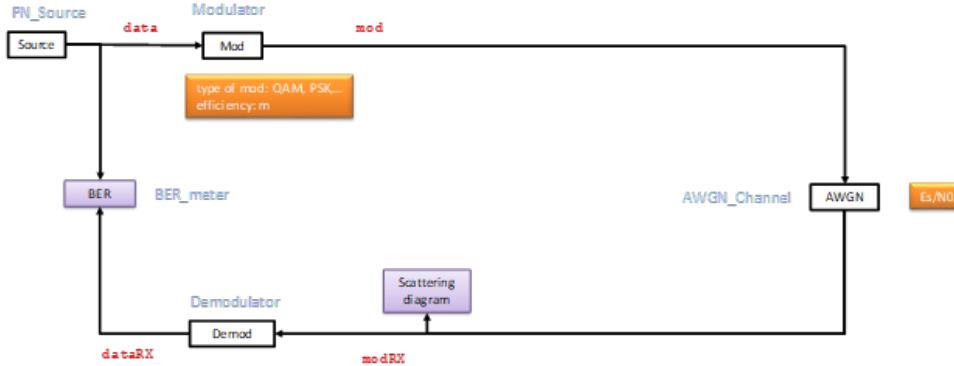


Figure 1.1: The block diagram without filters

- Source/PN Source: generates the originally transmitted bits of the signal.
- Modulator: originally indicates the circuit which is able to superimpose the low-frequency signal which carries information, onto a high-frequency signal, in order to realize the wireless transmission. Here, the modulator is corresponding with the process of associating waveform to a stream of bits or discrete symbols. This is represented in the code with the modulator class, which implements the modulation functions. The following utilized modulators are the PAM modulator, QAM modulator, and PSK modulator with different modulation bits.
- AWGN channel: implement the additive white Gaussian noise (AWGN), as a simulation of the real signal in the transmitted environment. AWGN is a basic noise model, able to mimic the effect of many random processes that may occur in nature, which has uniform power across the frequency band for the information system (white noise) and conform to the normal distribution in the time domain with the average time domain value of zero (Gaussian noise). The output signal sample of this block is evaluated as,

$$s_{\text{out}}[i] = s_{\text{in}}[i] + \alpha n[i], \quad (1.1)$$

where  $s_{\text{in}}[i]$  is the noiseless input signal sample of the block,  $\alpha$  is the level of the noise as in (1.2), and  $n[i]$  is the Gaussian random variable with zero mean and variance equals to 1.

$$\alpha = \sqrt{\frac{N_0}{2}}, \quad (1.2)$$

$N_0$  is the one-sided noise power spectral density.

- Demodulator: generally indicates the implementation of the inverse function with respect to the modulators. The demodulator class here implements a generic one-dimensional or two-dimensional demodulator, where the module provides both hard and soft decoding. In the case of soft detection, their soft values are expressed as Log-Likelihood-Ratios (LLR).

The only characterizing parameter of this simple system is the BER, the bit-error rate, which evaluates the number of bits that have been transmitted incorrectly. The BER parameter is defined as in the equation (1.3),

$$BER = \frac{n}{N}, \quad (1.3)$$

where  $n$  is the number of the incorrect bits,  $N$  is the total number of comparing bits.

As shown in the diagram, the BER parameter here is evaluated as a comparison between the bits coming out of the source, as the original transmitted one, and the bits coming out of the demodulator, as the final received and processed one. The results of the BER meters from the implemented simulation codes are specifically discussed in the following section.

### 1.1.2 BER results

The initial set parameters in the input file of the built solution for the following plotted diagrams are as shown in Table 1.1,

Table 1.1: The set parameters of the BER plots without filters

Parameter	Value
<b>Maximum number of simulated intervals</b>	1000000
<b>Frame size</b>	1000
<b>E<sub>b</sub>/N<sub>0</sub>[dB]</b>	0:50:1

The explanations of the parameters are as follows,

- Number of simulated intervals: The minimum simulation step usually is equal to the maximum frame size of the system. In the case of multiple framing, it may be necessary to take as the minimum simulation interval the minimum common multiple of all the frames involved.

- Frame size: The frame size of the system often coincides with a codeword size if block coding is present.
- $E_b/N_0$  [dB]: is the normalized signal-to-noise ratio, as SNR per bit, which is evolved in the block of the AWGN channel. It is defined as,

$$SNR = \frac{E_b}{N_0}, \quad (1.4)$$

where  $E_b$  is the energy per transmitted bit, and  $N_0$  is the one-ended noise power spectral density.

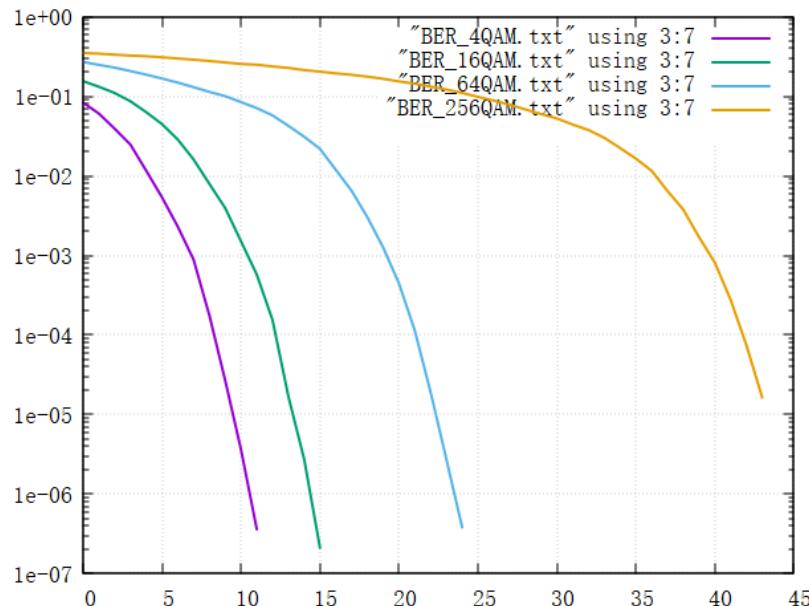


Figure 1.2: The BER for QAM with different modulation cardinalities

As shown in figure 1.2, the values of BER with the increased  $E_b/N_0$  are measured with different cardinalities of QAM, respectively are 4QAM, 16QAM, 64QAM, and 256QAM. The BER values are plotted in the logarithmic scale as the vertical axis, while the  $E_b/N_0$  is expressed in the unit of decibel as linear variation, as the horizontal axis.

QAM modulation indicates quadrature amplitude modulation, which is one of the most useful modulation types nowadays in digital telecommunication systems, such as in 802.11 WiFi standards, and widely uses for 4G and 5G cellular systems. It relates to a modulation way of changing the phase by quadrature and amplitude of the different carrier waves, to convey the analogue signals or digital bit streams.

It is obvious to see, that with the increase of  $E_b/N_0$  (or SNR), all the BER values with

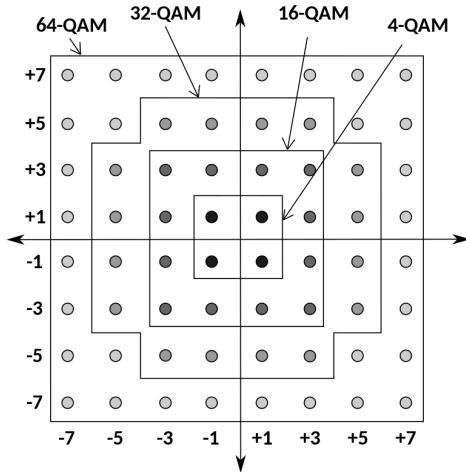


Figure 1.3: QAM Constellation

different types of QAM are decreasing. This can be well explained with the functions of  $E_b/N_0$  and BER, respectively as (1.4) and (1.3). When  $E_b/N_0$  is increasing, the one-ended noise power spectral density  $N_0$  is decreasing, while the energy per transmitted bit  $E_b$  remains unchanged with the same type of modulation, which causes the decrease of the number of the incorrectly transmitted bits  $n$ . As the total number of transmitted bits is constant, the value of the BER decreases.

Also, with fewer encoded bits, the BER values are smaller, and the variations show steeper. The previous phenomenon can be clearly discovered with the starting points of each modulation cardinality. 256QAM occupies the highest value of the BER, at about 0.5, with respect to the zero  $E_b/N_0$ , and the 4QAM shows the smallest, with almost 0.1 for the BER value. The larger the modulation bit is, the larger the BER value exists when  $E_b/N_0$  is zero. These tendencies result from, that the total transmitted energy remaining the same, the higher cardinalities modulation could be more susceptible to the noises. These can also be explained with the definitions of  $E_b/N_0$  and BER. As the power per transmitted bit is obtained from the power per transmitted symbol dividing the number of the modulation efficiency, as shown in (1.5), when the modulation efficiency  $m$  is lower, the power per transmitted bit  $E_b$  is higher, since the power per transmitted symbol  $E_s$  is constant. the one-ended noise power spectral density  $N_0$  is a constant, the  $SNR$  increases. So in same the power per transmitted symbol condition, high modulation efficiency has higher BER and lower Noise immunity.

$$E_b = \frac{E_s}{m}, \quad (1.5)$$

We can also get it from the *QAM Constellation*, in the same power condition, as the modulation efficiency increase, the distance between two adjacent points closer, so *BER* increases.

Moreover, during the simulation, as the  $E_b/N_0$  is set to vary from 0 to 50, with each step of 1, the 4QAM's simulation ends when the  $E_b/N_0$  reaches about 11, while the others are higher, with the final  $E_b/N_0$  equals to 15, 25 and 43, respectively for the 16QAM, 64QAM, and 256QAM. This is due to the limitation of the accepted error bit as initially set in the main code, which is 30. That is, when the number of the transmitted error bits is counted as low as 30, the transmission results become acceptable, and the simulation then operates much slower computations. So that the BER values end never exceed  $1e-7$  as shown in the figure.

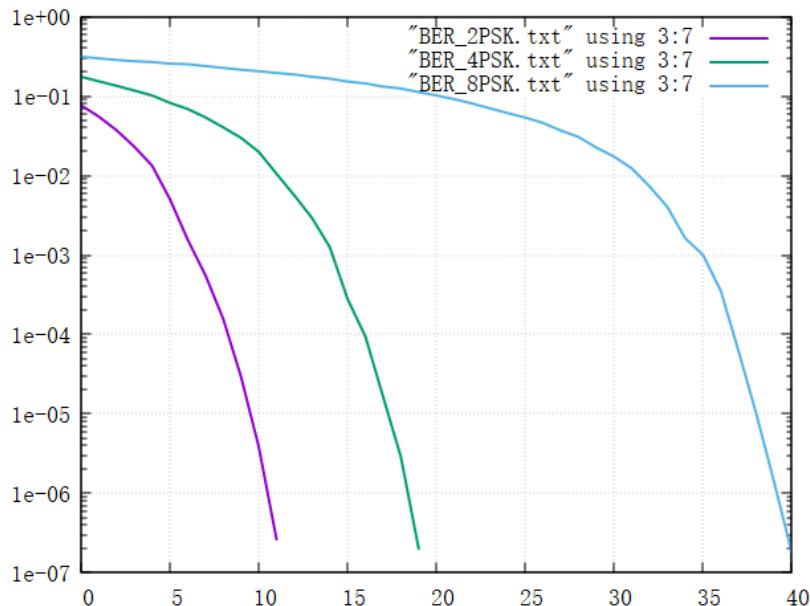


Figure 1.4: The BER for PSK with different modulation cardinalities

Similar tendencies have also shown in figure 1.4, which demonstrates the variety of different PSK with the increase of  $E_b/N_0$ . PSK modulation indicates phase-shift keying, which is also one of the most important types of modulation. In contrast to the QAM modulation, the modulation of the PSK changes only the phases of the signals. It is also widely used for current wireless signal transmissions, such as Bluetooth, RFID, and satellite broadcasting. Here, only three types of PSK have been discussed, which are, 2PSK, 4PSK, and 8PSK.

Again, with the increase of the  $E_b/N_0$ , the BER is always decreasing, and with the higher modulation efficiency, the levels of the BER results are higher.

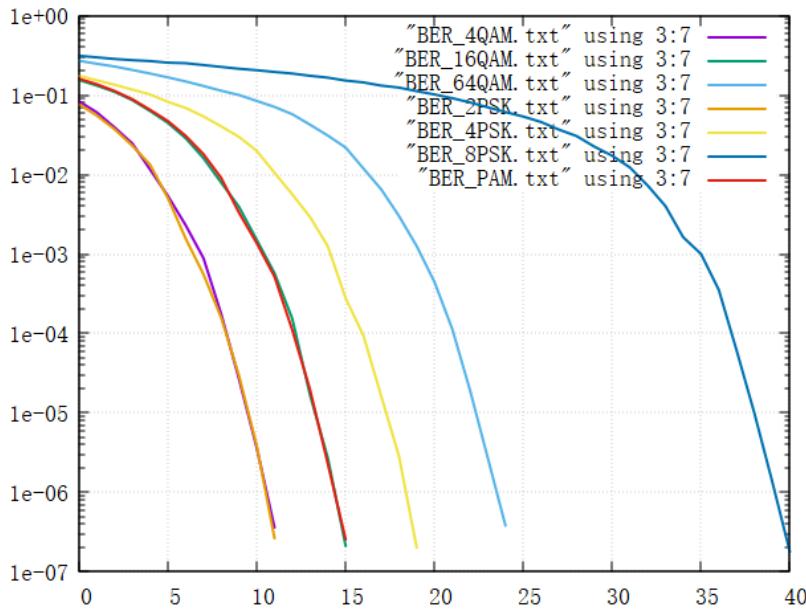


Figure 1.5: The BER for the different modulation types

In figure 1.5, the values of BER for the different modulations have been measured, which are different orders of QAM (4QAM, 16QAM, 64QAM), different orders of PSK (2PSK, 4PSK, 8PSK), and PAM. The PAM modulation represents the pulse amplitude modulation, in which the amplitude of the pulsed carrier signal is changed according to the amplitude of the message signal. As one of the most simple modulation technology, it is used in such as Ethernet communication, some micro-controller for generating control signals, and photo-biology.

The 2PSK and the 4QAM shows similar levels of bit-error rate, who possess the same modulation efficiency and have same constellations. However, this phenomenon doesn't extend to the higher order of the different modulations, while the 4PSK's BER is higher than the 16QAM's, and the same for the 8PSK and 64QAM. Besides, the BER values for the PAM modulation slightly coincide with the 16QAM, which starts from about 0.2, and ends at 2e-7 when  $Eb/N0$  is about 15dB.

## 1.2 System with filters

### 1.2.1 Block diagram

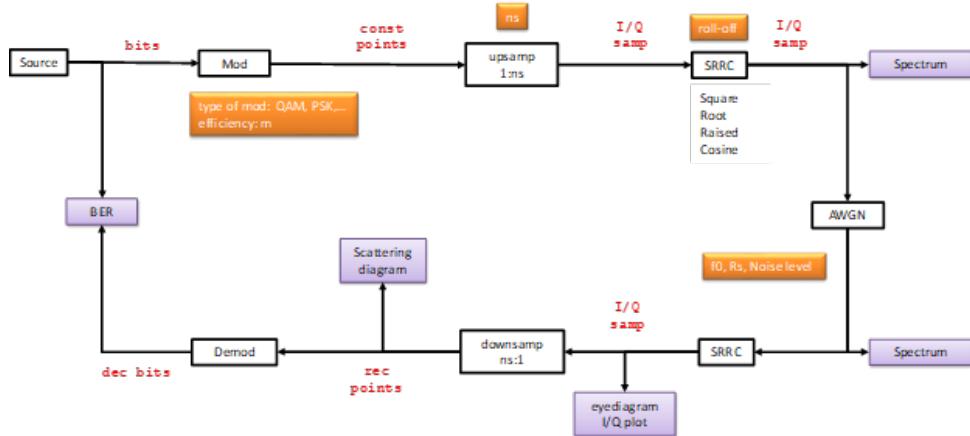


Figure 1.6: The block diagram with filters

As shown in figure 1.6, on the basis of the simple system without filters in figure 1.1, a more complete digital transmission system with the filters is discussed in this chapter, which has added the blocks of up-sampling, down-sampling, and two SRRC filters respectively placed in the transmission chain and receiving chain.

- SRRC filters: indicates the square root raised cosine filters, which is a frequently used type of filter for both transmit and receive in digital communication systems to improve Bit Error Rate (BER) performance. It is a practical structure nowadays, to solve the problem of wasting the power of the sending signal out of the bandwidth of interest. The square root raised cosine impulse response with different values of roll-off  $\beta$  is as shown in figure 1.7.

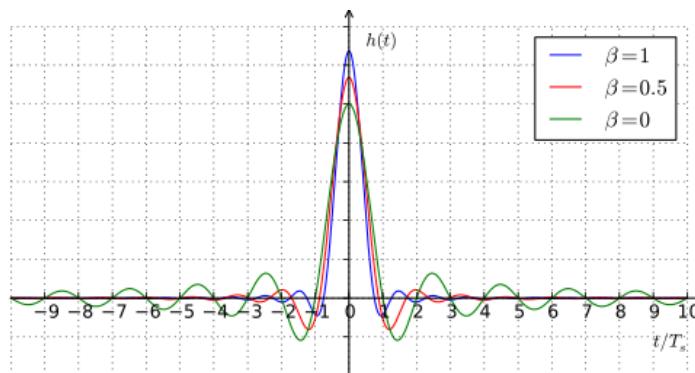


Figure 1.7: SRRC frequency response

- Up-sampling: is a method of interpolating zero-values by a certain interval into the real samples, in order to increase the sampling rate artificially.
- Down-sampling: is the inverse process of up-sampling, as the decimation of the signals, in order to reduce the sampling rate of the signal.

### 1.2.2 BER results

The values of the parameters implemented in this part are as shown in Table 1.2, which have been initially well set in the proper range according to the later simulations.

Table 1.2: The set parameters for the BER plots with filters

Parameter	Value
<b>SRRC roll-off (ro)</b>	0.5
<b>Length of filters (L)</b>	10
<b>Numbers of samples per symbols (ns)</b>	2
<b>E<sub>b</sub>/N<sub>0</sub></b>	0:50:1

The term of  $E_b/N_0$  as already explained in the former section is still set as an increasing value from 0 to 50 with a single step of 1 in this section. The explanations of other used parameters are as follows.

- Filter roll-off (ro): Roll-off is one of the most important parameters for the filters, which varies from zero to one. Basically, it is related to the steepness of a transfer function with frequency for the filters, where the lower roll-off values give a steeper slope to the attenuation.

Take the examples of raised cosine (RC) filters to illustrate the variation of the roll-off values, as shown in figure 1.8, where the parameter  $\alpha$  stands for the roll-offs. The SRRC filter's frequency response is the square root of the frequency response for the RC filters, with the difference lies on the amplitudes for the response waveform. The slopes are more directive to be witnessed with the RC filters, as the optimum spectrum occupation occurs when  $\alpha=0$ , which also corresponds with more oscillations in the time domain, thus increasing the inter-symbol interference (ISI). Therefore, the lower values of roll-offs allow for more efficient use of the spectrum but introduce more interference, which can be well visualized in the later sections of the scattering plots and spectrum of the signal.

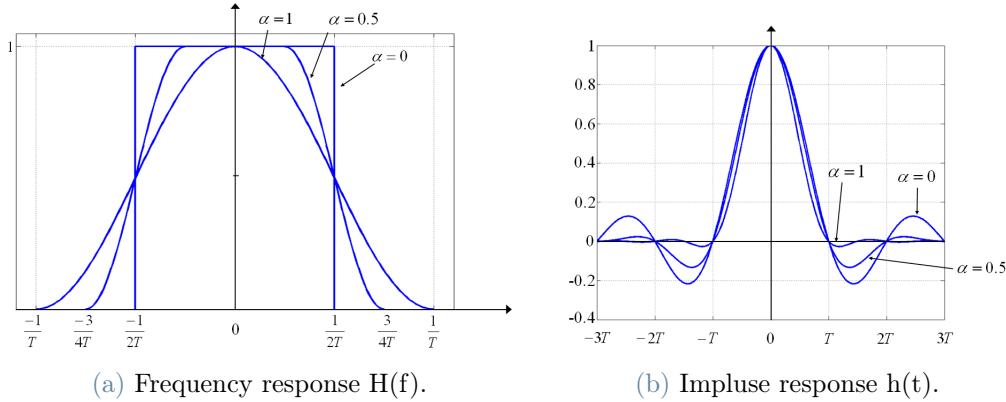


Figure 1.8: The frequency response and impulse response of the RC filters with different roll-offs ( $\alpha$ )

- Length of filters (L): indicates the number of coefficients of the digital filters.
- Numbers of samples per symbols (ns): represents how many samples are contained within a symbol, which can be calculated as the ratio of the sampling rate and symbol rate.

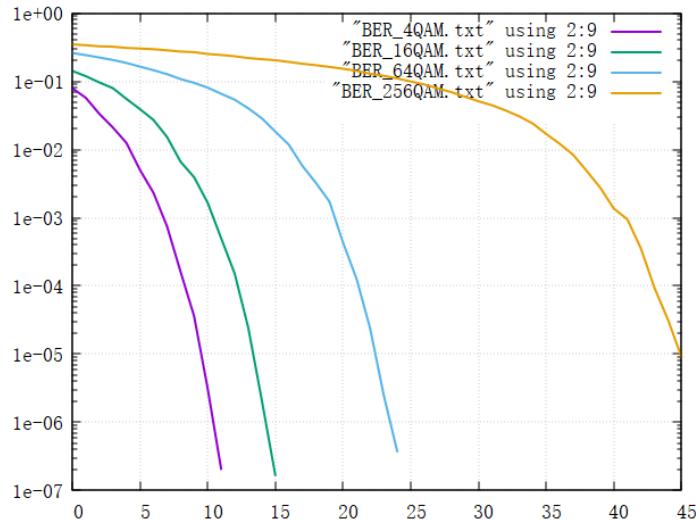


Figure 1.9: The BER for QAM with different modulation cardinalities

As shown in figure 1.9, the tendency of the variations of the BER parameters with regards to the increase of the  $Eb/N0$  for the QAM modulations is quite similar with the one without filters as in figure 1.2. The slight differences lay on a little bit lower values of the BER for each modulation when increasing the values of  $Eb/N0$ .

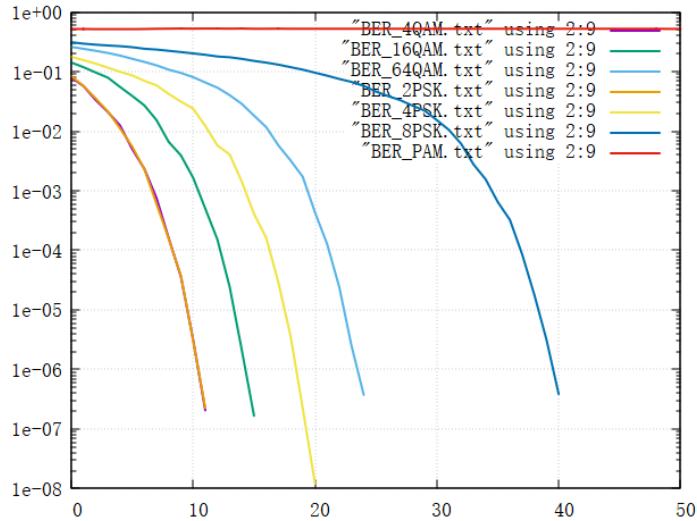


Figure 1.10: The BER with different modulation types

The comparison of different modulation types are as shown in figure 1.10, again, the tested types are, 4QAM, 16QAM, 64QAM, 2PSK,4PSK,8PSK, and PAM. The plots are generally same with the QAM modulations and PSK modulation, with compare with the plots without the filters as in figure 1.5. However, the PAM modulation shows a lot differences, which remains a rather high level of BER when adding the filters.

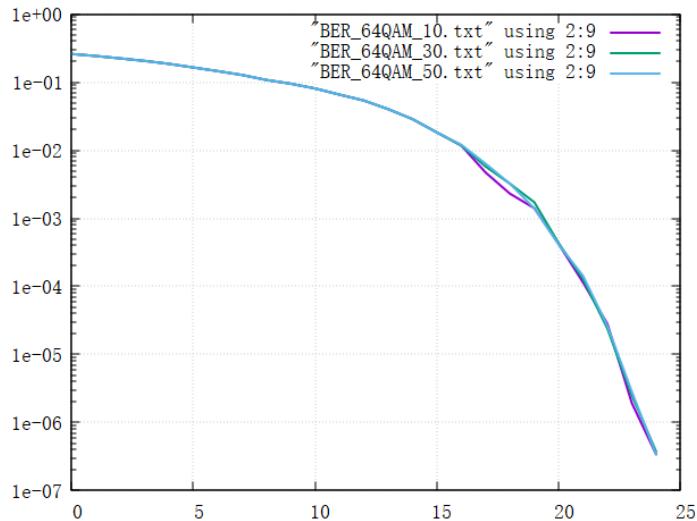


Figure 1.11: The BER with different error limits

Here we also tested the variation when changing the error limitation, to 10 bits, 30 bits, and 50 bits, as shown in figure 1.11. The plots are basically consistent, and the existing slight flaws may be caused by the random noises during the simulations.

### 1.2.3 Eye diagram, IQ plot, scattering diagram

In this section, the results are illustrated in the forms of eye diagrams, IQ plots, and scattering diagrams.

- Eye diagram: providing an overall view of the performances of the signal recovery correctness. It demonstrates a series of digital signals accumulated on the oscillator, to observe the effects of inter-symbol interference (ISI) and noise. The principle for generating an eye diagram is as shown in Figure 1.12. The vertical inputs indicate the repetitively sampled digital signals from the receiver, while the horizontal sweep stands for the data rate. The eye-opening area represents the correct operating area, and the errors in the time axis mainly result from the frequency error and drift, and the clock edge drift.

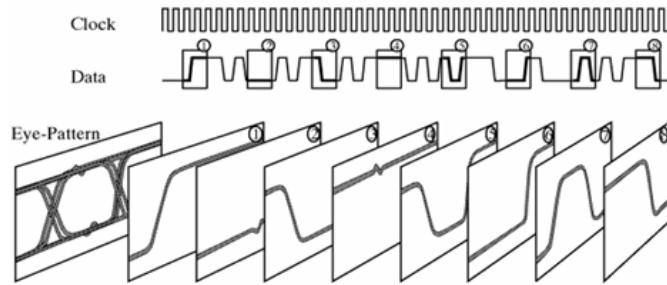


Figure 1.12: The principle of plotting eye diagram

- IQ plot: display the track of the in-phase and quadrature components of the modulated signal during the simulations.
- Scattering diagram: shows the position of each simulated point in the complex plane, which can be used for visualizing the constellation for the modulation and observing the deviation of the points from the ideal positions.

The set parameters for the diagrams of both the QAM modulation and PSK modulation are shown in Table 1.3, respectively for the figures of 1.13 and 1.14.

Table 1.3: The set parameters

Parameter	Value
SRRC roll-off (ro)	0.5
Length of filters (L)	10
Numbers of samples per symbols (ns)	12
Eb/N0 [dB]	25

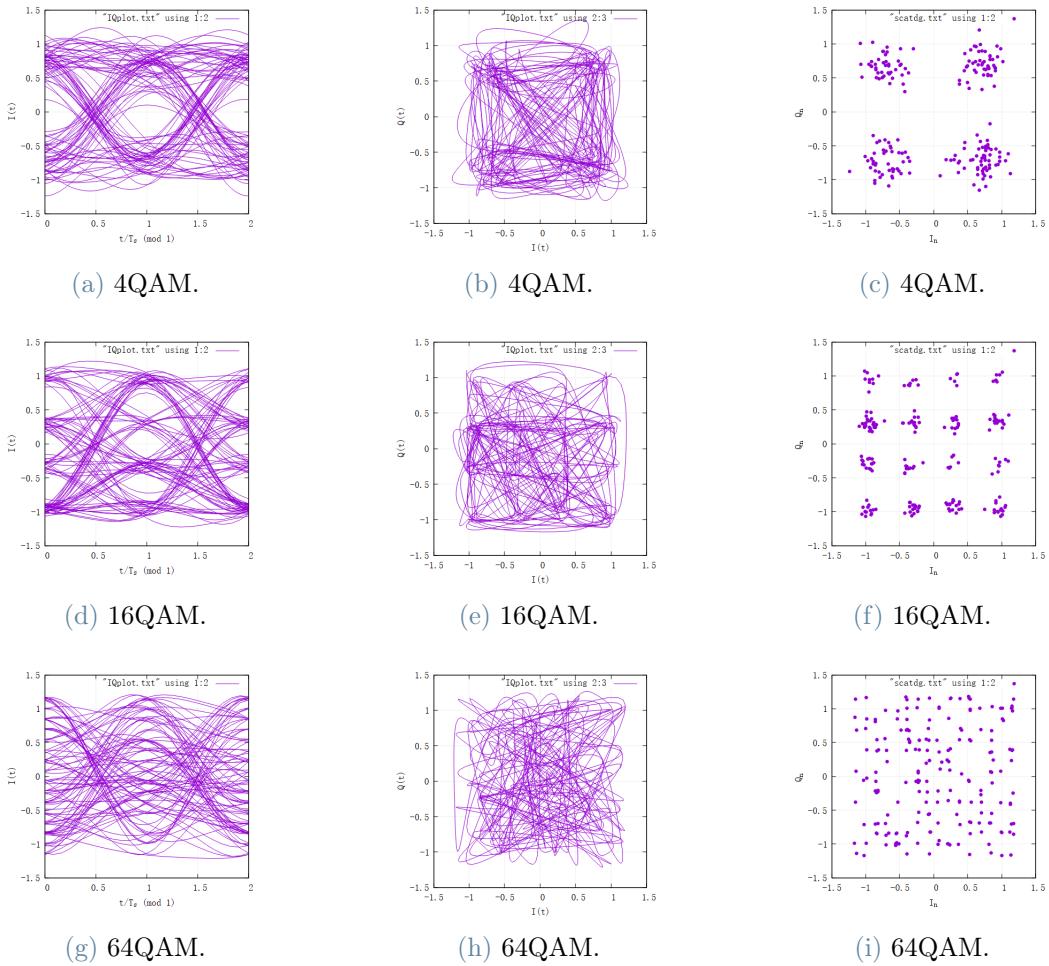


Figure 1.13: The diagrams of QAM

Figure 1.13 represents the eye diagrams, IQ plots, and scattering diagrams of different QAM modulation cardinalities, respectively as 4QAM, 16QAM, and 64QAM. The eye diagrams demonstrate different characters with various modulation efficiencies, relating to different numbers of sets of curves. The IQ plots, on the other hand, are able to display

## 1| Simulation of a Basic Digital Transmission System

the traces of the movement of the transmitting elements as previously explained. The scattering diagrams show obvious constellation patterns of the transmitted modulations, whereas for the 64QAM the pattern is not so clear which is due to a high level of errors corresponding with the set low value of the roll-off. It is evident to see, that for the QAM modulations, both amplitude and phase are changed.

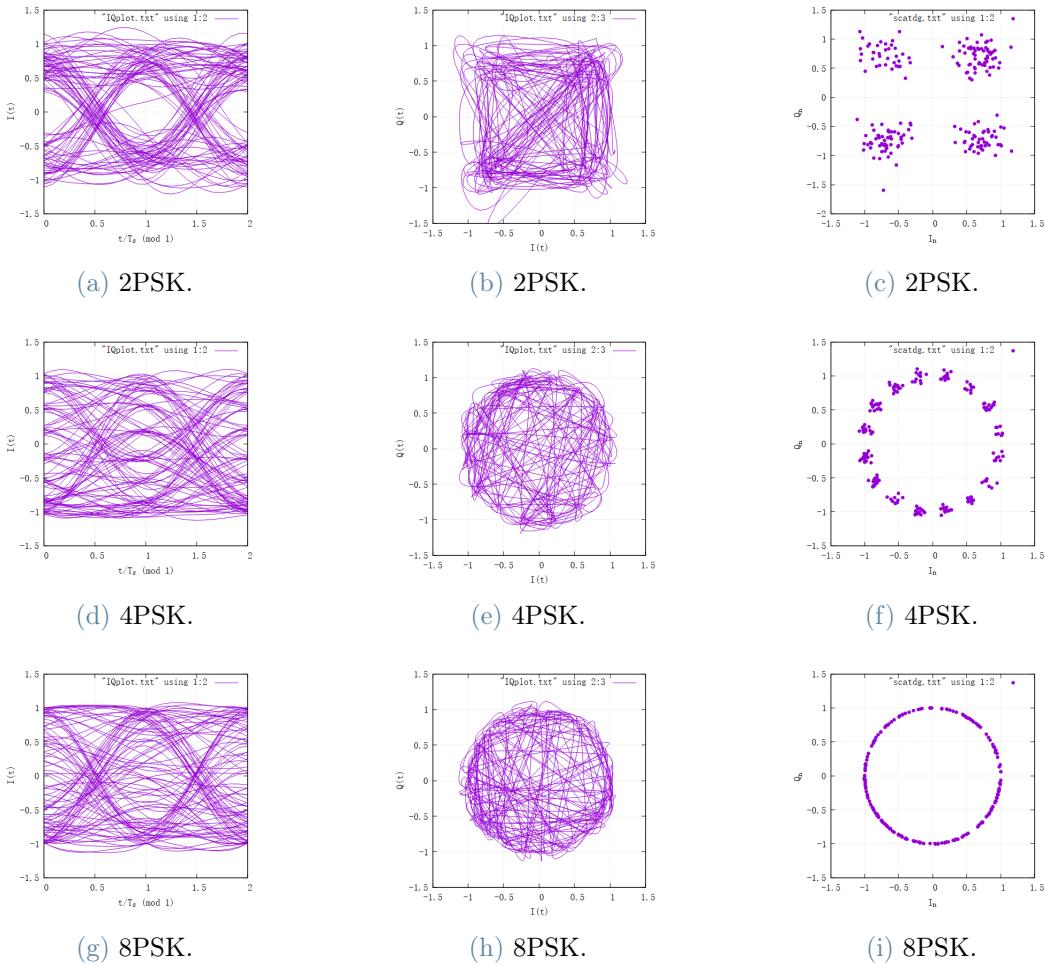


Figure 1.14: The diagrams of PSK

Figure 1.14 shows the diagrams for different PSK modulations. The character of this modulation as phase-shifting can be very clear to witness in the scattering diagrams, where the constellation points are forming as circles. Again, the number of horizontal lines in the eye diagrams is related to different numbers of modulation cardinality.

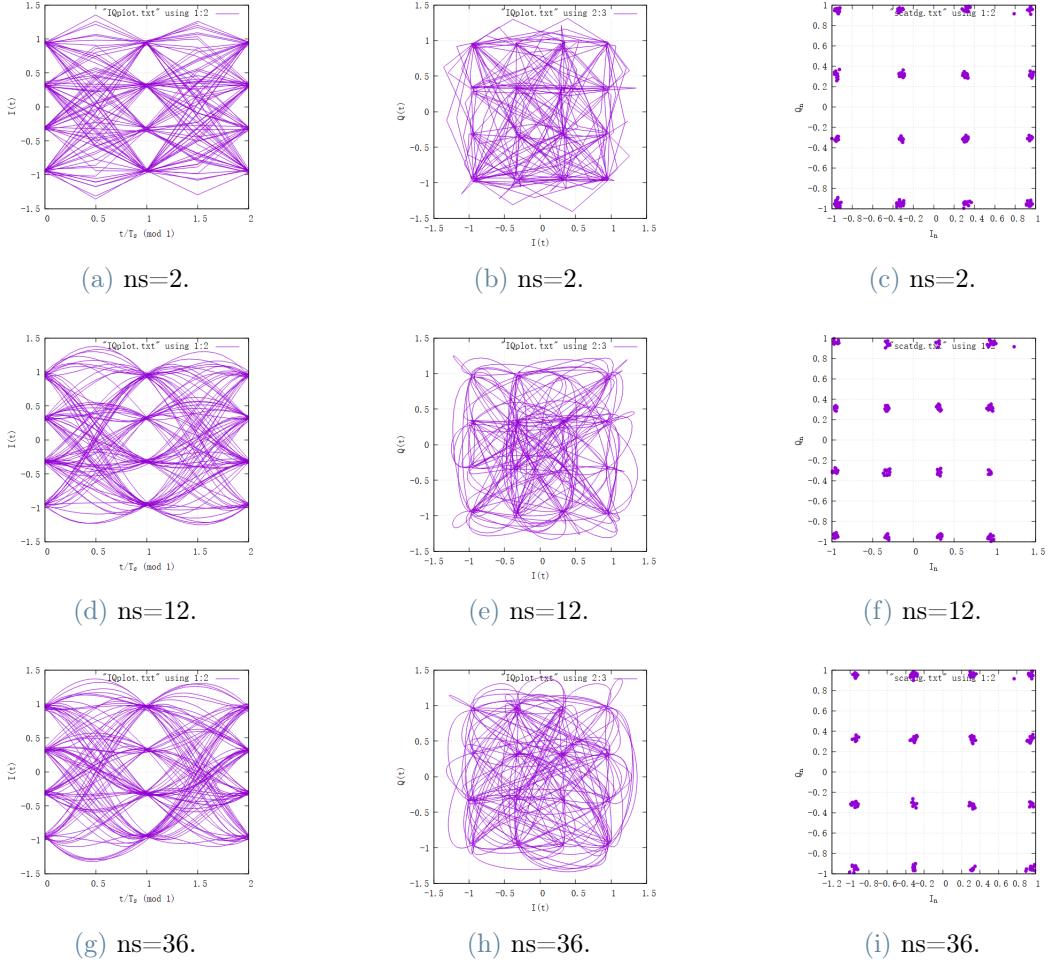


Figure 1.15: The diagrams with different Samples per symbols (16QAM)

Table 1.4: The set parameters for variation of ns

Parameter	Value
<b>Modulation bits</b>	4
<b>SRRC roll-off (ro)</b>	0.5
<b>Length of filters (L)</b>	10
<b>Eb/N0 [dB]</b>	25

Figure 1.15 shows the eye diagrams, IQ plots, and scattering diagrams of 16QAM with different samples per symbol, while the other set values are as shown in Table 1.4. With a higher level of samples per symbol, the curves for the eye diagrams and the IQ plots are smoother, as more samples have been transmitted within a single symbol.

## 1| Simulation of a Basic Digital Transmission System

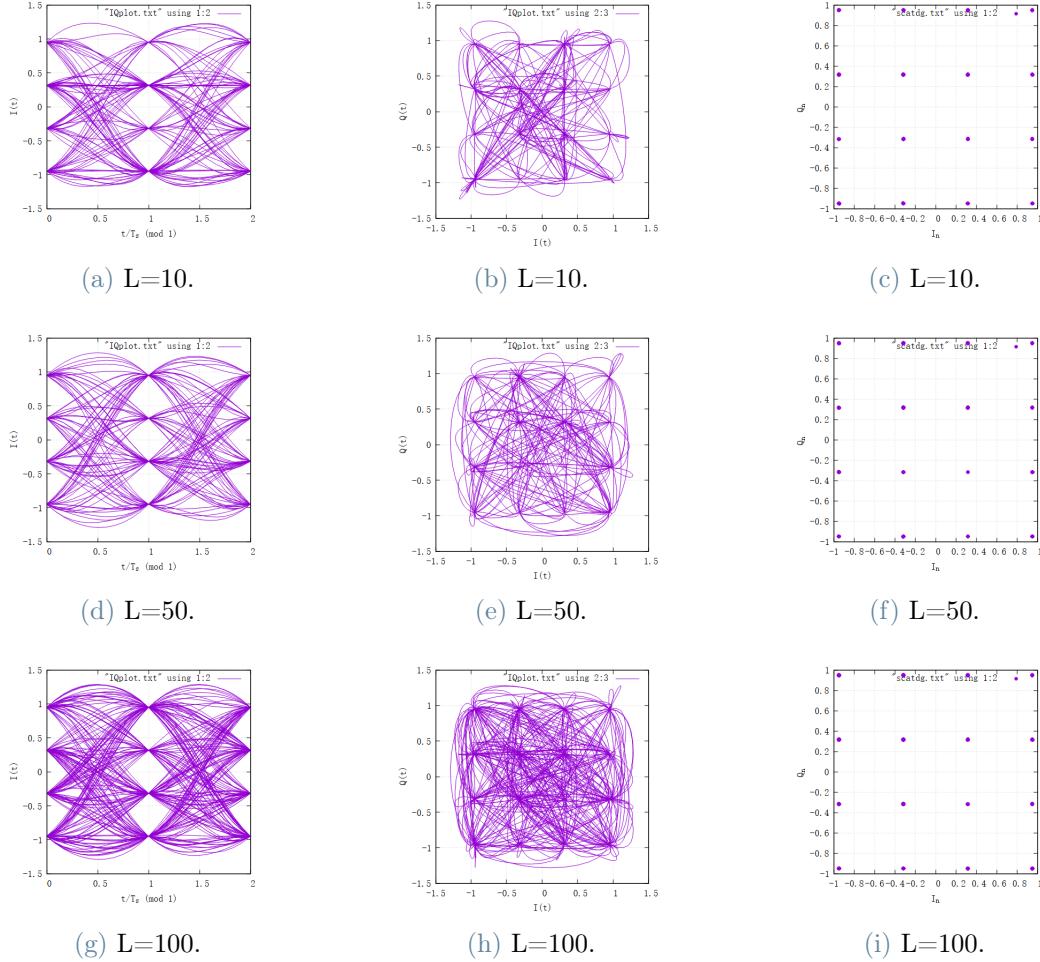


Figure 1.16: The diagrams with different lengths of the filter (16QAM)

Table 1.5: The set parameters for variation of L

Parameter	Value
<b>Modulation bits</b>	4
<b>SRRC roll-off (ro)</b>	0.5
<b>Number of samples per symbols (ns)</b>	12
<b>Eb/N0 [dB]</b>	25

Figure 1.16 shows the diagrams of 16QAM with different lengths of the filter. The other parameters are set as shown in Table 1.5. With the deeper length of the filter, the order of the filter increase, with means to get the output signal, we need to sample more input values in the past which is reflected in the eye diagram and IQ plot as it becomes denser.

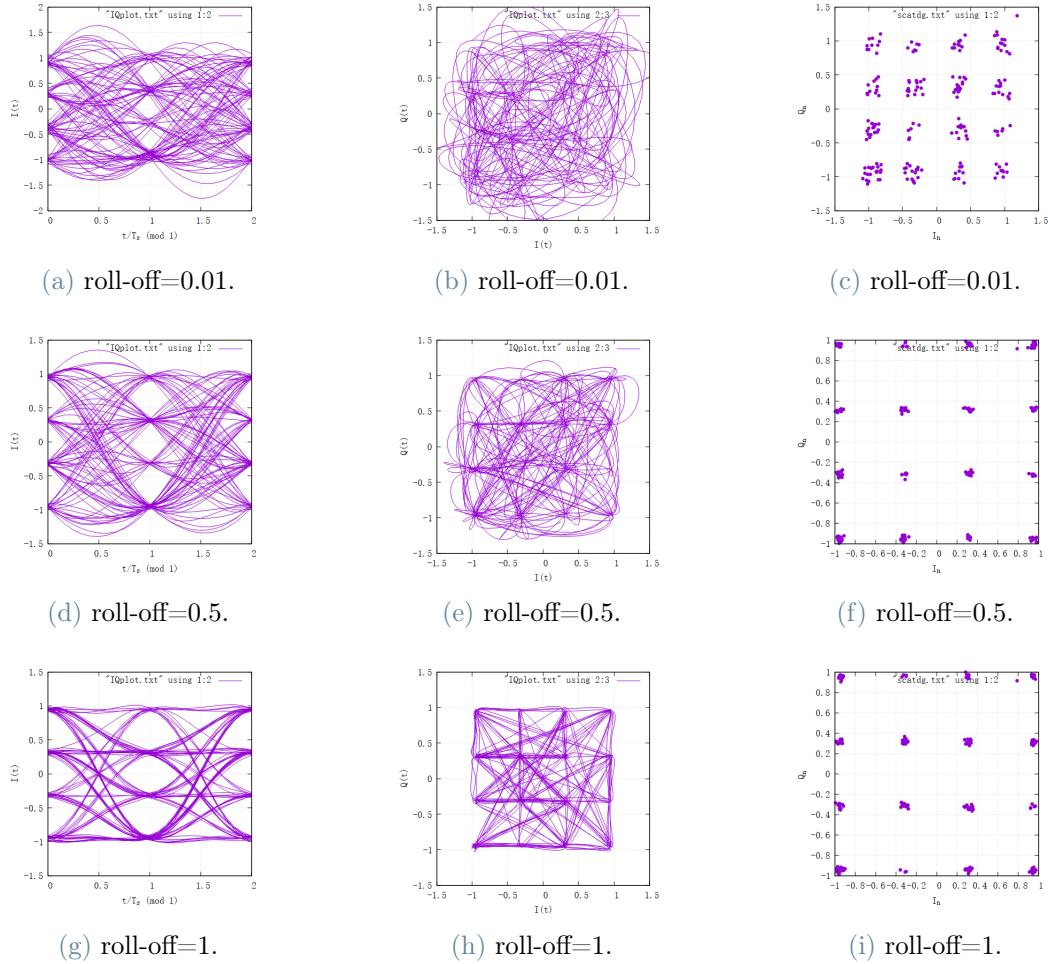


Figure 1.17: The diagrams with different roll-offs of SRRC filters (16QAM)

Table 1.6: The set parameters for variation of roll-offs

Parameter	Value
Modulation bits	4
Length of filters (L)	10
Number of samples per symbols (ns)	12
Eb/N0 [dB]	25

Figure 1.17 shows the eye diagrams, IQ plots, and scattering diagrams of 16QAM with different SRRC filter roll-offs, based on the set parameters in Table 1.6. It is clear that, with the higher roll-off, the eye diagram becomes wider and clearer, which corresponds with a lower level of errors and interference. So for the IQ plots and scattering diagrams,

## 1| Simulation of a Basic Digital Transmission System

the distributions of the traces and points are more intensive when roll-off equals to 1, compared with the lower values.

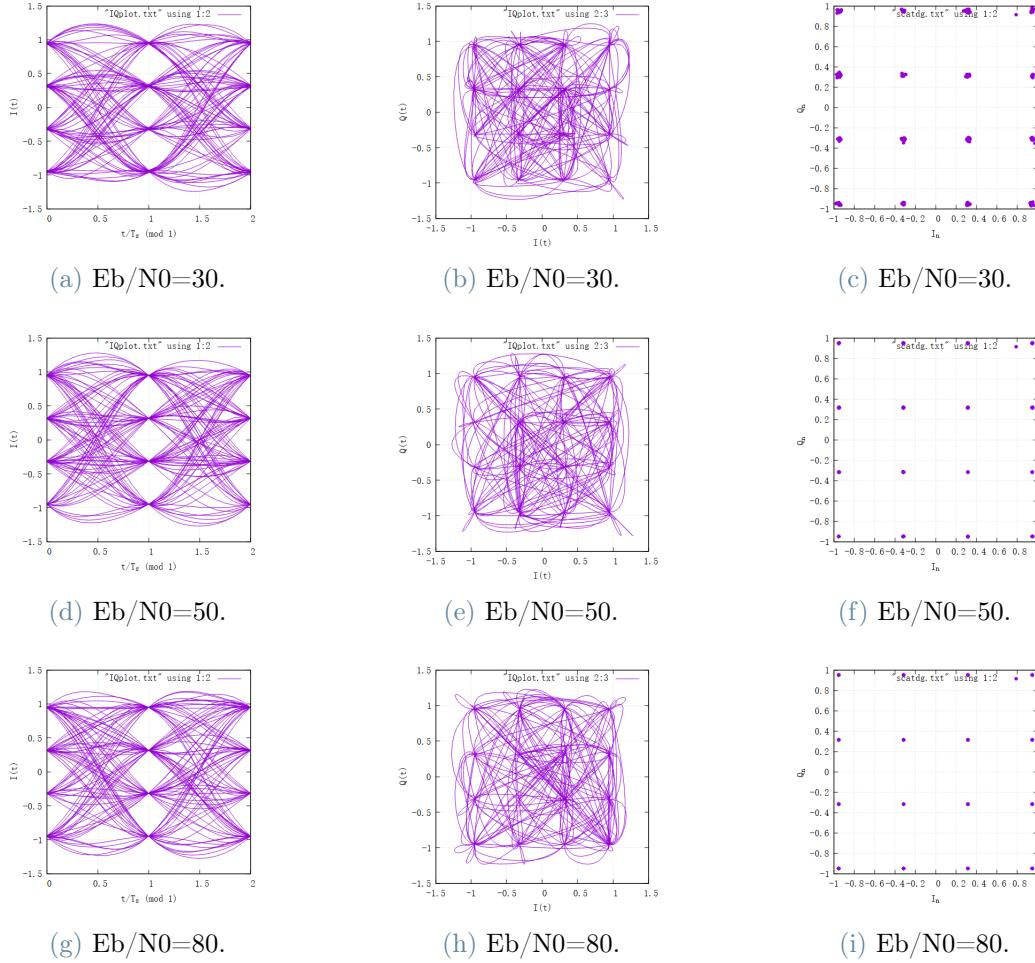


Figure 1.18: The diagrams with different Eb/N0 (16QAM)

Table 1.7: The set parameters for variation of Eb/N0

Parameter	Value
Modulation bits	4
SRRC filter roll-off (ro)	0.5
Length of filters (L)	10
Number of samples per symbols (ns)	12

Figure 1.18 shows the eye diagrams, IQ plots, and scattering diagrams of 16QAM with different Eb/N0, using the parameter as shown in Table 1.7. It is evident that, with the

increase of  $E_b/N_0$ , the deviation of the transmitted points is smaller, especially when observing the scattering diagrams. The eye diagrams and IQ plots aren't showing many differences with this set of parameters, but still can be witnessed with slightly clearer plotted curves.

### 1.2.4 Spectrum of the signals

Generally, the signal spectrum describes the characteristics of the signal's magnitude and phase as a function of frequency. In this section, the spectrum of signals with different parameters has been specifically discussed. In our terms, the spectrum implements the power spectral density (PSD) probe, with the horizontal axis as  $f/R_s$ , as the ratio of frequency and the sampling rate, and the vertical axis represents the power of the signal spectrum in the unit of decibels. Each spectrum is displayed by three different color lines, respectively indicating the signal from the receiver (purple), the signal from the transmitter (green), and SRRC filter processed signal from the receiver (blue).

- Spectrum with different modulations:

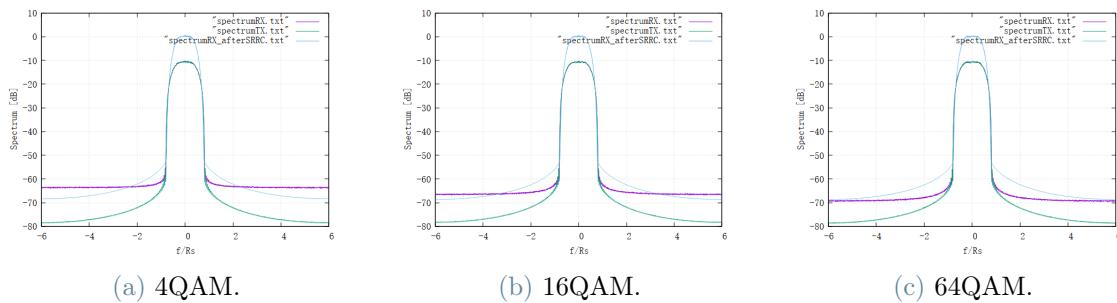


Figure 1.19: The spectrum of the QAM signal

Table 1.8: The set parameters for spectrum of QAMs and PSKs

Parameter	Value
<b>SRRC filter roll-off (ro)</b>	0.6
<b>Length of filters (L)</b>	10
<b>Number of samples per symbols (ns)</b>	12
<b><math>E_b/N_0</math> [dB]</b>	50

The first discussed spectrums are, again, with the different modulations of the signal,

QAMs and PSKs, respectively shown in Figure 1.19 and Figure 1.20. The initial set parameters for both of them are displayed in Table 1.8.

For the QAM modulations, the levels and shapes of the spectrum with 4QAM, 16QAM, and 64QAM are quite similar. The only slight difference depends on the spectral level of the initially received out-band signals, which becomes more negative (the purple one) with increasing modulation efficiency. As mentioned before, the larger modulation efficiency leads to a larger BER existing. These tendencies result from, that the total transmitted energy remaining the same, the higher cardinalities modulation could be more susceptible to the noises. So when the cardinality increases, the transmission system is becoming more vulnerable to resistant interference. This situation can be well eliminated with the SRRC filters.

The same situation also occurs with the PSK modulation. The spectrum of the RX signals varies from about -64dB to -70dB, from 2PSK to 8PSK.

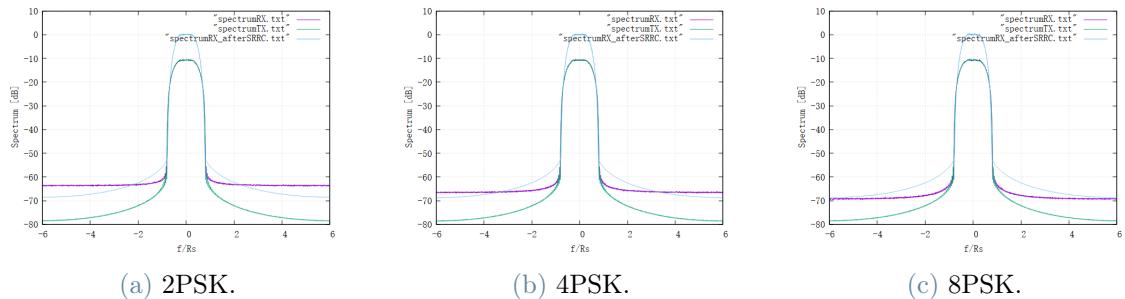


Figure 1.20: The spectrum of the signal with different modulation cardinalities

- Spectrum with different samples per symbol:

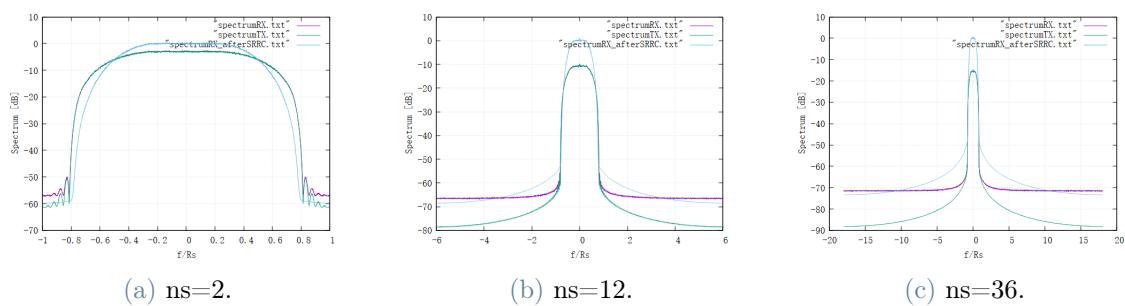


Figure 1.21: The spectrum of the signal with different samples per symbols (ns)

Table 1.9: The set parameters for spectrum of different ns

Parameter	Value
Modulation bits	4
SRRC filter roll-off (ro)	0.6
Length of filters (L)	10
Eb/N0 [dB]	50

Table 1.9 and Figure1.21 shows the Spectrum of 16QAM with different Samples per symbols (ns). Samples per symbol determine the acquisition bandwidth. With the increase of symbols per symbols, we can get more spectrum information regards the frequency. IF NS is not big enough, the systems can't see if the signal is inside the bandwidth. Only when the NS is big enough, then we can distinguish it.

- Spectrum with different filter length:

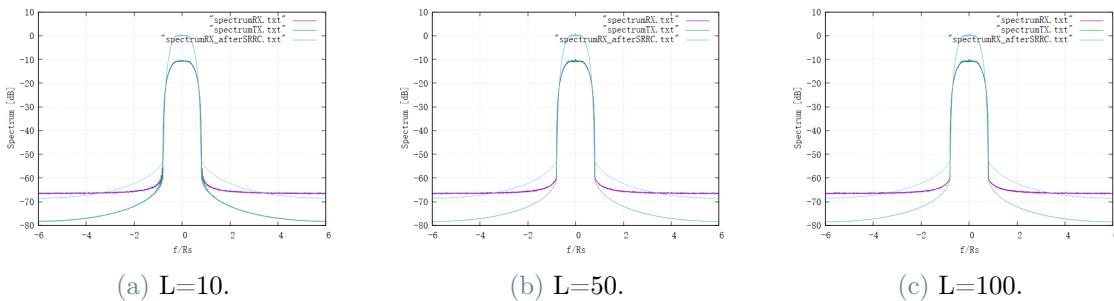


Figure 1.22: The spectrum of the signal with different lengths of filters

Table 1.10: The set parameters for spectrum of different L

Parameter	Value
Modulation bits	4
SRRC filter roll-off (ro)	0.6
Number of samples per symbols (ns)	12
Eb/N0 [dB]	50

Figure 1.22 shows the Spectrum of 16QAM with different lengths of filters, with the other set parameters in Table 1.10. The values of the filter length ( $L$ ) vary from 10 to 50, and 100, with the increment of  $L$ , the number of ripples can be witnessed as decreasing. The signal of the transmitter after the SRRC filter is more obvious. And a filter with a long length will obtain a better performance, which is more attenuation of the sidelobes, shorter transition bandwidths and fewer ripples.

- Spectrum with different roll-off:

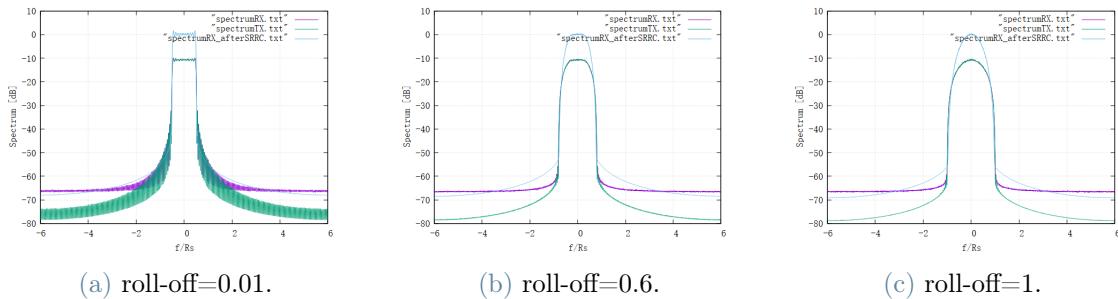


Figure 1.23: The Scattering diagram for different roll-offs of SRRC filters (16QAM)

Table 1.11: The set parameters for spectrum of different roll-offs

Parameter	Value
Modulation bits	4
Length of filters (L)	10
Number of samples per symbols (ns)	12
Eb/N0 [dB]	50

Figure 1.23 shows the Spectrum of 16QAM with different SRRC filter roll-offs, respectively 0.01, 0.6 and 1. The other set parameters are as shown in Table 1.11. The lower value of the roll-off is related to better spectral efficiency, but there is a trade-off relationship, as shown in 1.23a. The behavior of ripple with roll-off = 0.01 is also heavier than the other higher values, even if the slope of the band is the steepest one, and the spectrum is rather flat inside the band. The system will introduce more error and interference when the roll-off is low.

- Spectrum with different Eb/N0:

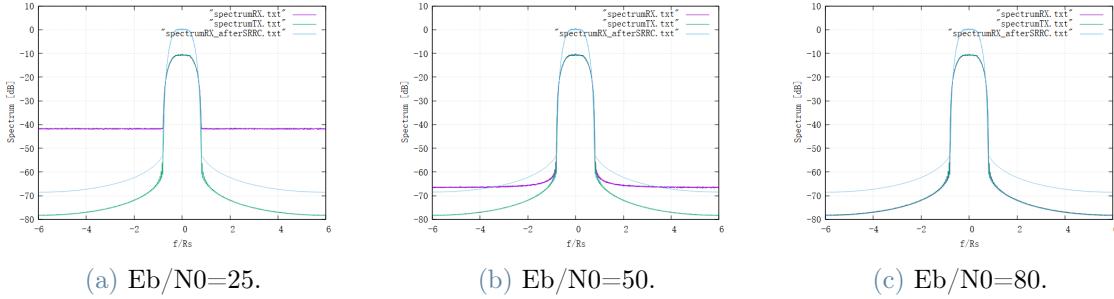


Figure 1.24: The spectrum of the signal with different Eb/N0

Table 1.12: The set parameters for spectrum of different Eb/N0

Parameter	Value
Modulation bits	4
SRRC filter roll-off (ro)	0.6
Length of filters (L)	10
Number of samples per symbols (ns)	12

Figure 1.24 shows the Spectrum of 16QAM with different Eb/N0. The related parameters are in Table 1.12. With the increase of the Eb/N0, the spectrum level of the out-of-band receiving signal moves more on the negative, and when the value of Eb/N0 is larger enough as shown in Figure 1.24c, the spectrum of RX and the TX one are nearly coincident. Whereas the spectrum of TX and the RX one after the SRRC filters does not change much during the variation of Eb/N0. This phenomenon is due to the lower level of errors and noises for larger Eb/N0.



# 2 | Implementation of a DSP C++ Class

During the third and fourth laboratories, we analyzed a given digital transmission system to implement a certain particular block by defining the corresponding C++ class, starting from scratch. We also checked the class functioning by comparing Spectra, Eye diagrams, Scattering diagrams and BERs between the original system and the one containing the block defined by us, which in our case is the PN\_Source block.

## 2.1 BER example

In Figure 2.1, we implemented the "my\_ber\_meter" as the beginning of the study on how to implement the c++ code. The variations of the parameter BER are shown, respectively with the QAM modulations and the PSK modulations. Among the plotted curves, the one for 4QAM and 2PSK coincide, which results from the same pattern of the two constellations, as discussed in chapter one.

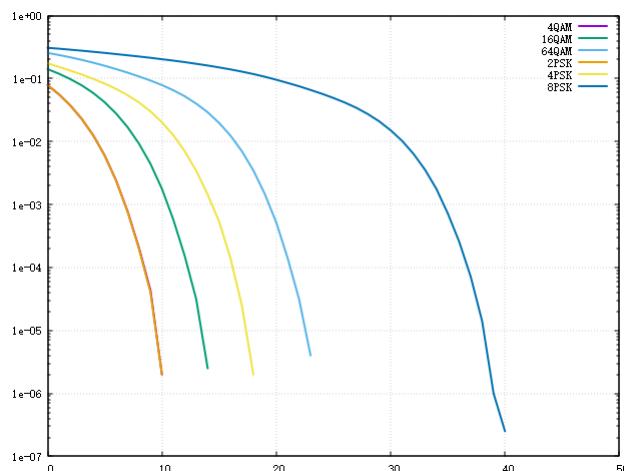


Figure 2.1: BER of MY\_BER\_METER

The represented BER results as shown in Figure 2.2 are simulated with  $E_b/N_0$  as 20,

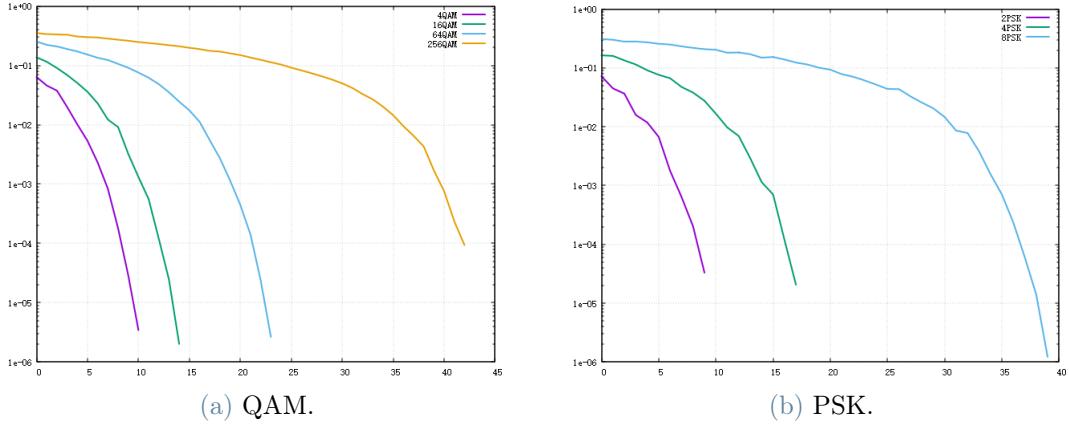


Figure 2.2: BER results with different modulations of reference

roll-off as 0.8, length of the filter as 10, and the number of samples per symbol as 6.

Similar to the previous BER results, the general statistics of BER with all the modulation types are decreasing with the increase of the  $E_b/N_0$ , and with the higher modulation efficiency the BER values are higher.

## 2.2 PN\_Source

### 2.2.1 Algorithm description

For this part, we implement a PN\_Source. Essentially, PN\_Source can be seen as a liner feedback shift register. We choose to apply an 8th-order LFSR using a primitive polynomial.

$$1 + x + x^5 + x^6 + x^8 \quad (2.1)$$

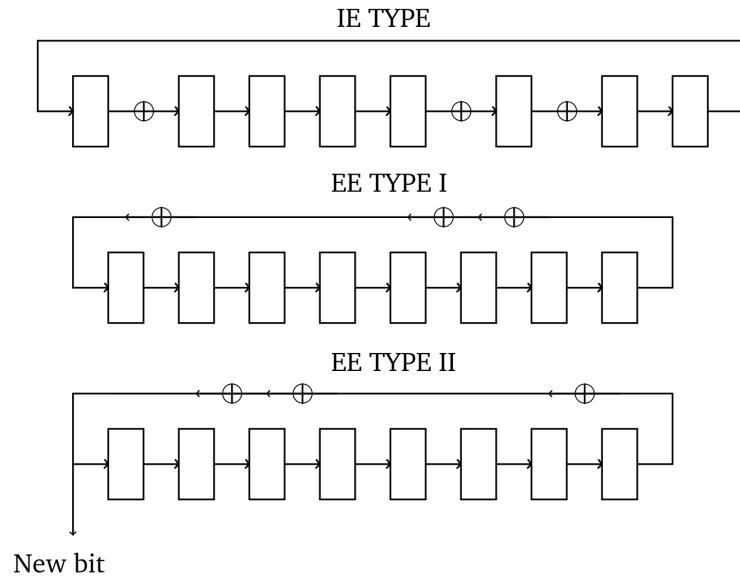


Figure 2.3: PN\_source use a primitive polynomial of 8th order

The structure implemented in the block PN\_source is the right side one in Figure 2.3. The IE TYPE is the direct structure using the primitive polynomial. Then it can transform to the EE TYPE I, their function is the same. The EE TYPE II is the algorithm we apply in our PN source. It is slightly different from EE TYPE I, just loop direction.

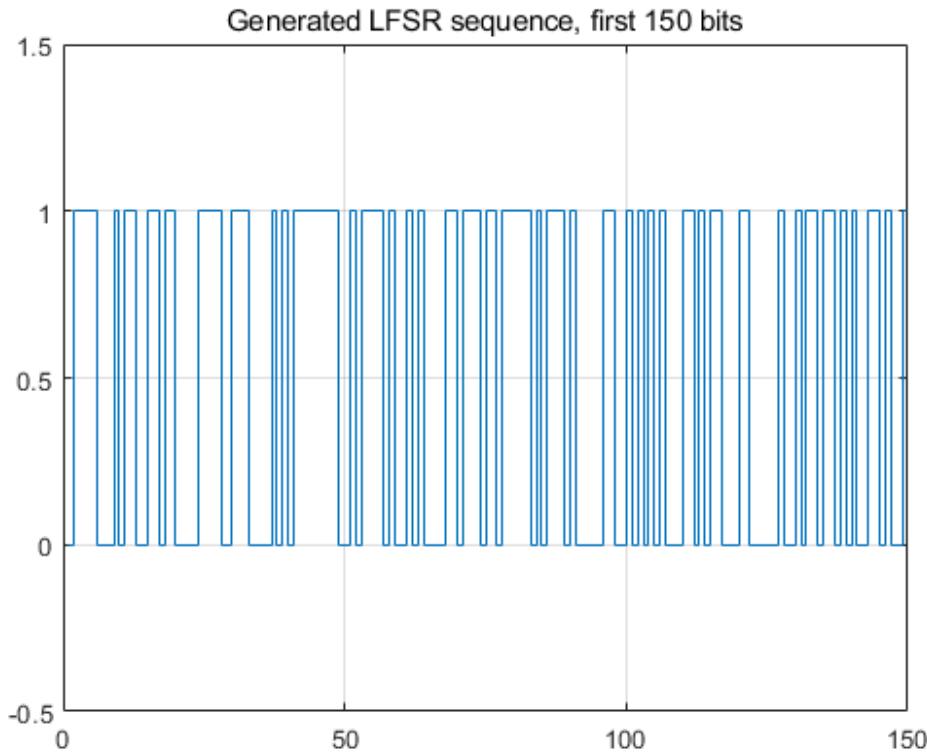


Figure 2.4: First 150 bits generated from PN\_source

The generated results from the above block are as shown in Figure 2.4, where only the first 150 bits are presented due to the quite large size of the full data.

### 2.2.2 Functionalities

The original functionality of PN\_source provide in lab 4 is to provide a stream of random value. The value type is the integer type, but only store 0 and 1. And every time, it provides  $N*m$  values.  $N$  is the frame size, and  $m$  is the modulation efficiency. Inside it, having two key parameters. One is the  $N*m$  indicates the number of bits we need to produce. Another one is a pointer, which has a fixed address value and needs to be used to indicate the address of the space where values will be stored in. Based on the previous one, our PN\_source realize that function.

### 2.2.3 Source code (header and implementation)

- Header file

The header file contains the declaration of all the variables and the methods of the

class. class is a user-defined data type that we can use in our program, and it works as an object constructor or a "blueprint" for creating objects. In class MY\_PN, we only define a function run, which will be called in the main function and used to generate N\*m random numbers. N is the frame size, and m is the modulation efficiency.

```
class MY_PN
{
public:
    static int* addout;

    MY_PN();

//! Run the PN source generating random bits
void Run(
    int tics,
//!< Number of generated bits
    int* addout
//!< Pointer to buffer to store generated bits
);

private: // Place here variable of the class
    int tapbit;
    int seed;
    int i;
    int tics;
};

• Run
```

The Run method is contained in MY\_PN.cpp file. Inside the Run method, all the computational operations are performed: generate the tapbit, and store it in the space to which the pointer points to.

```
void MY_PN::Run(
    int tics,
//!< Number of generated bits
    int* addout)
```

```

//!< Pointer to buffer to store generated bits
{
    static int seed = 30;
    //The initial value of the data stream (8bit trigger)
    for (i = 0; i < tics; i++)
    {

        //Generate new bit which will be used as output bit
        tapbit = (seed ^ (seed >> 1) ^ (seed >> 5) ^ (seed >> 6)) & 1;
        //New value of the data stream (8bit trigger)
        seed = (seed >> 1) | (tapbit << 7);
        //Store newbit to the memory space pointed to by the pointer
        *(addout + i) = tapbit;
    }
}

```

#### 2.2.4 Validation tests

- Eye diagram, IQ plots, scattering diagram

With the use of the newly written code of the block *PN\_Source*, the generated diagrams and plots with different parameters are shown in the figures below. Compared with the plots generated with the original codes, the variations of the figures are similar, which proves the code's validation.

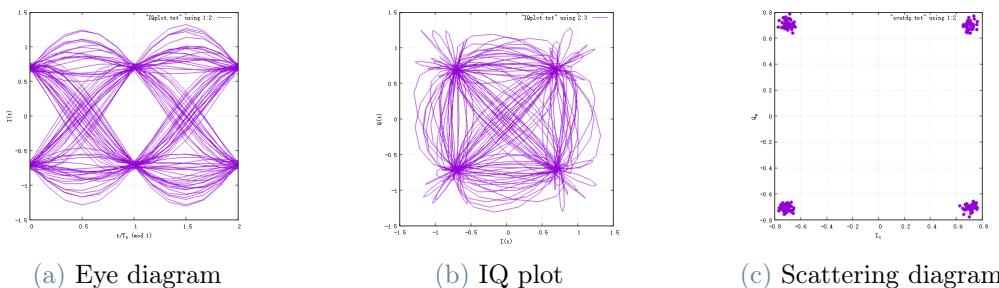


Figure 2.5: The diagrams of 4QAM with samples per symbol=6, length of filter=10, roll-off=0.25, Eb/N0=25

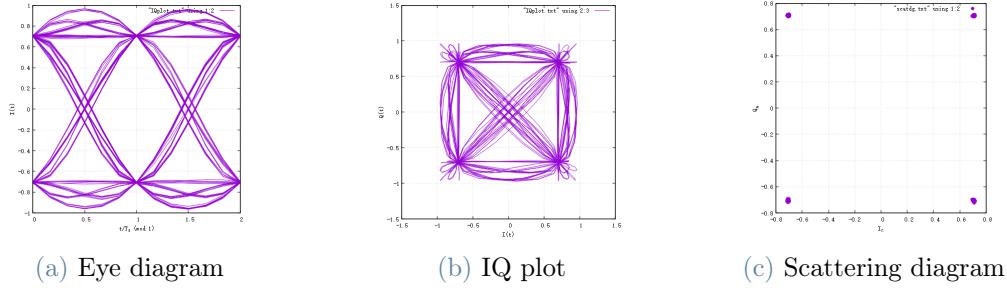


Figure 2.6: The diagrams of 4QAM with samples per symbol=6, length of filter=10, roll-off=0.6, Eb/N0=40

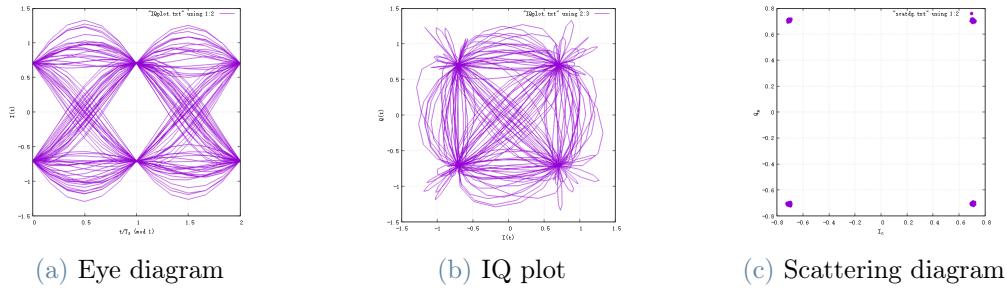


Figure 2.7: The diagrams of 4QAM with samples per symbol=6, length of filter=10, roll-off=0.25, Eb/N0=40

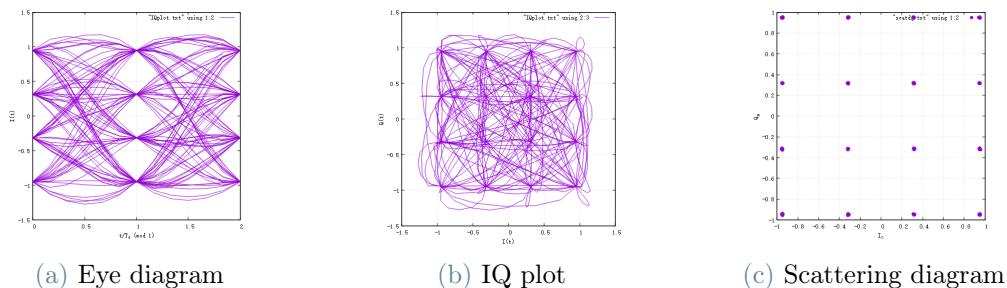
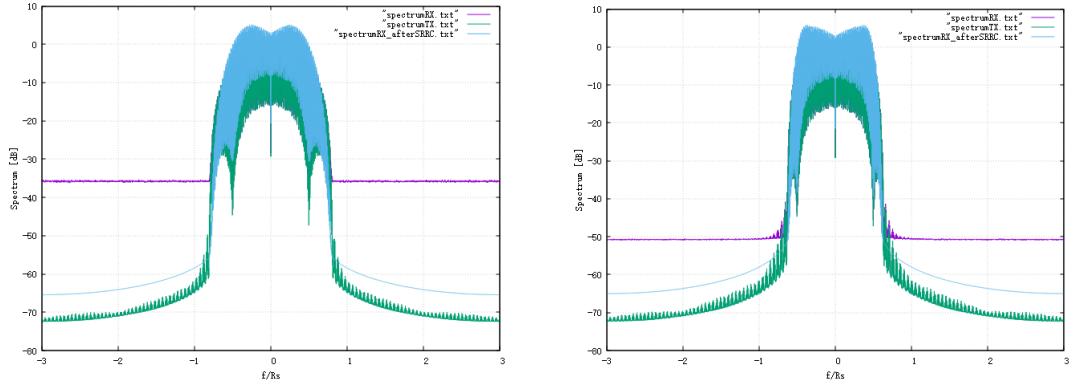


Figure 2.8: The diagrams of 16QAM with samples per symbol=6, length of filter=10, roll-off=0.6, Eb/N0=40

- Spectrum



(a) Spectrum of 4QAM, with  $E_b/N_0=25$ , roll-off=0.6, length of filter=10, samples per symbol=6

(b) Spectrum of 4QAM, with  $E_b/N_0=40$ , roll-off=0.25, length of filter=10, samples per symbol=6

Figure 2.9: Generated spectrum

The generated examples of the spectrum are shown in Figure 2.9. We can find that there are many spectral components in the transmission of signals. In the case of 4QAM modulation, with the increasing roll-off, there are more components and lower signal reception power

# 3 | A Realistic Receiver for a Realistic Wireless Channel

In this chapter, the parameters from lab 5 have been involved. Focusing on a more realistic receiver, this task is performed with several experiments with changing the additional parameters for the narrow-band model channel.

The TX, RX and narrow-band block diagrams for implementation are first discussed in this chapter, followed by some relevant plots from the original codes. The frame structure with the pilot has then been discussed. According to the arrangement of tasks, our tests are about Costa's Loop with variations of phase estimate, the updating step of PLL ( $\alpha_p$ ), the weight of decision-aided error for PLL ( $w_p$ ), and the Pilot length, which have been discussed in the last part of this chapter.

## 3.1 Narrowband channel model results

### 3.1.1 The block diagrams of TX and RX

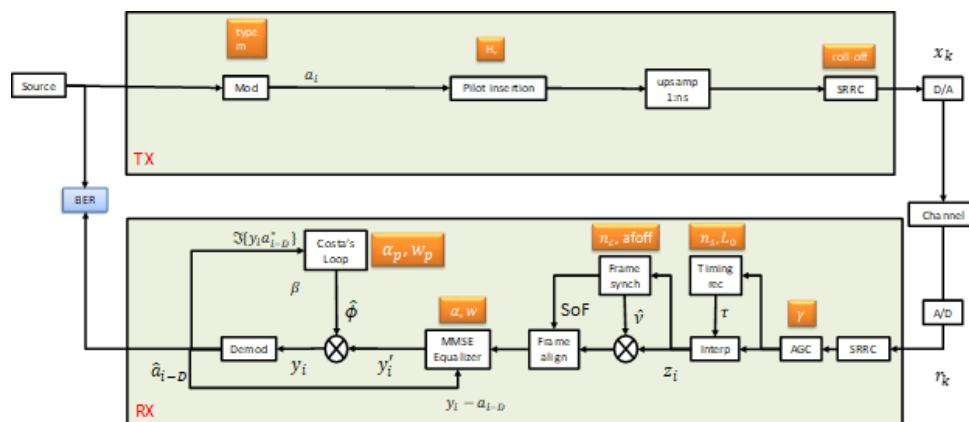


Figure 3.1: The block diagram of the complete TX and RX system to be simulated

In figure 3.1, the block diagram of transmitting and receiving implemented in this section is displayed. Actually, a realistic receiver must deal with several additional impairments introduced by the wireless channel and by the non-ideal RF front end at TX and RX.

- Mismatched and unstable oscillators at the TX and RX (frequency and phase jitters)
- Mismatched and unstable A/D and D/A converters at the TX and RX (timing jitters)
- Additive noise
- Linear channel distortions
- Time varying attenuations and phase rotations introduced by the channel
- Unknown delay introduced by the channel

In this system, several new blocks have been inserted.

- Pilot insertion: as the first one to be encountered, and the pilot insertion is corresponding with a change in the transmitting frame structure. It implements the functionality of a transmitter block that inserts pilots into a defined packet with useful data. This new frame structure will be specifically discussed in section 3.1.4.
- AGC (automatic gain control): is situated after the A/D block in the receiving chain. The AGC block is in charge of equalizing the running average power of a signal to a desired level. By using this block, the undesired numerical problems can be avoided.
- Timing recovery: is used to get the optimum sampling instant  $\tau$ , for the following *interpolating* block.
- Interpolating: creates the sequence  $Z_i$  by interpolating the sequence of the incoming A/D samples, which is behaving as the former down-sampling blocks in the simpler systems.
- Frame synchronization: implements the finding of the position of the header (the starts of the frames, SoF), but sometimes the frame synchronizer is still affected by these impairments. We need to consider the frequency offset, so the frequency offset  $\hat{v}$  need to be estimated at the same time.
- Frame align: is to align the frame base on the former blocks.
- MMSE equalizer: the minimum mean square error equalizer, acting as a digital filter, the purpose of this block is to estimate the varied amplitude between the aligned

frame and the output of the demodulation after Costa's loop and compensate for the linear distortion introduced by the wireless channel.

- Costa's loop: is known as the phased locked loop (PLL) in the digital domain, is able to estimate and remove the residual frequency and phase error on the received data.

### 3.1.2 Description of narrow-band channel model

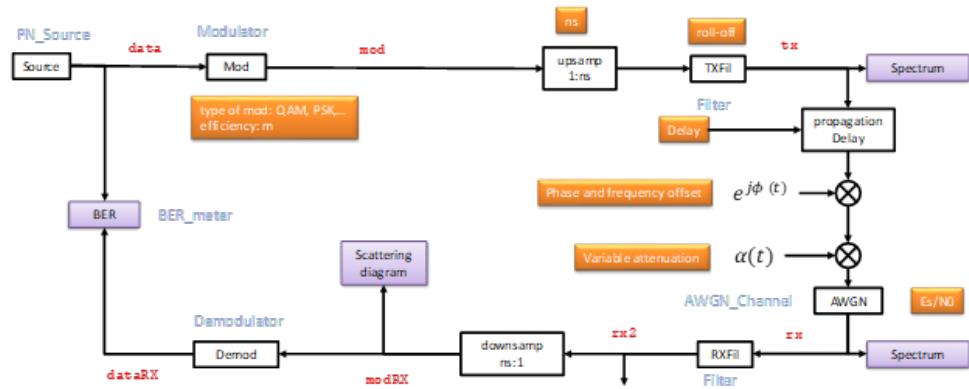


Figure 3.2: Block diagram of the narrow-band model

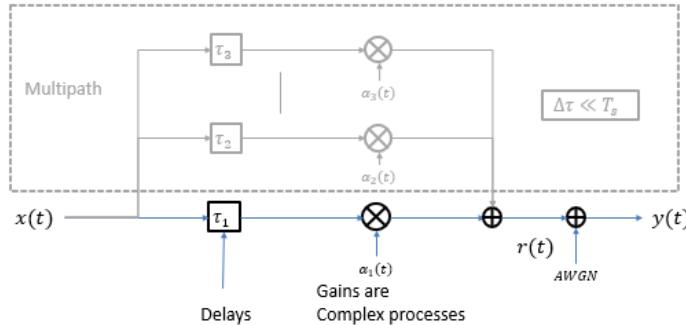


Figure 3.3: Single-tap model for the narrow-band channel system

In this narrow-band channel model, the transmitting channel between the transmitting and receiving chain is not only represented by the AWGN channel but also affected by the unknown propagation delay (D). This results from the finite propagation speed of light when waveform travels across space, which introduces the propagation delay.

Moreover, the frequency and phase offset have also been considered, which introducing a product of complex sinusoidal waveform,  $e^{j\phi(t)}$ , as a rotation of the complex value for the

phase, in order to mix the in-phase and quadrature components. The term  $\phi(t)$  can be written as,

$$\phi(t) = 2\pi v_f t + \theta \quad (3.1)$$

Also, a variable attenuation  $\alpha(t)$  is involved, which is real and positive. During the simulation, this can be expressed as,

$$\alpha(t) = 1 + A \sin(2\pi v_a t) \quad (3.2)$$

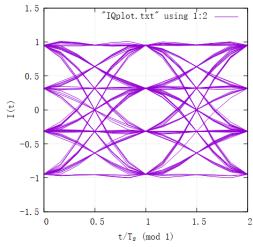
Therefore, the receiving signal can be expressed as,

$$r(t) = \alpha(t) e^{j\phi(t)} x(t) + n(t) \quad (3.3)$$

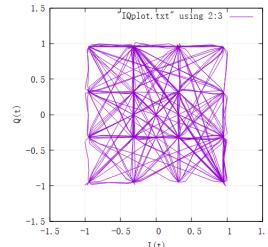
### 3.1.3 Some relevant plots with comments

Here, several plots are displayed with the given narrow-band model program, with the variation of the following three parameters, while the other parameters have been set as reasonable values in the 'input' file.

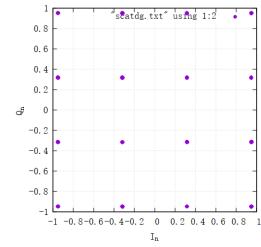
- Normalized frequency offset ( $nuf$ ): inserting a level of frequency offset in the system corresponds to the  $v_f$  in the above formulas.
- Amplitude fluctuation frequency ( $nua$ ): corresponds to the  $v_a$  in the above formulas.
- Amplitude peak-to-peak fluctuation ( $A$ ): adding fluctuations in the amplitude signal.



(a) Eye diagram.



(b) IQ plots.



(c) Scattering diagram.

Figure 3.4: Plots with  $nuf=0$ ,  $nua=0$ ,  $A=0$  for 16QAM

Figure 3.4 shows the eye diagram, IQ plot, and scattering diagram without any fluctuations or offsets, for 16QAM modulation. All the figures are in a well and decent condition.

It shows a standard performance for this communication system, and is to be compared with the following figures.

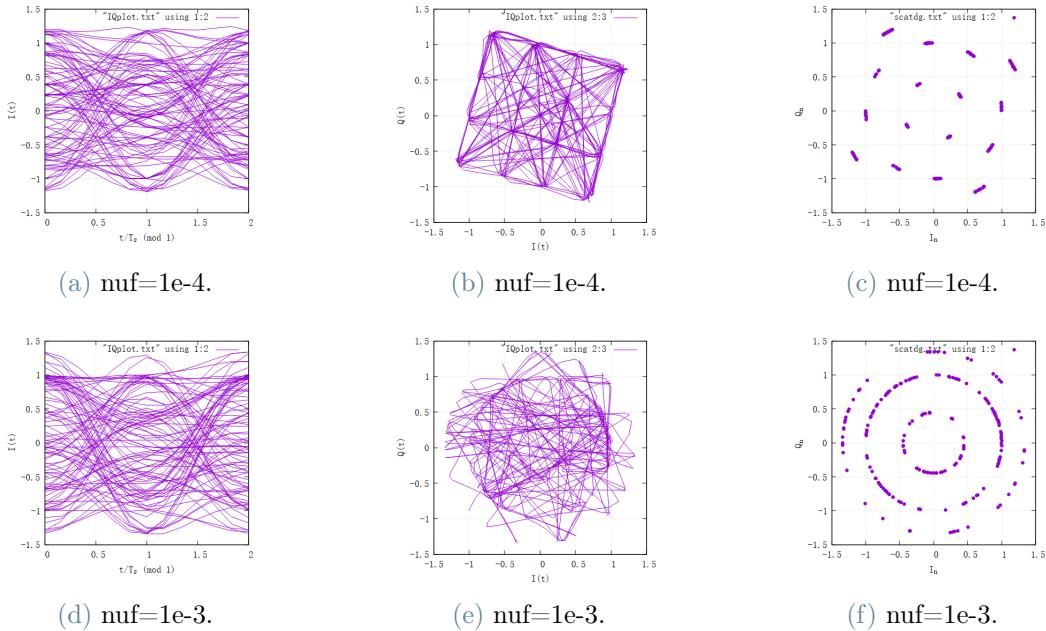


Figure 3.5: Plots with different normalized frequency offset

Figure 3.5 shows the diagrams with the variations of the normalized frequency offsets (nuf), while the other values are set to zero.

Observing from the IQ plot and scattering plot, the pattern increasingly rotates with a change of the frequency offset, from 1e-4 to 1e-3. Persisting chaos is also happening in the eye diagrams. For the eye diagram, the opening of the eye in the middle is hard to be visible which means the fluctuations are huge.

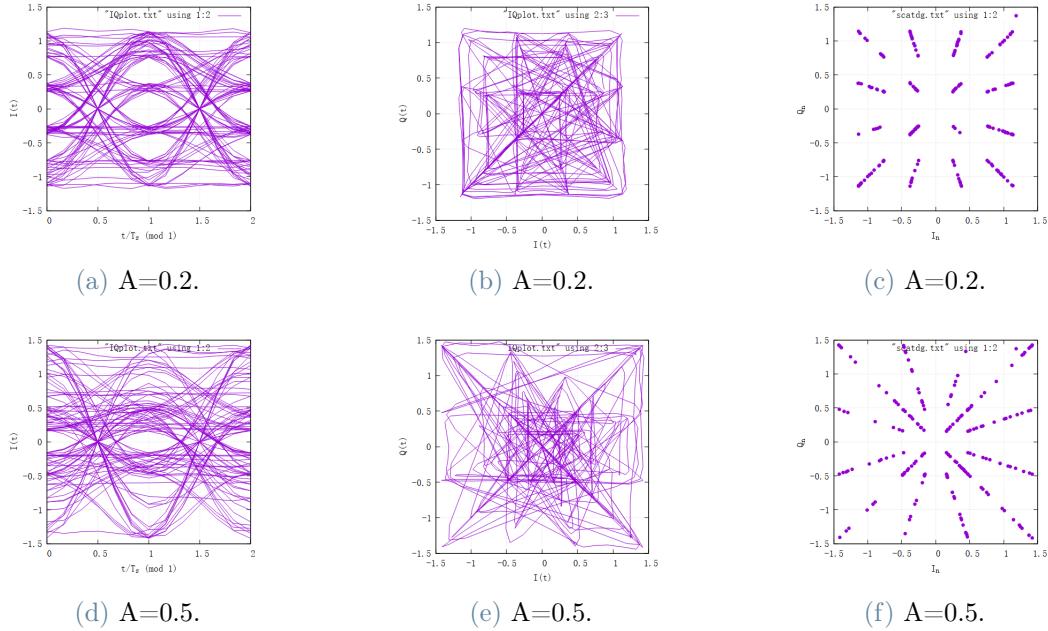


Figure 3.6: Plots with different amplitude peak-to-peak fluctuations

Figure 3.6 shows the changes with the variations of the amplitude peak-to-peak fluctuations ( $A$ ), while the normalized frequency offset ( $nuf$ ) is zero and the amplitude fluctuations frequency ( $nua$ ) is 0.01. The curves and elements in the IQ plots and scattering parameters show behaviors of radial variation and it would be intensified when the value of the amplitude fluctuation rises. While for the eye diagram, the lines also messed up with more branches and less opening area. with the increase of the fluctuations, those points become more like lines which means the system is not able to compensate for the fluctuation we introduced. A possible solution is to increase pilot density.

### 3.1.4 Description of frame structure (header, pilots..)

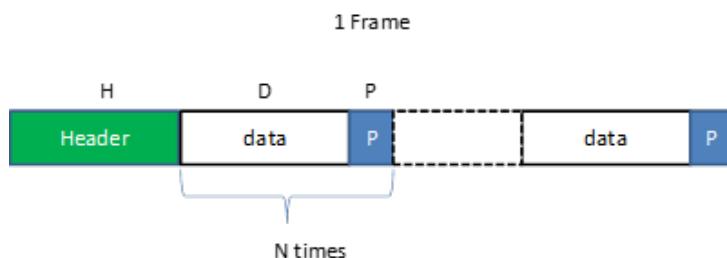


Figure 3.7: Pilot structure

The frame structure with the pilot is as shown in figure 3.7.

The definition of "pilot": some periodically inserted symbols, to compensate for the effect of the channel.

The Pilot can be seen as a part of the defined packet that does not contain any data, but there will be some characteristic symbols for identification.

Usage:

1. Locate the location of the data packet through the header, which provides a periodic update of the channel estimation.
2. The change of channel makes influences the length of the Pilot.

The pilots are mainly used for:

- Frame Synchronization
- Frequency Synchronization

Pilot overhead: 1. Pilots are a waste of energy and throughput, as they don't carry information 2. Their use must be reduced as much as possible, compatibly with the requirement of having a reliable channel estimation

$$\text{Pilot overhead} = \frac{\text{number of pilots per packet}}{\text{packet length}} = \frac{H + NP}{H + N(P + D)} \quad (3.4)$$

N should be large and P << D.

## 3.2 Test of the considered receiver block

### 3.2.1 Algorithm description

The test to be performed is about Costa's loop for our group. This block behaves as the PLL (Phase Locked Loop) in the digital domain, which is aimed at phase synchronization. It performs the estimating and removing of the residual phase error on the received data, which corresponds with the 'rotation' behavior of the IQ plot and scattering diagrams as shown in Figure 3.5, with the variation of the normalized frequency offsets. Figure 3.8 shows the algorithmic diagram of Costa's loop, which can be generally expressed as in 3.5, where  $y'_j$  is the observation of the input,

$$y'_j = e^{j\phi_j} a_{j-D} + n'_j \quad (3.5)$$

Finally, the algorithm would like to reach the function 3.6, as an elimination of the phase variation term  $e^{j\phi_j}$ ,

$$y_j = a_{j-D} + n''_j \quad (3.6)$$

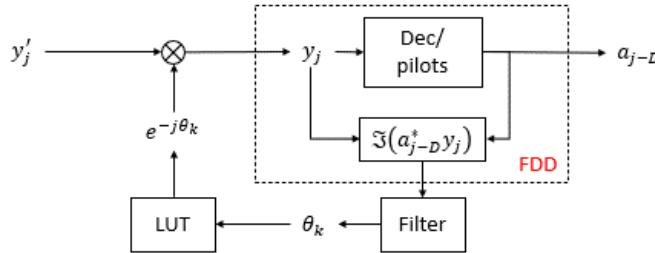


Figure 3.8: The block diagram of the Costa's loop

The phase estimation is as in the expression below, where  $\hat{\theta}_{k+1}$  is the new estimation of the compensation phase while  $\hat{\theta}_k$  represents the previous estimation, and  $a_{j-D}^*$  is the transmitted point.

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha_p \operatorname{Im} \left\{ a_{j-D}^* y_j e^{-j\bar{\theta}_k} \right\} \quad (3.7)$$

This algorithm is driven by the pilots and the decisions, as discussed in the following section.

### 3.2.2 Description of performed tests and motivations

The parameters to be performed in this part are as follows, and the performances are checked with the BER and variance of the phase ( $V_p$ ).

- Updating step of PLL ( $\alpha_p$ ): is used to control the quality and the tracking capacity, as the updating step of Costa's loop. When it is very small, the quality of the estimation of the phase is well, but the system is not able to track the variation of the phase fluctuation. When it is too large, either will the loop be not converged, or the quality of the estimation is not good enough.
- Weight of decision aided error for PLL ( $w_p$ ): When the error is evaluated using the decision, the updating step can be multiplied by the weight of the decision error for Costa's loop. When this value is set to zero, the updating of Costa's loop will only depend on the pilot's section. When it is set to one, both phase and amplitude will be tracked with the pilot and decision with the same weight.
- Pilot length (length of pilot sequence) ( $P$ ): indicates the bit length of the pilot.

The other parameters are set as in Table 3.1.

Table 3.1: The set parameters of the performed tests

Parameter	Value
<b>Modulation efficiency</b>	2
<b>Fixed SRRC filter roll-off</b>	0.6
<b>Amplitude PP fluctuations</b>	0
<b>Amplitude fluctuation frequency normalization</b>	0
<b>AGC bandwidth</b>	0

### 3.2.3 Tests results and comments

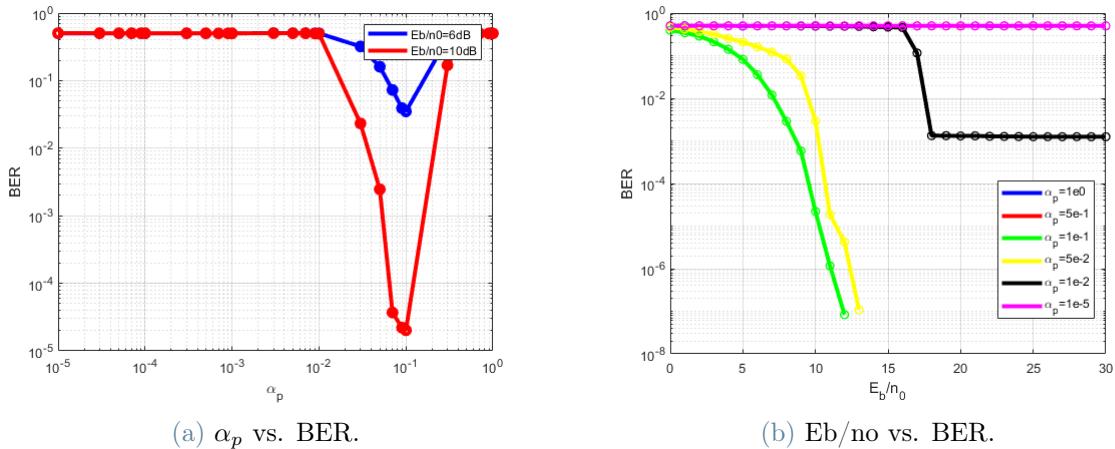
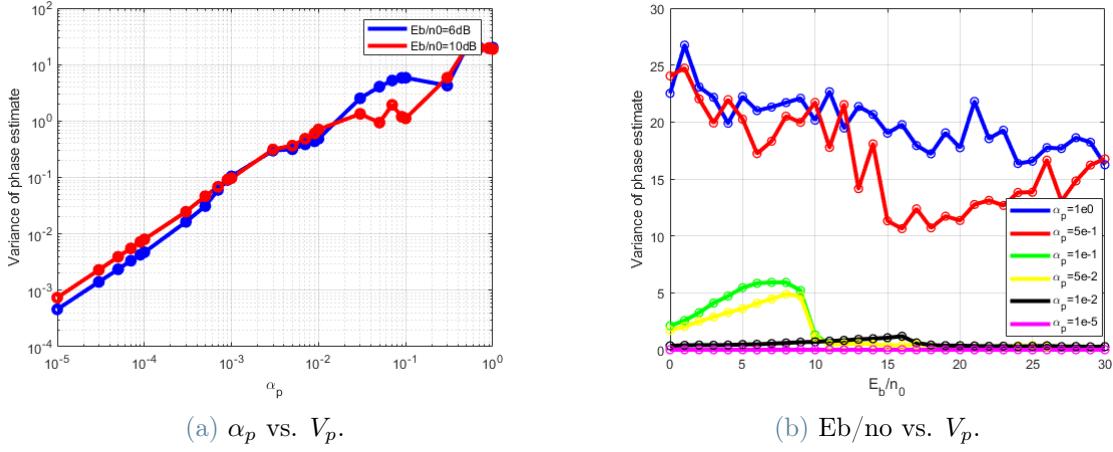


Figure 3.9: The variance of BER with  $w_p$

In figure 3.9, a) this figure indicates that for the same PLL step ( $\alpha_p$ ), a higher energy-to-noise power ratio (En/No) has lower BER and leads to better performance. b) it is visible from this figure that  $\alpha_p=10^{-1}$  (the green line) has the lowest BER compared to others thus it shows higher performance. so we can conclude that the system has the best performance when En/No = 10 dB and  $\alpha_p= 10^{-1}$ .

Figure 3.10: The variance of phase variance with  $\alpha_p$ 

In figure 3.10, here we tested the variance of phase ( $V_p$ ) against the change of PLL step ( $\alpha_p$ ) and Energy to noise power ratio ( $E_b/N_0$ ). we can observe a slight difference between the two values of  $E_b/N_0$ , except for  $10e-2$  and  $10e-1$ . However in general the higher value of  $E_b/N_0$  leads to less phase variance and for higher values of  $\alpha_p$  the phase variance seems to be very fluctuating.

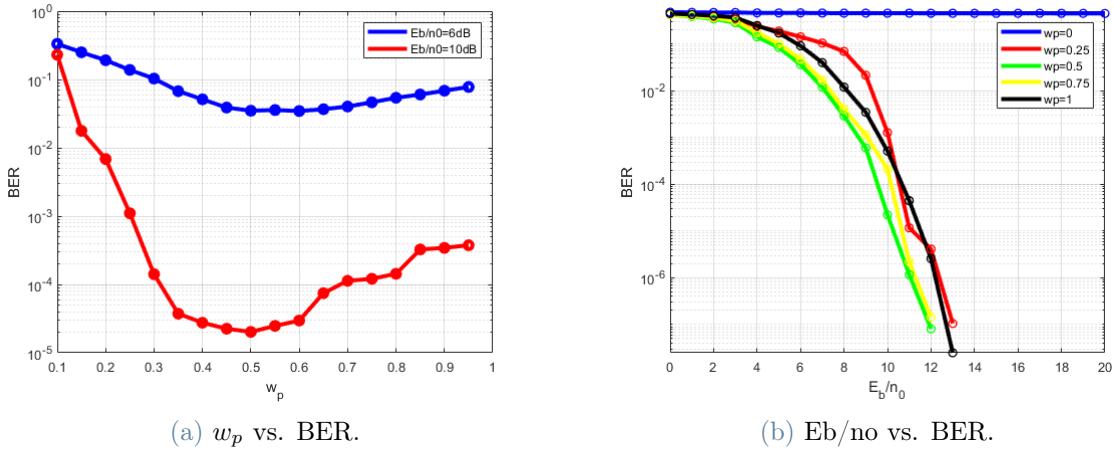
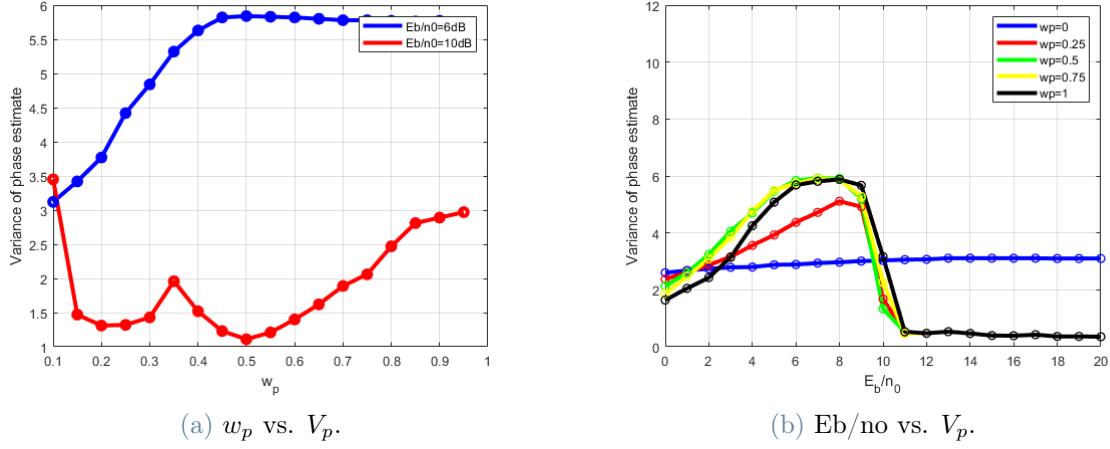
Figure 3.11: The variance of BER with  $w_p$ 

Figure 3.11 illustrates the change in BER according to the weight of decision-aided error for PLL ( $w_p$ ) and the energy-to-noise power ratio ( $E_b/N_0$ ). there is a huge difference between the two values of  $E_b/N_0$ , and 10dB has a less BER and accordingly leads to higher performance. on the other hand,  $w_p=0.5$  has the lowest BER among others so we can conclude that the best performance for the system is at  $E_b/N_0 = 10dB$  and  $w_p = 0.5$

Figure 3.12: The variance of phase variance with  $w_p$ 

In figure 3.12, we tested the variance of phase according to Eb/No and wp. Again here we see a huge difference between the two plots of Eb/No and the higher one has a better performance. Meanwhile, wp=0.25 seems to have a lower variance than the rest. however, all the non-zero values show a pick between Eb/No = 2 and 8.

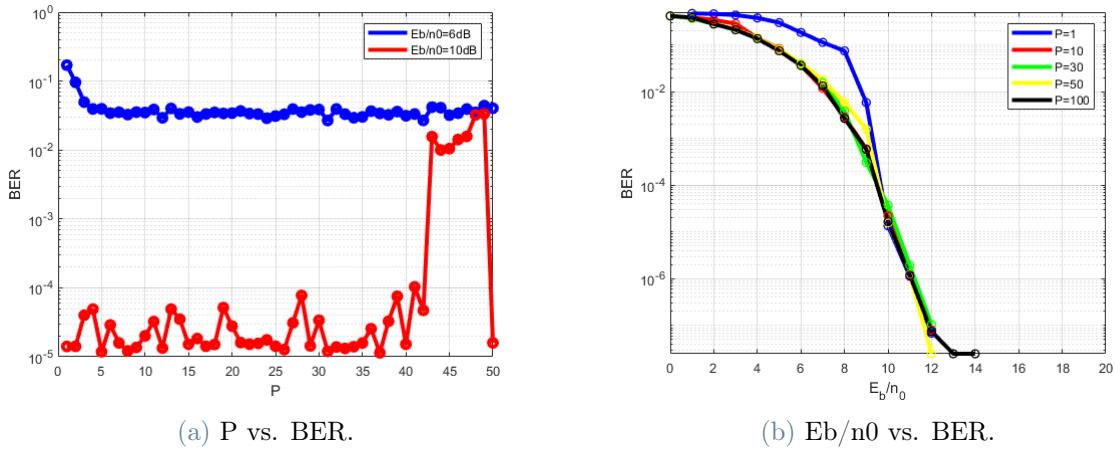


Figure 3.13: The variance of BER with P

As shown in figure 3.13, here we examined the changes of BER against the length of Pilot sequence (P) and Eb/No. As expected for  $E_b/n_0 = 10$  dB we see fewer values of BER but rather fluctuating while 8 dB leads to a higher level of BER but less fluctuating. On the other hand for all the values of P seems to be a slight difference. however, P=30 has the best performance among the rest.

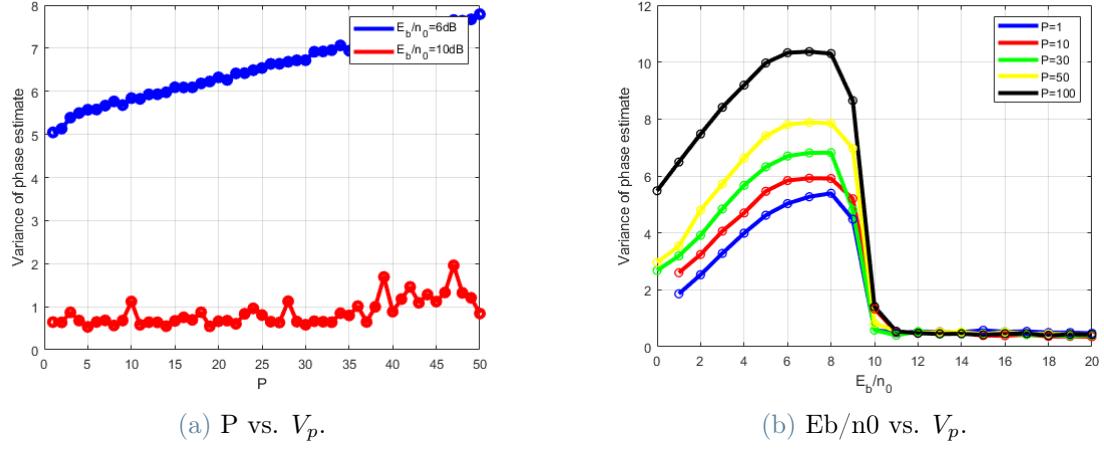


Figure 3.14: The variance of phase variance with P

Figure 3.12 shows the changes in phase variance  $V_p$  according to the length of the pilot sequence. A lower but fluctuating variance was achieved for  $E_b/\text{No}=10\text{ dB}$  and for  $E_b/\text{No}=8\text{ dB}$  it is much higher but with a stable steep. We can also see that the lowest level of phase variance is related to  $P=1$ . However, between  $E_b/\text{No}=2$  and 8 all the  $P$  values have a pick.

# 4 | USRP Results

In this chapter, a real Software Defined Radio (SDR) wireless transmission system has been implemented by using a Universal Software Radio Peripheral (USRP), which is related to the performed tests in Lab 7, about the maximum sampling rate and checking of the bottlenecks of the system.

Afterward, the wireless transmission systems using USRP from lab 8 were tested with two main measurements. One of them corresponds with an SDR wireless Broadcast system, and the other is related to an SDR wireless point-to-point Frequency Division Duplex system.

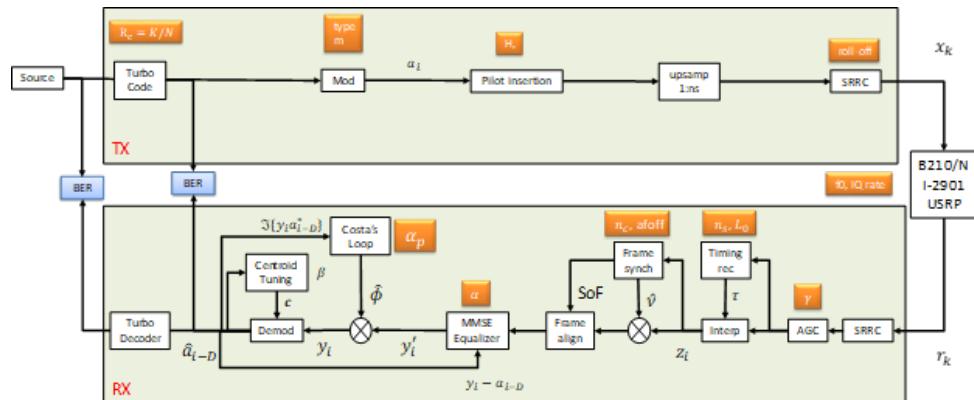


Figure 4.1: The SDR wireless communication system

As Figure 4.1 shows it is the Software Defined Radio (SDR) transmission system, we replace the simulated channel with the USRP device, which is operating between 70MHz to 6GHz.

The USRP device performs the following signal processing at the TX section:

- Analog to digital conversion with a given Sampling Rate (IQ rate). This establishes the actual bandwidth of the TX signal.  $BW = \frac{IQ \times (1+\alpha)}{n_s}$ ,  $n_s$  is the number of samples per symbol. Increasing it improves accuracy.
- Up conversion to a desired nominal carrier frequency  $f_0$ .

- Power Gain control at the TX.
- Transmission over the air through an antenna (or to cable when present).

The USRP device performs the following signal processing at the RX section:

- Reception of signal from an antenna (or from the cable when present).
- Power Gain control at the RX.
- Down conversion from a desired nominal carrier frequency  $f_0$ .
- Digital to Analog conversion with a given Sampling rate (IQ rate).

## 4.1 Closed loop with cable

The communications between the PC and USRP are performed through the USB cable and are managed through suitable device drivers (USRPManger) that transfer data between the internal memory of the PC and the internal memory of USRP. The operating frequency is at 9.6MHz.

- Check the maximum sustainable IQ rate of your system:

As shown in Table 4.1, both the value of the IQ sampling and the IQ rate are the smallest while RX and TX are activated. And there is no difference in the value of the IQ sampling and the IQ rate under these two operating modes when either of the two is activated. The calculated bandwidth uses the formula of  $BW = \frac{IQ \times (1+\alpha)}{n_s}$ .

RX(State)	TX(State)	IQ sampling	IQ rate	Bandwidth
RX(1)	TX(1)	$2.436 \times 10^6$	$609 \times 10^3$	121.8kHz
RX(0)	TX(1)	$8.8 \times 10^6$	$2200 \times 10^3$	440kHz
RX(1)	TX(0)	$8.8 \times 10^6$	$2200 \times 10^3$	440kHz

Table 4.1: The maximum sustainable IQ rate of our RX and TX system

- Check the spectra for 16QAM, in the state of RX(1) and TX(1):

As we can see when the gain is moving from positive to negative, the in-band level of the receiving signal is increasing, as in Figure 4.2 and Figure 4.3.

When the gain of TX is more negative, the distance between the line of the RX spectrum and the zero horizontal line of the coordinate axis is greater and the SNR is smaller, the received signal with more noise, and the BER becomes bigger, as shown Figure 4.3 and Figure 4.4 below.

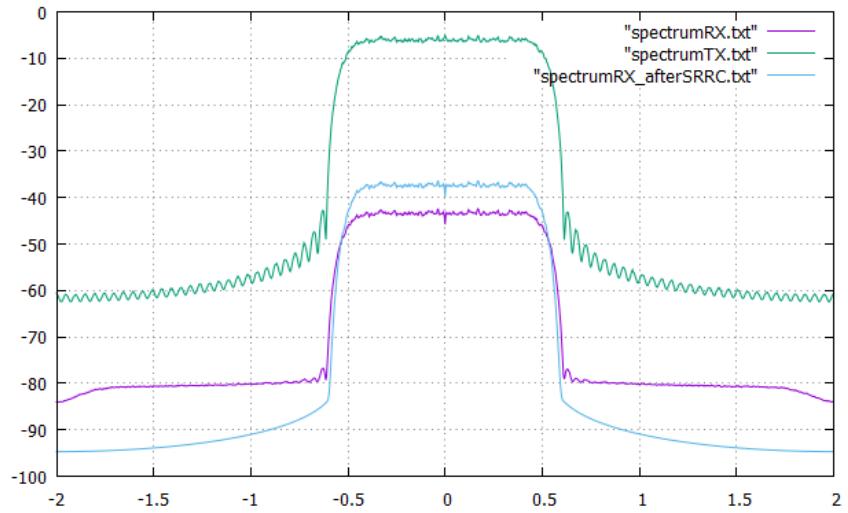


Figure 4.2: The spectrum of the TX gain with 10 dB in 16QAM

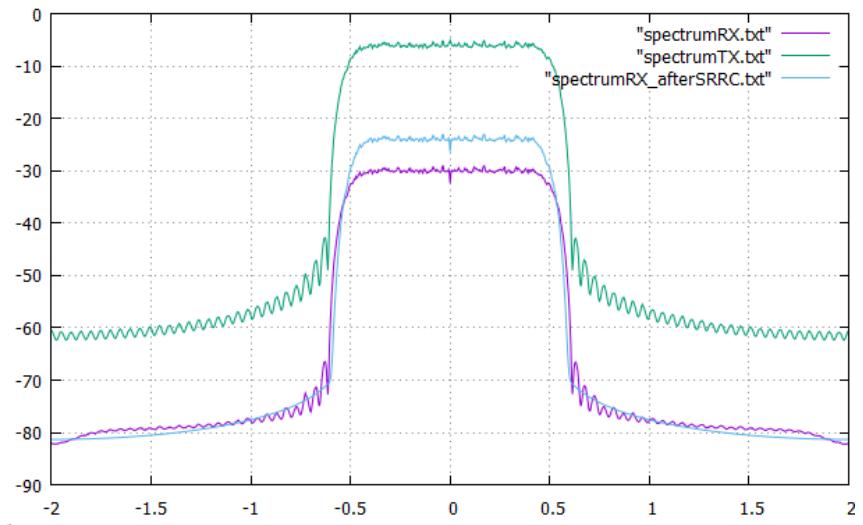


Figure 4.3: The spectrum of the TX gain with -3 dB in 16QAM

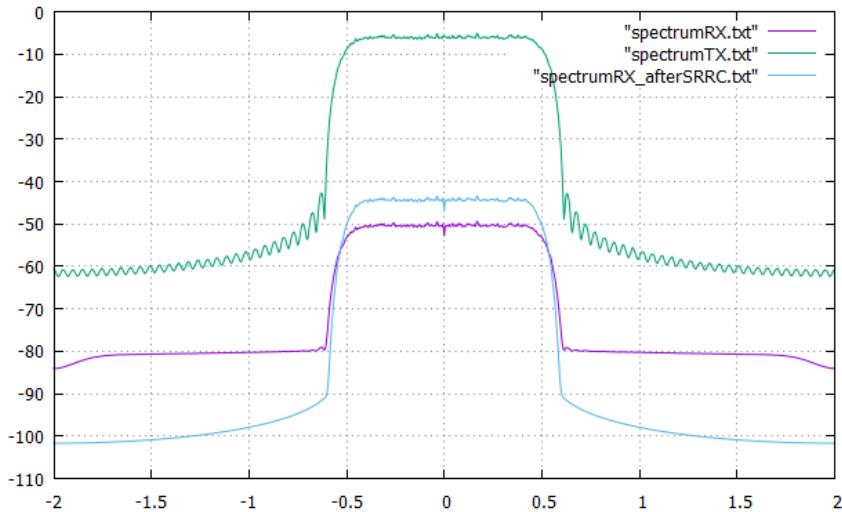


Figure 4.4: The spectrum of the TX gain with -10 dB in 16QAM

## 4.2 Broadcast wireless channel

In Figure 4.5, this is the given broadcast system, where only one main computer is used as the transmitting end, and our computer is used as the receiving end.

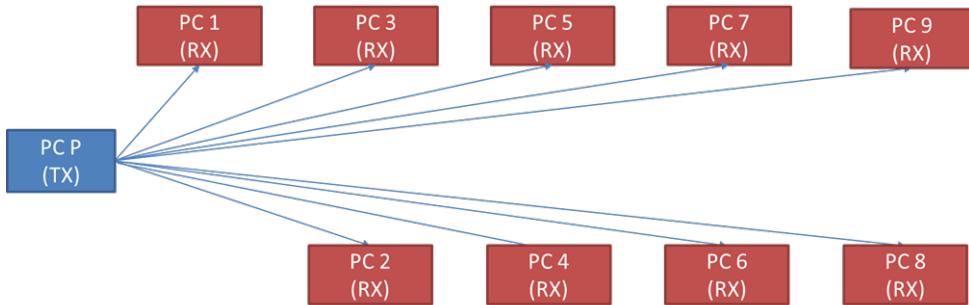


Figure 4.5: The Broadcast system

- The Carrier frequency: 960MHz
- The up-sampling factor: 4
- The header length (H): 100
- The number of the data sections in a frame (N): 20
- The data length (D): 100
- The Pilot length (P): 10

According to the above parameter settings, we turn off our own transmitter. When we started the simulation, we checked the first block for RX, which is AGC with a value greater than one. but the value is not too big. And then the standard deviation is quite low which is good for us. We found that the Equalizer theta does not repeat due to the phase noise. Then the main computer transmitter starts to reduce the GTX(the gain of TX) from 20 dB after the simulation Maximum frequency offset equals 0.05 and we set the symbol rate to 200 k. We collected 4 sets of experimental data, in Figure 4.6, Figure 4.7, Figure 4.8 and Figure 4.9.

Combining with Table 4.2, we can obtain that the phase changes in this system, but the amplitude does not. And in other groups, the SNR is very small when GTX is set to 0db. Our group needs GTX to be more negative (such as our result -12 dB), which may be due to different equipment and different errors introduced.

By comparing these two spectrums of GTX-20dB and GTX-0 dB, the smaller the gain, the more broadband noise there will be at the RX, and there will be many glitches around the RX spectrum. As mentioned before, When the gain of TX decreases(positive value), the position of the spectrogram away from the horizontal line at coordinate 0 will become higher and the SNR is smaller.

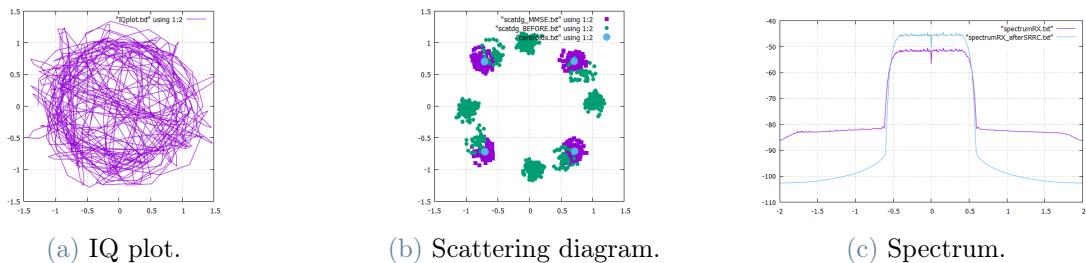


Figure 4.6: Broadcast channel with  $\text{GTX} = 20\text{dB}$

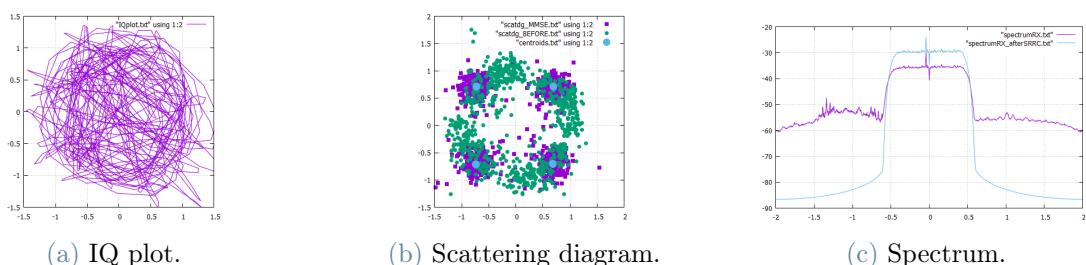


Figure 4.7: Broadcast channel with  $\text{GTX} = 0\text{dB}$

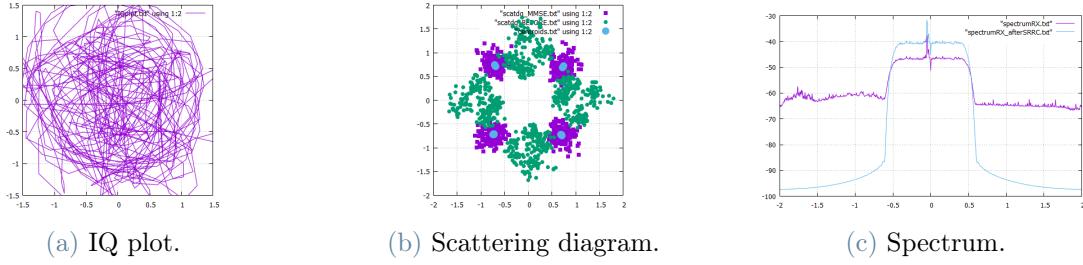
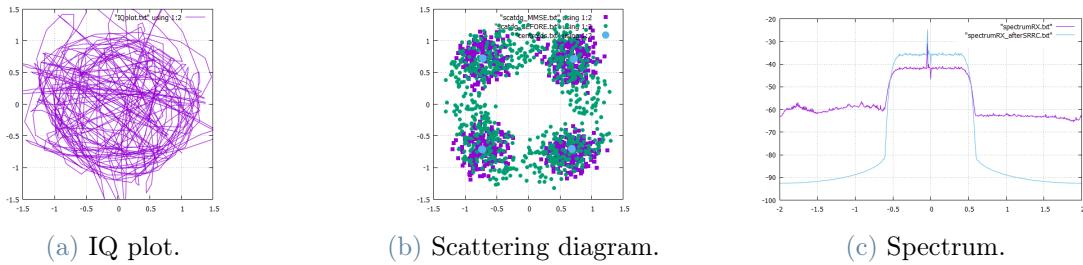
Figure 4.8: Broadcast channel with  $\text{GTX} = -3\text{dB}$ Figure 4.9: Broadcast channel with  $\text{GTX} = -12\text{dB}$ 

Table 4.2: Measured parameters for each transmitting gain

GTX	20dB	0dB	-3dB	-12dB
<b>AGC</b>	3.48	22.54	12.20	84.86
<b>AGC std.dev</b>	0.0039	0.052	0.0064	0.64
<b>Frame synchronization</b>	75	552	393	757
<b>Frequency off-set</b>	-0.0096	-0.0099	-0.0095	0.0097
<b>Equalizer power [dB]</b>	-0.085	-0.19	-0.030	-0.25
<b>Equalizer theta [deg]</b>	51.6	-145.1	-78.0	238.2
<b>SNR [dB]</b>	21.24	17.71	21.02	6.76

### 4.3 Frequency Division Duplex wireless channel

In Figure 4.10, our group is Group-4 and the group label is  $G_1$  with Group-13. According to the given equation, our carrier allocation frequency for the two-way channels is 962 MHz.

$$F_0(G) = 910MHz + G \times 2MHz \quad (4.1)$$

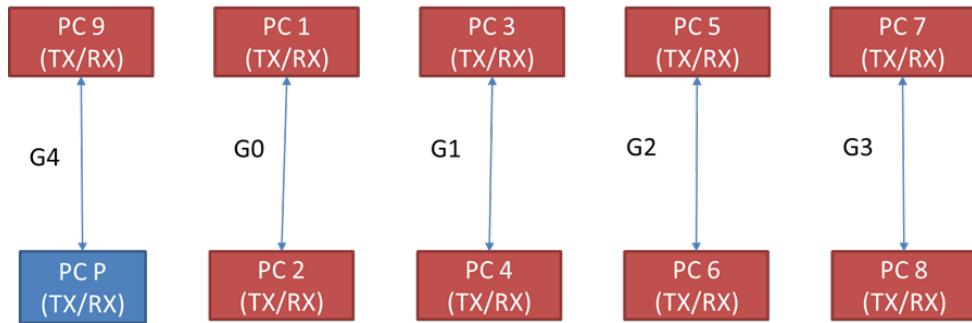


Figure 4.10: The Duplex links of Frequency Division Duplex

An additional parameter is available that specifies the carrier spacing between TX and RX. The carrier spacing is normalized to the baud rate. In this system, another group and our group use the same carrier spacing to do the simulations. When we set the carrier spacing to 2, we could obtain the spectrum in Figure 4.11. And we find the smallest carrier spacing is 1.2 for our system.

As we can see, when our RX side starts losing information in Figure 4.12.

The final transmitting gain is 32dB during this simulation with Group-13, in order to preserve the lossless transmission.

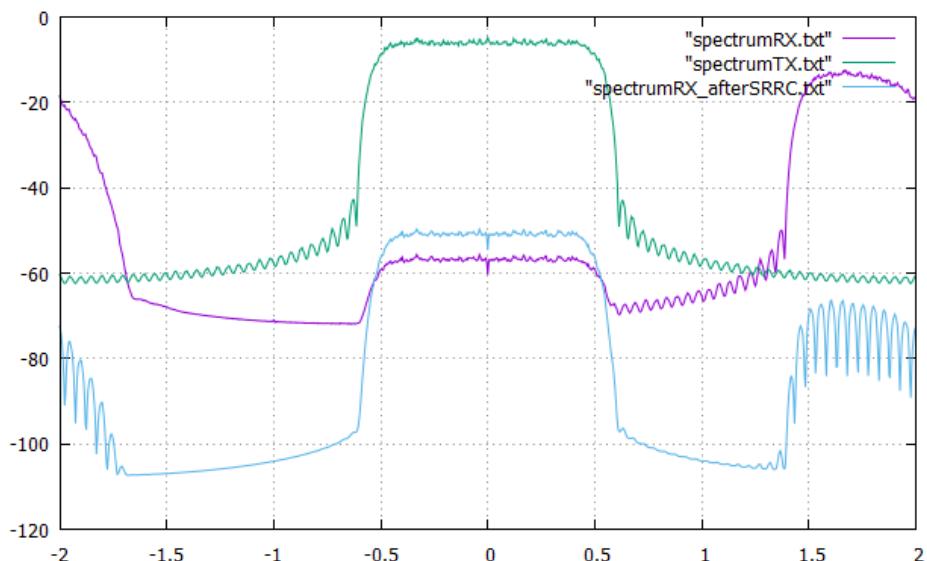


Figure 4.11: The spectrum with the carrier spacing of 2

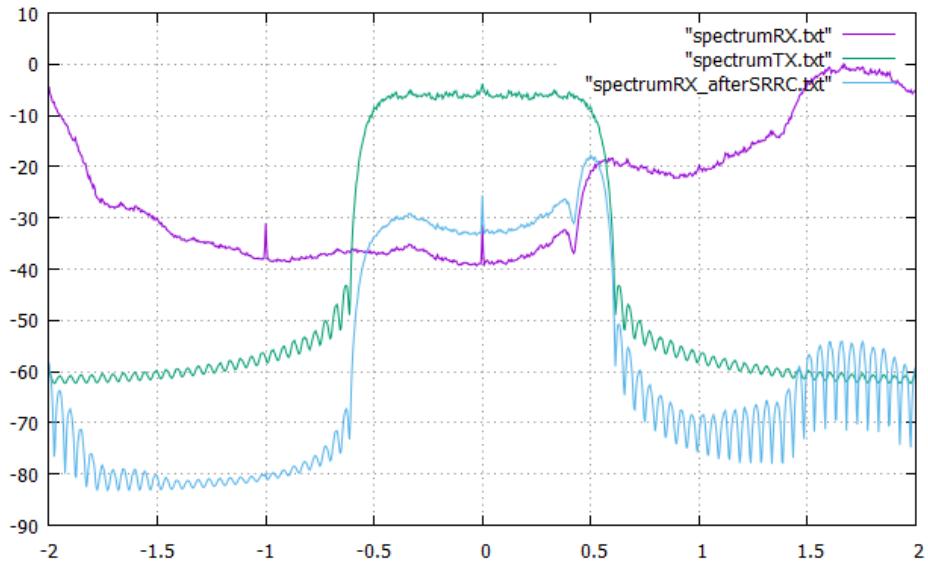


Figure 4.12: The LOST saturation

# 5 | GNU Radio Lab Results

This chapter discusses what we obtain based on our GNU Radio flow graph, involving the simulated parameters from the given Lab9.

## 5.1 Tuning simulation

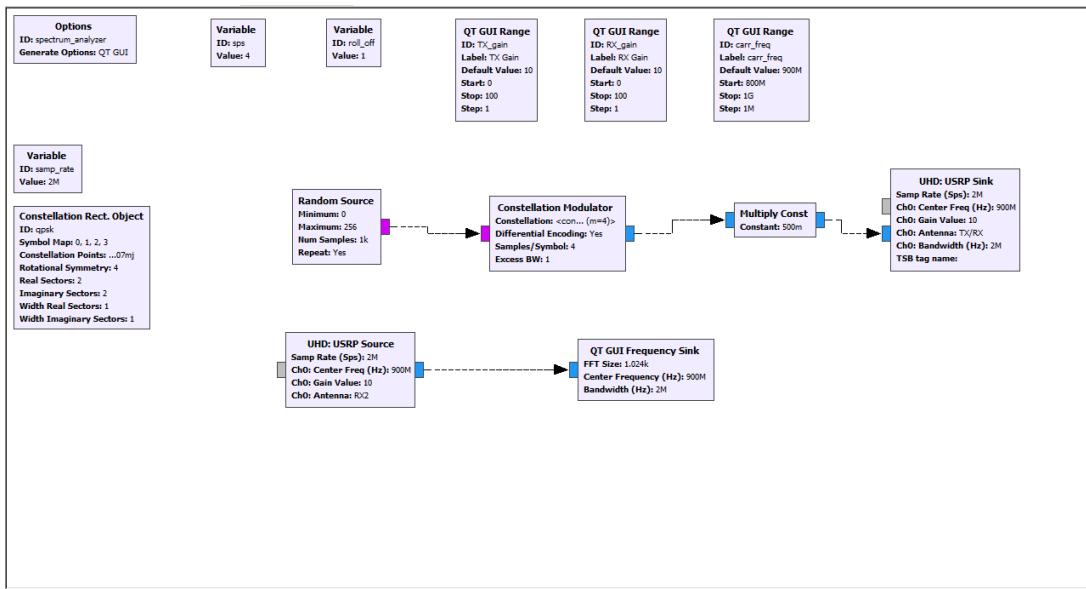


Figure 5.1: The block of design

In this task, the design block is above Figure 5.1.

In the given GNU Radio flow graph, the amplitude plots are almost red, and there are no blue parts. So we find the optimal delay is 58 samples in this tuning simulation, as Figure 5.2 shown.

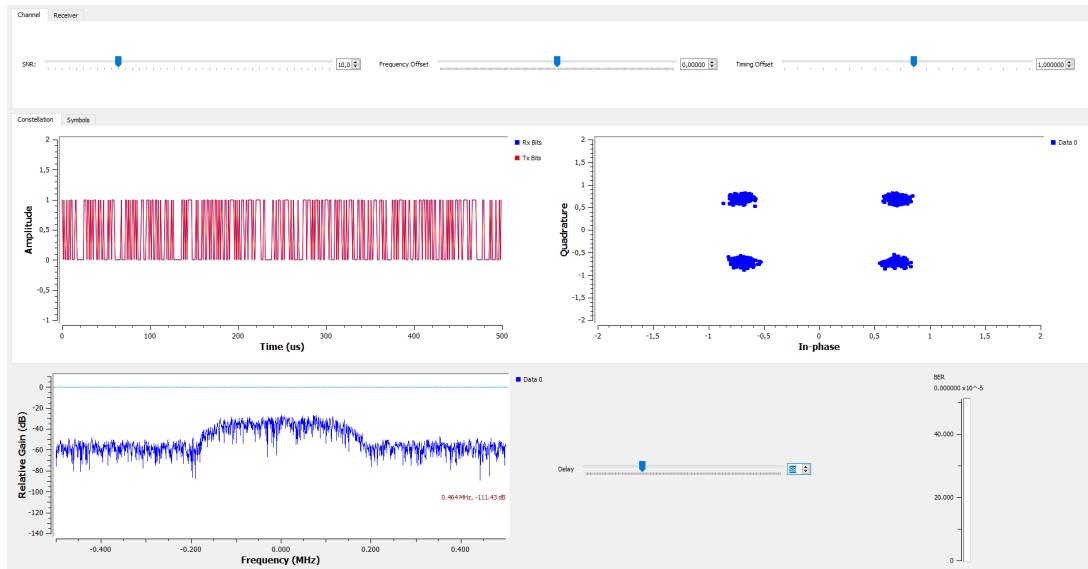


Figure 5.2: The optimal delay

In Figure 5.3, we read the value of the maximum frequency offset when the scatterers of the In-phase Quadrature plot start to rotate.

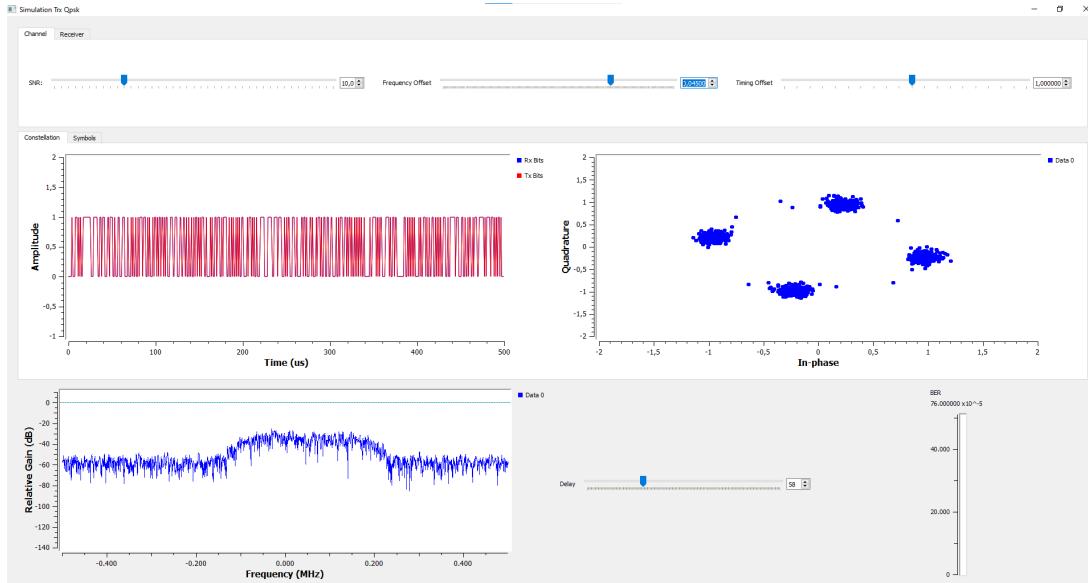


Figure 5.3: The maximum frequency offset is 0.045

When we change the position of the slider of SNR to approach a BER of approximately  $10^{-2}$  and  $10^{-1}$ , we obtain the values of SNR are 6.3 dB and 3.0 dB respectively, as Figure 5.4 and Figure 5.5 shown.

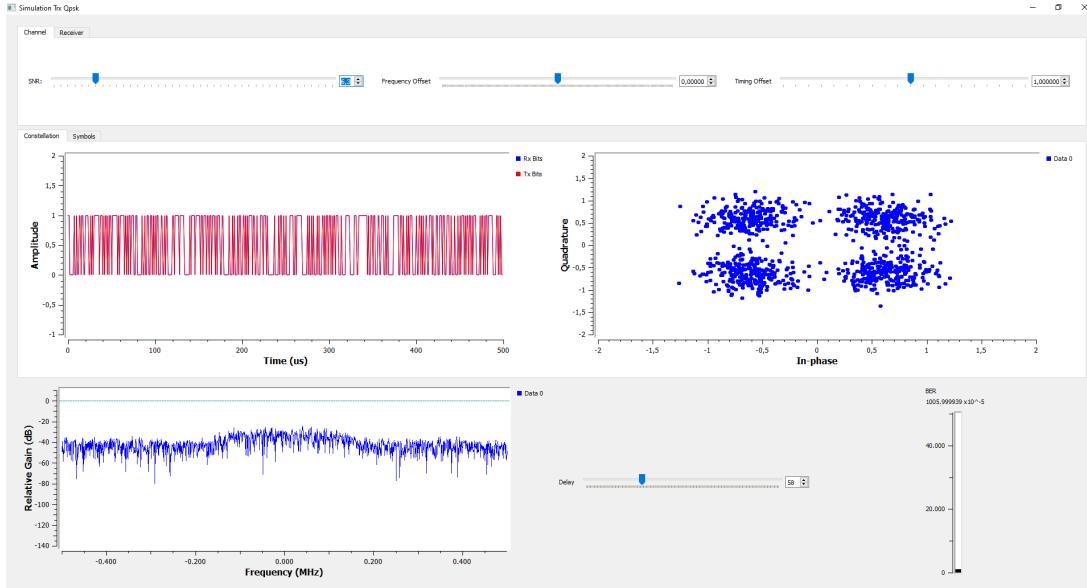


Figure 5.4: A BER of approximately  $10^{-2}$  for  $\text{SNR} = 6.3 \text{ dB}$

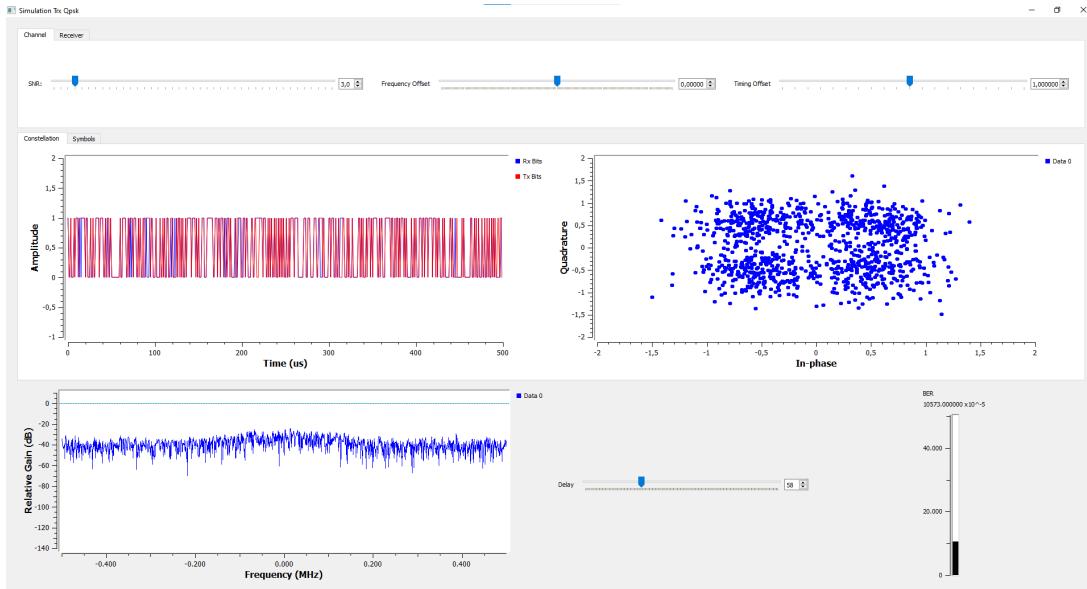


Figure 5.5: A BER of approximately  $10^{-1}$  for  $\text{SNR} = 3.0 \text{ dB}$

By comparing Figure 5.6 with Figure 5.8 and Figure 5.10, we conclude that as the roll-off factor (no more than 1) increases, the spectrum has some changes which the bandwidth of the SRRC (Square Root Raised Cosine) filter becomes larger. The bandwidth is from about 0.3MHz to about 0.4MHz to about 0.5MHz when the roll-off factor is changed as aforementioned with the same value of SNR. And when SNR is fixed setting, the smaller roll-off factor will have a better spectrum graphic, because the spectral efficiency

is higher. On another hand, in Figure 5.10 with Figure 5.11 and Figure 5.12, when the roll-off factor (no more than 1) is relatively large, we need to improve SNR. So the filter becomes less efficient.

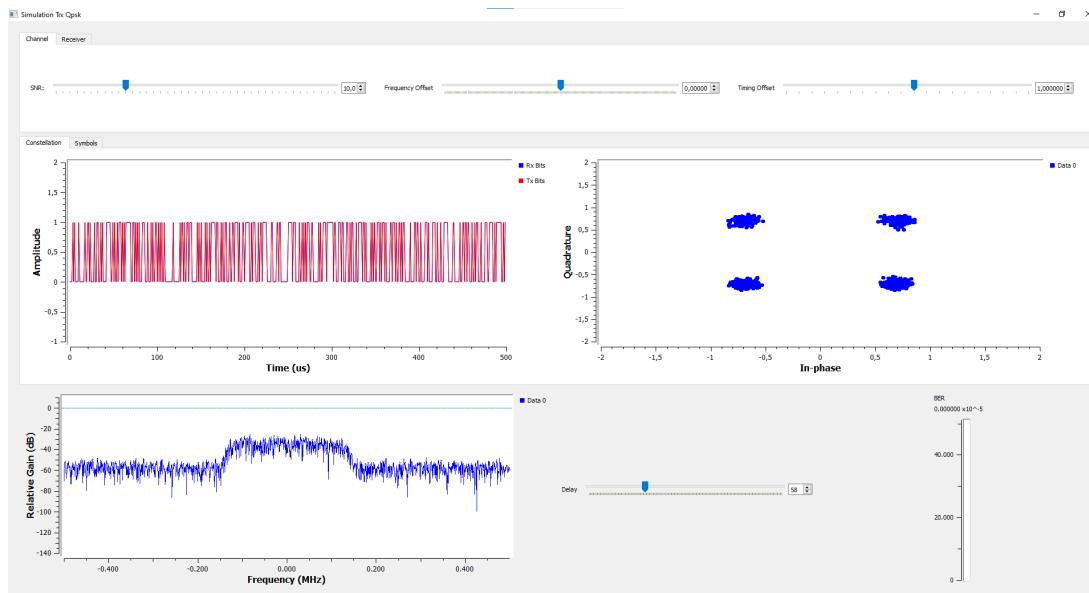


Figure 5.6: The roll-off is 0.2 for  $\text{SNR} = 10 \text{ dB}$

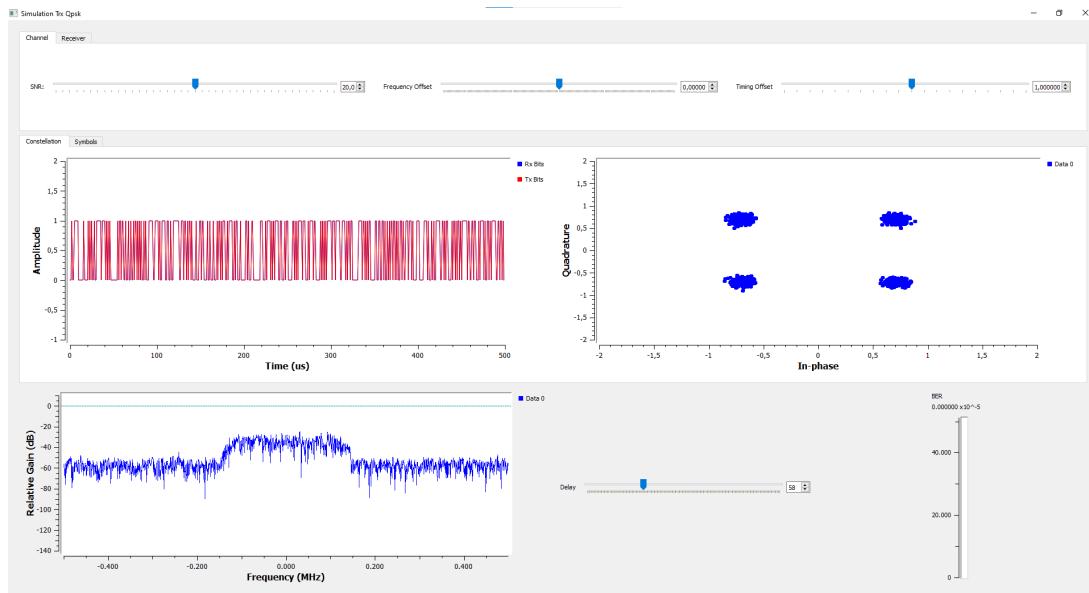
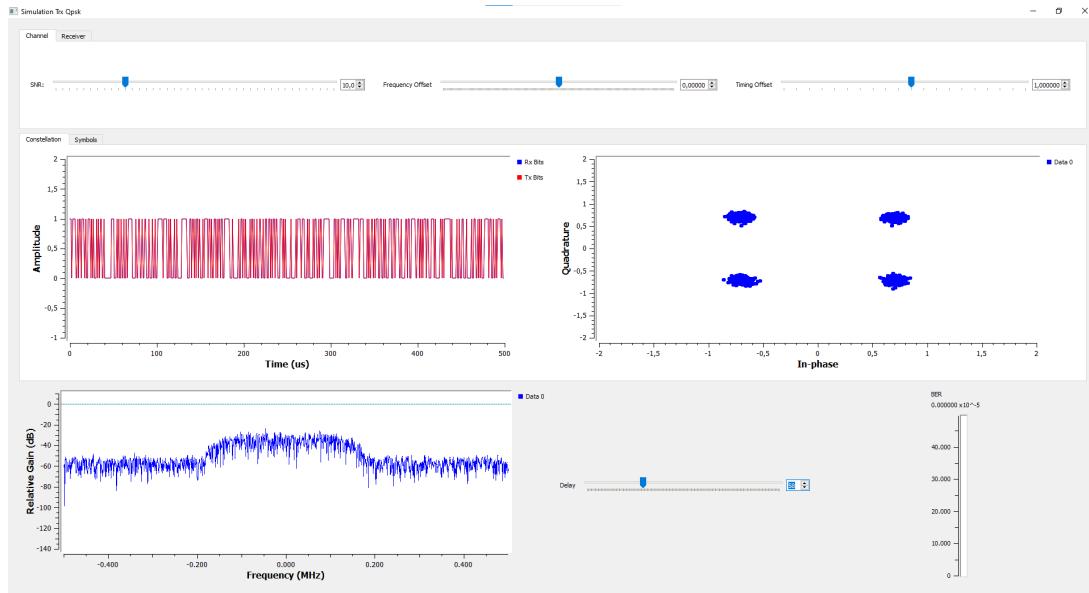
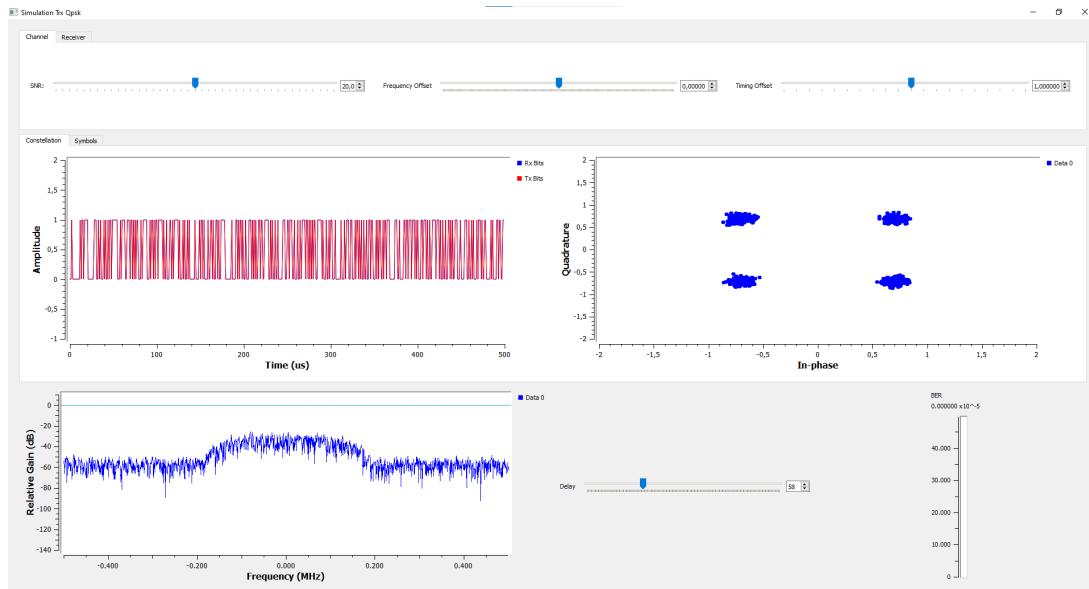


Figure 5.7: The roll-off is 0.2 for  $\text{SNR} = 20 \text{ dB}$

Figure 5.8: The roll-off is 0.5 for  $\text{SNR} = 10 \text{ dB}$ Figure 5.9: The roll-off is 0.5 for  $\text{SNR} = 20 \text{ dB}$

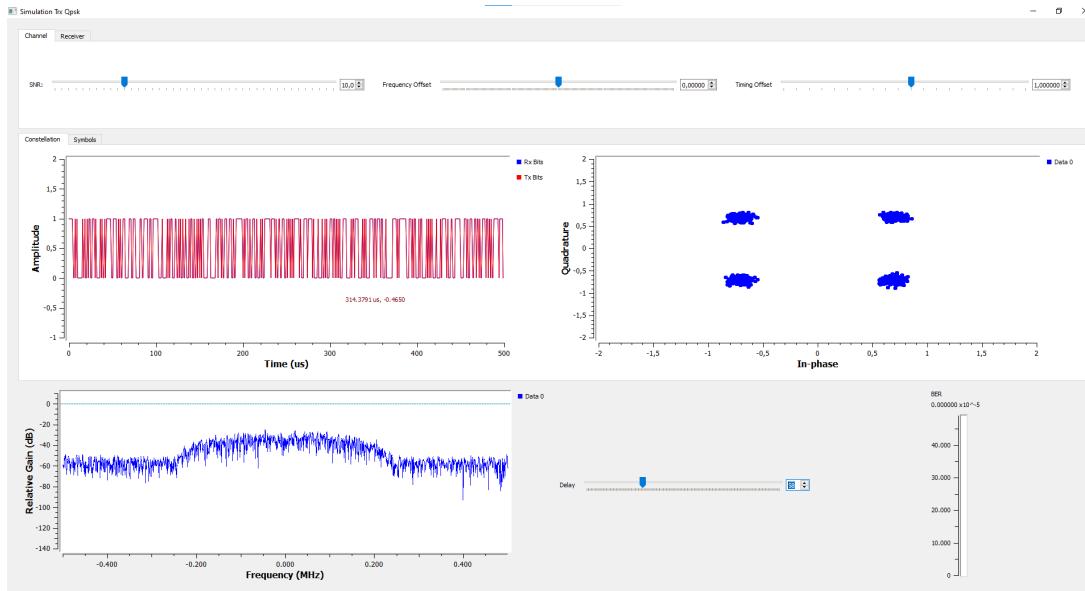


Figure 5.10: The roll-off is 1 for  $\text{SNR} = 10 \text{ dB}$

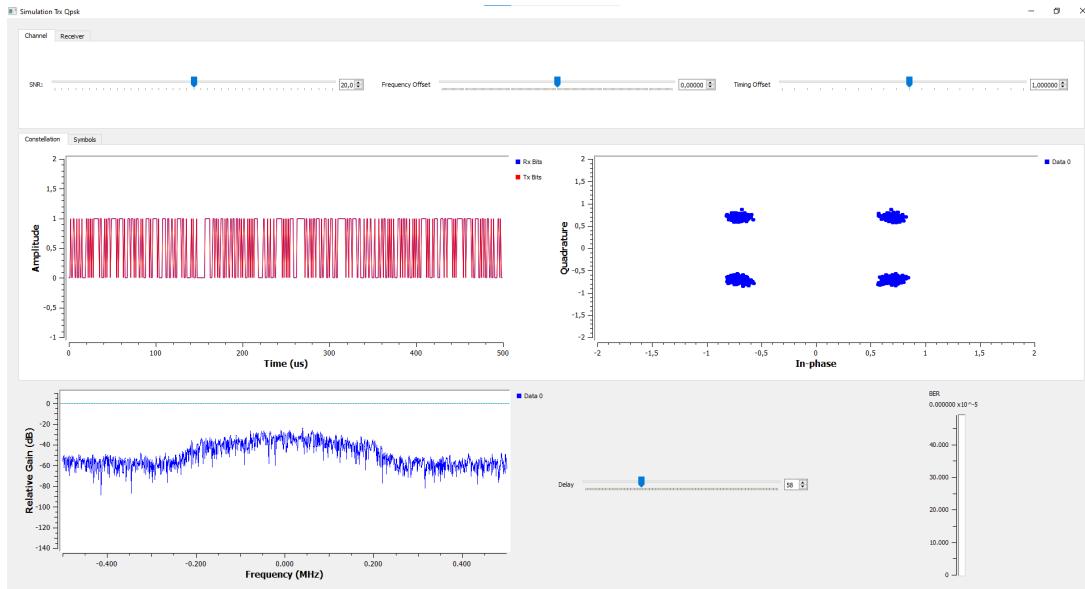
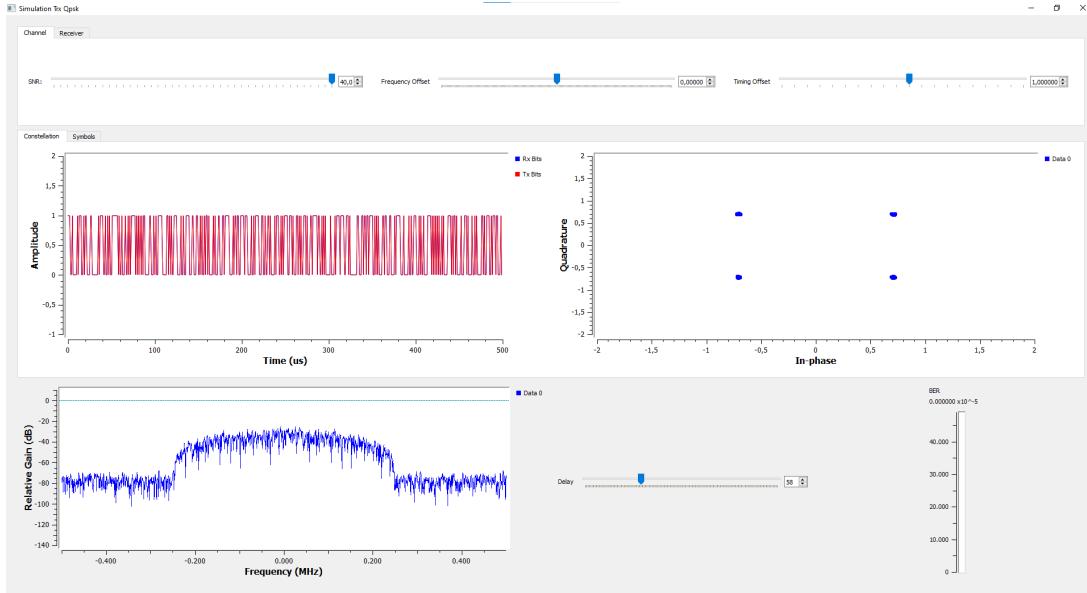


Figure 5.11: The roll-off is 1 for  $\text{SNR} = 20 \text{ dB}$

Figure 5.12: The roll-off is 1 for  $\text{SNR} = 40 \text{ dB}$ 

## 5.2 QPSK transmission

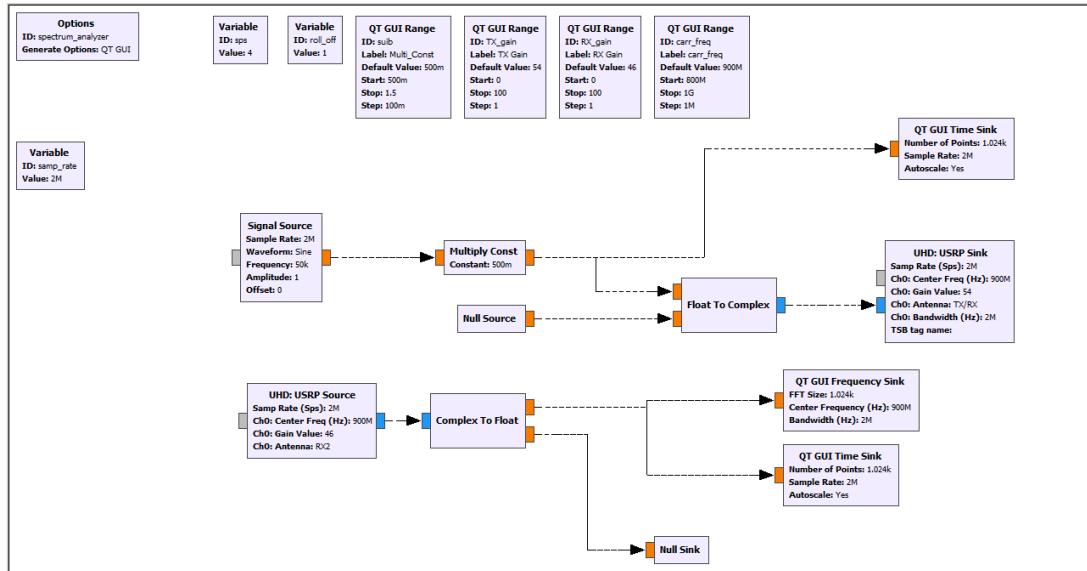


Figure 5.13: The block of design

In the second task, the design block is above Figure 5.13. We create a system with a QPSK transmission connected to a USRP sink and use a Multiply Const block between Constellation Modulator and USRP Sink block and set it to 0.5.

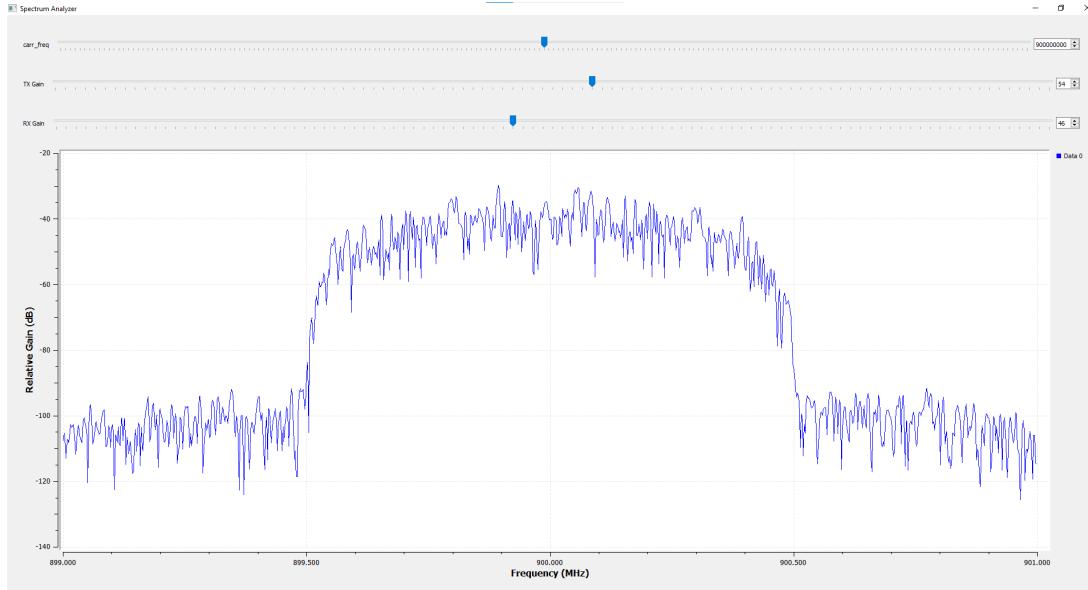


Figure 5.14: The optimal gain values of TX and RX

The carrier frequency is 900 MHz. we can observe that the optimal gain values of TX and RX are 54 dB and 46 dB respectively, as Figure 5.14 shown.

### 5.3 Real sinusoidal transmission

As Figure 5.15, Figure 5.16, Figure 5.17, Figure 5.18, Figure 5.19 and Figure 5.20 shown, the value of the Multiply Constant block (parameter Constant) is from 0.5 to 1.5 and the amplitude is set from 1 V to 1.5 V. And the plots of relative gain almost present a symmetry centred around 900MHz

In Figure 5.15, there are two main peaks around 900 MHz in the bottom plot. We also can observe that as the increment of the Multiply Constant, more peaks appear in the plot of relative gain between 899 MHz to 901 MHz when our amplitude is fixed(1 V). On the other hand, the second amplitude plots are distorted when the Multiply Constant block is set to more than 0.5.

If the amplitude of the sinusoid is bigger than 1, we found that even if this constant is 0.5, there is a small distortion in this second amplitude plot and there are many small peaks appearing in the plot of relative gain.

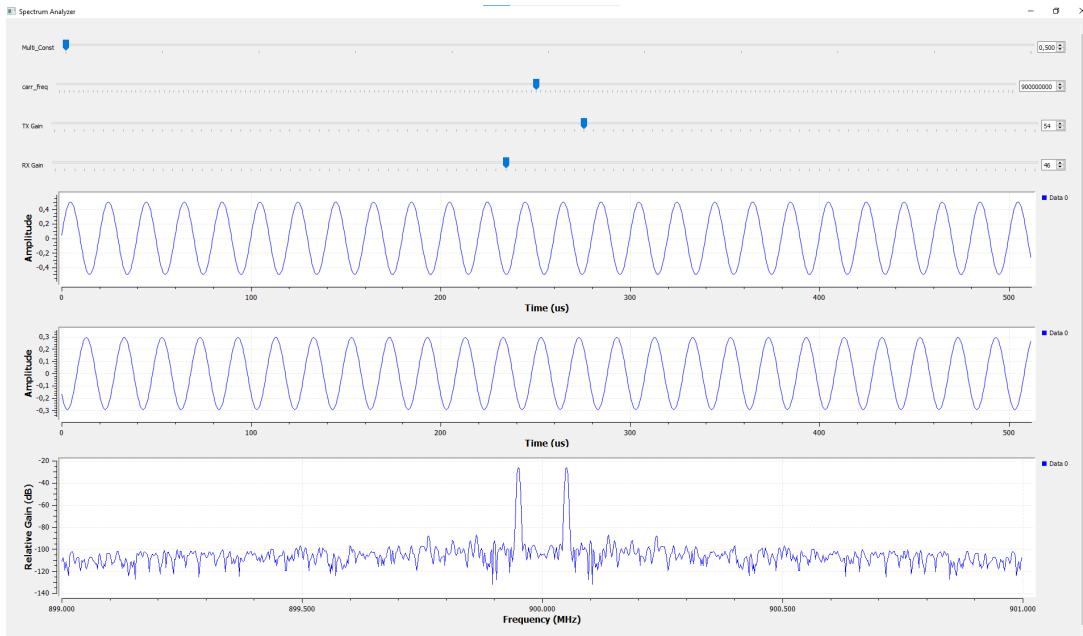


Figure 5.15: The Amplitude = 1 and the Multiply Constant = 0.5

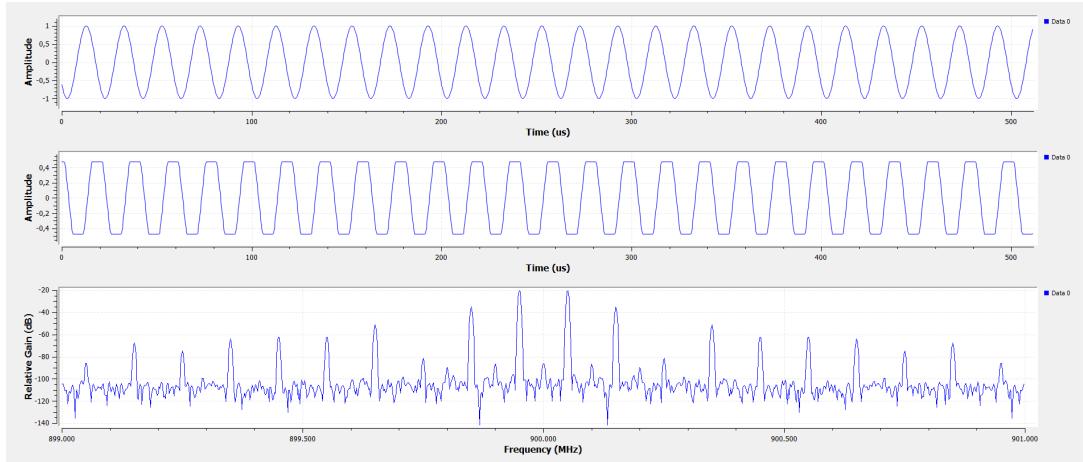


Figure 5.16: The Amplitude = 1 and the Multiply Constant = 1

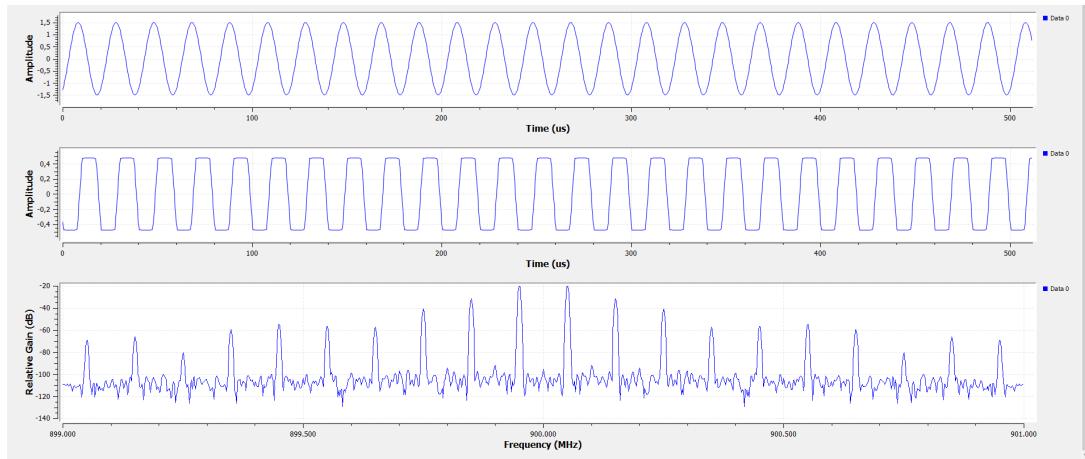


Figure 5.17: The Amplitude = 1 and the Multiply Constant = 1.5

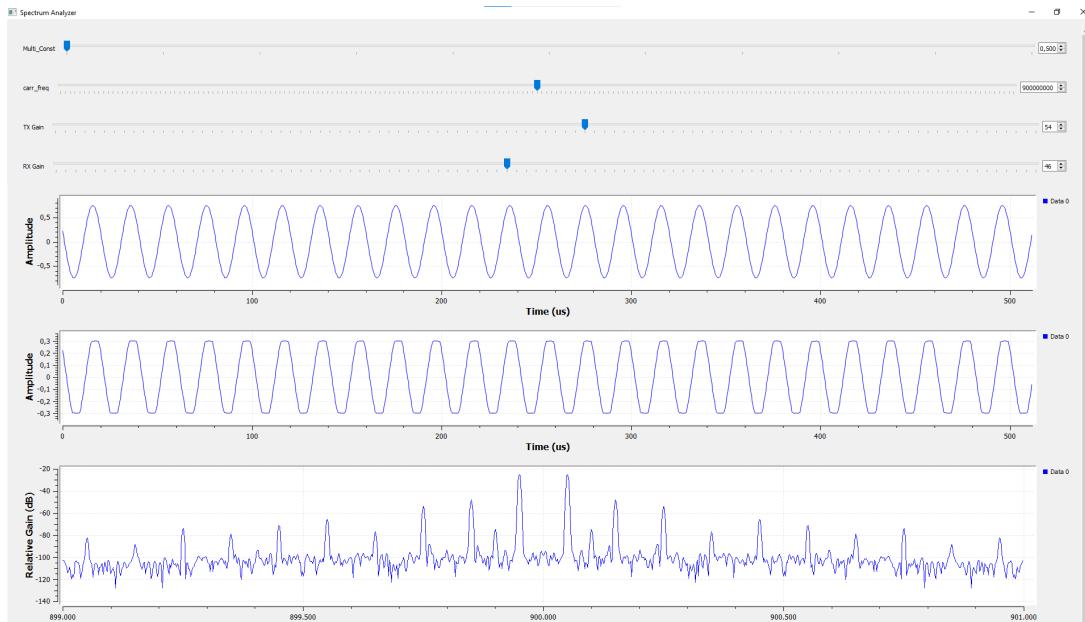


Figure 5.18: The Amplitude = 1.5 and the Multiply Constant = 0.5

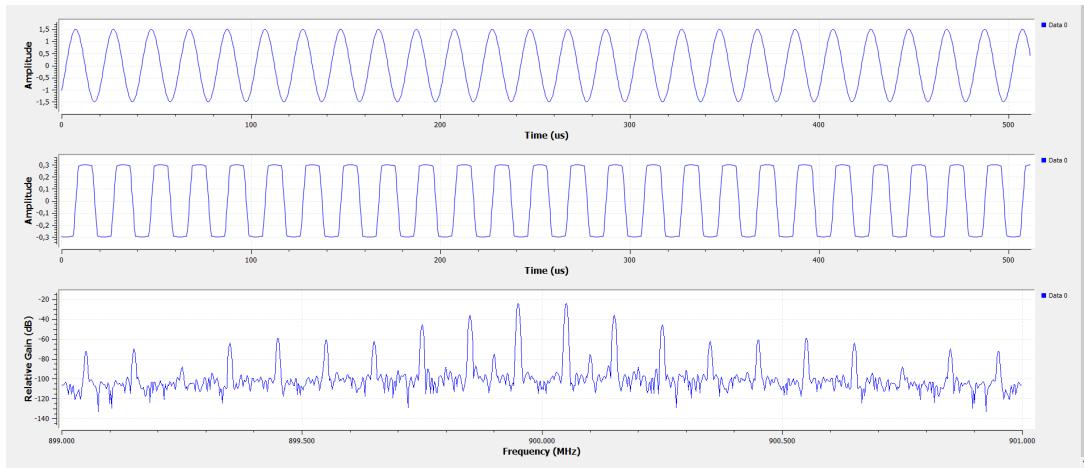


Figure 5.19: The Amplitude = 1.5 and the Multiply Constant = 1

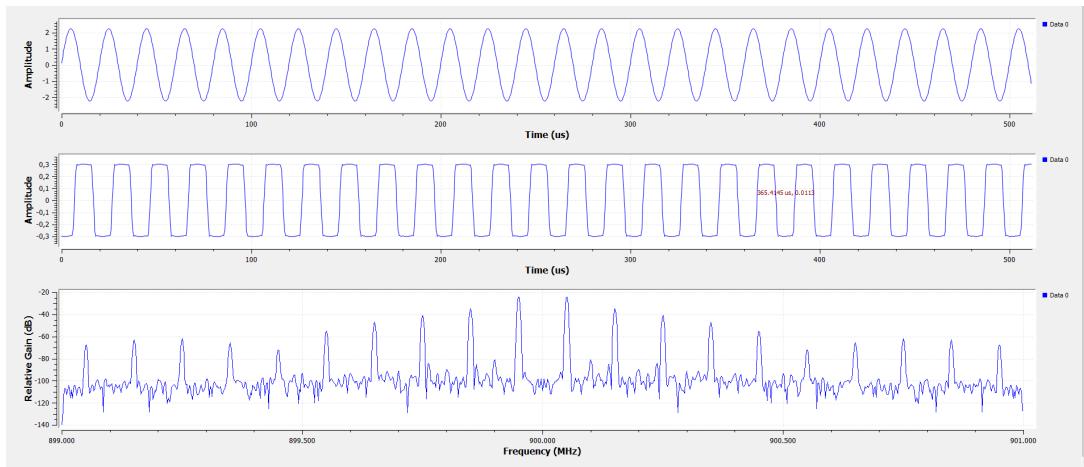


Figure 5.20: The Amplitude = 1 and the Multiply Constant = 1.5



# 6 | Appendix

## 6.1 Code

### 6.1.1 My\_Ber\_Meter.h

My\_Ber\_Meter.h:

```
class My_BER_Meter
{
public:
    My_BER_Meter();
    ~My_BER_Meter();

    ///! Set the BER parameters
    void SetParameters(
        const int delay,           //!< Delay of second stream
        const int n = 30            //!< Number of errors to declare reliable
    );

    ///! Compare two bit streams. Returns 1 if errors are present
    int Run(
        const int sizein,          //!< Number of compared bits.
        const int* ref,             //!< First stream (source).
        const int* decoded          //!< Second bit stream (decoded).
    );

    bool IsReliable();           //!< Return true if enough bits or frame errors have occurred
    void Reset();                //!< Reset the error counters

    ///! Display the BER statistics on specified output.
    void Display(FILE* stream = stdout) //!< Output stream
```

```

    );

//! Display the BER statistics on a single line of the specified output.
void Display_on_Line(FILE* stream = stdout);

private: // Place here variable of the class
    int nerr;           //!< Number of counted errors
    int nbits;          //!< Number of counted bits
    int delay;          //!< Length of delay line
    int nmin;          //!< Minimum number of error to declare reliable the measure

    int* line;          //! delay line
    int pointer;        //! Pointer within the delay line
    bool count;          //! Flag for starting measurements

};


```

### 6.1.2 My\_Ber\_Meter.cpp

```
My_BER_Meter::My_BER_Meter()
```

```
{
    delay = 0;
    nmin = 0;
    nerr = 0;
    line = 0;
    count = false;
}
```

```
My_BER_Meter::~My_BER_Meter()
```

```
{
    delete[] line;
}
```

```
void My_BER_Meter::SetParameters(const int delay, const int n)
```

```
{
    this->delay = delay;
    this->nmin = n;
```

```

    delete[] line;  line = 0;
    int i;
    count = true;
    if (delay > 0)
    {
        line = new int[delay];
        for (i = 0; i < delay; i++)line[i] = 0;
        pointer = 0;
        count = false;
    }
    Reset();
}

int My_BER_Meter::Run(const int sizein, const int * ref, const int * decoded)
{
    int i,bit;
    for (i = 0; i < sizein; i++)
    {
        if (delay > 0)
        {
            bit           = line[pointer];
            line[pointer] = ref[i];
            pointer      = (pointer + 1) % delay; // Circular pointer
            if (!count)
            {
                if (pointer == 0)count = true;// Full line, Start B
                continue;
            }
        }
        else
        {
            bit = ref[i];
        }
        nerr += bit ^ decoded[i];
        nbits++;
    }
    return nerr;
}

```

```
}

bool My_BER_Meter::IsReliable()
{
    if (nerr >= nmin) return true;
    else           return false;
}

void My_BER_Meter::Reset()
{
    nerr = nbits = 0;
}

void My_BER_Meter::Display(FILE * stream)
{
    fprintf(stream, "BER=(%d/%d)= \t%e\n", nerr, nbits, (double)nerr / nbits);
}

void My_BER_Meter::Display_on_Line(FILE * stream)
{
    fprintf(stream, "%d\t%d\t%e\t", nerr, nbits, (double)nerr / nbits);
}
```

### 6.1.3 PN\_Source\_Matlab

PN\_Source\_Matlab code:

```
clear all;
close all;
clc

m=2; % modulation bits
N = 500; %frame size
D= m*N; % data bits
pnSequence = comm.PNSequence('Polynomial',[8 6 5 1 0], ...
    'SamplesPerFrame',D,'InitialConditions',[0 0 0 1 1 1 1 0]);
x1 = pnSequence();
x1b=2*x1-1; % bipolar version 0 to -1, 1 to +1

i = 1:D;
figure
stairs(i, x1)
title('Generated LFSR sequence, first 150 bits');
axis([0 150 -0.5 1.5]);
grid on
```

## 6.2 Screen shots and results of GTX

Corresponding to chapter 4:

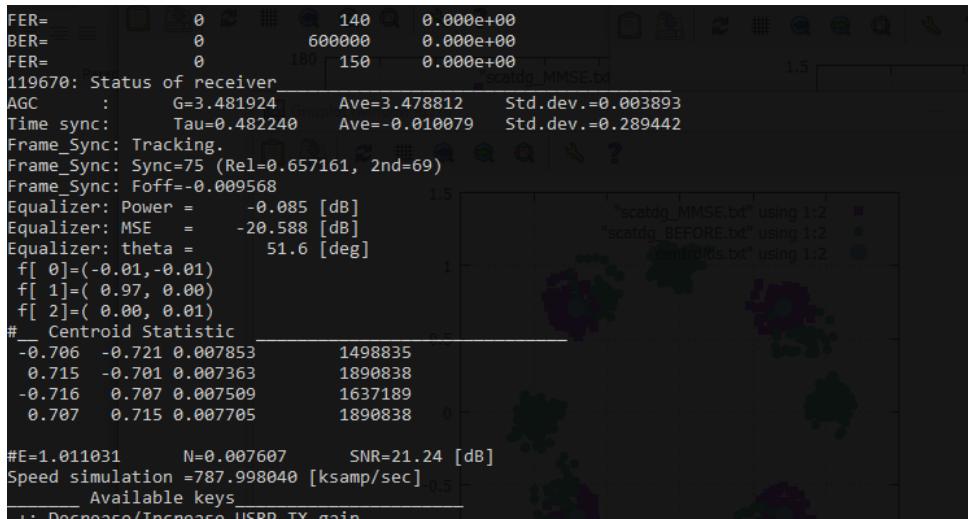


Figure 6.1: The results for GTX equals 20 dB

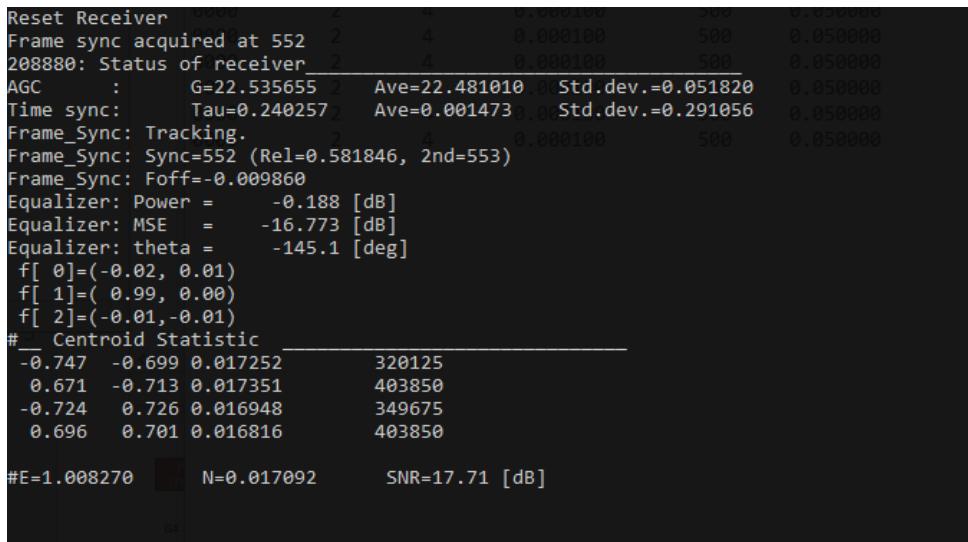


Figure 6.2: The results for GTX equals 0 dB

```
184430: Status of receiver_
AGC      :      G=12.204479      Ave=12.208242  Std.dev.=0.006419
Time sync:   Tau=-0.417578    Ave=-0.005310  Std.dev.=0.286892
Frame_Sync: Tracking.
Frame_Sync: Sync=393 (Rel=0.593431, 2nd=394)
Frame_Sync: Foff=-0.009545
Equalizer: Power =      -0.030 [dB]
Equalizer: MSE   =      -20.386 [dB]
Equalizer: theta =      -78.0 [deg]
f[ 0]=(-0.01,-0.01)
f[ 1]=( 0.97, 0.00)
f[ 2]=( 0.00, 0.01)
#__ Centroid Statistic _____
-0.721  -0.722 0.008212      778375
  0.697  -0.719 0.007853      981950
-0.717   0.704 0.007915      850225
  0.702   0.697 0.007874      981950

#E=1.007884      N=0.007963      SNR=21.02 [dB]
BER=      1514035      77460000      1.955e-02
FER=      2040          19365       1.053e-01
Append BER stat in BER_results.txt
BER=      1514035      77740000      1.948e-02
FER=      2040          19435       1.050e-01
Append BER stat in BER_results.txt
BER=      1514035      77900000      1.944e-02
FER=      2040          19475       1.047e-01
Append BER stat in BER_results.txt
BER=      1514035      78060000      1.940e-02
FER=      2040          19515       1.045e-01
Append BER stat in BER_results.txt
```

Figure 6.3: The results for GTX equals -3 dB

```

f[ 1]=( 0.93, 0.00)
f[ 2]=(-0.03, 0.01) in Notepad
#__ Centroid Statistic
-0.724 -0.705 0.050627      2653625
  0.686 -0.708 0.049962      200000.000000 3347650
  0.704  0.724 0.048098      2898575
  0.710  0.716 0.049620      3347650

#E=1.006912      N=0.049577      SNR=13.08 [dB]
BER=          8368      2280000      3.670e-03
FER=            30        570      5.263e-02
BER=          8368      2560000      3.269e-03
FER=            30        640      4.688e-02
BER=         10489      6680000      1.570e-03
FER=            60        1670      3.593e-02
Reset Receiver File Edit Format View Help
Frame sync acquired at 757      4      100000000      100      20
BER=          694523      14908000      4.659e-02
FER=            1357      3727      3.641e-01
248350: Status of receiver      4      100000000      100      20
AGC :      G=84.862512      Ave=85.545743      Std.dev.=0.640684
Time sync:      Tau=-0.131779      Ave=0.003867      Std.dev.=0.286826
Frame_Sync: Tracking.      4      100000000      100      20
Frame_Sync: Sync=757 (Rel=0.806565, 2nd=1722)      4      100000000      100      20
Frame_Sync: Foff=-0.009705      4      100000000      100      20
Equalizer: Power = -0.247 [dB]      4      100000000      100      20
Equalizer: MSE_Da = /8 R=-10.529 [dB]      4      100000000      100      20
Equalizer: theta = /8 Re=-238.2 [deg]      4      100000000      100      20
f[ 0]=(-0.07, 0.00) Results.txt      4      100000000      100      20
f[ 1]=( 0.74, 0.00) Results.txt      4      100000000      100      20
f[ 2]=(-0.05, 0.01) Results.txt      4      100000000      100      20
#__ Centroid Statistic
-0.752 -0.684 0.225702      18590
  0.655 -0.694 0.201306      23452
  0.703  0.734 0.204206      20306
  0.719  0.693 0.207218      23452
Data/8_Results.txt      4      100000000      100      20
#E=0.993520      Da N=0.209608 Results.txt SNR=6.76 [dB] 100000000
BER=          829761      16584000      5.003e-02
FER=            1668      4146      4.023e-01
Speed simulation =785.238927 [ksamp/sec]
Available keys

```

Figure 6.4: The results for GTX equals -12 dB