Comunicações Digitais
Practical Assignment n°2

QPSK transmission using Simulink and SDR

Objective: In this practical assignment, we will analyse the performance of Quadrature Phase Shift Keying (QPSK) through Simulink simulation. The second part of this project involves the investigation and characterization of a real QPSK system implemented with Software-Defined Radio (SDR).

Duration: This script will be performed in three classes.

Deadline for uploading the report: 15/11/2023. <u>Reports sent via email will not be accepted</u>.

Part I
QPSK performance assessment using Simulink simulation

1) Building Simulink Model of a QPSK transmission link

The Simulink model of QPSK modulator and demodulator is shown below

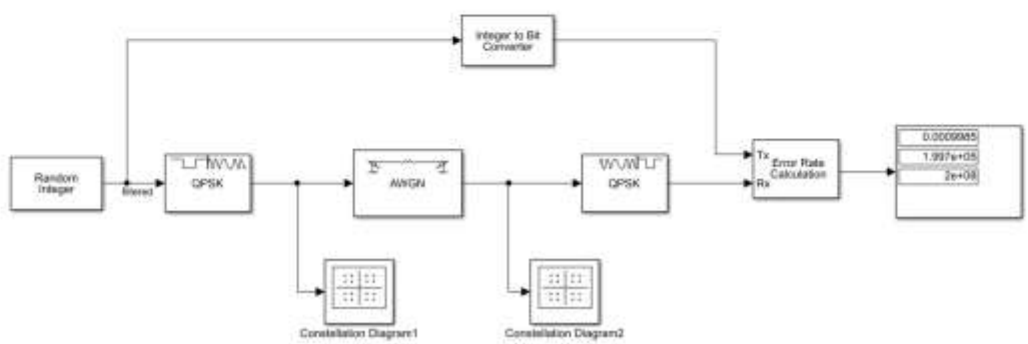

Figure 1. QPSK transmission link

1.1 Build the Simulink model shown in Figure 1.

1.1.1 Double-click on the Random Integer Generator and adjust the set size to a proper value (Remember that the input to the QPSK modulator should be either 0, 1, 2, or 3).

1.1.2 In the Random Integer Generator block, set the Sample Time to 1e-6 (i.e. 1 μs) and the Samples per frame parameter to 1024.

1.1.3 The QPSK modulator and demodulator should have the following parameters:

Figure 2 Block Parameter of QPSK baseband a) modulator, b) Demodulator

1.2 In the AWGN block, set the Symbol period parameter to 1e-6 (i.e. 1 μs).
1.3 Integer to Bit Converter: Change the number of bits per integer for the correct value.
1.4 For the Error Rate Calculation block, set the Output data field to “port” so you can connect the Display block.
1.5 The Display Block will show you three values. The first value is the BER, the second value is the number of incorrect bits, and the third value is the total number of bits received.
1.6 Set the simulation time to 100 seconds.

1.7 Fill the following table:

$P_{e_{bit}}$	Theoretical $\frac{E_S}{N_0}(dB)$	Measured $P_{e_{bit}}$	Measured $EVM_{(average)}$ (dB)
10^{-1}			
10^{-2}			
10^{-3}			
10^{-4}			
10^{-5}			
10^{-6}			

1.8 Plot BER vs. $\frac{E_S}{N_0}(dB)$ and compare with the theoretical values.
1.9 Acquire the constellations at output of the AWGN channel for $P_{e_{bit}} = 10^{-1}, 10^{-3}, 10^{-6}$.
1.10 Comment on the results obtained in 1.7,1.8 and 1.9

SDR-Part II

Introduction to SDR ADALM-PLUTO

The ADALM-PLUTO SDR (PlutoSDR) provides a high-performance software defined radio (SDR) transceiver that was designed specifically for students. It allows simultaneous transmit and receive operations. PlutoSDR communicates with a host PC through a USB 2.0 interface. It utilizes the libiio drivers, which are supported on Windows, Linux, and Mac OS X operating systems. The local oscillator Tuning Range and Bandwidth depend on the RF transceiver configuration as depicted in the table below.

RF Transceiver	LO Tuning Range	Bandwidth
AD9363 (factory default)	325 - 3800 MHz	20 MHz
AD9364	70 - 6000 MHz	56 MHz

The Transmitter Power Output is approximately 7 dBm. The ADALM-PLUTO Kit includes two antennas, both working in frequency bands [824 894] MHz and [1710 2170] MHz.

Very Important: ADALM-PLUTO operates in frequency ranges that cover most commercial and non-commercial communications. You should only transmit signals over unlicensed bands and at low power to avoid jamming existing communications, some of them critical! **Transmission on unauthorized frequencies is a crime punishable by law**.

Suggestion: Use the unlicensed bands for Short Range Devices (SRD): 863-870 MHz that are compatible with the antennas frequency bands.

2) ADALM-PLUTO installation

Install the Support Package for Analog Devices ADALMPLUTO Radio

1 On the MATLAB Home tab, see Figure 3, in the Environment section, click Add-Ons > Get Hardware Support Packages.

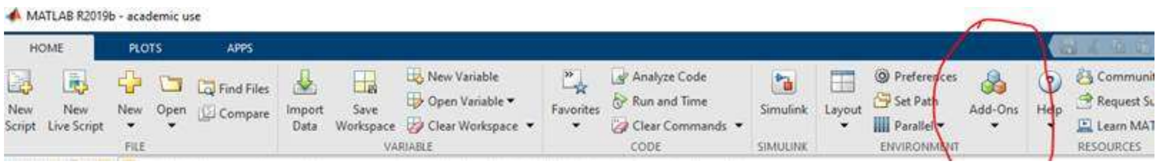

Figure 3 - MATLAB Home tab

2 In the Add-On Explorer window, browse or search for the Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio.
3 Select the support package and then click Install.
4 Follow the installation instructions.
5 When the support package installation is complete, you are prompted to configure the hardware.
6 Verify the PlutoSDR Installation

Connect a PlutoSDR device to your laptop. It may take Windows a few minutes to recognize the hardware and configure the necessary drivers. Type *findPlutoRadio* at the MATLAB prompt. After a few moments, you should see a message beginning with

```
>> findPlutoRadio
RadioID: 'usb:0'
SerialNum: '100000235523730900160013090216107d'
where the SerialNum entry will vary depending on your specific PlutoSDR.
```

Very Important: Do not upgrade the firmware

3) Real-Time Spectrum Analyser

We will now create a real-time spectrum analyser using the SDR hardware as a data source. Open a new Simulink model and create the model shown in Figure 4. The specific components you will need are:

- **ADALM-Pluto Radio Receiver** (found in Communications System Toolbox Support Package for ADALM-PLUTO Radio)
- **Spectrum Analyser** (found in DSP System Toolbox → Sinks)
- **Constant** (found in Simulink → Sources)

Hint: You can construct models more quickly by clicking in the white space of the model and typing the name of the block you would like to add. For example, to add a **Spectrum Analyzer** block, start typing ”Spectrum”, and you will see a list of blocks that have ”Spectrum” in the name. You can then choose the **Spectrum Analyzer** block.

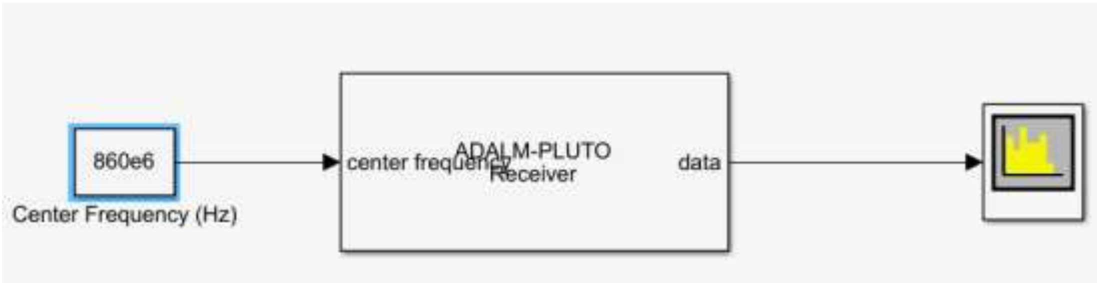

Figure 4 Real-Time Spectrum Analyser

Set the Simulation stop time to inf.

Configure the **ADALM-PLUTO Radio Receiver** block as shown in Figure 5. The key settings and their meaning are as follows:

- ***Center frequency (Hz) / Source of center frequency*:** controls the center frequency of the SDR hardware. Selecting Input Port allows the setting to be easily adjusted while the Simulink model is executing. Set the corresponding Constant block to 860e6.
- ***Source Gain*:** AGC Slow Attack
- ***Baseband sample rate (Hz)*:** controls the sample rate of data delivered to the PC over the USB bus. This rate corresponds to the instantaneous bandwidth of the receiver. Set this parameter to 2e6.
- ***Samples per frame*:** specifies the length of each data frame that will be processed by Simulink. Choice 3660 should work reasonably well for this example.

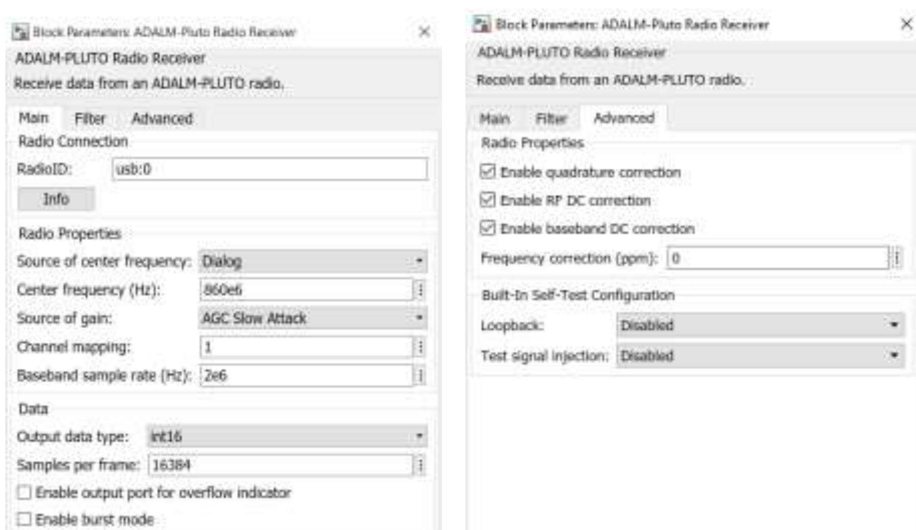

Figure 5 Block parameters ADALM-Pluto Receiver

Begin execution by clicking Run (the play button). After a few moments, you should see a plot in the Spectrum Analyser window.

3.1 Acquire and analyze the spectrum obtained

3.2 Experiment to change the values of the *Center Frequency, Gain and Baseband sample rate* in the following range:

Center Frequency: [-859.5 860.5] MHz,

Gain: [-1 70] dB (*Source of Gain* must be in *Dialog* option)

Baseband sample rate: [0.4 4] MHz

Comment and justify the results obtained by changing the above results

4) QPSK receiver based on SDR

4.1) Introduction: In this part of the practical Assignment, you will observe and investigate coherent digital modulation techniques using real-world wireless signals.

The first part of this Assignment and Assignment nº 1 have addressed the performance evaluation using a baseband representation of a QPSK communication system. That is, the process of modulating the message information onto a high-frequency sinusoidal carrier and afterward converting to baseband at the receiver, as is typically required in wireless communication, was omitted from our simulation.

As we have seen in the lecture, the frequency upconversion process (translating the baseband message to a high-frequency carrier) done in the transmitter can be exactly undone by the frequency downconversion process (translating the message information imparted on the high-frequency carrier down to the baseband) in the receiver.

Therefore, our simulations are valid for wireless systems, assuming the frequency translation processes are ideal. For coherent communication systems, the critical challenge is synchronization between the transmitter and receiver. Ensuring coherent operation can be a significant challenge.

This part of the laboratory will give an introduction to DSP demodulation techniques used in real-world optical and wireless transmission systems. In particular, you will observe:

- Carrier Synchronization: ensuring the correct downconversion of the received signal to the baseband.
- Symbol Timing Recovery: ensuring the correct sampling time of the received signal (usually in baseband)

Both of these processes are essential for coherent communication.

Because we will use SDRs, all demodulation processes are implemented through DSP algorithms, some of them adaptive. In traditional / legacy radio systems, they were realized through analogue circuitry. Our focus will be on studying the importance of these processes within a coherent digital communication system and understanding how they work.

4.2) Download from Elearning, the QPSK receiver similar to the one shown in Figure 6. The different blocks were pre-configured for this laboratory work with the exception of the “Frequency correction (ppm)” parameter.

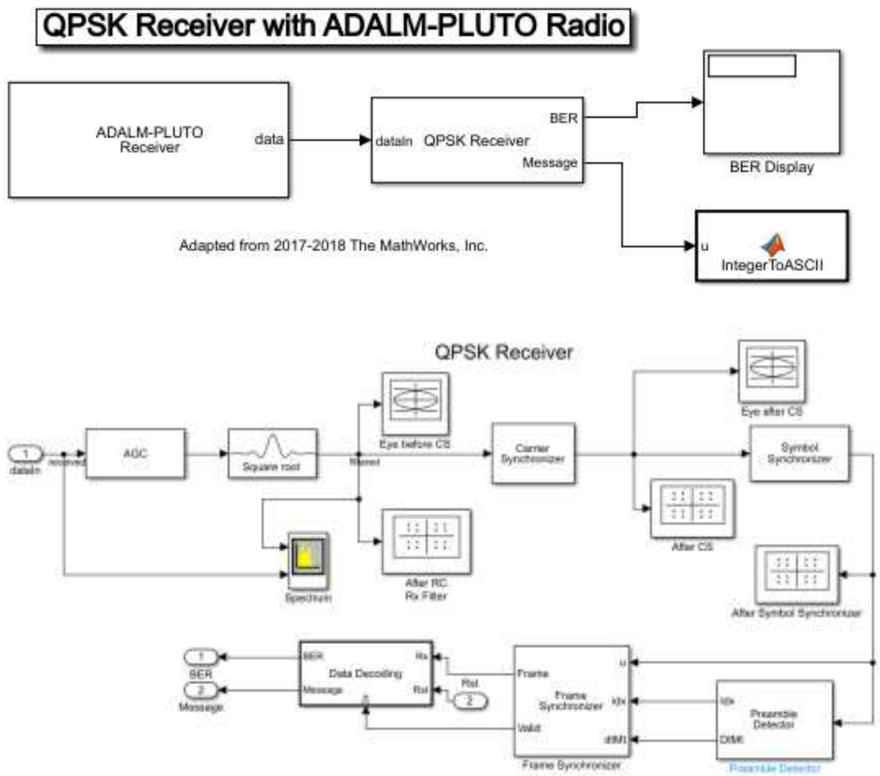

Figure 6: QPSK receiver implemented in Simulink

The functionalities of the various sub-blocks:

AGC - implements an automatic gain control system. This block adaptively adjusts its gain to give a constant signal level at its output. Such a block is common in systems in which the receiver input signal level changes due to variation in the communication channel.

Square root- filters the input signal using a raised cosine pulse shape filter. The block is configured to match the pulse shape used in your instructor's transmitter, and therefore implements a matched filter.

Carrier Synchronizer- Adaptively compensates for frequency deviation and phase difference between the transmitter and receiver oscillators.

Symbol Synchronizer - adjusts for symbol clock drift in the incoming signal, ensuring that incoming pulses are sampled (for purposes of deciding which symbol was sent) at the "opening" of the eye. The block outputs exactly one sample per symbol interval.

Preamble Detector- Detect the beginning of the frame that contains the transmitted information.

Frame Synchronizer-Synchronizes the plot that contains the information

Decode Data - Decodes the text message

4.3) Frequency Correction

4.3.1) Introduction: For a correct transmission between two PlutoSDRs, they should operate at the same center frequency and baseband sample rate. The baseband sample rate and the center frequency of a PlutoSDR are physically derived from the same oscillator. Although each PlutoSDR is factory-calibrated to use appropriate PLL settings to digitally compensate for component variations of the oscillator, the compensation is imperfect because of quantization effects and changes in operating conditions (temperature, in particular). Consequently, two PlutoSDRs will operate at slightly different baseband sample rates and center frequencies even if the corresponding System objects for the two PlutoSDRs use the same values for the `BasebandSampleRate` and `CenterFrequency` properties.

This difference in frequencies will be compensated by the “Carrier Synchronizer”, as will be studied later. However, if the frequencies of the local oscillators were higher than a few KHz, this block would not be able to make the correction. To reduce the frequency deviation between the two oscillators, a pilot (sinusoid) with a known frequency will be transmitted together with the QPSK signal. Reading this frequency in the baseband signal at the receiver will allow estimating the deviation in frequency between the two local oscillators. The frequency deviation will be corrected at the receiver through a fine adjustment of the local oscillator frequency, using the FrequencyCorrection property of ADALM-Pluto, as will be explained in the following steps.

4.3.2) Run the simulator and analyse the spectra at the input of the receiver and at the output of the “Square Root” filter as shown in Figure 7. It is possible to observe, added to the QPSK signal spectrum, a pilot signal that should be at a frequency of $f_{ref} = 175KHz$.

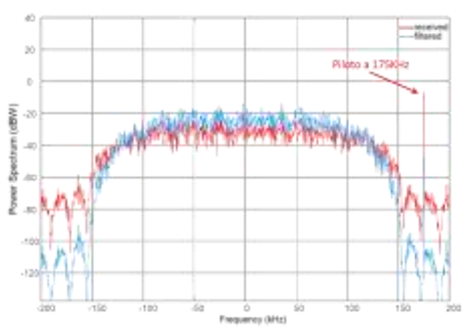

Figure 7: Spectra at the input of the receiver and at the output of the “Square Root” filter.

4.3.3) Using the “Peak Finder” functionality of the spectrum analyser, measure the frequency of this pilot, $f_{received}$.

Figure 8: Spectrum Analyser tool bar

4.3.4) The correction parameter “p”, in parts per million, for the central frequency is given by:

$$p = \frac{f_{received} - f_{ref}}{f_c + f_{ref}} * 10^6.$$

where: $f_{ref} = 175 \times 10^3\ Hz$ and $f_p = 860 \times 10^6 Hz$.

4.3.5) To access the “Block Parameters ADALM-Pluto Radio”, illustrated in Figure 9, double-click on the “ADALM-PLUTO Receiver”. In the “Advanced” TAB, place the calculated value of the deviation parameter p in “Frequency correction (ppm)”. Next, check that the pilot frequency is correct (175KHz).

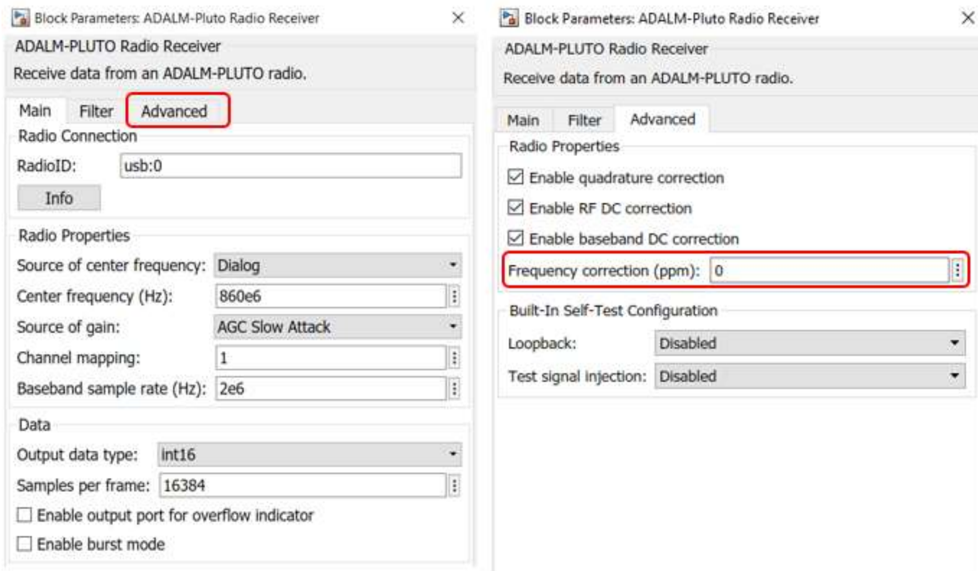

Figure 9: ADALM-PLUTO receiver configuration parameters

4.4) QPSK receiver characterization

4.4.1 Observe and acquire the spectrum at the output of the ADALM-Pluto Receiver and at the output of the square root raised cosine filter. Estimate the bitrate from the spectra bandwidths.
Suggestion: Observe the roll of factor parameter used at the Square Root block

4.4.2 Observe the Constellation Diagram plots at the input and output of the Carrier Synchronizer block. Take screen captures of each. Explain and interpret the plots.

4.4.3 Observe the Eye Diagram plots at the input and output of the Carrier Synchronizer block. Take screen captures of each.

4.4.4 Observe the Constellation Diagram plots at the input and output of the Symbol Synchronizer block. Take screen captures of each. Explain and interpret the plots.

4.4.5 From EVM measurement at the input of QPSK Demodulator estimate the BER.

4.4.6 Reduce the SNR to obtain an EVM at the input of QPSK Demodulator in range [-10 -5] dB. Take screen capture of the Constellation Diagram and estimate de BER.

Note: You can reduce the SNR in several ways, such as:

- decreasing the Gain of the Adalm-Pluto SDR Receiver
- placing an object near/over the receive antenna
- move your receiver to a different location

Carefully examine the effects of reduced SNR on Constellation Diagram and Eye Diagram plots.

4.4.7 Explain how Carrier Synchronizer and Symbol Synchronizer work.
Suggestion: Consult references [3] to [5]

Bibliography

[1] Communications Toolbox Support Package for Analog Devices® ADALM-Pluto Radio User's Guide, 2019.
[2] Communications Toolbox Support Package for Analog Devices® ADALM-Pluto Radio Reference, 2019
[3] Robert W. Stewart, Kenneth W. Barlee, Dale S. W. Atkinson, Louise H. Crockett, “Software Defined Radio using MATLAB & Simulink and RTL-SDR”, 2nd edition, January 2023.
[4] Rice, Michael. Digital Communications: A Discrete-Time Approach. Upper Saddle River, NJ: Prentice Hall, 2008.
[5] Mengali, Umberto and Aldo N. D’Andrea. Synchronization Techniques for Digital Receivers. New York: Plenum Press, 1997.