CS-330 Assignment 2
Modulation, Channel Impairments

Deadline: 20/11/2016 23:59 via turnin

November 9, 2016

General Information

The goal of this assignment is to become familiar with different modulation schemes, the impact of noise and frequency offset, as well as pulse shaping. Furthermore some up-sampling issues are investigated to better understand the challenges of sampling theorem.

Exercise 1

In this exercise you will investigate the accuracy of the three different up-sampling schemes proposed by students during the lectures. Consider the case of doubling the sample rate, thus for each input sample two are output.

1. Build the logic of the three schemes:
   (a) For each input sample a second sample is produced with zero amplitude (zero padding).
   (b) For each input sample a second sample is produced with the exact same amplitude (duplication/repetition).
   (c) For each input sample a second sample is produced with the average amplitude of the current and next input samples.
2. Use the proper block that allows the different input and output rates and implement the three aforementioned schemes.
3. Test the three schemes by using the following parameters:
   (a) Input sample rate 48000 Hz.
   (b) Output sample rate 96000 Hz.
   (c) Test signals 1 & 10 kHz (not simultaneously). Ideally use a slider to adjust the frequency in real time between 100 and 20000 Hz.
4. Report and discuss your findings. How is the frequency domain affected? Which system is the best?

   NOTE: You can evaluate all the schemes by using a spectrum analyzer block and by hearing the input and output signals with a soundcard.
5. After evaluating the three schemes, do you consider of a better scheme to up-sample? If yes, report the reasoning, build the scheme, test it against the others, and report your findings.

Exercise 2

This exercise will help you to get familiar with the modulation and constellation, as well as the impairments of noise and Carrier Frequency Offset (CFO).

1. Open the *modulation_example.grc* located in *examples* directory and save it at *lab2_2.grc* at the *apps* directory.

2. This flowgraph uses a BPSK modulation that maps data into constellation points. There are two sliders. The first controls the amplitude of noise, whereas the second inserts a Carrier Frequency Offset (CFO) at the system. Note that CFO is a common hardware impairment between the TX and RX device and telecommunication standards should always deal with it. Play with the sliders by adjusting their values and **report** what you observe. How each one of the impairment (noise, CFO) affects the constellation points?

3. Modify properly the flowgraph in order to support other modulations schemes too. More precisely the flowgraph should support:

   - BPSK
   - QPSK
   - 16QAM
   - 64QAM

   The constellation point mappings are depicted at Figures 1-4.

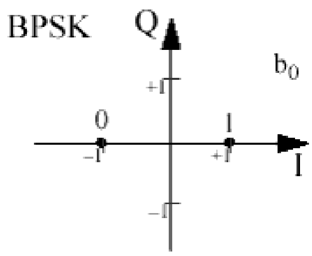

Figure 1: BPSK constellation points

4. You may observe that the maximum IQ amplitude of each modulation scheme is different. This means that the mean energy of each modulation scheme is different. In general this is not desirable. For this reason, perform normalization at the constellation points of each modulation scheme. The normalization factors can be retrieved from the table below. Report a screenshot for each one of the modulations.

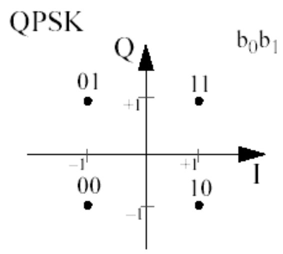

Figure 2: QPSK constellation points

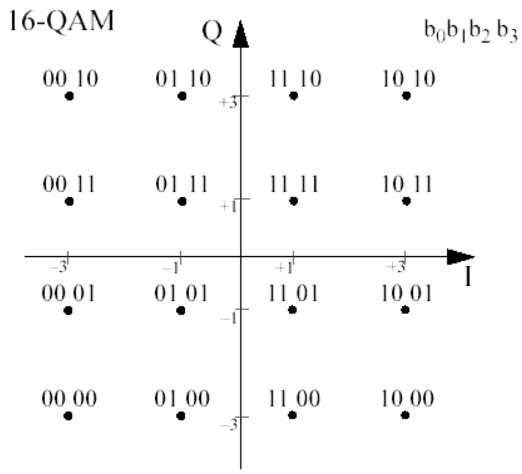

Figure 3: 16-QAM constellation points

Modulation	Normalization factor
BPSK	1
QPSK	$1/\sqrt{2}$
16-QAM	$1/\sqrt{10}$
64-QAM	$1/\sqrt{42}$

5. For each one of the modulations report the maximum possible noise amplitude that can be achieved without errors. Actually without a demodulator this value can not be precise. Just provide an empirical estimation. Compare your findings and provide a brief explanation.
6. Do the same for the CFO. How each of the modulations is affected. Is there any similarity with the previous task or not? Discuss.

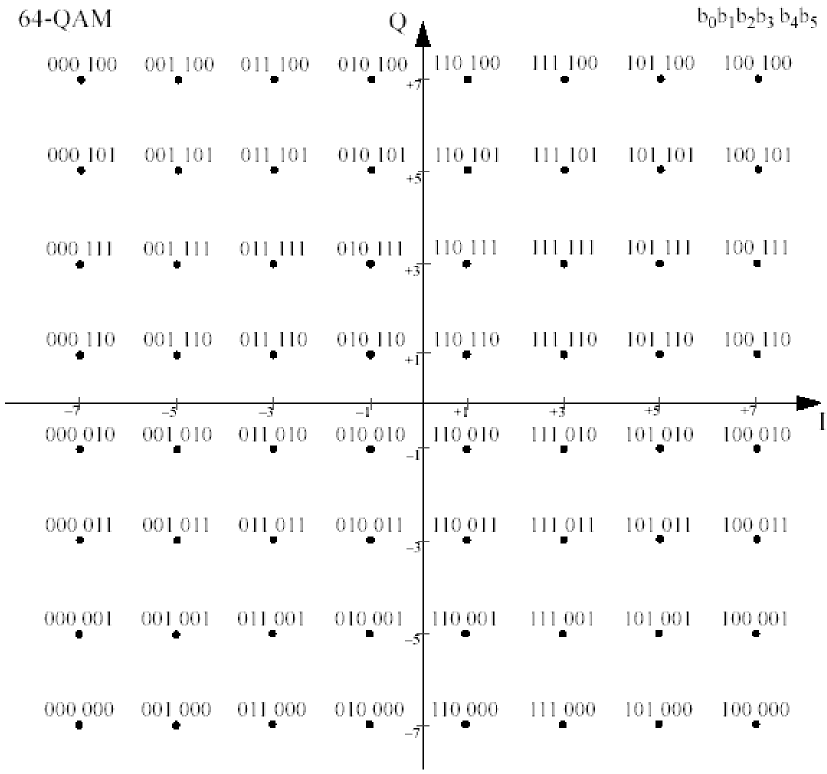

Figure 4: 64-QAM constellation points

Exercise 3

The system of the previous task occupies a bandwidth of 1 MHz, due to the 1 MSPS sampling rate. In general this is not the case. The hardware supports specific sampling rates (bandwidth) and the telecommunication system uses only a portion of it. In this exercise you are asked to make the proper modifications to the flowgraph of the previous exercise to support configurable occupied bandwidth configurations.

1. Open the *lab2_3.grc* flowgraph and integrate the modifications of the previous task that you made to support all the required modulations.

2. After the *Chunks to symbols* block there is a Polyphase arbitrary resampler. This block makes resampling and filtering at the same time. The filter taps are derived from the *taps* variable. You should **not** alter the filter parameters. To control the occupied bandwidth the *sample_per_symbol* variable should be changed. This variable controls how many samples the resampler will produce for each incoming constellation point. Why the resampling controls the occupied bandwidth of the system? Discuss, providing a simple but valid explanation.

3. Provide a screen-shot of the system occupying 400 KHz bandwidth. How many samples per symbol did you use and why?

4. Does the resampling altered the noise and CFO immunity of the system or not?

Exercise 4

Alter the flowgraph of the previous task, in order to support 8-PSK modulation too.

1. Assuming that one of the constellation points is the $(1+0j)$ find the others. How did you compute the coordinates of the rest of the constellation points? Report.

2. Provide a screenshot with the constellation.

3. Compare the 8-PSK with the QPSK, regarding the immunity in noise and CFO.

4. Compare the spectrum of 8-PSK and 16-QAM. What do you observe?

Exercise 5

In this exercise you will develop a block that performs the demodulation of the received constellation points.

1. Using the *gr_modtool* we have created a block called **constellation_demodulation**. This block has one complex input stream and **two** output ports. The first port is a byte output stream and the second is a float output stream. The names of the output streams are **out** and **evm** respectively. Furthermore, the block takes two arguments in order to identify the modulation scheme that is trying to demodulate. The number of K bits that each constellation point can carry and a boolean *is_psk* indicating if the constellation that is used is PSK or QAM.

2. For each complex number the block should decide which constellation point was sent. There are many ways to implement this. One naive implementation is the following.

   (a) For every received constellation point, compute the euclidean distances between the received point and all the known constellation points of the modulation scheme.

   (b) Find the minimum distance and the corresponding constellation point p of the modulation scheme.

   (c) Assume that p was actually transmitted and produce the corresponding K bits that represents.

3. Use the existing GNU Radio blocks to pack $K \leq 8$ useful bits into bytes of 8 bits. Then save the resulting bytes into a file using a file sink block.

4. The **evm** port should produce the Error Vector Magnitude (EVM) of each received constellation point. The EVM can be calculated using the below formula:

$$EVM(dB) = 10\log_{10}\left(\frac{P_{error}}{P_{reference}}\right)$$

   where P_{error} is the magnitude of the error distance between the received point and the reference constellation point, and $P_{reference}$ the magnitude of the reference constellation point.

5. You do **not** need to edit the **.xml** files for this block. We have done it for you!

6. Install the OOT module and ensure that is accessible from GRC.

7. Use the *lab2_2.grc* flowgraph and save it as *lab2_5.grc*. Connect the output of the multiply block with the input of **constellation_demodulation**. Use proper GUI sinks for both outputs. Play with the flowgraph parameters (Noise, CFO) and see how your demodulator performs. What do you observe? What about the EVM values?