ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Δίκτυα Καθοριζόμενα από Λογισμικό

Άσκηση 4

Ξενοφώντας Δημητρόπουλος

Τμήμα Επιστήμης Υπολογιστών

University of Crete
Computer Science Department
Lecturer: Prof. Dr. X. Dimitropoulos
TAs: Dimitrios Gkounis, George Nomikos
M. Lakiotakis, G. Vardakis

Autumn Semester 2014
HY436 - Software Defined Networks
Tasks of **Assignment 4**
Assigned: 22 November 2014
Due: 2 December 2014, 23:59

Assignment 4

Contents

1. What this exercise is about

In this exercise you will use the GNU Radio framework [1] in order to familiarize with basic telecommunication concepts and the most popular SDR software. Therefore there will be two quite different tasks regarding:

- FM Radio reception/decoding
- Error Vector Magnitude (EVM) estimation

Your tasks are to:

1. Decode and store in separate sound-files as many FM station transmissions as possible from a real-life 4 MHz spectrum snapshot incorporating a DDC.
2. Implement an EVM estimator and correlate the results to the applied SNR and the resulting bit error rate, for multiple constellations.

Most required modules are provided by the GNU Radio.
You will have to implement only a fraction of the required components.

2. Brief Introduction to GNU Radio

GNU Radio is a free & open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in hobbyist, academic and commercial environments to support both wireless communications research and real-world radio systems.

3. Initial Steps - Prerequisites

Before proceeding with the exercise, you are required to go through the following stages:

1. Visit and check out a custom version of GNU Radio appropriate for this lesson: `https://github.com/surligas/cs436-gnuradio`.
2. Install the dependencies
3. Compile and install the GNU Radio
4. Play and get familiar with the GNU Radio Companion IDE
5. Explore the FM Receiver module.

4. Task 1: FM receiver & Digital Down-Converter

(a) Overview of the Setup

Obviously, the FM Radio broadcasting uses the Frequency Modulation (FM) technique. Usually the region of the spectrum from 87.5 to 108.0 MHz is used and the stations are tuned at exact multiples of 0.1 MHz. The signal is modulated such as the required bandwidth is about 53 kHz for the stereo broadcasts [mono requires only 15kHz]. A real-world captured signal is provided to you here `http://139.91.68.99/~surligas/hy436/radio_capture_complex_96MHz.dat`. The USRP B210 SDR device was used for the recording. The SDR receiver was tuned in 96 MHz and 4 MHz sampling rate has been used, therefore the captured sample covers 94 to 98 MHz. In `http://radiomap.eu/gr/irakleio` you can spot which radio stations broadcast in the particular

range. There are about 10 stations that should have been recorded concurrently and your task is to tune into each one and try to save the result in an audio file. Although the FM Radio receiver (with all the proper filters, multipliers, etc.) is already implemented in GNU Radio, you need to implement a digital down conversion (DDC) system in order to be able to "tune" into a specific broadcast/channel/station. The DDC is based on the same principal of modulation/demodulation where we multiply a signal with a carrier frequency to either shift its frequency to a higher region, or shift it back to the original region (up-conversion & down-conversion). Instead of using for example an analog heterodyne system to perform this in hardware, you will apply the technique using software. By simple multiplication of the signal with a complex cos/sin waveform of the desired frequency [plus low-pass filtering] you can obtain the baseband signal.
Therefore your tasks will be to:

- create a complex cosine/sine waveform (DDC f_c)
- multiply the captured signal with the DDC f_c
- use a low-pass filter to acquire only the baseband part of the signal (is this necessary?)
- route the DDC signal to the FM receiver
- route the FM receiver output to your sound-card output / PCM audio file
- calculate the appropriate f_c to tune into the desired station - beware there is a high probability that there will be an offset between the frequency of the station and the SDR: do a fine-tuning by listening to the result (extra bonus for the ones who will implement an auto-fine-tuning system [hint: look for pilot signals in the FM broadcast spectrum]) - Is this offset the same for all the stations?

(b) Implementation details

A reference flowgraph can be found at:

`gr-cs436/examples/fm_receiver.grc`

This flowgraph provides most of the functionality of an FM receiver, among with a simple spectrum analyzer that will help you to locate the FM stations.
In order to dynamically change the target frequency of the 'tuner' you can use the *QT GUI Range* block. Due to the large bandwidth, the accuracy of a single GUI Range chooser will be low. For this reason, two different *QT GUI Range* blocks can be used. The first will help provide a coarse frequency selection, whereas the second one can help for a more accurate frequency adjustment.
Note: This task requires to use only the GNU Radio companion tool and the pre-installed available blocks. No custom code is required.

5. Task 2: EVM and BER block

(a) Overview

In this task you need to implement a synchronous GNU Radio block that calculates the EVM and BER of an input stream of symbols. The implementation files that you have to alter are the following:

- `gr-cs436/lib/evm_impl.h`
- `gr-cs436/lib/evm_impl.cc`

- `gr-cs436/include/gnuradio/cs436/evm_impl.h`

  (optional)

All the other necessary components, are implemented. If you want to compile and install the new code just execute the following inside the *build* directory:

```
make
make install
```

The parameters of the *EVM and BER* blocks are:

- The vector length of the input ports. In other words, this parameter specifies the number of constellation points that each input item includes. You may take a look at *gr-cs436/lib/constellation_source_impl.cc* in order to get the idea.
- The sampling rate. As we discussed at the exercise session, in SDR development we measure time in a per sample basis. Sampling rate is necessary to calculate the BER per second.
- The modulation scheme. This is necessary for the de-mapping process that is described below. Of course, the modulation scheme of the block that produces the symbols should be the same with the modulation scheme of this block. For convenience, modulation scheme can be specified through a drop down menu from the GNU Radio Companion IDE. Again for a working example you can refer to the *gr-cs436/lib/constellation_source_impl.cc* file.

You can test your block through the GNU Radio Companion (GRC) IDE. There under the *CS 436 Module* category, you will find the *EVM and BER* block. By using it, your code is added at the flowgraph and executed. For your code testing you can use the flowgraph located at:

`gr-cs436/examples/evm_simulation.grc`

This program includes also many graphical sinks that will help you to visualize your results.

(b) Implementation details

The block should have two input complex streams. Each item of these streams should be *vlen* complex numbers, where the *vlen* is a user defined parameter. The first input corresponds to the actually transmitted symbols and is the reference for calculating the EVM. The second input corresponds to the received symbols after they have been distorted by the wireless channel's impairments, i.e. noise.
Furthermore, the block provides two output floating point ports. The first port produces the EVM results, whereas the second provides the BER measurement. Note that each item of the output ports is exactly one float number instead of *vlen* complex numbers of the input ports. That means that for each input item of *vlen* complex numbers, one float number will be produced at each of the output ports.

i. EVM Calculation

In order to calculate the EVM, the distance of the received constellation point from the corresponding reference constellation point should be computed as depicted in Figure 1. Due to the fact that each input symbol has *vlen* constellation points, the mean of *vlen* EVM calculations should be produced.

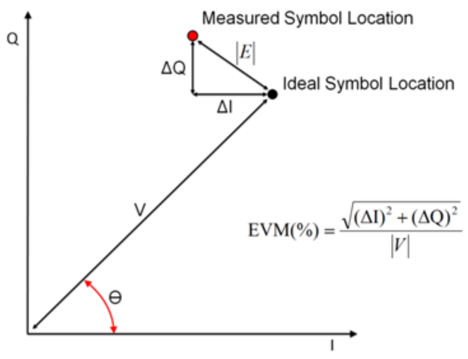

Figure 1: EVM calculation

ii. BER Calculation

For the BER measurement, the initial transmitted bit stream and the recovered at the receiver bit stream should be computed. This is a process known as constellation de-mapping.
A simple process to de-map the constellation points is the following. For every received constellation point, calculate the distance with each of the constellation points of the modulation scheme. Select as the transmitted constellation point, that point of the modulation scheme with the smallest distance from the received one.
The bit mapping and the constellation points of each modulation scheme can be found at the header file located at:

`gr-cs436/include/gnuradio/cs436/constellation.h`

The original transmitted sequence can be derived after de-mapping the constellation points of the reference input port. After obtained the reference and received bit sequence the BER can be retrieved easily.

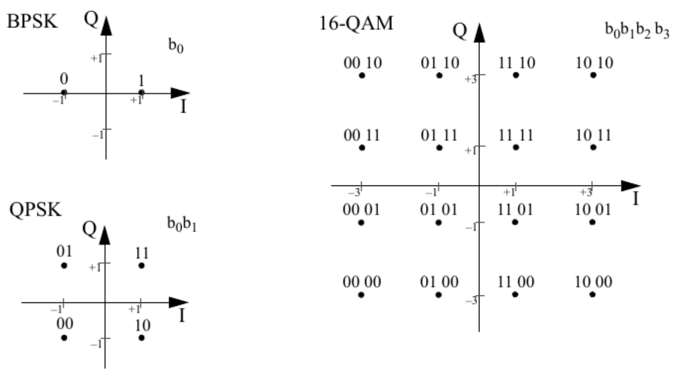

Figure 2: Constellations mapping

iii. Evaluation

Using the *evm_simulation.grc* flowgraph evaluate the performance under different noise amplitudes of each modulation scheme.

- Report the maximum EVM and the noise amplitude with BER equals to zero, for every modulation scheme
- Report the EVM for every modulation scheme when the BER is 50%

Asking for help

For any issues you may encounter regarding the exercise, please ask the teaching assistants during the exercise sessions (14:00-16:00) at K.206, or send an email to the course mailing list (hy436-list [at] csd.uoc.gr).

Before contact, please check the code and the comments that have been already provided within the code by the assistant.

Good luck!

References

[1] GNU Radio. The GNU Software Radio. *Available from World Wide Web: https://gnuradio.org*, 2014.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης