Teaching DSP and Telecommunications with GNU Radio and Low Cost SDR Hardware

Joary Paulo, Francisco C. B. F. Müller, Andrey Silva and Aldebaro Klautau

Abstract—**Software defined radio (SDR) is an important tool for communication research nowadays. It allows for better spectrum utilization and leads to smarter communication systems. Besides, due to its flexibility, research groups have been using SDR hardware platforms for quicker implementation of new algorithms and development of products. Therefore, the availability of flexible and relatively low cost platforms has been increasing and can also be explored in the teaching environment. This work describes the GNU Radio platform and presents some GNU Radio scripts for public use in teaching basic and intermediate level aspects of communications systems. The goal of this initiative is to contribute to the telecommunications community with easy-to-follow educational material that is made freely available.**

Keywords—**Education, GNU Radio, Software Defined Radio, Low Cost SDR Platforms.**

I. INTRODUCTION

Software defined radio (SDR) is a powerful tool for research in the communications area, as it allows for quick practical implementations. There are some examples of SDR systems implemented using GNU Radio, such as the OpenBTS (SDR GSM Network) and the OpenLTE, as can be found at the GNU Radio Archive Network [1].

GNU Radio is an open source software tool developed to simulate digital systems and generate/process digital data. It is also possible to transmit and receive real communication signals using a hardware platform, in the same fashion a sound card does with voice and music, for instance. Recently, the availability of cheaper platforms make it possible to use them not only for research, but also for teaching.

SDR is actively being used in teaching as shown in the 2012 IEEE Globecom conference, which had a session dedicated to software radio tools for teaching. Educational material has been developed and distributed by several universities around the world (e. g., UT Austin and RWTH Aachen). In Brazil for example, previous SBRT conferences described SDR implementations for teaching [2], [3].

This paper describes a project, called *TelClass*, which benefits from low cost SDR hardware. The project makes available at GitHub [4] a set of GNU Radio scripts to be used for education in the communications area, covering important techniques such as synchronization.

II. SDR HARDWARE PLATFORMS

Simulations are an important option but, in order to generate and analyze real-world signals, the use of proper equipments bring several advantages. According to the GNU Radio

nomenclature, this equipment is denoted *hardware interface* and is responsible for converting the digital signal "inside" GNU Radio into a real-world signal and vice versa. A typical hardware interface can be divided in three main system parts: the analog-to-digital (ADC) / digital-to-analog (DAC) conversions; the analog tuner, capable of up/down converting signals; and finally the transport, responsible for streaming the signal to/from the converter, constrained by the stream capability (S_c) of the connection interface (typically, USB or Ethernet).

Table I compares four distinct SDR hardware platforms. In the past years, the Universal Software Radio Peripheral (USRP) has been the most popular hardware used with GNU Radio. It can be purchased from National Instruments (NI) or directly from Ettus (acquired by NI). NI commercializes USRP kits that include educational material. Table I includes only two models: *N210* and *B100* and assumes the current prices suggested by Ettus. Nowadays, there are cheaper yet powerful alternatives to USRPs such as the *Jawbreaker* board developed within the HackRF project. If only a SDR receiver (without a transmitter) is useful, a DVB-T Dongle is the cheapest alternative. Table I lists the *Ezcap EZTV645* dongle, but there is a large number of models and a list of devices compatible with GNU Radio at [1].

TABLE I
EXAMPLES OF SDR HARDWARE PLATFORMS.

	U. N210	U. B100	Jawbreaker	EZTV645
F_s (Msps)	100	64	22	4
b (bits)	14	14	8	8
Δf (GHz)	0-6*	0-6*	0.03-6	0.064-1.7
S_c (Msps)	50	8	20	2.4
Price (US$)	1700	650	300	20
Obs.	full-duplex	full-duplex	half-duplex	just receiver

* Δf depends on the daughter boards used.

As indicated in Table I, GNU Radio can be used to analyze real-world signals obtained with hardware as cheap as 20 US dollars.

Fig. 1 shows a typical physical setup used in the Telclass project: the USRPs (*A* and *B*) generate signals that can be captured by students using DVB-T dongles (*C*).

III. TELCLASS SOFTWARE DEFINED RADIO SCRIPTS

One strategy adopted in the on-going Telclass project is to build scripts with increasing level of complexity and, consequently, some scripts share significant percentage of code. The parameters are described on the README.md of each script [4].
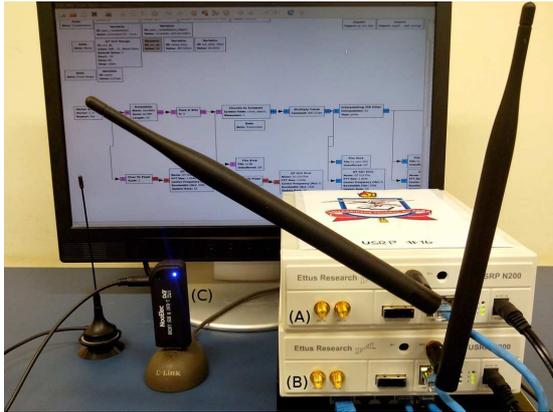
Fig. 1.   Setup with two USRPs and a DVB-T dongle.

To provide a concrete example of the kind of material that has been progressively made available, this section describes an example based on a general configuration: a signal flow graph composed by *Transmitter*, *Channel* and *Receiver*. Due to the lack of space, a very simple AWGN model is assumed, given that the number of parameters per script may be large depending on the communication techniques.

In the assumed example, the transmitter script can handle different modulation schemes: QAM, PAM and PSK.

A pulse shaping technique is applied, and different pulse shapes can be adopted: square, sinc and root raised cosine. The simulation script calculates and applies the proper power normalization factors.

An example of the general transmitter form is presented in Fig. 2, which depicts the block diagrams in GNU Radio Companion (a GUI for GNU Radio) of a transmitter using modulation 256-QAM and sinc pulse shape with oversampling of 2 samples per symbol.
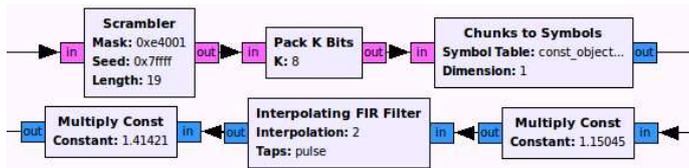
Fig. 2.   Transmitter flow graph for a system using 256-QAM modulation with sinc pulse shaping.

In the example, the channel is AWGN with unknown signal attenuation and its signal flow-graph is shown in Fig. 3
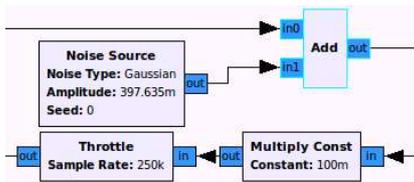
Fig. 3.   Channel Flow Graph.

Then, the receiver applies power normalization (automatic gain control), symbol extraction by matched filtering, and

decision. An example of such receiver is shown in Fig. 4.
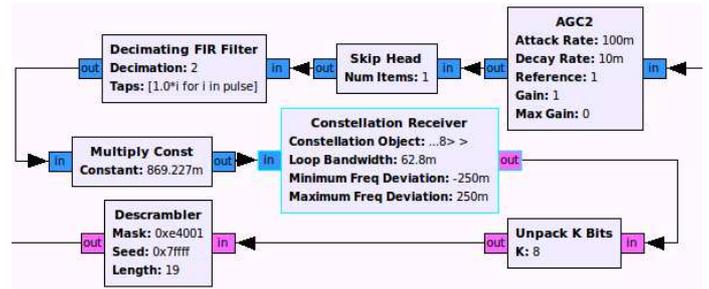
Fig. 4.   Receiver flow graph for a system using 256-QAM modulation with sinc pulse shaping.

Due to its internal Scheduler, GNU Radio allows to analyze the block output in real time. This way, the user can better understand the effects of each block along a communication system. The set of scripts made available at Telclass already define a graphical interface to show the signals after each block.

The scripts also save all the processed data in files, so that they can be used to generate new plots in tools such as Matlab or Octave. This was the process used to generate Fig. 5, which shows the PSD of the transmitted signal. Similarly, Fig. 6 show received symbols for a 256-QAM modulation after AWGN.
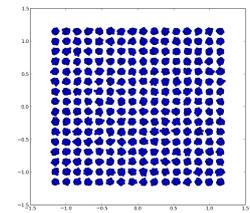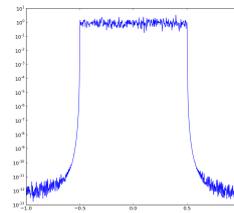
Fig. 5.   Transmitted Signal PSD.   Fig. 6.   Received constellation.

IV. CONCLUSIONS

This work briefly described low cost SDR hardware platforms and the Telclass project. This project has been publishing easy-to-use scripts for experiments in telecommunications using the GNU Radio platform and the SDR platforms. The results obtained along the project clearly indicate the positive impact that these tools have on students and general users.

REFERENCES

[1] CGRAN, ," The Comprehensive GNU Radio Archive Network - https://github.com/laps/telclass.

[2] F. Fernandes, C. Panazio, and D. Zanatta Filho, "Implementação de um software-defined radio com processamento em tempo real usando sinais acústicos e matlab," *Anais do XXVI SBrT*, Set 2007.

[3] Luciana Rêgo, Igor Almeida, Eduardo Medeiros, Lilian Freitas, and Aldebaro Klautau, "Uma base de sinais pública para ensino e pesquisa em telecomunicações," in *Anais do XXVII SBrT*, Set 2009.

[4] Telclass, ," https://github.com/laps/telclass.