Application of Digital Signal Processing in USRP Satellite Signal Detection

Giti Javidi, University of South Florida, College of Business, Sarasota, FL, USA
Ehsan Sheybani, University of South Florida, College of Business, Sarasota, FL, USA

ABSTRACT

The Universal Software Radio Peripheral development technique is designing and implementing radio frequency based systems. The distinctiveness originates from the interchangeable daughterboard within the USRP. The system is designed around the Xilinx Vertex 3 FPGA chip. This means C++, Python, and VHDL can be used to program this device. The project consists of creating a receiver. The objective of the project is to research and comprehend the hardware functionalities of the USRP. The purpose is to create codes in C++ and Python to implement receiving capabilities of the device. The goal of this project was to design a GPS receiver that is capable of recording the L1 signal from a DirecTV satellite. The USRP is used a lot for research. This project consisted of more than just one method. We used GNU Radio Companion and Matlab/Simulink. GNU Radio is open source for building software defined radios. It is also known as GRC. While using GRC the USRP1 was the device used. This software has rapid development. It runs in Ubuntu, a Linux operating system. Within this software there are logic blocks. Each block consists of information to create a flow graph. The flow graph builds and generates the program. Simulink can be compared to GRC. They both have logic blocks that have to be connected to run. Simulink can be used to create a transmitter or a receiver for software radio development and signal processing. Software-defined radio can only be defined if its baseband operations can be completely defined by software. A SDR converts digital to analog signals. The USRP can also convert digital signals from a computer to Radio Frequency Signals (RF). This software is one way to communicate between hardware and software.

KEYWORDS

Detection, DS, GNU, GRC, Satellite Signal, USRP

INTRODUCTION

What follows is a research report detailing the steps taken to apply USRP in detecting satellite signals. One should be able to follow these steps with the specified set of hardware and software to get the same configuration on the proper USRP device. These projects were implemented under the supervision of NASA scientists and faculty advisors in dedicated NASA labs. Following lessons learned here and authors’ previous experiences in data visualization and signal processing research and training, (Javidi & Sheybani, 2008; Sheybani & Javidi, 2006; Sheybani, Ehsan; Garcia-Otero, Adnani, Javidi & Deshpande, 2012; Ouyang et al., 2010; Sheybani, Javidi & Garcia-Otero, 2008; Javidi
& Sheybani, 2010; Varde et al., 2007; Sheybani, Javidi, Hardy, Denton & Campbell, 2007; Sheybani & Javidi, 2007; Badombre-Wanta & Sheybani, 2010; Sheybani & Sankar, 2002; Sheybani & Arora, 1992, Sheybani, 2011; Garcia-Otero & Sheybani, 2011) relevant labs were designed to enhance the Computer Engineering program at the Virginia State University (VSU).

The Universal Software Radio Peripheral development technique is designing and implementing radio frequency based systems. The distinctiveness originates from the interchangeable daughterboard within the USRP. The system is designed around the Xilinx Vertex 3 FPGA chip. This means C++, Python, and VHDL can be used to program this device. The project description consists of creating a receiver. The objective of the project is to research and comprehend the hardware functionalities of the USRP. The purpose is to create codes in C++ and Python to implement receiving capabilities of the device. The goal of this project was to design a GPS receiver that is capable of recording the L1 signal from a DirecTV satellite (Anon, 2014; Ettus, 2014; Mathworks, 2014.

The USRP is used a lot in RF communications research. This project consisted of more than just one method. We used GNU Radio Companion and Matlab/Simulink. GNU Radio is open source for building software-defined radios. It is also known as GRC. While using GRC the USRP1 was the device used. This software has rapid development. It runs in Ubuntu a Linux operating system. We had to get familiar with this software, GNU Radio Companion, Python, and Linux. Within this software there are logic blocks. Each block consists of information to create a flow graph. The flow graph builds and generates whatever is programmed, if programmed correctly. Simulink can be compared to GRC. They both have logic blocks that have to be connected to run. Simulink can be used to create a transmitter or a receiver for software radio development and signal processing. Software-defined radio can only be defined if its basebands operations can be completely defined by software. A SDR converts digital to analog signals. The USRP can also convert digital signals from a computer to Radio Frequency Signals (RF). This software is a way to communicate hardware systems with software.

**METHODOLOGY**

This project consists of plenty of research. We began researching general information on the Universal Software Radio Peripheral and GRC. The setup to this was simple and did not require much. We used a USRP1 and also the USRP2, a function generator and a computer. Figure 1 displays the setup with the USRP2. While doing this research, we came across some simple examples that we could conduct to get a feel of the USRP and GNU Radio. The first thing we came across was to create a dial tone. A dial tone (Figure 2 and Figure 3) has two different frequencies, one in which is a high one and another one that is low. The standardized frequencies for a dial tone are 350 Hz and 440 Hz.

To create a dial tone in GNU Radio we had to use several different blocks. Below you will find the flow graph and the dial tone signal. To create the dial tone, the blocks you will need are UHD: USRP source block, WX GUI Scope Sink, two signal source blocks and an audio sink block. The USRP source block provides the receiver data. The scope block provides a graph like so we can see the signals. The two signals are for the two different frequencies that make up a dial tone. Of course, the audio sink block outputs the sound. We were able to hear the dial tone because of the audio sink block. Changing its frequencies affects the pitch of the sound.

Another example was the FM radio receiver. This is a more intense signal than the dial tone. The FM radio receiver signal is similar in shape to the GPS signal. To receive a signal, we had to connect an antenna to the USRP. Once we configured and connected the GNU blocks and generated it we received the signal. We were able to hear the station clear if the antenna was in the right direction of arrival. After the receiver started to work we recorded and replayed it back. The purpose of this
An Intelligent Scheduling Architecture for Mixed Traffic in LTE-Advanced Networks
www.igi-global.com/chapter/an-intelligent-scheduling-architecture-for-mixed-traffic-in-lte-advanced-networks/99344?camid=4v1a

Peer-to-Peer IP Traffic Classification Using Decision Tree and IP Layer Attributes
www.igi-global.com/article/peer-peer-traffic-classification-using/1448?camid=4v1a

A GPS Based Deterministic Channel Allocation for Cellular Network in Mobile Computing
www.igi-global.com/article/gps-based-deterministic-channel-allocation/37529?camid=4v1a