On-line Alternation of Modulation Techniques using a Dynamic Switch

RadioPlus

Shashank Gaur, Sankalp Agarwal, Ammar Almasri, Anas Bennani

Masters Students at

ECE (Ecole d'Ingenieurs), Paris

gaur@ece.fr, sagarwal@ece.fr, almasri@ece.fr, bennani@ece.fr

Abstract: - In the ongoing era, the need of wireless communication is increasing at an exponential rate, alongside the development requirement is also rising in order to provide efficient, uninterruptable and effective services all around the globe. We propose an intelligent system, which would be able to switch in-between, the modulation techniques during on-line transmission in order to communicate with the surrounding environment and to provide efficient SNR and throughput.

Keywords: - USRP, SNR, BER, GNURadio, RadioPlus, SDR, GRC

1. Introduction

This paper describes a software solution in implementing a reconfigurable system without changing the hardware. RadioPlus (our project) provides a switching mechanism of modulation techniques to communicate efficiently according to the varying network conditions and to communicate with the surrounding networks (running on different modulation techniques).

Next generation wireless communication systems would have two important features: heterogeneity and adaptation. Heterogeneous wireless communication platforms require interoperability among different networks such as wireless local area networks, wireless personal area networks, etc. In a wireless network, the networking parameters (such as change in modulation techniques or network protocols varies continuously, so it is necessary to adapt with fluctuations in the network, while transmission and reception are in progress. The main problem is of maintaining synchronization in between Tx and Rx, without loosing data and connection (online). This is why an intelligent communication technique, which adapts itself to the prevailing channel conditions is sought in order to deliver best possible performance.

With the development of technologies, more and more real-time applications, such as Intelligent Clothing [14], Sensor network for Aircrafts [9], military (battlefield surveillance, reconnaissance of enemy forces, nuclear, biological and chemical attacks), environmental applications (forest fire and flood detection, monitoring of drinking water and level of air pollution), home applications (intrusion detection, home automation), commercial applications [11] etc. could come into existence.

In response to the above demands for the next generation wireless communication system, this paper aims at providing a timely and concise solution, which is focused on the current technology and research activities in the wireless communication fields.

The technique we are proposing is based on GnuRadio Companion (Software used) and the platform is known as Software Defined Radio. The GRC helps in providing software interface to the hardware (USRP-1), which works as a digital modulator and demodulator for transmission and reception respectively. The ability to select the required modulation scheme is a major advantage in a wireless network. Our system closely keeps a track of the network behavior and can take a decision regarding the change of modulation techniques (BPSK and QPSK in our case) [A] during on-line transmission, while maintaining the real time objective.

Various approaches and algorithms regarding adaptation have been proposed. Researchers have worked with many system architectures and techniques for the adaptive modulation and demodulation. We are taking references of the following two techniques:

1. Modulation Recognition Technique, which can help in recognizing the modulation at the receiver using signal space partitioning and pattern recognition approach, which can be useful in order to synchronize the modulation at the receiver end. [5]

2. Implement TTSAS algorithm to change the Modulation technique at the receiver, So that it will automatically change the Modulation technique according to the transmitter. [13]

2. System Architecture

As we know, different networks have different modulation techniques, for example a blue-tooth wireless network uses GFSK, DQPSK, 8PSK etc. where as a ZIGBEE network requires OQPSK and DSSS for the purpose of communication. But in any general case if we need to communicate between two different networks, Network1 should be able to demodulate the signal for that particular modulation technique, which is been used in Network2. For example In a medical facility, employed with ZIGBEE or IEEE 802.15.4 network, if there are changes in the status of a patient's health, these changes can be sent directly to the respective doctor's phone/device, no matter in which network doctor is, at that moment.

The system proposed here is basically a combination of Hardware and Software. The Hardware is, Universal Software Radio Peripheral (USRP-1) [16] and Software Platform is GnuRadio. The idea is to develop an autonomous system having an intelligent switch, which can take decision itself regarding the switching of modulation techniques during on-line transmission, taking care of synchronization between TX and Rx. By using this switch, we would be able to change the modulation techniques according to the network requirement. Even user would be able to switch between modulation techniques according to his/her needs; this provides additional flexibility to user.
In case of low performance due to a particular modulation technique i.e. high BER, system should be able to do the calculation of parameters (BER, SNR etc.) and switch to another modulation techniques (with less BER, which results to high SNR). After switching, it should compare the new parameters to the parameters of previous modulation technique, if the performance of system improves, it should continue transmitting on the current modulation else it can switch to other techniques or switch back to previous one. Figure 1 provides the rough overview of our system, with a switch using - GPSK and BPSK modulation techniques.

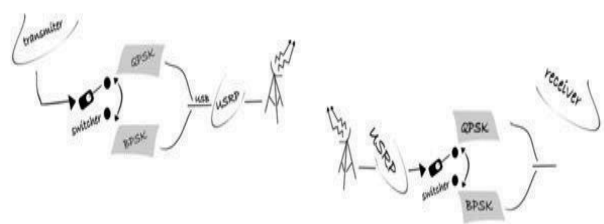

Figure 1

This idea can also be implemented in applications, where data security is important (Figure 2).

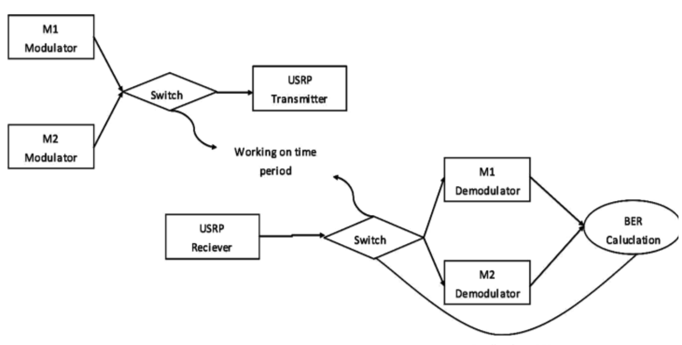

Figure 2

The switch can change modulation techniques after a random period of time and only the receiver having knowledge of that change could demodulate and use the data.

2.1 HARDWARE ARCHITECTURE

USRP-1 (Fig. 3) is our hardware platform. It is used as digital modulator - demodulator, digital baseband and IF section of Radio Communication. The inputs and outputs are connected through FPGA. FPGA is also connected to the USB Interface chip. Basically a motherboard has 4 input and 4 outputs. These are high-speed 64 bit digital I/O ports and they are divided in groups for RX and TX. Each side of motherboard (A and B) has 16 bit I/O pins for RX and TX. There are 4 high speed 12 bit and 14 bit ADCs and DACs

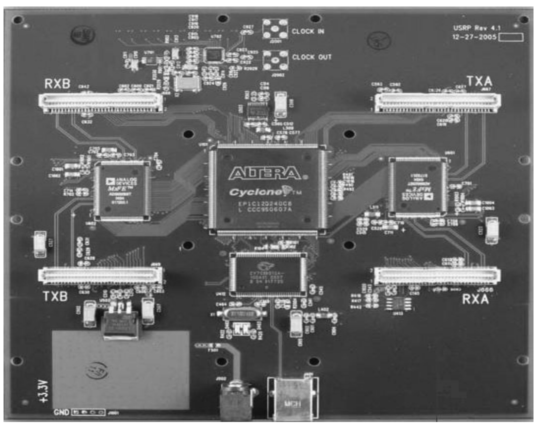

Figure 3 [17]

respectively. The FPGA is the main part of USRP. It performs complex calculation and minimizes the data rate so that, USB cable becomes compatible with it.

The motherboard uses different daughter-boards (Fig 4) to transmit and receive on different frequencies. A basic TX daughterboard is interface to DAC of USRP and basic RX daughterboard is interface to ADC of USRP. Mostly RFX 900 and RFX 2400 daughter-boards are used which works on 900 MHz and 2400 MHz respectively.

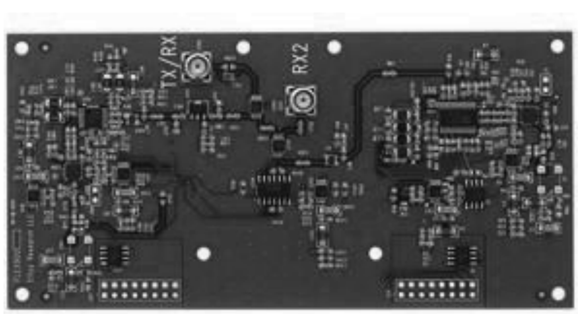

Figure 4

2.2 SOFTWARE ARCHITECTURE

GnuRadio [1] is a free/open-source software toolkit for signal processing. It uses a combination of C++ and Python language. The computationally intensive processing blocks are implemented in C++ while application, control and coordination of these blocks are defined in Python. Benefit of software radio is that it can replace the traditional radio. Instead of expensive fixed devices, the flexibility of the software is more attractive. It is interesting to see that one can use just one portable device to listen radio, watching TV etc. [2]. We implemented BPSK and QPSK modulation techniques, within our system (dynamic Switch) in order to achieve the alternation between these two techniques during online transmission.

Before detailing the working mechanism, we need to introduce some core software modules in Gnu Radio.

GRC blocks: In Gnu radio there are many forms of GRC blocks that depend on the methods, a block handles signal/data processing. Briefly, each GRC block gets one or multiple data input streams, it processes them and generates one or multiple data output streams (.py files). However there are following two exceptions:

- GRC block (a source of data) in a flow graph has only output stream(s) conversely.
- The second GRC block (a sink) has only input stream(s).

Different GRC blocks, as shown in Figure a, vary in detailed implementation of the work function. These blocks are written in C++ and are executed by using Python. Python provides glue between C++ and functionality of block.

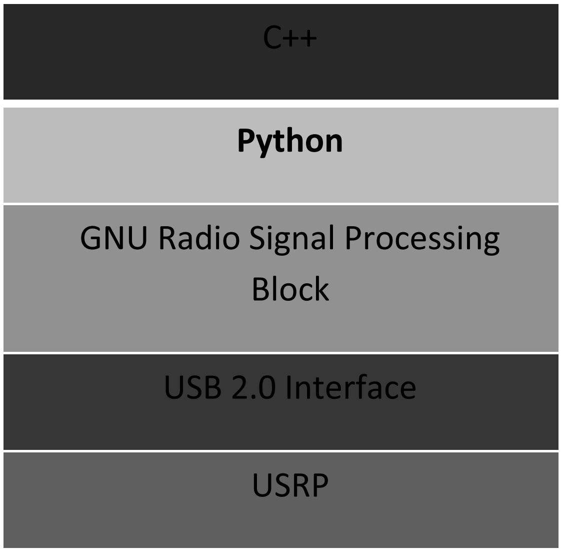

Figure a

For online alteration of modulation techniques, there are two solutions:

1. After certain period of time we can change the modulation technique at transmitter, and accordingly the receiver will acknowledge the change. For the ease of synchronization between Tx and Rx, time period algorithm has been used.
2. We can send a specific stream of bits and the receiver will acknowledge the stream in order to demodulate accordingly, So that it can get synchronized with the transmitter.

3. SYSTEM DESCRIPTION

Our system depends upon complex mathematical calculations and their practical implementation using python.

Signal-to-noise ratio (SNR) is the platform on which our system depends. It is defined as the power ratio between a signal and noise. It can be derived from the formula: -

SNR = (Psignal)/(Pnoise) = Es/N0

SNR is typically expressed logarithmically in decibels (dB).

Bit-Error-Rate (BER) is the ratio of the number of bit errors to the total number of bits transferred over an infinite time interval. Mathematically, we can express this definition as: -

$$BER \triangleq \lim_{n \to \infty} \frac{\varepsilon(n)}{n}$$, Where

"n" is the number of bits transferred and the "ε" is the number of errors among those n bits.

Relation between SNR and BER: -

Better the SNR, lower is the BER. Mathematically,

$$BER = \frac{1}{2} erfc(\sqrt{2Eb/N0})$$. Where

"erfc" is the error function (also called the Gauss error function).

QPSK: - It is known as Quadrature phase-shift keying. The analysis shows that QPSK can be used either to double the data rate compared with a BPSK system while maintaining the same bandwidth of the signal, or to maintain the data-rate of BPSK but halving the bandwidth needed.

$$\alpha_{QPSK} = \{\pm 1 \ \pm 1j\}$$

The theoretical plot representing values for QPSK in all the quadrants has been shown in Fig 5.

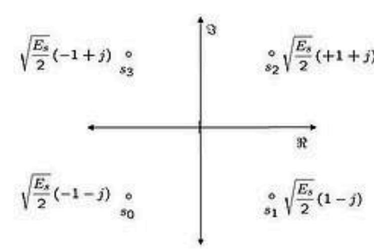

Figure 5

The practical plot, which has been obtained during the implementation, has been shown in Fig 6.

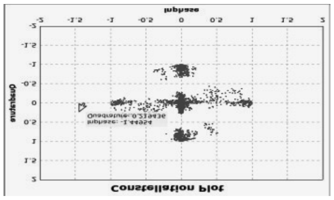

Figure 6

Above plot descries the QPSK modulation with 4 constellation points.

BPSK: - It is the simplest form of phase shift keying (PSK). It uses two phases, which are separated by 180° and

$$\alpha \text{ BPSK} = \{-1\ 1\}$$

The practical plot, which has been obtained during the implementation, has been shown in Fig 7.

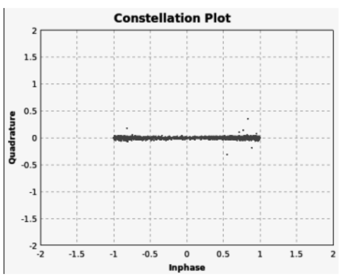

Figure 7

Above plot descries the BPSK modulation with 2 constellation points.

The GRC flow graph for Transmitter is been represented in the Fig 8.

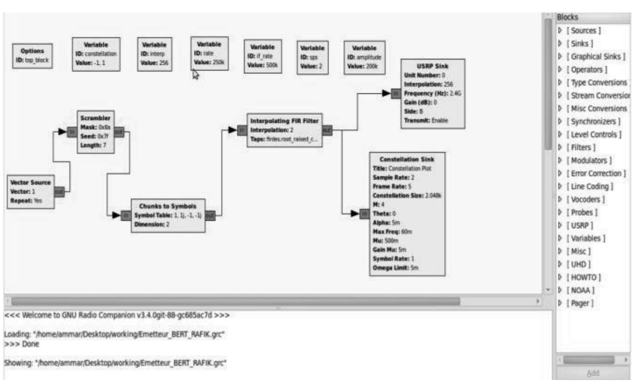

Figure 8

Here switching between BPSK and QPSK has been done by using the algorithmic logic of python programming at backend. The developed python file can be used in same structure but with additional functionality of switching. Here switching can be done randomly between the modulation techniques, after a predefined period of time.

Once the transmitter changes the modulation, it is necessary to synchronize Transmitter and Receiver, so that Receiver could demodulate respective data stream sent by Transmitter.

For the same purpose we have used an algorithm in transmitter which sends a predefined bits, suppose 16 zeroes, whenever there is a change in modulation techniques by Transmitter.

In our Transmitter at each interval of 10 seconds, there should be a change in modulation technique and at each change it first transmits the pseudo bits i.e. 16 zeroes and then it transmits the data with the changed modulation techniques.

Although our technique is time dependent, but it is able to switch the modulation techniques and sent out pseudo bits to detect the change.

The GRC flow graph for Receiver has been represented in the Fig 9, as follows

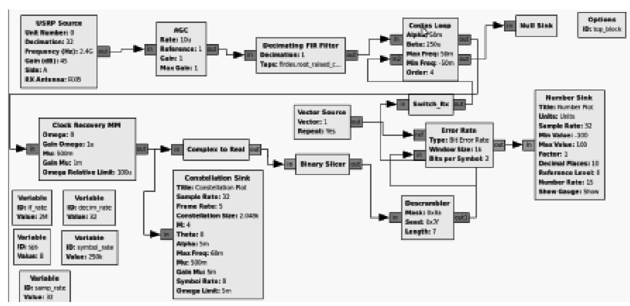

Figure 9

This flow graph has been made with the help of GRC and the block of switch, which is our main system, has been designed and implemented in Python and C++.

The receiver works as a normal receiver, which demodulates the data and compares it with the standard data and calculates the Bit Error Rate for the transmission and reception quality analysis, except one special feature which is switch_rx. This block has been designed with one input and one output. It takes input from Descrambler and gives output to Costas Loop. Its input to Costas Loop can change the order of the Costas Loop. As we know that order (M) for BPSK should be 2 and for QPSK should be 4.

So if we wish to read the data received by our hardware after demodulation, it can be done at the output of Descrambler. Switch_rx continuously reads the output of descrambler and if it detects the pre defined pseudo bits, in our case 16 zeroes, it changes the order of Costas Loop from previous to a new modulation technique.
In this way if the receiver was demodulating on QPSK, with Costas Loop's order equal to 4, on receiving the pseudo bits the switch will change the value of order to 2 and will start demodulation on BPSK. All of this synthesis has been implemented on C++ and Python level of the programming.

The switch has been developed in C++. The algorithm behind the switch is to compare the predefined pseudo bits to its input and give the output to Costas Loop block. We have also developed an extra input in Costas loop to connect the same output of switch. The change in the value of Costas Loop's order has been implemented in C++. As previously it was predefined using the "Costas_Order()" function but now it can be changed by the other input from switch block. Although we have kept this input as an optional input so that the block can also be used in normal circumstances.
The connection of switch block to the inputs and outputs of Costas Loop and Descrambler respectively has been done in the python.

4. EVALUATION

Before proceeding further it is necessary to calculate the BER for some modulation techniques, So that we can choose some of the best output providing modulation techniques to switch in between the online transmission.

	M	Amplitude	BER
1.	4	5000	0.001500
2.	4	1500	0.04800
3.	4	1000	0.215499
4.	4	500	0.249500
5.	4	250	0.250499
6.	4	10	0.36700
7.	2	5000	0.0010
8.	2	1500	0.00300
9.	2	1000	0.0400
10.	2	500	0.04199
11.	2	250	0.3930
12.	2	10	0.4580

Figure 10

As previously mentioned that order for BPSK, M= 2 and for QPSK, M= 4. The experimental results we obtained have been mentioned in Fig 10 and graphical comparison has been shown in Fig 11.

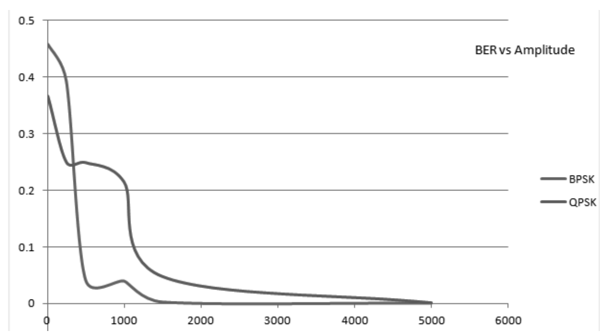

Figure 11

As we mentioned earlier, our system works upon the principal of qualitative comparison in between BER of various modulation techniques.
The main goal was to achieve a perfect qualitative working model which compares the current BER value with the previous values, If the current value (BER) is more than the previous one, then receiver sends the pseudo bit to Transmitter in order to switch to another modulation technique which has less BER.
For the same purpose we are developing another block switch_tx, which can acknowledge reception of the pseudo bit from receiver and

change the modulation technique at transmitter, instead changing it after a predefined time period.

Transmission:

After the switch GRC flow graph has been made, by using the command prompt, we would provide the necessary parameters such as frequency, M (Order i.e. 2 for BPSK and 4 for QPSK), bandwidth etc. to run Transmitter.py file. Initially it [B] runs on BPSK and then after a predefined duration of time (10 seconds), transmitter switches to QPSK. The benefit is that one needs not to disconnect and reconnect the USRP during the whole process. Activity has been shown in Fig 12.

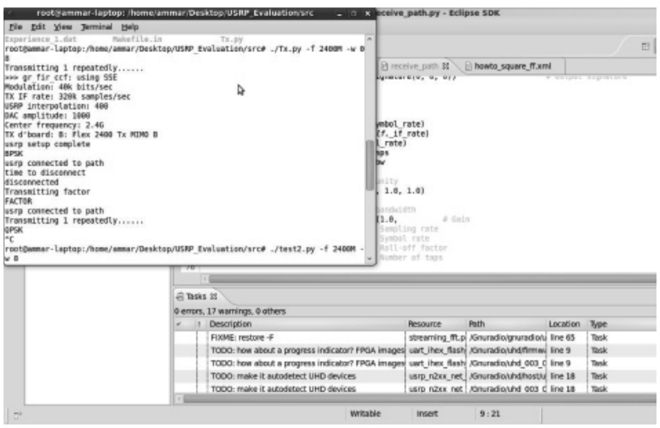

Figure 12

Reception:

On the receiver side, according to the transmitted signal, the receiver demodulates. As we discussed above [B] initially signal modulated with BPSK has been sent. So, receiver demodulates the BPSK signal and after that the demodulation should de done as per the signal sent by transmitter for ex: In our case, QPSK signal. Here for the ease of implementation we just took BPSK and QPSK signals, as described in the introduction [A]. But with some minor changes one can add a number of modulation techniques. Activity has been shown in Fig 13.

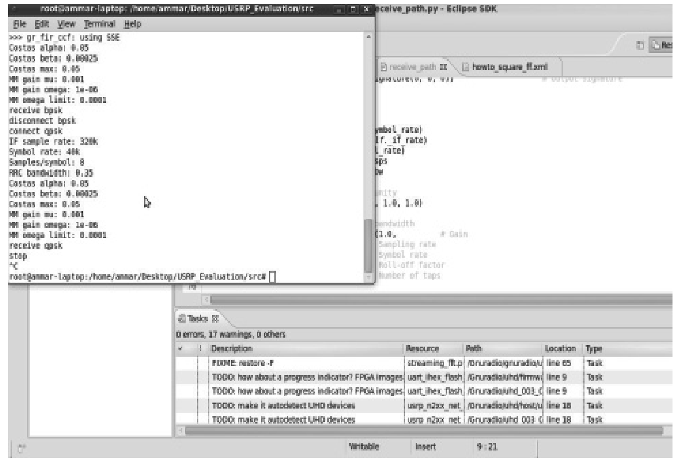

Figure 13

5. Conclusion

We have discussed and successfully demonstrated the purpose and implementation of dynamic switch, which is a necessary requirement of the adaptive wireless transmission network. The results for different modulation techniques (BPSK, QPSK) have been represented. The switch is just a simple implementation of C++ and python language but plays an important role for unperturbed and continued communication not only with in its network but also with surrounding environment having different networking parameters.

Acknowledgment

We thank Mr. Rafik ZITOUNI for being the source of motivation and support through out our entire project. We like to thank Mr. Stefan ATAMAN for providing us time-to-time suggestions. We are also obliged to LACSC research lab of ECE, Paris, for providing all the required resources.

However we can never forget the inspiration accommodated by Dr. Mohammed Ikbal BENAKILA. We are highly thankful to Prof. Marcus D. Leech, for the guidance through the forum for “GNU Radio”. Last but not the least we are thankful to all our friends and family for the support.

REFERENCES

[1] Ke-Yu, Chen (8818-0493), Zhi-Feng Chen (1218-1197) “GNU Radio” Dept. of Electrical Computer Engineering University of Florida Gainesville, Florida

[2] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker, “Sora: High performance software radio using general purpose multi-core processors,” in Symposium on Networked Systems Design and Implementation (NSDI), 2009.

[3] Veljko Pejovic and Elizabeth M Belding “A Context-aware Approach to Wireless Transmission Adaptation”

[4] Jefferson L. Xu, Weo Si and MengChu Zhou “Software-Defined Radio Equipped With Rapid Modulation Recognition”.

[5] “Modulation Scheme Recognition Techniques for Software Radio on a General Purpose Processor Platform” by Keith Nolan, Linda Doyle, Donal O’Mahony, Philip Mackenzi, Networks and Telecommunication Research Group, Trinity College, Dublin

[6] A. Kaul, “Software defined radio: The transition from defense to commercial markets,” in 2007 Software Defined Radio Technical Conf. Product Exposition, Nov. 2007.

[7] F. Daneshgaran and M. Laddomada, “Transceiver front-end technology for software radio implementation of wideband satellite communication systems,” Wireless Personal Communications, vol. 24, no. 2, pp. 99–121, 2003.

[8] M.N. O. Sadiku and C. M. Akujuobi, “Software-defined radio: A brief overview”, IEEE Potentials, Oct.-Nov. 2004, Vol. 23, No. 4, pp. 14-15

[9] Jorge M.Pereira ”Reconfigurable Radio: The evolving Perspectives of different players”. Personal, 12th IEEE International Symposium on In- door and Mobile Radio Communications, 2001

[10] Livia Ruiz, Gerard Baldwin and Ronan Farrell Centre for Telecommunications Value-Chain Research “A PLATFORM FOR THE DEVELOPMENT OF SOFTWARE DEFINED RADIO”

[11] Vincenzo Mancuso and Sara Alouf INRIA Sophia Antipolis Me´diterrane´e, France “Reducing costs and pollution in cellular networks”

[12] H. Lu “Making green telecom real” Communicate Huawei online publications, no. 45, Jan. 2009

[13] Negar Ahmadi “Modulation Recognition Based on Constellation Shape Using TTSAS Algorithm and Template Matching”

[14] Paul KIEKENS, Lieva VAN LANGENHOVE, Carla HERTLEER, Ghent University – Department of Textiles “SMART CLOTHING: A NEW LIFE”

[15] S. Srikanteswara, R. C. Palat, J. H. Reed, and P. Athanas, “An overview of configurable computing machines for software radio handsets,” Commun. Magazine, IEEE, volume 41, number 7, pp 134–141, 2003.

[16] Matt Ettus, “USRP USER’S AND DEVELOPER’S GUIDE”, www.ettus.com, http://home.ettus.com/usrp/usrp guide.html, (10 Sep 2011)

[17] Comsec, “USRP INSTALL”, www. comsec.com, http://comsec.com/wiki?UsrpInstall, (10 Sep 2011