Building Software-Defined Radios in MATLAB Simulink – A Step Towards Cognitive Radios

Ahmad Ali Tabassam, Farhan Azmat Ali[1]
[1]Department of Information Technology (INTEC)
Gent University
9000 – Gent, Belgium
a.tabassam@gmail.com, farhan.azmatali@intec.ugent.be

Sumit Kalsait[2], Muhammad Uzair Suleman
[2]Business Unit Automation, Research & Development
Phoenix Contact Electronics GmbH
32657 – Lemgo, Germany
skalsait@phoenixcontact.com, uzairengr@gmail.com

Abstract – Software Defined Radio (SDR) **is a flexible architecture which can be configured to adapt various wireless standards, waveforms, frequency bands, bandwidths, and modes of operations. This paper presents a detailed survey of the existing hardware and software platform for SDRs. It also presents prototype system for designing and testing of software defined radios in MATLAB/Simulink and briefly discusses the salient functions of the prototype system for *Cognitive Radio (CR)*.**

A prototype system for wireless personal area network is built and interfaced with a *Universal Software Radio Peripheral-2 (USRP2)* main-board and RFX2400 daughter board from Ettus Research LLC. The philosophy behind the prototype is to do all waveform-specific processing such as channel coding, modulation, filtering etc. on a host (PC) and general purpose high-speed operations like digital up and down conversion, decimation and interpolation etc. inside FPGA on an USRP2. MATLAB has a rich family of toolboxes that allows building software-defined and cognitive radio to explore various spectrum sensing, prediction and management techniques.

Keywords – MATLAB/Simulink; software defined radio; cognitive radio; universal software radio peripheral-2; frequency hopping spread spectrum; modulation; radio frequency (RF)

I. INTRODUCTION

Software Defined Radio is a flexible architecture that is applicable to many radio standards. Joseph Mitola coined the term software radio, to signal the shift from digital radio to multiband multimode software-defined radios [1][2]. In an introduction of reconfigurable logic and the coining of the term SDR, the dominant implementation architecture used for RF Front-Ends (FEs) was the super-heterodyne architecture [3]. The SDR is a radio communication system, which provides software control for a variety of modulation method, filtering, wideband or narrowband operations, spread spectrum techniques and waveform requirements etc. The frequency bands are still constrained at the RF Front-Ends. An SDR allows an implementation of signal processing functionality in software instead of dedicated hardware circuitry. So, the most obvious benefit is loading an appropriate program instead of having to build extra circuitry for different types of radio signals and communication protocols. The development of an SDR system implies to achieve two main goals [4][5];

- To move the border between the analog and digital world (in TX and RX Paths) as much as possible toward radio frequency (RF) by adopting analog-digital (A/D) and digital-analog (D/A) conversion as near as possible to the antenna.
- To replace the application specific integrated circuits (ASICS) dedicated hardware, with the re-configurable computing (FPGA) for baseband signal processing.

The FPGAs are mainly used in SDR RF Front-Ends (FEs) to improve the performance of DSP-chip-based systems. There is currently a wide range of FPGA products being offered by many semiconductor vendors; Xilinx, Altera, Atmel and AT&T etc. The architectural approaches used in these FPGAs are as diverse as their manufacturers [6]. The obvious benefits of wireless transmission have led to a number of radio systems.

Table I shows some of the IEEE 802 wireless family of standards. The family contains different types of access networks namely [7]; wireless personal area network (WPAN) IEEE-802.15.1 called Bluetooth, IEEE-802.15.4 is ZigBee, wireless local area network (WLAN) the IEEE-802.11 and wireless metropolitan area network (WMAN) the WiMax/IEEE-802.16. These standards apart from the channel bandwidth and the transmit power have different modulation techniques and transmission mechanisms, which can be implemented easily in a software; the RF Front-Ends may require for different frequency bands. The development of the IEEE 802.22 WRAN standard was aimed to use the cognitive radio techniques to allow sharing of geographically unused spectrum allocated to the television broadcast service [8]. An IEEE-1900.1 is a first standard that defines the concepts for dynamic spectrum access, terminologies relating to emerging wireless networks, system functionality and spectrum management [9].

The paper presents a detailed survey of the existing software radio platforms and an experimental prototype system that is built in MATLAB/Simulink and interfaced with USRP2. The MATLAB has a rich family of signal processing block, unless we want to do a something fancy; the mex (MATLAB executable) can be used to build an executable functions for C++ code for stand-alone MATLAB engine. The prototype system is low priced solution to realize the software-defined radios techniques.

TABLE I
IEEE 802 WIRELESS FAMILY STANDARDS

Specifications Standards	IEEE Standard (PHY Layer)	Frequency Band	Channel Bandwidth	Transmission Mechanism	Modulation Techniques	Transmit Power	Range (Distance)
Wireless - PAN	IEEE 802.15.1 (Bluetooth)	2.4 GHz	1 MHz	FHSS TDMA	GFSK	10 dBm	< 10 m
	IEEE 802.15.4 (ZigBee)	868/915 MHz, 2.4 GHz	5 MHz	DSSS CDMA/CA	ASK / BPSK / O-QPSK	3 dBm	< 10 m
Wireless - LAN	IEEE 802.11 (Wi-Fi)	2.4 GHz, 5 GHZ	20 MHz	DSSS/OFDM CSMA/CA	QAM / O-QPSK	16 dBm	< 150 m
Wireless - MAN	IEEE 802.16 (WiMax)	2.3 GHz, 2.5 GHz, 3.5 GHz	10 MHz	SOFDMA OFDA/CDMA	BPSK / QPSK / 16-QAM / 64-QAM	23 dBm	< 5 km
Wireless - WAN	IEEE 802.20 (BWAs)	below 3.5 GHz	5, 7, 20 MHz	OFDM OFDMA/CDMA	QPSK / 8-PSK / 16-QAM / 64-QAM	25 dBm	< 15 km
Wireless - RAN	IEEE 802.22 (WRANs)	54 - 862 MHz	6, 7, 8 MHz	OFDM OFDMA	BPSK / QPSK / 16-QAM / 64-QAM	24 dBm	< 100 km

The rest of the paper is organized as; Section II presents a detailed survey of current SDR solutions. This is followed by a prototype system design in Section III and the prototype system implementation is spread-over Section IV to VI. The cognitive radio is built on top-of-the prototype system in Section VII to wrap up everything in order to draw a sufficient conclusion and future directions in Section VIII.

978-0-7695-4376-5/11 $26.00 © 2011 IEEE
DOI 10.1109/UKSIM.2011.100

IEEE computer society

II. Background & Related Work

There are various hardware (HW) platforms and the software (SW) architectures that are used for defining the software radios. Many individuals, various research-groups and different companies have been working on software defined radio concept for years, pushing it to make a reality. This section presents a survey of the current SDR hardware platforms followed by the software architectures. The military solutions [10] [11] are not in the scope of the paper.

A. SDR Hardware Platforms

The hardware aspects of a SDR platform consist of the radio-frequency (RF) parts, communications links to the software-based signal processing elements (mostly a Host-PC). The rest may consist one or more of the following;

- ASICs (application-specific integrated circuits).
- FPGAs (field-programmable gate arrays).
- DSPs (digital signal processors).
- GPPs (general-purpose processors).

The ASICs are non-reprogrammable that contradicts the principle of SDR, but still used as a part for special characteristics. The FPGAs provide high computing power due to quasi-parallel processing nature while the DSP and GPPs are essentially serial in operation. The main strengths of DSPs and GPPs are their flexibility and easy configurability [12]. The various SDR hardware solutions (RF Front-End) are available in commercial and academic area, providing potential opportunities in the radio communication industry.

Universal Software Radio Peripheral 2 (USRP2): It is a brainchild of Matt Ettus (Ettus Research LLC) [13]. The USRP family of products has been nominated "Technology of the Year" award from the Wireless Innovation Forum, 2010. The USRP2 is a second generation of Universal Software Radio Peripheral, its platform consist Xilinx Spartan-III FPGA and general purpose AeMB processor [14]. The USRP-E100 is its next revision that's a stand-alone system powered by the combination of an ARM Cortex A8 processor & TI C64x+ DSP and a Xilinx Spartan 3A-DSP1800 FPGA.

Rice Wireless Open-Access Research Platform (WARP): The wireless open-access research platform of Rice University is a scalable and extensible programmable platform, built for prototyping advanced wireless networks [14]. The Xilinx Virtex-4 FX100 FPGA is used to enable programmability of both physical and network layer protocols on a single platform.

Berkeley Emulation Engine 3 (BEE3): BEE3 is new generation of Berkeley Emulation Engine-2 [16]. It is jointly developed by Microsoft Research, UC Berkeley and BEEcube Inc. The platform contains four Virtex-5FPGAs, emulates over 64 RISC processor cores concurrently at a 100 MHz rate. It is useful for most computationally intensive real-time applications, a high-speed multiple FPGA and validation solution. BEE3 is suited as a real-time, real-world prototyping and development platform.

Kansas University Agile Radio (KUAR): The KUAR hardware employs a Xilinx Virtex II Pro P30 FPGA along with 1.4 GHz Pentium M processor [17][18]. It has been promoted through the defense advanced research projects agency (DARPA) next generation (XG) program. The complete system was developed in Simulink, implemented in Xilinx VHDL, by generating the VHDL code from Simulink model(s) using a Modelsim of Mentor Graphics.

Small Form Factor Software Defined Radio (SSF-SDR): The Xilinx Inc. in collaboration with Lyrtech and Texas Instruments incorporated a SFF-SDR development platform for developing the handheld and mobile radios [19]. The Xilinx Virtex-IV FPGA, TI DSP TMX320 and TI MSP430 MCU are used in addition with ARM926 embedded processor [20].

Intelligent Transport System (ITS); National Institute of Information and Communications Technology (NICT) of Japan, developed a software-defined radio platform so-called NIST-ITS. It is specially designed for mobile communication, wireless LAN and digital terrestrial TV [21]. The platform contains Xilinx Virtex-4 FPGA, and two general purposes process (GPP) 430 MIPS (240 MHz).

Table II shows a detailed survey of existing SDR hardware platforms and their performance. The USRP2 is a cheaper and fast enough; the Xilinx Spartan-3 FPGA (XC3S2000) contains; 2M system-gates, 46080 equivalent logic cells. The larger FPGA and general purpose AeMB processor allows the USRP2 to be used as a standalone system without a host computer in many cases; the USRP2 is used in the prototype.

Table II
A Survey Of SDR Hardware Platforms [13-21]

	USRP2	KUAR	WRAP v2.2	SSF-SDR	BEE3	NICT-ITS

Architecture (GPP, DSP, FPGA)	FPGA (Xilinx Spartan III)
	GPP (AeMB Processor)
		FPGA (Xilinx Virtex II)
		GPP (Pentium M)
			FPGA(Xilinx Virtex-4)
			GPP(PowerPC)
				FPGA (Xilinx Virtex-4)
				GPP (ARM926)
				DSP (TMX320xx)
					4 FPGAs
					(Xilinx Virtex-5)
					Quad-Core OpenSPARC
						FPGA(Xilinx Virtex-4)
						GPP(ARM11)

RF Bandwidth	100 MHz	30 MHz	30 MHz	22 MHz	100 MHz	5 MHz
RF Range	2.4 GHz & 5 GHz (multi M/G-Hz)	2.4 GHz & 5 GHz ISM/UNII	2.4 GHz & 5 GHz ISM/UNII	0.2–1.0, 1.6–2.2 GHz 2.5/3.5 GHz (WiMAX)	Ultra-wide-band (multi-GHz)	2 GHz, 5 GHz and VUHF
RF Channels	2	16	4	2	16	5
Connectivity	Gig Ethernet (National Semiconductor)	USB 2.0, Gig Ethernet	USB 2.0 / Gig Ethernet (Marvel)	RS232/USB 2.0, Ethernet	RS232/USB, Gig Ethernet(Broadcom)	RS232C, USB, Ethernet
ADCs	14-bit, 100 MS/s (LTC2284)	14-bit, 105 MS/s (LTC2284)	14-bit, 65MS/s (AD9248)	14-bit, 125MS/s (TI ADS5500)	8-bit 3GS/s (BEE3-ADC-D3G)	12-bit, 170MS/s (ADCs N/K)
DACs	16-bit, 400 MS/s (AD9777)	16 bit, 100 MS/s (AD9777)	16-bit, 160MS/s (AD9777)	16-bit, 500MS/s (TI DAC5687)	12-bit, 2GS/s (BEE3-DAC-D2G)	12-bit, 500MS/s (DACs N/K)
Power	6 volt	12 volt	12 volt	12 volt	12 volt	3.3 volt
Price	$ 1,400.00	$4,000.00	$6,500.00	$9,900.00	$20,000.00	N/A

B. SDR Software Platforms

GNU Radio: is an open source software development toolkit that provides the signal processing runtime and processing blocks to implement software radios [14]. The radio applications are written in Python, while the performance-critical signal processing components, implemented in C++ using processor floating point extensions where available. SWIG glues C++ classes into Python. In GNU Radio Python (*gr.flow_graph*) library of signal processing blocks is used to tie together the signal processing blocks of the waveforms. GNU Radio Companion (GRC) is a graphical tool for creating signal flow graphs and generating flow-graph source code.

Open-Source SCA Implementation - Embedded (OSSIE): is a Virginia Tech's open source, the core framework is based on the JTRS software communications architecture (SCA); the CORBA based communication model for SDR [22]. The OSSIE is an object-oriented SCA operating environment, where signal processing components are written in C++. The operating environment, often referred to as the core framework, implements the management, configuration, and control of the radio system. Every OSSIE's component is considered having two parts: one part realizing the signal processing and another managing the SCA infrastructure. The OSSIE waveforms are described in an XML that is used to describe component properties and interconnections between components in a waveform.

Wireless Open-Access Research Platform for Network (WARPnet): is an SDR framework that is built around client-server architecture in Python [23]. The WARPnet uses PCAP (packet capture) API to communicate with the WARP board directly. To allow the Python-based client/server to access PCAP, the Pcapy module is required. With WARPLab, one can interact with WARP nodes directly from the MATLAB workspace and signals generated in MATLAB can be transmitted in real-time over-the-air using WARP nodes.

Cognitive Radio Open Source System (CROSS): is an open source cognitive radio architecture [24]. It consists of five core components categories (modules); cognitive radio shell (CRS), cognitive engine (CE), policy engine(PE), service management layer (SML), and software-defined radio host platform. The CROSS is a modular cognitive radio system framework that uses socket connections for inter-component communication. The cognitive radio shell library and API are implemented in C++, the other modules can be implemented in any language that supports a TCP/IP socket interface.

The High Power SDR (HPSDR) and FlexRadio Systems are not considered in the paper due to space constraints.

III. SDR Prototype System Design

In our previous work [25] the prototype system was built in GNU radio framework and interfaced with a universal software radio peripheral (USRP) and RFX2400 RF Front-End from Ettus Research LLC. The current prototype system as shown in Fig. 1, is built in MATLAB/Simulink R2010b and interfaced with a universal software radio peripheral-2 (USRP2) and RFX2400 RF Front-End from Ettus Research LLC for better performance, reusability and further extension for cognitive radio [26]. MATLAB/Simulink has a variety of signal processing blocks and it can generate various type of code: C/C++ for MCU, DSP; VHDL/Verilog for FPGA/ASIC; and SPICE for Analog Devices. The Simulink HDL Coder generates HDL code (VHDL/Verilog) for FPGAs from Simulink models, State-flow charts, and MATLAB code for various device vendors. The basic design philosophy behind the prototype is, to do all of the waveform-specific processing; like encoding, modulation, cognitive decisions on the host CPU (in Simulink) and all of the high-speed operations like digital up and down conversion, decimation and the interpolation inside the FPGA of USRP2. The RFX2400 is used for frequency constrain to operate in 2.4 GHz band.

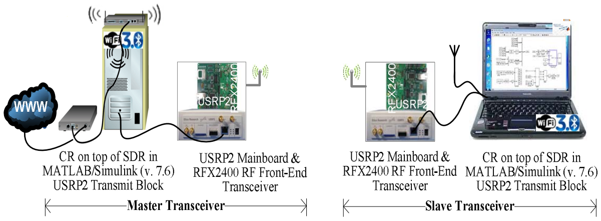
WWW
WiFi 3.0
USRP2
RFX2400
CR on top of SDR in MATLAB/Simulink (v. 7.6) USRP2 Transmit Block
USRP2 Mainboard & RFX2400 RF Front-End Transceiver
Master Transceiver
USRP2 Mainboard & RFX2400 RF Front-End Transceiver
CR on top of SDR in MATLAB/Simulink (v. 7.6) USRP2 Transmit Block
Slave Transceiver

Fig. 1. Testbed prototype system

In USRP2, the main ADC is a Dual 14-bit LTC2284, 100 MS/s, 2 auxiliary channels, 12-bit ADC (AD7922) for each daughter-board connector. The main DAC is a Dual 16-bit AD9777, 400 MS/s, 2 auxiliary channels, 12-bit DAC (D5623). The interpolation [26] happens from 100 MS/s to 400 MS/s inside the DAC chip itself, but he FPGA talks to the DAC at 100 MS/s just like it talks to the ADC at 100 MS/s. The TX rate may be different from the RX rate. The USRP2 has a faster FPGA, faster ADCs and DACs with a higher dynamic range. It only supports Gigabit Ethernet connection; does not work with a 10/100 Mbps interface. It is much faster than the USB interface that was used in our previous work. A Gigabit Ethernet interface allows applications to simultaneously send 50 MHz of RF bandwidth in and out of the USRP2. The Gigabit Ethernet uses the pause frames for flow control; its interface is managed by an open source MAC implemented in FPGA along with National Semiconductor PHY (DP83865). The USRP2 is plugged directly into a dedicated ethernet interface; multiple USRP2s cannot be used with a switch or a hub. USRP2 communicates at the IP/UDP layer over the gigabit ethernet. Communication over UDP is easy over cross platform, unlike a raw ethernet interface. The default IP address of the USRP2 is 192.168.10.2, so we should configure the host's ethernet interface with a static IP address 192.168.10.1 and a subnet mask of 255.255.255.0 to enable communication. If the play-plug feature does not work properly, the xpcexplr - open xPC target explorer or the xpctargetspy - open real-rime xPC targets spy can be used to configure the IPs in Simulink.

The system development contains three sections for simplicity and reusability. The system uses a frequency-hopping spread spectrum, using a pseudorandom sequence known to both transmitter and receiver. The transmitter and the receiver are properly synchronized so that they can hop together from channel to channel. The hop sequence is derived from the device address of the master; the phase in the hopping sequence is determined by its clock [25]. The prototype system hops over 79 frequencies, each 1MHz wide, the bandwidth of the spectrum over which the hopping occurs is called the total hopping bandwidth that ranges 2.402-2.480 GHz [26].

IV. Packet Formulation Model

The packet consists of three entities: the access code, the header, and the payload as shown Fig. 2. These three entities are processed separately and concatenated together to build a raw packet. The access code is 72 bits and used to detect the presence of a packet. The header is 54 bits that contains information associated with the packet and the link control. The payload ranges from zero to a maximum of 2745 bits.

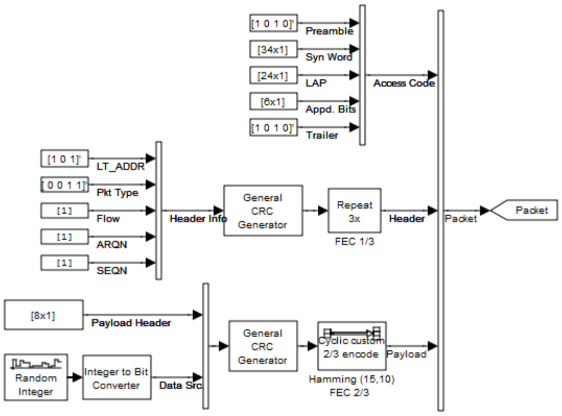

Fig. 2. Packet/payload formulation model

In packet access code, the preamble is a fixed zero-one pattern of 4 symbols used to detect edges of received data. It should be 1010 or 0101, the synchronization (SYNC) word is used for time synchronization, while the trailer is combination of 6-append-bits and 4-trailer-bits should be 1100101010 or 0011010101. In packet heard; the LT_ADDR is 3-bit active member address that is used to distinguish active members (slaves) in a piconet, the 4-bit packet type filed is used to distinguish between 16 different types of packets. The 1-bit flow used to control the flow over logical link, the flag is asserted when device is unable to receive any more data due to full receiver buffer. The 1-bit acknowledgement indication of success of transfer of the packet, the SEQN bit provides a sequential numbering scheme to order the data packet stream. The header error check (HEC) is an 8-bit word, used to check the header integrity [26]. In payload encoding block the 136 bits are input from the source is overlapped by 8 bit payload header which is protected by CRC -16-CCITT, the resulting 160 bits sent to the encoding block where the 2/3 forward error correction (FEC) is applied to produce a total of 240 encoded bits. The 10-bit header info is protected by the 8 bit CRC which is then encoded by 1/3 FEC results in total of 54 bits. The header information is then concatenated with the 72 bits access code and the 240 payload encoded bits resulting in completion of 366 bits packet. The 72 bits access code is generated by lower address part of the master [26].

V. Frequency Hopping Spread Spectrum Model

Spread spectrums techniques are signal structuring techniques that employ a transmission bandwidth which is several orders of magnitude greater than the minimum required signal bandwidth. These techniques are used for multiple access and/or multiple functions [28].

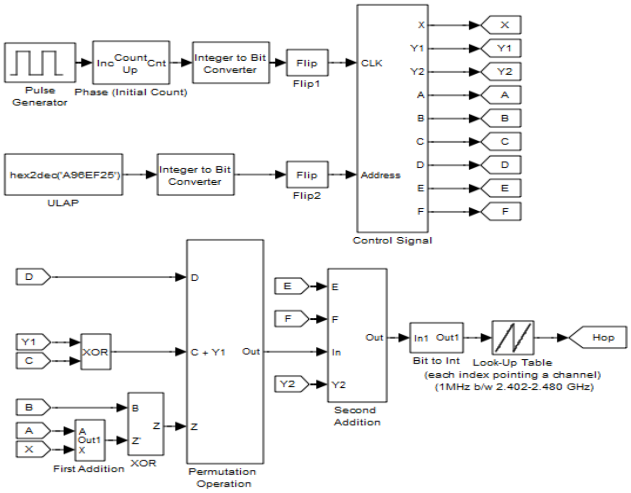

Fig. 3. Frequency hopping spread spectrum model

The frequency hopping involves a periodic change of transmission frequency, deterministically at a rate of 1600Hz; the hopping is derived from the master device address and its clock. The set of possible carrier frequencies is called the hopset that are 79, while the bandwidth of a channel used in the hoset is called the instantaneous bandwidth that is 1 MHz.

According to the section 15.247 (a) of the FCC regulations for FHSS devices requires devices to hop over at least 75 channels and limit the maximum bandwidth of each hopping channel to 1MHz. The hop selection kernel as shown in Fig. 3, addresses a register containing the RF channel indices those refer to 79 RF channels ranging 2.402GHz - 2.480GHz. The inputs A to D determines the ordering within the segment, the inputs E and F determine the mapping onto the hop frequencies. The X input determines the phase in the 32-hop segment, whereas Y_1 and Y_2 select the time slots between master-to-slave and slave-to-master for time division duplex [29].

VI. Modulation & Transmission Model

Modulation is a process by which some characteristics of a carrier wave, is varied in accordance with an information-bearing signal. The information will be less vulnerable to noise or interference and it also permits the use of multi-access techniques. The basic aim of digital modulation is to transfer a digital bit stream over an analog Bandpass channel. The system uses a Gaussian frequency shift keying (GFSK) modulation, before the baseband pulses (-1, 1) go into the FSK modulator, it is passed through a Gaussian filter to make the pulse smoother so to limit its spectral width. The frequency shift keying involves binary signaling by using the two frequencies separated by Δf Hz, where Δf is a frequency deviation that is smaller as compared to the carrier frequency f_c. The modulation index μ is defined as [30];

$$\mu = \Delta f T \tag{1}$$

Where T is a symbol time (that is the inverse of the data rate for binary schemes), the frequency deviation (Δf) is the maximum frequency shift with respect to the carrier frequency, if a ’0’ or ’1’ is being transmitted, and it can vary between 140 KHz and 175 KHz. In GFSK modulation, the symbol '0' shall be modulated on -Δf and symbol '1' shall be modulated on +Δf.

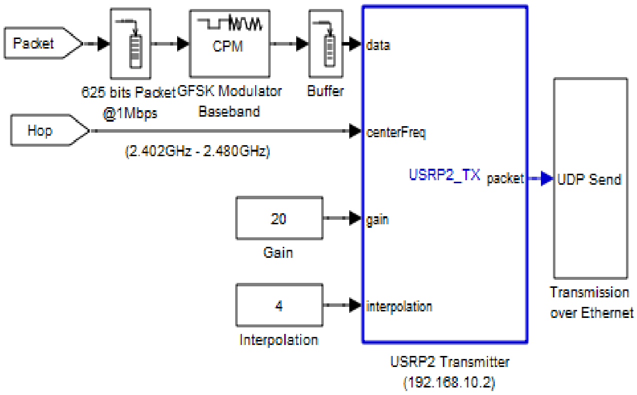

Fig. 4. Packet modulation & transmission over USRP2 model

According to FCC rule the system follows the modulation characteristics; the modulation index μ is between 0.28 and 0.35, with a bandwidth-bit period product BT=0.5, for a symbol rate of 1 Ms/s, which corresponding to a data rate of 1 Mb/s, while the minimum frequency deviation is greater than 115 kHz [28].

The raw packet is modulated first before it is transmitted at a specific frequency. Fig. 4 shows a modulation of a packet and it transmission over an USRP2. In the USRP2 we can control center frequency, transmit power, and interpolation/decimation factor. The range for the center frequency and the power are dependent on the attached daughterboard (RFX2400). The interpolation factor in the transmit path and decimation factor in the receive path must be properly configured. Interpolation (also known as up sampling) increases the sampling rate and requires that we somehow produce values between the samples of the signal. The interpolation by a factor *q* is accomplished by inserting *q-1* zeros in between each sample of x[n] and then discrete time low-pass filtering [31]. The sampling process is critical for radio receivers that digitize signals. Sampling an analog signal at IF or RF results in replicas of the signal's spectrum repeated at uniform intervals [5]. The choice of the sampling rate of such signals is dependent on the signal's bandwidth and the IF or RF center frequency [32]. The sampling frequency (Nyquist frequency) or sampling rate f_s is defined as the number of samples obtained in one second; $f_s = 1/T$. The time interval between successive samples is referred to as the sampling interval ($T = 1/f_s$), the Nyquist sampling rate or Nyquist rate can be calculated by the following equation 2.

$$Sample\ Rate = \frac{ADC\ Rate}{Decimation\ Rate} \tag{2}$$

The Nyquist rate can be used to calculate the maximum RF bandwidth as, the uniform sampling rate f_s (in samples per unit time) should be at least twice the maximum bandwidth.

$$B = \frac{f_s}{2} \tag{3}$$

In USRP2 interpolation and decimation is an integer value between 1 and 512. The interpolation and decimation rates, in combination with the sampling rate of the DAC/ADC, determine the data rate in Simulink. The sampling rate of the USRP2 device is 100 MHz, so an interpolation rate of 512 results in a Simulink sample time of 512/100e6.

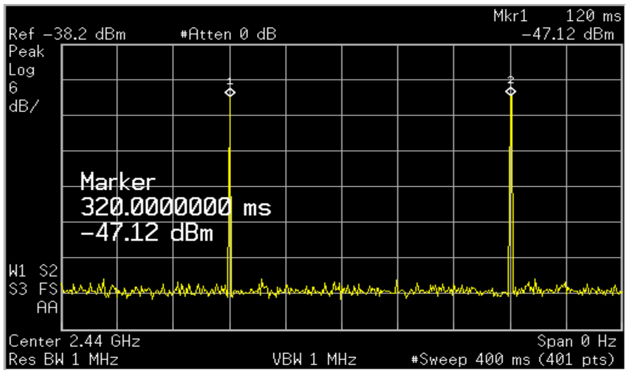

Fig. 5. Transmission spectrum view

Fig. 5, shows a transmission spectrum view, where two packets are captured successful one at 120 msec. and second at 320 msec. during a single sweep. The time difference between two captures is 200 ms, shows a channel reuse twice (hopping). The frequency band 2.4 GHz, central frequency 2.44 GHz, channel width is 1 MHz, the signal strength –47 dBm.

The processing elements of an SDR RF-front-end have their associated strong and weak points. The system performance is mainly constrained with the re-configurable computing power (FPGA); high sample rate processing, like digital up- and down conversion, higher-speed and higher precision ADCs and DACs allow for wider band signals while at the same time increasing dynamic range, and high-speed interface between the front-end and the host PC [33]. The USRP2 Gigabit Ethernet interface in MATLAB/Simulink communicates at UDP layer. The raw Ethernet interface for USRP2 in GNURadio is only a Linux specific, the MATLAB provides a platform independent solution; there is a break-down of 52 bytes overheads on-top of the ethernet header (IP+UDP+VRT = 52 bytes).

There are still many open issues; a flexibility of an SDR as the possibility to load and install new software on an SDR unit (i.e. mobile phones) and also the security issues. The major challenge with SDR is how to achieve sufficient computational capacity [34], in particular for small handheld units where a number of operations may be carried out in parallel and the deterministic behavior allows low-level optimization of implementations. The IEEE standards committee on next generation radio and spectrum management and the ETSI reconfigurable radio systems technical committee have made the principles for the software defined radios and especially for cognitive radios [35][36].

VII. Cognition for Interference Estimation

The cognitive radio (CR) is an emerging new technology, which is currently far from mature in terms of real applications. The CR is considered an intelligent wireless communication system that should be aware of its surrounding environment. It mainly consists of three modules as shown in Fig. 6; spectrum sensing, spectrum predicting, and spectrum management [38]. A survey of spectrum sensing algorithms for cognitive radio applications is presented in [39].

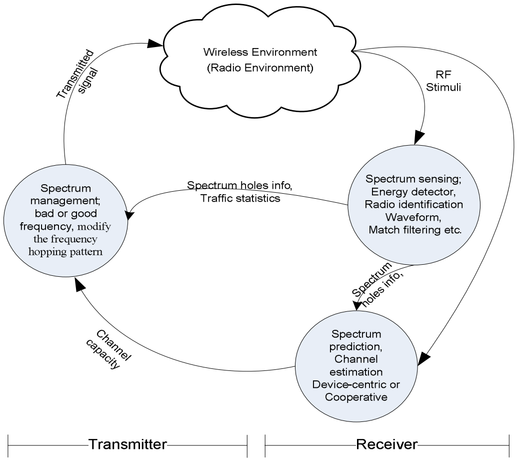
Wireless Environment
(Radio Environment)
Transmitted signal
RF Stimuli
Spectrum management; bad or good frequency, modify the frequency hopping pattern
Spectrum holes info, Traffic statistics
Spectrum sensing; Energy detector, Radio identification Waveform, Match filtering etc.
Spectrum holes info,
Channel capacity
Spectrum prediction, Channel estimation Device-centric or Cooperative
Transmitter
Receiver

Fig. 6. Basic cognitive cycle model

For re-configurability a cognitive radio looks naturally to software- defined radio to perform its task. The prototype system has some intelligence to allow a radio terminal to automatically sense, recognize, and make wise use of available radio frequency spectrum at a given time within the frequency band. The method for interference estimation consists of measuring the bit error rate and percentage of packets loss per-channel (frequency).The prototype system uses an interference mitigation technique with the ability to detect the presence of other systems operating in the same band (ISM). The master device controls all packet transmission. The measurements collected by a slave device are sent to the master to avoid data transmission on a "bad" channel. The statistics are based on the packet lost measurements per-channel. The channel is listed as bad, when the packet lost is below the threshold. The master device modifies the frequency hopping pattern that is called an adaptive frequency hopping [40]. It is used for determining whether there are other devices present in the ISM band or not and to avoid them.

VIII. Conclusion And Future Work

In this paper firstly we presented a survey of SDR HW architectures and SDR SW platforms and secondly developed a prototype system (testbed) for experimenting and emulating the software defined radios. The motivation for a testbed is provided by the need to validate various coding schemes, modulation techniques, and spread spectrum algorithms to measure the quality of service. The platform independent SDR prototype system is developed in MATLAB/Simulink and interfaced with an USRP2 main-board and RFX2400 daughter-board from Ettus Research LLC. The prototype system opens the doors to experiment different spread spectrum techniques, various coding and modulation schemes for software-defined radios to realize the cognitive radios. Future work would be practical implications of various statistical and artificial intelligence techniques for cognitive radios; metaheuristic algorithms, auto-correlation/regression, hidden Markova model, genetic algorithms, and neural networks etc.

References

[1]. J. Mitola, "The software radio architecture," *IEEE Communications Magazine*, pp. 26–38, May 1995.

[2]. J. Mitola III, "Software radio architecture: a mathematical perspective," *IEEE Journal on Selected Areas in Communication.* vol. 17, no. 4, April 1999.

[3]. Enrico Buracchini, "The software radio concept," *IEEE Communications Magazine,* vol. 38, no. 9, Sept. 2000.

[4]. R.H. Walden, **"**Analog-*to*-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, April 1999.

[5]. Agilent Application Note, "Software Defined Radio Measurement Solutions," *Agilent Technologies, Inc.* July 13, 2007.

[6]. C. Dick, "FPGAs: The high-end alternative for DSP applications," *DSP Engineering,* Spring 2000.

[7]. WILLIG et al., "Wireless technologies in industrial network," *IEEE Special Issue on Industrial Communication Systems*, vol. 93, June 2005.

[8]. IEEE-802.22 - Standard for Wireless Regional Area Networks (WRAN) - Specific requirements - Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and procedures for operation in the TV Band.

[9]. IEEE-1900.1-2008 - Standard Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management, Sept 2008.

[10]. R&S M3SR Series 4100 Software Defined Radios: HF Radio Family for Stationary and Ship-borne Communications, R&S GmbH, Germany.

[11]. Digital Modular Radio (DMR): *Software-Defined Communication System, General Dynamic - C4 Systems,* D-DMR-1-0409, 2010, Arizona, USA.

[12]. T. J. Rouphael, *RF and Digital Signal Processing for Software-Defined Radio – A Multi-Standard Multi-Mode Approach,* Elsevier Inc, 2009.

[13]. M. Ettus, Universal Software Radio Peripheral (USRP) Rev. 2, *Matt Ettus. Ettus Research LLC,* Internet http://www.ettus.com/, [Dec. 2010].

[14]. C. Clark, *Software Defined Radio: with GNU Radio and USRP,* McGraw-Hill Professional, Nov. 2008.

[15]. P. Murphy, A. Sabharwal, and B. Aazhang, "Design of WARP: Wireless open-access research platform," in *Proc. of European Signal Processing Conference*, June 2006.

[16]. C. Chang, J. Wawrzynek, and R. W. Brodersen, "Bee2: a high-end reconfigurable computing system," in *Proc. of IEEE Magazine, Design and Test of Computers,* vol. 22, no. 2. April 2005.

[17]. G. J. Minden, et al, "KUAR: A flexible software defined radio development platform," in *Proc. of IEEE Dynamic Spectrum Access Networks Symposium*, April 2007.

[18]. G. J. Minden, et al, "Cognitive radios for dynamic spectrum access - an agile radio for wireless innovation," *IEEE Communications Magazine*, vol. 45, no. 5, Dublin, Ireland, May 2007.

[19]. R. Sathappan, M. Dumas, and M. Uhm, "A new architecture for development platforms targeted to portable radio applications," *Lyrtech Technical Paper,* 2007.

[20]. L. Belanger, "Advanced SDR platform eases multiprotocol radio development," RF Design Magazine, January 2007.

[21]. H. Harada, "Software defined radio prototype toward cognitive radio communication systems," *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN-05),* Maryland, Nov. 2005.

[22]. *C. R. A. González et al.,*"Open-source SCA-based core framework and rapid development tools enable software-defined radio education and research," *IEEE Communication. Magazine.*, vol. 47, no. 10, pp. 48– 55, Oct. 2009.

[23]. C. Hunter, J. Camp, P. Murphy, A. Sabharwal, and C. Dick, "A Flexible Framework for Wireless Medium Access Protocols," Invited Paper in *Proc. of IEEE Signals, Systems and Computers Conference*, Asilomer, Pacific Grove, CA, Nov. 2006.

[24]. T.R Newman, S.M.S. Hasan, D. Depoy, T. Bose, and J.H. Reed "Designing and deploying a building-wide cognitive radio network testbed," *IEEE Communications Magazine*, vol. 48, no 9, Sept. 2010.

[25]. A.A. Tabassam and S. Heiss, "Bluetooth clock recovery and hop sequence synchronization using software defined radio," in *Proc. of IEEE Region 5 Technical, Professional and Student Conference (5PBASICS2)*, Kansas KS, USA, April 2008.

[26]. A.A. Tabassam, *Bluetooth Security in Industrial Networks; concepts, components and applications of software defined radio,* VDM Verlag Dr. Müller, 2010.

[27]. W.Y. Yang et al, *MATLAB/Simulink for Digital Communication*, A-JIN Publishing, Korea, 2009.

[28]. T. S. Rappaport, *Wireless Communications: principles and practices*, 2nd edition, Pearson Educations Inc. 2003.

[29]. Bluetooth Special Interest Group, "*Specification of the Bluetooth System,*" ver. 4.0, Dec. 2009.

[30]. J. D. Oetting, "A comparison of modulations techniques for digital radios," *IEEE Transactions on Communications.* vol. 27, no.12, Dec. 1979.

[31]. R. E. Crochiere and L. R. Rabiner, "Interpolation and decimation of digital signals - A tutorial review," in *Proc. of the IEEE,* vol. 69, no.3, March 1981.

[32]. P. B. Kenington, *RF and Baseband Techniques for Software Defined Radio*, Artech house, Boston, London, 2005.

[33]. B. Jerker and V. Seppo, "Convergence of hardware and software in platforms for radio technologies," *IEEE Communications Magazine,* vol. 44, no. 11, 2006.

[34]. T. Ulversøy, "Software defined radios: challenges and opportunities," *IEEE Communications Surveys & Tutorials,* vol.12, no. 4, April 2010.

[35]. R.V. Prasad, P. Pawelczak, J.A. Hoffmeyer, and H.S. Berger, "Cognitive functionality in next generation wireless networks standardization efforts," *IEEE Communications Magazine,* vol. 46, no. 4, April 2008.

[36]. M. Mueck et al., ETSI reconfigurable radio systems: status and future directions on software defined radio and cognitive radio standards*, IEEE Communications Magazine*, vol.48, no 9, Sept. 2010.

[37]. M. Laddomada, F. Daneshgaran, M. Mondin, and R.M. Hickling, "A PC-based software receiver using a novel front-end technology," *IEEE Communications Magazine,* vol.39, no.8, Aug. 2001.

[38]. S. Haykin, Cognitive radio: brain-empowered wireless communication, *IEEE Journal on Selected Area in Communications,* vol. 23, no. 2, 2005.

[39]. T. Yücek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communication Surveys & Tutorials*, vol. 11, no. 1, first quarter 2009.

[40]. N. Golmie and O. Rebala, "Bluetooth adaptive techniques to mitigate interference," In proc. of *IEEE Global Telecommunications Conference (GlobeCOM03),* San Francisco, USA. Dec. 2003.