社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SR2009-57(2009-10)

Development of Spectrum Sensing System with GNU Radio and USRP to Detect Emergency Radios

Md.Abdur RAHMAN†, Santosh KHADKA†, Iswandi†, Mutsawashe GAHADZA†, Azril HANIZ†, Minseok KIM†, and Jun-ichi TAKADA†

† Graduate School of Engineering, Tokyo Institute of Technology
2-12-1 O-okayama, Meguro-ku, Tokyo, 152-8550 Japan
E-mail: †{abdur,santosh,iswandi,mutsa,azril,mskim,takada}@ap.ide.titech.ac.jp

Abstract Scarcity of wireless channels leads the researchers to develop opportunistic spectrum usage systems. Spectrum sensing is the most important and complex job for the cognitive radios. Spectrum sensing may play a vital role to detect the presence of unknown emitters in a noisy and multi-path environment. In emergency situations (i.e. after earthquake) a number of rescue teams come to the field and setup their own radio communication systems. If the teams have no collaboration among themselves (which is most likely), they may interfere with each other by using same RF channels. One of the possible way to detect the presence of emitters is spectrum sensing. To utilize the advantages of Software Defined Radio (SDR) GNU Radio and USRP are used to capture the signal samples and store to a data file. Matlab communications toolbox is used to analyze the data and examine the spectrum. A number of theoretical analyses are available in literatures to detect the presence of signal. To investigate some of them from experimental perspective, we developed a spectrum sensing system with GNU Radio and USRP. GNU Radio based spectrum analyzers can be a cost effective solution for blind channel estimations also.

Key words Spectrum Sensing, GNU Radio, USRP, Energy Detector, Emergency Radios

1. Introduction

Spectrum sensing is one of the challenging and exciting techniques of modern wireless systems to utilize the vacant spaces. It is the most important and complex part of the cognitive radios. Spectrum sensing may also be used to detect the presence of unknown radio emitters in an area. Human civilization has always been devastated by natural disasters like earthquakes, tsunamis, hurricanes etc. Most obvious outcomes of such natural disasters are loss of human lives and collapse of infrastructure. Although these natural hazards can not be avoided, valuable human lives can be saved by quick and efficient rescue efforts. Usually telecommunication infrastructure falls apart during most of the disasters. Telecommunication has become the part and parcel of almost all of the national and international rescue teams. Wireless technologies are more preferable for post disaster rescue operations.

After a big disaster, generally emergency rescue teams from all over the world come to the scene and try to participate in the rescue activities. Usually each agency sets their own wireless networks to coordinate among the mobile teams. As the number of wireless channels for emergency communications are limited, it is quite possible that two teams are using the same wireless standards with same physical specifications. The result is severe interference among the networks that hinders the rescue efforts. A well informed wireless environment may help the systems to avoid interferences and perform rescue operations smoothly.

Geolocation based spectrum usage information may help the rescue teams to avoid the unwanted interferences. A well planned spectrum sensing network may scan the geolocation based channel information and form a central database to inform the rescue teams beforehand. The database may contain some PHY/MAC layer parameters as well as the geolocation of each of the radio emitters active on the scene with different standards. Sensing the spectrum and extracting the PHY/MAC parameters as well as geolocation estimation of an active radio emitter is a very big challenge. Spectrum sensors may not have any pre-knowledge about the radio emitters and the radio channels. Blind channel estimation techniques are very useful to identify them.

The concept of software defined radio (SDR) has been generating a lot of interest from various quarters, from the pure hobbyist through to academics and business minded people. The benefits to SDR include reconfigurability and possibility of rapid prototyping among others. The GNU Radio project has emerged as one of the most exciting SDR development project. The GNU Radio technology provides an opensource software platform which together with low cost hardware called USRP (Universal Software Radio Peripheral) can be used to develop and implement various software radio applications. GNU Radio and USRP can be used to design and implement a wireless spectrum sensor to develop and maintain the geolocation based channel database. Detail of the architecture can be found in [1].

The key challenges of spectrum sensing (to detect the presence of unknown emitters in emergency situations) are little sensing time and mobility of the emitters. Narrowband radio emitters with different standards will communicate side by side in the spectrum. Sensing all the emitters at the same time is another big challenge. A number of techniques have already been developed to detect the primary users in Cognitive Radio (CR) environments. But most of the CRs have some prior knowledge of the primary users location and PHY parameters. In case of the detection of emergency radios the sensors have no prior knowledge of the location and PHY parameters of radio emitters. Energy detection can be a simple and effective solution in this case. In this study GNU Radio and USRP are used to develop energy detector based spectrum sensing system.

Remaining part of this study includes some introductory discussions about the related technologies. Framework of the spectrum sensing system has been presented in the next section followed by some preliminary results and analyses. Subsequently conclusions and future works finishes the study.

2. Technical Overview

To develop a spectrum sensing system for emergency communications energy detection, cyclostationarity analysis and matched filter techniques are widely used. However, energy detector can be suitable for blind channel detection for assumed situation.

2.1 Energy Detection

In the energy detection approach, the energy of a received signal is compared to a predetermined threshold to determine whether the channel is active or not. This is called energy detection based on the χ^2 hypothesis test. One of the advantages of energy detection is that it has low complexity and can be implemented simply. It is also suitable when the CR user does not have any prior information about the received signal [2].

In the time domain, the energy detection is based on the following hypothesis

$$\begin{cases} \mathcal{H}_0 : r(n) = v(n) \\ \mathcal{H}_1 : r(n) = h(n) \cdot s(n) + v(n) \end{cases}, \quad (1)$$

where $r(n)$ denotes the received signal, $s(m)$ and $v(n)$ denote a transmitted signal and the band limited additive white Gaussian noise (AWGN). The linear time varying characteristics $h(n)$ of the propagation channel is herein assumed to be constant without loss of generality [3].

The test statistics are given by

$$\mathcal{T} = \sum_N |r(n)|^2 \quad (2)$$

and the detection rule with threshold γ is as

$$\begin{cases} \mathcal{T} \geqq \gamma & \Rightarrow \quad \text{signal present} \\ \mathcal{T} < \gamma & \Rightarrow \quad \text{signal absent} \end{cases}. \quad (3)$$

γ is assumed as Gaussian random process with variance ${\sigma_\mathrm{r}}^2$ and zero mean under the $\mathcal{H}_0$ hypothesis. When the signal is absent, the test statistic follows the central χ^2 distribution with N degrees of freedom [2]. According to the central limit theorem, as the number of samples N increases, the test statistics $\mathcal{T}$ can be approximated by the normal distribution

$$\mathcal{T} \approx \mathcal{N}\left(N\sigma_r^2, 2N\sigma_r^4\right) \text{ under } \mathcal{H}_0 \quad (4)$$

The probability of false alarm P_{FA} can be calculated by :

$$P_{\mathrm{FA}} = Q\left(\frac{\gamma - N\sigma_r^2}{\sqrt{2N}\sigma_r^2}\right), \quad (5)$$

where the detection threshold γ is usually obtained by predetermined P_{FA} as

$$\gamma = \left[Q^{-1}(P_{\mathrm{FA}})\sqrt{2N} + N\right]\sigma_n^2 \quad (6)$$

The probability of detection P_{D} can be calculated in the same way as P_{FA}. When the signal is present ($\mathcal{H}_1$), the only difference from $\mathcal{H}_0$ is the variance of the total input signal if the signal can also be treated as independent random process to the noise with zero mean. The signal variance can be written as $\sigma_r^2 + \sigma_s^2$. From this fact, the probability of detection can be calculated by [4].

$$\begin{aligned} P_{\mathrm{D}} &= Q\left(\frac{\gamma - N(\sigma_n^2 + {\sigma_s}^2)}{\sqrt{2N}(\sigma_n^2 + \sigma_s^2)}\right) \\ &= Q\left(\frac{Q^{-1}(P_{\mathrm{FA}})\sqrt{2N}\sigma_n^2 - N\sigma_s^2}{\sqrt{2N}(\sigma_n^2 + \sigma_s^2)}\right) \end{aligned}. \quad (7)$$

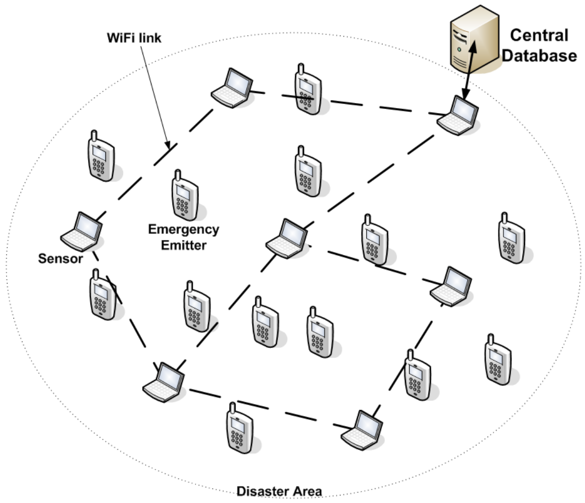


Figure 1 Spectrum sensing framework

Table 1 Daughterboard (RFX900) specifications

Parameter	Value
Type	Tranceiver
Frequency range	750 MHz. ~ 1050 MHz.
TX/RX bandwidth	30 MHz.
AGC range	-30 ~ 39.5 dB(69.5dB)
TX power	200mW (23dBm)
Control	Software or FPGA
Mode	Full-duplex

3. Framework

Assumptions for the framework designs are as follows [1]:

- Earthquake in a major city (10.0 Richter Scale)
- Telecommunication Infrastructure collapsed
- Rescue teams from all over the world are coming and setting their own networks
- Each emergency rescue team will be responsible to maintain their own networks.
- Sensors will not communicate with any of the emitters in operation.
- Sensor network will try to find the frequency and bandwidth used by the emitters.

Framework of the proposed system is shown in Figure 1. Sensors will be placed randomly in the disaster area. Each sensor will detect the presence of emitters by sensing the spectrum within its range. An emitter can be detected by multiple sensors. Sensors will transfer the sensed information to a central server to form a cooperative database.

4. Developed System

We used the RFX900 daughterboard to receive the transmitted signals. Table 1 shows the brief description of the daughterboard. Frequency tuning is a two step process. The Phase Locked Loop (PLL) tunes to the closes value of the

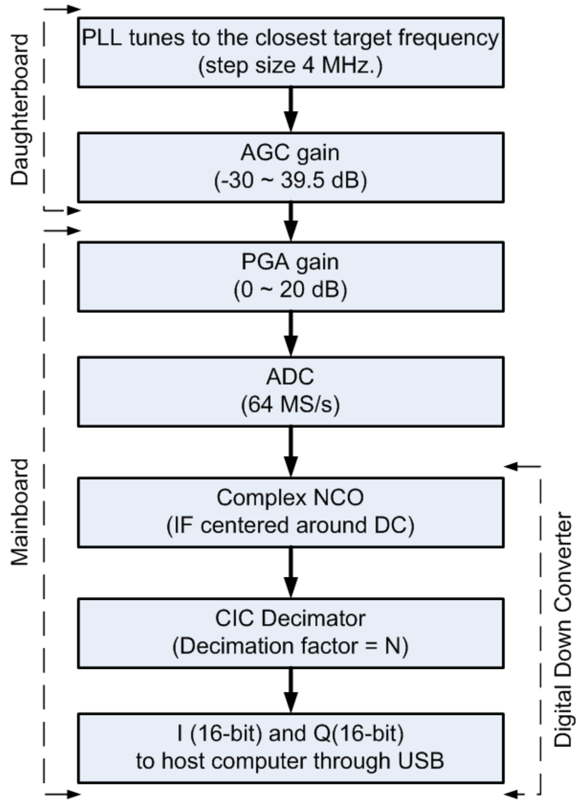


Figure 2 Signal flowchart

target frequency. Afterwards the fine tuning is done by NCO inside the DDC. A complete chart of the signal flow in the receiver is given in Figure 2. Signal is received by the RF front ends (daughterboards) and transferred to the FPGA through a high-speed 12-bit Analog to Digital Converter (ADC). The specifications of the ADC are given in Table 2. Maximum frequency allowed for the sampled signal is 150 MHz but the higher the frequency of the sampled signal, the more the SNR will be degraded by jitter. 100 MHz is the recommended upper limit.

The ADC transfers the signals to the FPGA after the sampling operations. The FPGA performs high bandwidth math to reduce the data rates to something transferable over USB2.0 (32 MByte/s). The FPGA connects to a USB2.0 interface chip, the Cypress FX2. FPGA circuitry and USB Microcontroller are programmable over the USB bus.

Standard FPGA consists of digital down converters (DDC) implemented with 4 stage cascaded integrator-comb (CIC) filters. CIC filters are high-performance filters using only adds and delays. Each DDC has two inputs of I and Q. The resulting signals are passed through 31 tap halfband filters to shape the spectrum and reject the out of band signals [8].

Figure 3 shows the block diagram of the USRP's receive path and the diagram of the digital down converter. The MUX is like a router or a circuit switcher. It determines which ADC is connected to each DDC input. DDC is one

Table 2 Mainboard specifications (ADC part)

Comp.	Parameter	Value
ADC	Resolution	12 bits
	Input Feq. range	up to 150 MHz.
	Input Voltage	$2V_{\text{pp}}$
	Input Impedance	50 ohm
	Sampling rate	submultiples of 128 MHz
		(64, 42.66, 32, 25.6 and 21.33 MS/s)
	Sampling type	Possible, (IQ) sampling
PGA	Gain	$0 \sim 20$ dB

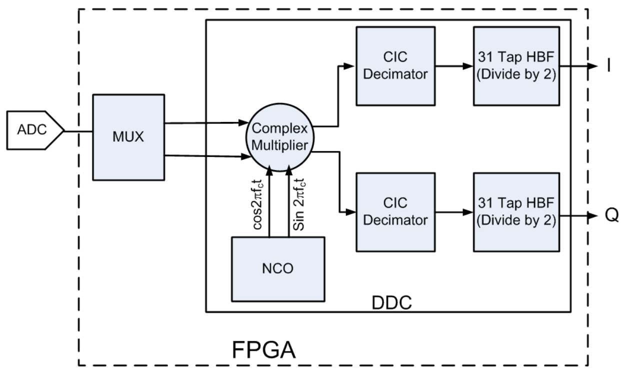


Figure 3 Block Diagram of FPGA for Rx Path

of the most important components of the FPGA. First, it down converts the signal from the IF band to the base band. Second, it decimates the signal so that the data rate can be adapted by the USB2.0 and is reasonable for the computers' computing capability. The complex IF input signal is multiplied by the constant frequency (usually also IF) exponential signal. The resulting signal is also complex and centered at 0. Then the signal is decimated with a factor N. The decimator can be treated as a low pass filter followed by a downsampler.

Regarding the bandwidth, we can sustain 32 MB/sec across the USB. All samples sent over the USB interface are in 16-bit signed integers in IQ format, i.e. 16-bit I and 16-bit Q data (complex), resulting in 8M complex samples/sec across the USB. This provides a maximum effective total spectral bandwidth of about 8 MHz for realtime transmission. Narrower ranges can be achieved by changing the decimation rate. The decimation rate must be set within the range [1, 256]. Finally the complex IQ signals enter to the computer via the USB interface.

Following Table 3 is showing the receiver parameters used in this experiment [5]

5. Experimental Setup

The spectrum sensor is implemented by using the GNU Radio and USRP in conjunction with Matlab Communications toolbox. A built in function of GNU Radio (usrp_rx_cfile.py) is used to store the received complex samples in a data file. Sensors need to scan a wider band to detect the presence of different narrowband emitters in the area. The USRP can scan bandwidth of as low as 250 KHz to a maximum of 8 MHz at a time. To detect the emergency radios wider spectrum is needed to be scanned. For example in Japan some bands of 150 MHz to 900 MHz is allocated for emergency communications. Most of the rescue teams use analog modulations like FM, while some systems use digital communications also. In the sensor implementations we have scanned narrowband spectrum sequentially and stored them to a data file. Figure 5 represents the setup used for the experiments. In this study we considered 890 MHz as the carrier to transmit and sense the Frequency modulated baseband signal of which bandwidth is 128 KHz.. To eliminate the effect of the propagation channel the output of the signal generator is connected to the USRP input through a coaxial cable.

Table 3 Receiver parameters

Parameter	Value
Carrier Frequency	890.5 MHz
Modulation	FM
Signal bandwidth	128 KHz.
Decimation	64
ADC Sampling Rate	1 MS/s
Sensing Bandwidth	1 MHz.
No. of Samples	8 M
Receiver Gain	10 dB

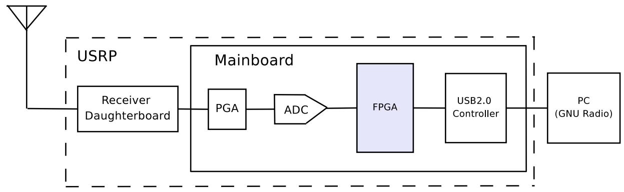


Figure 4 Receiver architecture

USRP outputs the complex IQ data that is transferred to the PC through a USB cable. Later the stored data is analyzed in Matlab to scan the whole spectrum. At first we took the noise data to calculate the noise power and energy threshold for the energy detector. Afterwards we transmitted the FM signal from a signal generator and applied the threshold value to detect the presence of emitters. Once the emitters are detected, we will try to extract the PHY and MAC parameters of the signal.

The USRP is the hardware that incorporates some AD/DA converters, RF front end (daughterboards), and an FPGA block. Received data are transferred from USRP to PC with a USB2.0 interface. The FPGA block is responsible for some very important but computationally expensive preprocessing tasks of the received signal. The USRP is low-cost

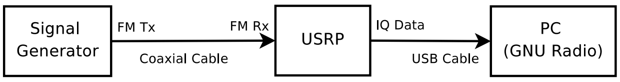


Figure 5 Experimental Setup

and relatively high speed, which is the best choice for a GNU Radio user to implement some real time applications at the moment and can be purchased from Ettus [6]. A basic diagram of the receiver is shown in Figure 4.

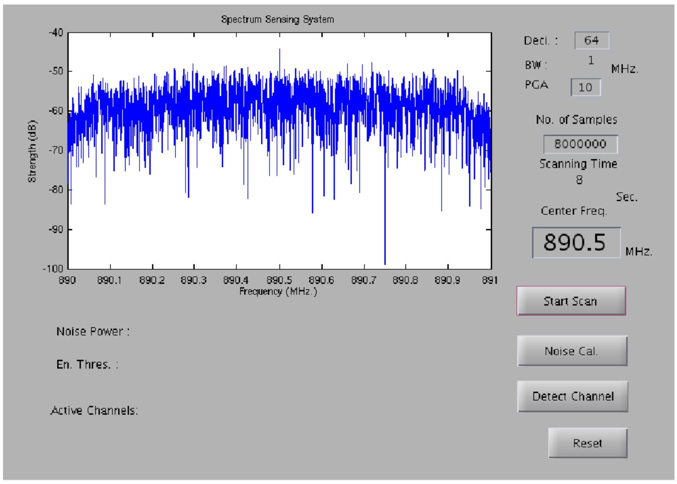
6. Measurement example

As mentioned above, we evaluated GNU Radio function, the usrp_rx_cfile.py. This function collects samples taken from the USRP and stores them in a data file (.dat). Input parameters can be adjusted easily to set the central frequency, the LNA gain, the number of samples, decimation value, output data format etc. An implementation example is described below:

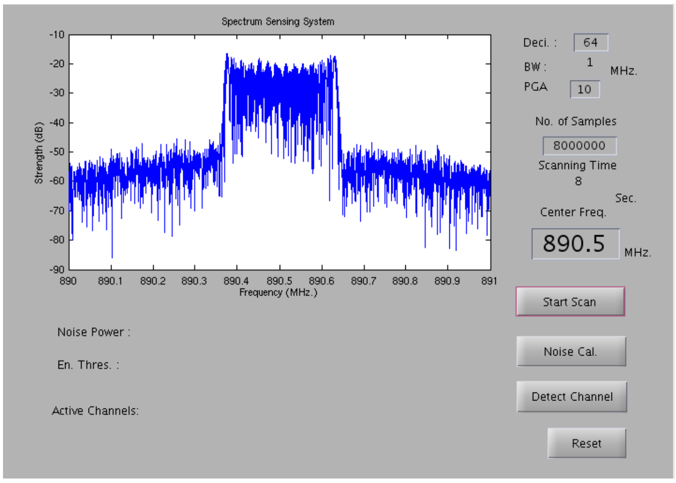
```
./usrp_rx_cfile.py -d 64 -f 890.5M -g 10 -N 8M rxsigdata.dat
```

A matlab function is created to automatically store the desired bandwidth data with appropriate parameters. Same script is used to evaluate the whole spectrum data. A graphical interface as shown in Figure 6 is developed for convenience. The interface takes the center frequency and other related parameters and gives command to the USRP through GNU Radio. Once all the desired channels are scanned, it shows signal power of the whole band. Figure 6(a) is showing the scanned signal strength of 1 MHz band with only noise centered at 890.5 MHz. The average noise floor is around -50 dBm. We calculated the noise power and the energy threshold with the techniques mentioned above. Afterwards, We transmitted a FM signal of bandwidth 128 KHz from the signal generator. In Figure 6(b) we can see the presence of the signal in the same frequency. The measurements were taken inside the laboratory. We calculated the energy of the signal and compared it with the noise threshold to detect the presence of signal.

This experimental study mainly aims to evaluate and verify the theoretical results on the performance of energy detector in a USRP based system. Energy detection theories discussed earlier has been implemented. We measured the probabilities of detection with respect to SNR for FM data transmitted with different relative power level (-70 dBm to -50 dBm). As the average noise is -50 dBm, the SNR varied from -10 to +10. These measurement values are taken in respect to a reference value. The experimental study gives us the ideas about some practical conditions like hardware imperfections, noises, wireless environment etc.



(a)



(b)

Figure 6 Spectrum Sensing System

7. Conclusions

This study investigated the preliminary implementation issues of spectrum sensing system to detect the presence of RF emitters in a disaster area. GNU Radio and USRP was used to collect the sensed data. These data were further analysed in matlab to identify the signals. We found the USRP to be a potential candidate for implementing the spectrum sensing systems. But, some calibration and controlling issues are still needed to be investigated. We are trying to investigate the limitations at the moment and hope to finalize the spectrum sensing system in near future.

References

[1] A. Rahman, S. Khadka, Iswandi, M. Gahadza, A. Haniz, M. Kim and J. Takada, "Design and Implementation of a Cognitive Radio Based Emergency Sensor Network in Disaster Area", IEICE Technical Report, SR2009-27 (2009-07), Software Radio, pp.31-36, July, 2009.

[2] D. Cabric, A. Tkachenko, and R.W. Brodersen , "Experimental Study of Spectrum Sensing based on Energy Detection and Network Cooperation", Proc. Intl. Workshop on Technology and Policy for Accessing Spectrum (TAPAS), 2006.

[3] P. J. Green, and D. P. Taylor, “A Real Time Cognitive Radio Test Platform for Public Safety Physical Layer Experiments”, IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC07), pp. 1-5, 2007.

[4] M. Kim, R. Ryo, K. Po, S. Maharajan and J. Takada,“Development of Spectrum Sensing Evaluation System”, IEICE Technical Report, SR2008-35, July 2008.

[5] D. Shen, “Tutorial 4: The USRP Board”, August, 2005. Available at: www.snowymtn.ca/gnuradio/gnuradiodoc-4.pdf

[6] http://www.ettus.com.

[7] M. Ettus,“USRP Users and Developers Guide”, Available at: www.olifantasia.com/gnuradio/usrp/files/usrp_guide.pdf

[8] F. A. Hamza,“The USRP under 1.5X Magnifying Lens!”, Rev. 1.0, June, 2008. Available at: http://gnuradio.org/trac/attachment/wiki/UsrpFAQ/USRP_Documentation.pdf