Implementation and Performance Evaluation of Spectrum Sensing Using Software Defined Radio

UNITED KINGDOM · CHINA · MALAYSIA

Thesis submitted to the University of Nottingham for the degree of

Master of Research (MRes)

Electrical and Electronic Engineering

Written by: Huang Yan

Principal Supervisor: Steve Greedy

Additional Supervisor: Angela Nothofer

George Green Institute for Electromagnetics Research

Department of Electrical and Electronic Engineering

The University of Nottingham, University Park, UK

September 30, 2016

Abstract

As the spectrum becomes more and more congested, an accurate understanding of its usage is required. Software Defined Radio (SDR) systems provide a flexible prototyping platform to develop a wide range of transceivers operating different protocols which can be changed in real time conveniently. SDR is a promising technique with low requirement of hardware devices which made it popular in recent years.

In this project, a SDR test bed has been built as an analysis tool for mapping the use of the spectrum and identifying potential sources of interference. In particular, the project will make use of the Ettus SDR platform (USRP B210) and the GNU Radio development toolkit.

In order to map the use of the spectrum, a system needs to be set up to work as a tool to implement this function. There are some previous simulation results demonstrating that the spectrum sensing schemes have a good performance though these are implemented in software. Whereas, spectrum sensing implementation with SDR in hardware, and the verification of the performance in different system configuration especially for wide bandwidth spectrum sensing has not been explored well. Evaluating the practical performance of spectrum sensing is very important in practice, including the deep understanding of the different spectrum sensing algorithms and the understanding of the practical limitations of the SDR hardware.

Therefore, the aim of this project is to research the use of a SDR platform to implement spectrum sensing and identify its performance due to potential electromagnetic interference. The comparison of the performances among different spectrum sensing schemes is presented in this thesis. For the practical use, a wide range of spectrum sensing methods also are presented to overcome the bandwidth limitation of the SDR hardware. In order to understand what will happen if adding interference to the system, the Gigahertz Transverse Electromagnetic (GTEM) Cell is used in this project to study susceptibility to electromagnetic interference.

The results of this project include the successfully built test bed for spectrum sensing with SDR using the USRP B210 hardware board and GNU Radio, a detailed technical documentation on how to implement an Out-of-Tree spectrum sensing module in GNU Radio, a comparison of Receive Operation Characteristic (ROC) curves for different spectrum sensing schemes, analysing of the system reaction when adding interference to the GTEM Cell and comparison of the results with the system under simulated noise.

The successful development of the spectrum sensing system with SDR gives plenty of opportunities for further research in Dynamic Spectrum Access (DSA) including spectrum sensing technique, for example, improving the threshold of energy detection as a dynamical and adaptive one which will increase the accuracy of sensing results, and using a more powerful platform (USRP X310) to improve the performance and decrease the sensing time for energy saving. These directions will be subjects for future research work.

Declaration

This work has been carried out by Miss. Huang Yan under the supervision of Dr. Steve Greedy and Angela Nothofer at the Department of Electrical and Electronic Engineering, The University of Nottingham, University Park, UK

Acknowledgements

At first, I want to thanks my principal supervisor, Dr. Steve Greedy, for introducing me to an interesting research area of wireless communication and encouraging me to explore my own research topic without any limitation for me. Special thanks also want to give to my additional supervisor Dr. Angela Nothofer for her continuous support and guidance. This thesis could not have been finished, if there is no help from them.

I also would like to express my sincere gratitude to all the staff in The University of Nottingham, as well as the school of Electrical and Electronic Engineering for their technical support and best working environment.

I am thankful to my colleagues Muhammed Diyar who gave me a lot of useful advice when I start learning SDR technology and help me to set up the basic communication system. Also I want to thanks my colleague: Dr. David Thomas, Meng Xuesong, Hayder Jahanger, Otobong Obot and Chen Ye.

Finally, I will give my sincere thanks to my parents and sister who give me encouragement and confidence while I feel frustrated during all my research progress. And I want to acknowledge all of my friends, they gave me a lot of assistance when I stay in Nottingham for the whole year.

Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analog to Digital Converter
API	Application Programming Interface
AWGN	Additive White Gaussian Noise
BRAN	Broadband Radio Access Network
CFAR	Constant False Alarm Rate
CP	Cyclic Prefix
CR	Cognitive Radio
CRN	Cognitive Radio Network
DSA	Dynamic Spectrum Access
ED	Energy Detection
EMC	Electromagnetic Compatibility
EME	Energy with Minimum Eigenvalue
EUT	Equipment Under Test
FC	Fusion Centre
FCC	Federal Communications Commission
FFT	Fast Fourier Transform
GNU	GNU's Not Unix
GRC	GNU Radio Companion
GTEM	Gigahertz Transverse Electromagnetic
IFFT	Inverse Fast Fourier Transform
ISI	Inter-Symbol Interference
ISM	Industrial Scientific Medical
LAN	Local Area Network
LO	Local Oscillator

LOS	Line-of-Sight
LTE	Long Term Evolution
MAC	Media Access Control
MAN	Metropolitan Area Network
MCM	Multi-Carrier Modulation
MIMO	Multiple Input Multiple Output
NI	National Instrument
OFDM	Orthogonal Frequency Division Multiplexing
OOT	Out-of-Tree
P/S	Parallel to Serial
PCIe	Peripheral Component Interconnect Express
PDF	Probability Distribution Function
PDU	Protocol Data Unit
PMTs	Polymorphic Types
PSK	Phase-Shift Keying
PU	Primary Users
QA	Quality Assurance
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RCO	Receiver Operating Characteristics
RMS	Root Mean Square
ROC	Receiver Operation Characteristics
S/P	Serial to Parallel
SDR	Software Defined Radio
SED	Sequential Energy Detection

SNR	Signal to Noise Ratio
SU	Secondary User
SWIG	Simplified Wrapper and Interface Generator
UHD	USRP Hardware Driver
USRP	Universal Software Radio Peripheral
WLAN	Wireless Local Area Network

1 Introduction

1.1 Overview

Because of the development of both portable devices and wireless network, there is an increasing demand for more radio spectrum channels [1]. To meet this great demand, more efficient spectrum management schemes are being generated.

The conventional spectrum management is a static allocation of the spectrum band which has been a prevalent method to implement spectrum management. In this traditional method, the licensed user (primary user) can use the spectrum band for a long time without sharing with other users. This static assignment can guarantee the communication quality, but creates an under-utilized spectrum. Such an inefficient utilization of limited wireless spectrum resources has motivated a lot of researchers and practitioners to search for innovative and advanced technologies to enable a more efficient and smarter spectrum resources utilization [2].

Cognitive Radio (CR) is one of the most discussed topics for up-to-date spectrum management [3]. It is an intelligent radio which can obtain the knowledge of radio operational environment and dynamically adjust its operational parameters or protocols to improve its performance. This feature can contribute to improving the spectrum utilization efficiency. It is always compared with Software Defined Radio (SDR), since both of them use software to set or alter the operation parameters of transceivers [4]. However, CR can be considered as a combination of SDR and intelligent signal processing with more flexibility.

Most current consideration of CR is used on an existing wireless communication system, which is therefore seen as a Secondary User (SU), coexisting with the Primary Users (PUs). However, CR could also improve the efficiency of spectrum band utilization. Dynamic Spectrum Access (DSA) is a new spectrum management paradigm that allows SUs to access the abundant spectrum holes in licensed spectrum band [5].

1.2 Motivation

1.2.1 Limitation of traditional hardware

The traditional hardware for spectrum management is based on radio devices which limit cross-functionality and can only be amended through physical intervention [6]. It means changing the components of the hardware to improve the existing system. This results in a higher cost of production and lower flexibility of multiple waveform standards. On the contrary, SDR platform provides a more comparatively, efficient, and inexpensive solution for the situation, which allows multi-functional wireless devices that can be upgraded easily.

1.2.2 Great demand in wireless communication

The tremendous increase in portable devices and computers has led to an ever-growing demand for greater data rates in wireless communication, and to an increasing demand for spectrum channels.

The benefits of SDR technology have a significant influence on the wireless communication industry. Cognitive Radio communication systems can be aware of their own inner environment and states. They have the ability to make sensible decisions about radio operating behaviour by mapping the predefined objective information. Using these elements is critical in allowing users to make an optimal use of available frequency spectrum and wireless networks by a common radio hardware toolkit.

1.2.3 The underutilization of spectrum band

Figure 1.1 illustrates the static spectrum allocation map in the UK from 2.35 GHz to 7.75 GHz, each different colours presents different usage of the spectrum band, which we can consider as a primary user. It can be seen from the picture that all the spectrum bands are allocated for different use. Therefore, it is nearly impossible to allocate a vacant frequency band to a new wireless communication for example the Long Term Evolution (LTE) and LTE-Advanced. Whereas, there are studies on spectrum allocation measurements that indicating this allocated spectrum is not fully utilized [7]. Dynamic Spectrum Access can significantly improve the utilization of spectrum.

It enables secondary users or unlicensed users to temporarily utilize the vacant frequency band without interference for primary users.

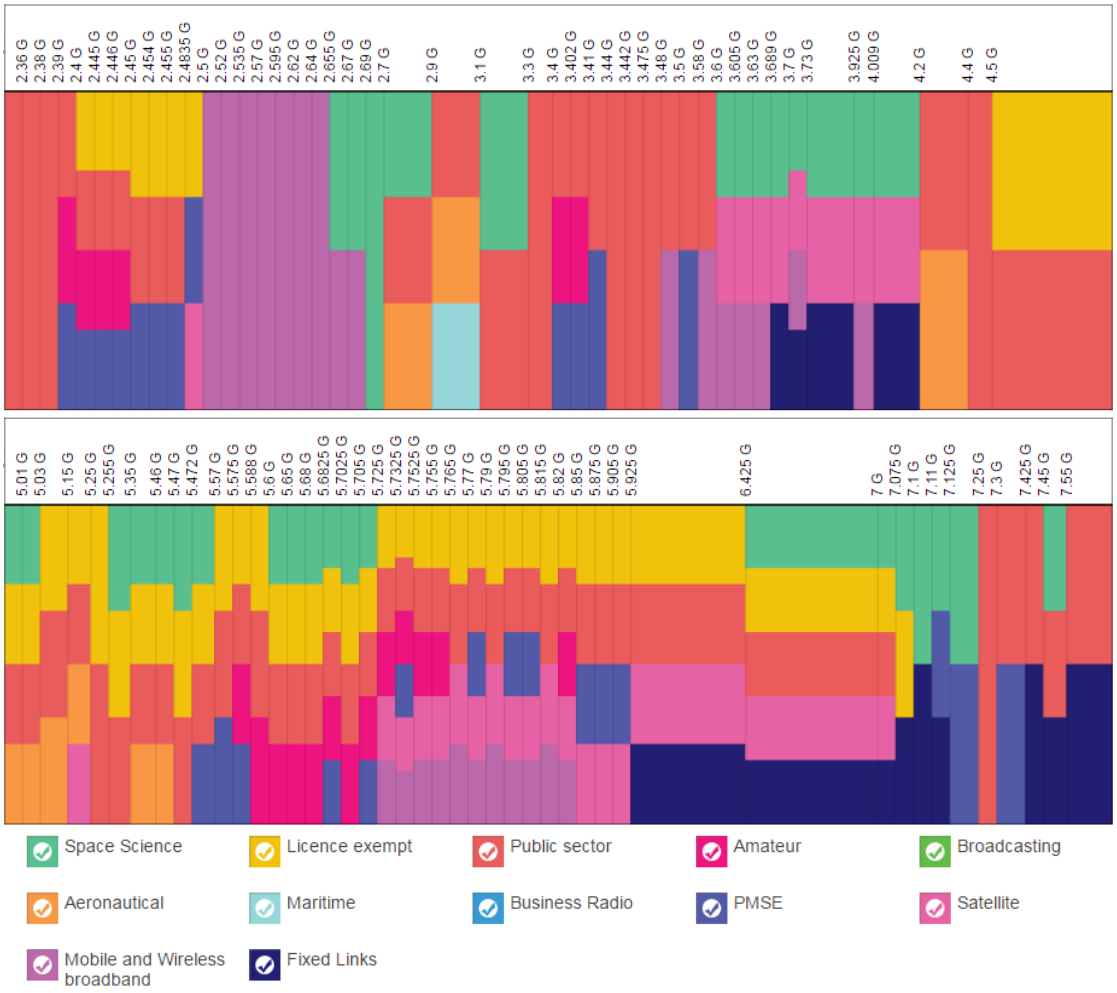

Figure 1.1 The UK spectrum map form 2.35 to 7.75 GHz [8]

1.3 Aim and objectives

The aim of this project is to research the use of a SDR platform as an analysis tool for mapping the use of the spectrum and identifying potential sources of interference. This interference could be the Additive White Gaussian Noise (AWGN) generated from GNU Radio software or other signals generated by vector signal generator. The project will make use of Ettus SDR platform and GNU Radio development toolkit. The detailed objectives are listed as following.

- Provide a literature review about Dynamic Spectrum Access, especially in spectrum sensing methods, OFDM system and cognitive radio techniques;

- Understand how to build OOT modules in GNU Radio;
- Understand how to use USRP B210 toolkits;
- Set up a wireless communication system using GNU Radio and USRP;
- Implement dynamic spectrum sensing methods in a wide bandwidth using GNU Radio Companion and USRP toolkit;
- Evaluate the performance of the dynamic spectrum sensing system with different modulation types;
- Compare the performance of different spectrum sensing methods under different SNR;
- Evaluate the performance of the spectrum sensing methods under interference using a GTEM Cell test environment.

1.4 Thesis structure

This thesis is organised as follows:

Chapter 1 introduces the motivation of the project on dynamic spectrum sensing and then put up with the aim, approach and objective of the project.

Chapter 2 presents a detailed literature review of the project. It covers the Cognitive Radio, Dynamic Spectrum Access, Wireless Communication Channel and OFDM techniques.

Chapter 3 introduces the software GNU Radio and hardware platform USRP which is used in this project. The architecture of GNU Radio is also presented in this part. Then introduce some basic parts of a flow graph to demonstrate how to build an OOT block to implement users' own function. In the hardware part details of different SDR boards are given. After comparing these boards, USRP B210 board and USRP X310 are chosen as the implementation platforms. It also details the architecture of these boards and the assemble steps of X310. The definitions of different bandwidths about USRP also are presented.

Chapter 4 sheds light on the Dynamic Spectrum Access that contains different spectrum sensing methods, mainly including knowledge-aware sensing, blind sensing

and semi-blind sensing. And it also refers to cooperative sensing and spectrum sharing.

Chapter 5 discusses the implementation of spectrum sensing on USRP board and GNU Radio, including the transmitter side and the receiver side. It also introduces the implementation of spectrum sensing modules with GNU Radio.

Chapter 6 provides the experimental results. There are three blind and semi-blind sensing methods are discussed in this chapter. It also explains how to realize a wide bandwidth frequency scanner based on the existing platform. And finally compare their outcomes in different configurations.

Chapter 7 identifies potential source of interference by using Gigahertz Transverse Electromagnetic (GTEM) Cell. At first, it details the function of the GTEM Cell and how to use GTEM Cell to set up the interference test environment is then presented. Finally, the implementation and results of interference estimation are discussed.

Chapter 8 makes a conclusion of the whole thesis. All the achievements will be presented and also discuss about the future research work.

2 Literature review

In this section, the following techniques will be introduced, which are involved in this project, as well as the research questions that are included in this project.

- Cognitive Radio
- Dynamic Spectrum Access
- Spectrum Sensing
- Wireless Communication Channel
- Orthogonal Frequency Division Multiplexing (OFDM)

2.1 Cognitive Radio

Cognitive Radio principles were first described by Joseph Mitola in 1999, in this paper Mitola described the cognitive radio could enhance the flexibility of personal service [9, 10], which is shown in Figure 2.1. It was thought as an ideal method to utilise software defined radio platform which is a reconfigurable wireless toolkit that can change its communication variables according to network and user demands [11].

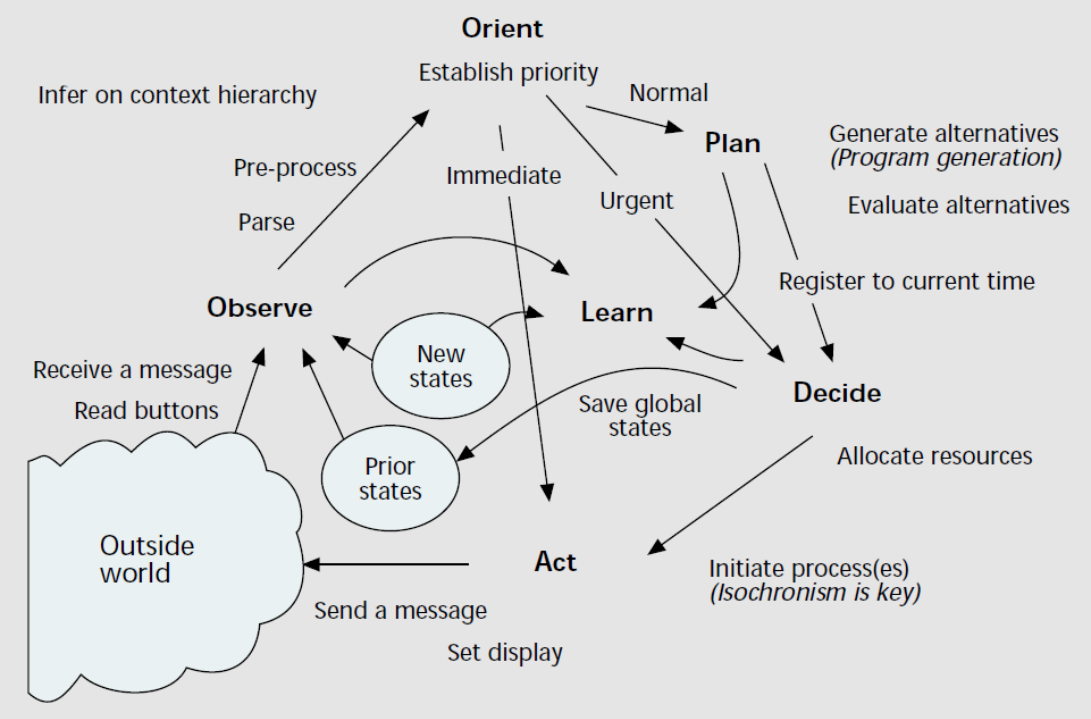

Figure 2.1 Joseph Mitola's view of cognitive radio [9]

In November 2002, the Federal Communications Commission (FCC) in the Unites States published a report about the spectrum usage models and interference avoidance. The aim of this report was to improve the way in which this precious spectrum resource was managed [12]. The concept behind cognitive radio is to exploit the unused spectrum opportunistically. Besides, from Symon Haykin's view, cognitive radio is an intelligent wireless communication system which can aware of the environment and learn from the environment by the methodology of standing-by-building [11].

2.1.1 The relation between CR and SDR

Cognitive Radio and Software Defined Radio are both fast developing technology, which can be used to reduce the spectrum shortage problems. To increase the spectrum utilisation and optimize the radio usage is the main motivation behind them.

The concept of SDR means a part of or all of the components that have been typically implemented in hardware are instead implemented by software which can be changed by users. A basic SDR system may consist of a personal computer equipped with analog-to-digital convertor, RF front end and so on. SDR is mainly concerned about the realize methods of signal processing in radio communication systems including flexible algorithm and software.

However, CR is an intelligent radio that can be programmed and configured dynamically according to the change of its working environments, such as wireless communication network. CR can be considered as a combination of SDR and intelligent signal processing with functional elements of radio flexibility, spectrum sensing and the intelligent decision-making. CR's adaptability to an optimal channel will benefit to other users and improve spectrum efficiency for wireless communications.

Although cognitive radio was initially thought as a software-defined radio extension, currently most research work is focusing on Spectrum Sensing Cognitive Radio. The essential problem of this is to design high quality spectrum sensing platform and algorithms for changing data between different sensing nodes. The basic energy detection cannot guarantee the accurate detection of existing signal. To operate the

signal detector in desired region, related work has been proposed with several more sophisticated sensing techniques. To further improve the sensing performance, cooperative spectrum sensing method decrease the probability of false detection[7].

2.1.2 Main functions

The main functions of spectrum management in CR networks are [13, 14]:

1) Spectrum Sensing: It refers to detect the unused spectrum and sharing it without harmful interference with other users. Spectrum sensing can be classified into three categories:
    a. Primary transmitter detection: CR must have the ability to determine whether a signal from a primary user is present in a certain spectrum range, there are several approaches proposed:
        - Matched filter detection
        - Energy detection
        - Feature detection
    b. Primary receiver detection
    c. Interference temperature management
2) Spectrum Management: It is to make decision and choose the best available spectrum to meet the communication requirements [15]. CR should decide on the best spectrum band to meet the Quality of Service (QoS) requirements in defined spectrum band, spectrum management techniques can be classified as:
    - Spectrum analysis
    - Spectrum decision
3) Spectrum Mobility: It is a key feature enabling continuous Second User (SU) data transmission. Cognitive Radio Networks (CRNs) aim to use the defined spectrum in a dynamic manner by making the terminal operating in the optimal frequency band.
4) Spectrum Sharing: It means to share their local spectrum scheduling method with other nodes in CRNs. It can also be classified by three categories:
    - The architecture:
        - Centralized
        - Distributed
    - Allocation behaviour:

        - Cooperative: form clusters to share interference information locally.
        - Non-cooperative: may reduce spectrum utilisation.
    - Spectrum access technology:
        - Overlay spectrum sharing
        - Underlay spectrum sharing

2.2 Dynamic spectrum access

Dynamic Spectrum Access (DSA) networks have been put forward since there is a poor utilization of the limited spectrum [11, 13, 16]. When considered with wireless communication technology, improving spectrum efficiency has become an important factor in both unlicensed and licensed frequency band [17].

There are some existing test beds to implement DSA using different toolkit. They can be found on the website [18]. Yan first implemented the cognitive radio network (CRN) test bed using USRP devices by energy detection methods for spectrum sensing and OFDM implementation with five channels [19]. It demonstrates the spectrum access and analyses the coexistence of primary users and secondary cognitive radio users in a common frequency band. It uses Quadrature Phase Shift Keying (QPSK) signal and OFDM signal as the primary and secondary user respectively. However, it only presents an interference-free case and lacks further analysis of the performance on different parameter, such as primary user traffic and noise level.

The majority of the recent work about implementing DSA test bed for CRNs have been concerned about the evaluation of spectrum sensing algorithms performance. Weber et al. evaluated the performance of three blind spectrum sensing methods [20]: energy detection, statistical covariance and prolate filter bank. All these methods are implemented in the 2.4 GHz ISM-Band on GNU Radio and USRP2 equipment. Furthermore, it presented a new perspective to compare different spectrum sensing algorithms by considering both detection accuracy and computational costs.

In [21], Rebeiz et al. proposed a novel algorithm to improve the robustness of the conventional cyclostationary detector under a cyclic frequency offset and use USRP

N210 to verify the validity of the spectrum sensing method. It also overcomes the SNR wall phenomenon. SNR wall is a SNR threshold. When the SNR value is below the SNR wall, it is impossible to detect the primary user. Mate et al. put forward two different spectrum sensing methods using time covariance matrix: covariance based method and eigenvalue based method, which don't need the primary users' information and have a better performance than conventional energy detection in [22]. This work also uses the USRP as the hardware platform. The maximum and minimum eigenvalue spectrum sensing method was first proposed in [23]. In [24], the author compared the cyclostationary-based algorithm (SPCAF) and Eigenvalues-based algorithms (MME and EME) by measuring the detection probabilities as a function of SNR for a given false alarm probability. It also compared different modulation types using two USRP N210 boards and GNU Radio.

To overcome the inaccuracy of static threshold in energy detection, the author proposed an histogram based adaptive threshold method in [25] and demonstrated the effectiveness of this histogram-based method by using USRP1 and GNU Radio. While another spectrum sensing method makes use of the traffic information about primary users in [26] which estimates the threshold by employing the statistical signal occupancy information. The used test bed includes a non-continuous ZigBee signal as a primary user and a USRP2 as the platform.

When evaluating energy detection based spectrum sensing methods, most approaches prefer to change the variables of the system. Budihal et al. implemented spectrum sensing algorithm considering to change the threshold of energy detection [27]. In [28] the energy detection method is investigated on a USRP test bed, and the author used MATLAB®/Simulink® as the software program environment. Jayawickrama et al. calculated the error of spectrum sensing when a SU implemented the energy detection method and considered both non-cooperative sensing and cooperative sensing scenario in GNU Radio and USRP [29]. They also consider adding two SUs and a centre to decide an optimal spectrum sensing threshold value. In [30], Syed-Yusof et al. presented a cooperative spectrum sensing scheme and gave a suggestion to add the Physical layer and the Media Access Control (MAC) layer together to achieve the synchronization. A TDMA-based cooperative sensing algorithm was tested on a

USRP board; it used the individual energy detection and then combined them together [30].

2.3 Wireless communication channel

The channel type of the system will affect the quality and the performance of wireless communication system significantly. Therefore, different communication channels will be discussed in this part. In [31-35], authors introduce about the behaviours of the wireless communication channel, effects and methods for battling these effect. Because of the signal scattering, reflection and refraction from high buildings or other objects, there will be more than one path between the transmitter and the receiver. There are two groups of multipath effect as following:

- Small scale effects
- Large scale effects

The first effect includes multipath fading; while the second effect includes path loss and shadowing. Shadowing fading is an average attenuation of the signal power effected by dense urban or high hills in transmit channel, which is described according to path loss.

2.3.1 Multipath fading

Multipath fading means the transmitted signals have many different transmitting paths before reaching the receiver. Figure 2.2 shows a scenario that the received signal is a combination of several parts of the original transmitting signal with different signal attenuation, delay and phase.

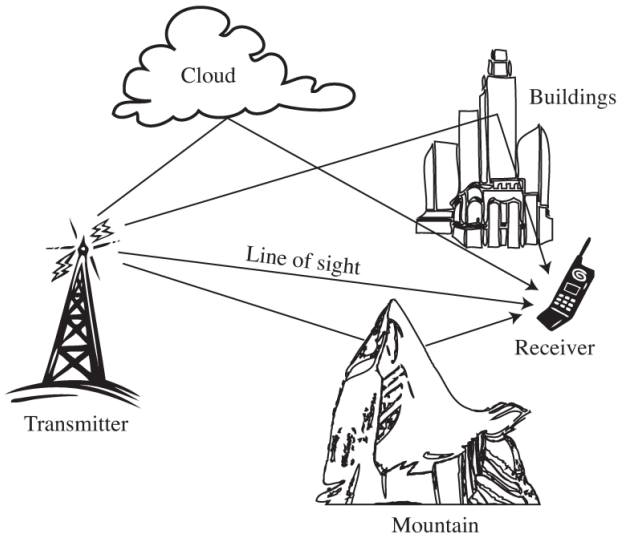

Figure 2.2 A Typical multipath fading scenario [36]

There are some important characteristic parameters about received signal in the multipath fading channel are shown as following:

- Root mean squared (RMS) delay spread: the difference between the first and final receiving signal transmitting paths;
- Doppler spread: the frequency change of the observed signal due to the relative movement of a transmitter and a receiver;
- Coherence time: the time over which the channel impulse response is independent;
- Coherence distance: the distance from the receiver antenna to the transmitter antenna;
- Coherence bandwidth: contiguous frequencies range;
- Number of multipath: the number of different paths between the transmitter and the receiver.

2.3.2 Rayleigh fading and Rician fading

Rayleigh fading and Rician fading are both prevalent channel models that are widely used in narrowband and wideband wireless communication channels. In practice, some phenomena of propagation (Doppler shifts, multipath scattering effects and time dispersion) are presented.

Rayleigh fading channel is a rational model while there are many objects in the channel which reflect and scatter the transmitted signal and no direct path (Line-of-sight) between the transmitter and receiver[32, 35, 37]. The multipath signal can be considered as a Rayleigh distribution, when there is no Line-Of-Sight (LOS).

The received signal is $s(t)$, at the Rayleigh channel:

$$s(t) = \sum_{i=1}^{N} \alpha_i \cos(\omega_c t + \phi_i) \tag{2.1}$$

Where N is paths number, α_i, ω_c, t and $\phi_i$is the amplitude, frequency, time and the phase of the i^{th} signal at the receiver respectively. If the relative motion direction of received antenna is angle ψ_i and it is assumed that the angle $\psi_i$and phase ϕ_i are both uniform distribution over [0, 2π]. The Doppler shift can be written as

$$\omega_{d_i} = \frac{\omega_c \upsilon}{c} cos\psi_i \tag{2.2}$$

The velocity of receiver is υ, the speed of light is c.

Considering Doppler shift, the received signal is given by

$$s(t) = \sum_{i=1}^{N} \alpha_i \cos(\omega_c t + \omega_{d_i} t + \phi_i) \tag{2.3}$$

Equation (2.4) can be expressed as in-phase ($I(t)$) and quadrature phase ($Q(t)$) form

$$s(t) = I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t) \tag{2.4}$$

Where

$$I(t) = \sum_{i=1}^{N} \alpha_i \cos(\omega_{d_i} t + \phi_i) \tag{2.5}$$

$$Q(t) = \sum_{i=1}^{N} \alpha_i \sin(\omega_{d_i} t + \phi_i) \tag{2.6}$$

The amplitude envelope R of this signal is given by

$$R = \sqrt{I(t)^2 + Q(t)^2} \tag{2.7}$$

The probability density function of the signal is given by the Gaussian density

$$f_Q(q) = \frac{1}{\sqrt{2\pi}\sigma} e^{-q^2/2\sigma^2} \tag{2.8}$$

Where σ^2 is the signal covariance of the in-phase ($I(t)$) components and quadrature phase ($Q(t)$) components. The marginal (envelop R) probability density function is

$$f_R(r) = \frac{r}{\sigma^2} e^{-r^2/2\sigma^2} \tag{2.9}$$

Where $r \geq 0$, the equation 上方 is Rayleigh probability density function.

While there is LOS component present between the transmitter and the receiver Rician channel will occur [35, 38, 39]. Typically the LOS signal is much stronger than other scattered signal. Therefore, the received signal can be written as

$$s(t) = \sum_{i=1}^{N} \alpha_i \cos(\omega_c t + \omega_{d_i} t + \phi_i) + k_d \cos(\omega_c t + \omega_d t) \tag{2.10}$$

The k_d is the direct signal amplitude of the component, ω_d is the frequency of Doppler shift, ω_{d_i} is the Doppler shift at the i^{th} direct path, the marginal probability density function of the received signal is obtained by

$$f_R(r) = \frac{r}{\sigma^2} e^{-(r^2 + {k_d}^2)/2\sigma^2} I_0\left(\frac{r k_d}{\sigma^2}\right) \tag{2.11}$$

Where

$$I_0(z) = \frac{1}{2\pi}\int_0^{2\pi} e^{-zcos\theta}d\theta \tag{2.12}$$

Equation (2.12) is called the zero-order modified Bessel function. While a random variable R having the Probability Distribution Function (PDF) given by equation (2.11) is said to be Rician distribution.

Rician fading is defined by K factor which is the ratio between the power of LOS path and all the other scattered components received at the receiver [39]. It is given by

$$K(dB) = 10\log\frac{{k_d}^2}{2\sigma^2} \tag{2.13}$$

For $K = -\infty$, it means there is no LOS path signal and the Rician distribution changes to the Rayleigh distribution; While for $K = +\infty$, the Rician distribution has no fading, channel has no multipath and only LOS component, so it becomes Gaussian distribution.

2.4 Orthogonal Frequency Division Multiplexing

In the thesis the wireless communication system is mainly based on OFDM architecture. Therefore, in this section some basic knowledge of OFDM is presented.

OFDM is a method of encoding digital data on multiple carrier modulation (MCM) frequencies, it modulates data symbols in parallel on orthogonal subcarriers [40]. The key point behind OFDM technology is divide the transmission channel to a set of orthogonal sub-channels which have flat transfer function and AWGN (Additive White Gaussian Noise). Then the transmission data is sent in parallel on all of the independent sub-channels. Figure 2.3 and 2.4 shows the structure of an OFDM transmitter and receiver. This system is implemented on discrete time domain.

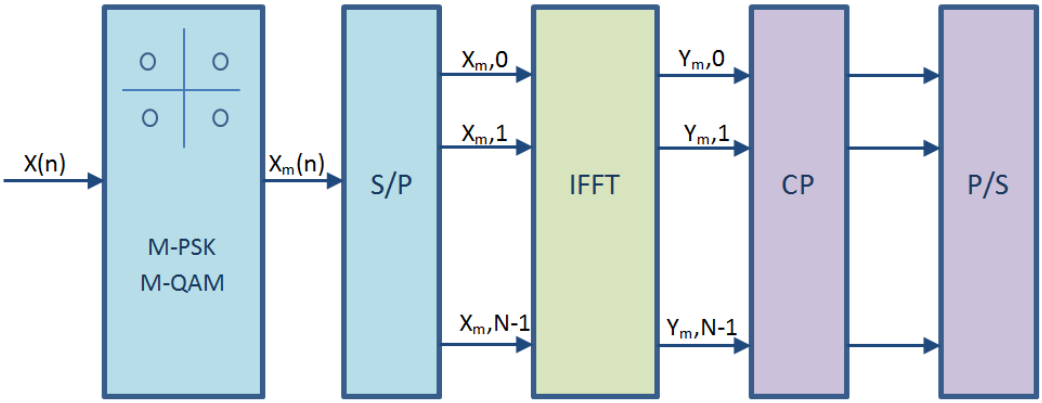

Figure 2.3 OFDM transmitter structure

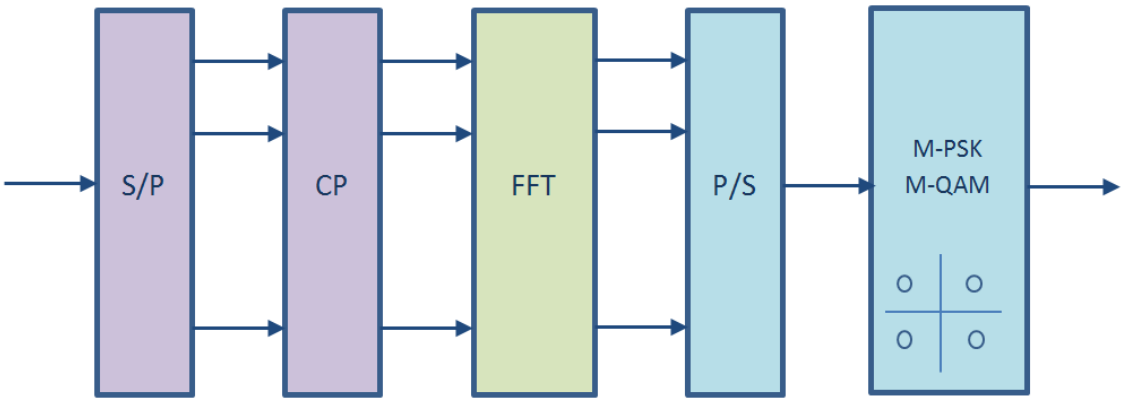

Figure 2.4 OFDM receiver structure

On the transmitter side, the OFDM works as below:

- Serial bit streams come from the source are divided into equal parts, each of them has M bits;
- Each part is mapped in the symbol mapping block, every M bits will mapped to one modulated complex symbol (M-PSK or M-QAM);
- Set the serial complex symbols to parallel complex symbols (S/P);
- Using Inverse Fast Fourier Transform (IFFT) to translate the symbols from frequency domain to time domain;

- Add Cycic Prefix (CP) and translate parallel to serial (P/S).

While on the receiver side, it works as follows:

- Receive the signals to translate serial to parallel (S/P) and remove Cycic Prefix (CP);
- Use Fast Fourier Transform (FFT) to translate the symbols from time domain to frequency domain;
- The resulting signals are fed to equalization;
- Translate parallel signal to serial signal (P/S);
- De-map the complex signal to the original one (M-PSK or M-QAM).

In the OFDM transmitter structure, after IFFT the symbols are added with Cycic Prefix to eliminate the residual Inter-Symbol Interference (ISI). On the receiver side, channel equalization becomes simpler than by using adaptive equalization using single carrier system. As OFDM has these advantages, it is prevalent to apply in digital communication system. For example, Wireless Local Area Network (WLANS), DVB-T system, High Performance Local Area Network type 2 (HIPERLAN-Ⅱ), and Broadband Radio Access Networks (BRANs) [32]. OFDM has been regarded as an important part of IEEE802.11 and IEEE802.16 standards for high bit rate data transmission over wireless Local Area Networks (LANs) [41].

There are still some disadvantages for OFDM. The most significant one is the Peak-to-Average Power Ratio (PAPR) will grow up while compared to the simple single carrier systems [42]. Another one is the power and capacity loss because of the Cycic Prefix, which can be significant. Furthermore, OFDM is sensitive with phase noise and frequency offset. Therefore, OFDM system needs more research to guarantee the received signal not to be distorted, the efficiency of bandwidth and low PAPR. Recently, there are some novel technologies to improve the performance of the OFDM-based systems. Such as bandwidth efficiency of OFDM without Cycic Prefix [43-45],PAPR reduction techniques [46, 47], channel estimation and synchronisation [48-52].

2.5 Summary

In this chapter, a detailed literature review is presented about most of the main basic concepts and techniques that are important in this thesis. They are Cognitive radio, Dynamic Spectrum Access, Wireless communication channel and OFDM–based system. It also comes up with some open research problems that are the motivation to implement the project. The next chapter will present an introduction of the software and hardware used in the project.

3 GNU Radio and USRP

3.1 Introduction

In order to implement spectrum sensing, we need to choose a compatible software and hardware combination. In our experiment, we choose GNU Radio as the software platform and Ettus USRP B210 as the hardware.

3.2 Software – GNU Radio

GNU Radio is a free software toolkit which provides signal processing blocks to implement signal processing and software defined radio system [53]. It can be used without hardware for simulation or connected with RF hardware to create software defined radio, which means it can use less hardware to develop wireless communication system. In other words, the digital modulation problems in wireless communication application can be treated using software.

3.2.1 GNU Radio architecture

The architecture of GNU Radio includes two components. The first one is an abundant library of signal processing blocks, which include coding/decoding, modulation/demodulation, FFT/IFFT, filtering and I/O operations. Another one is the Python scripts to control the stream flow of the data, which always are worked as glue to tie them all together. Being similar to connect physical FR building components to construct a hardware radio, users can easily build a SDR system by connecting blocks together using Python scripts. This can be done by Simplified Wrapper and Interface Generator (SWIG) using shared common library for both C++ and Python. In other words, SWIG is an interface compiler which connects programs that was written in C++ with Python. It is shown in Figure 3.1.

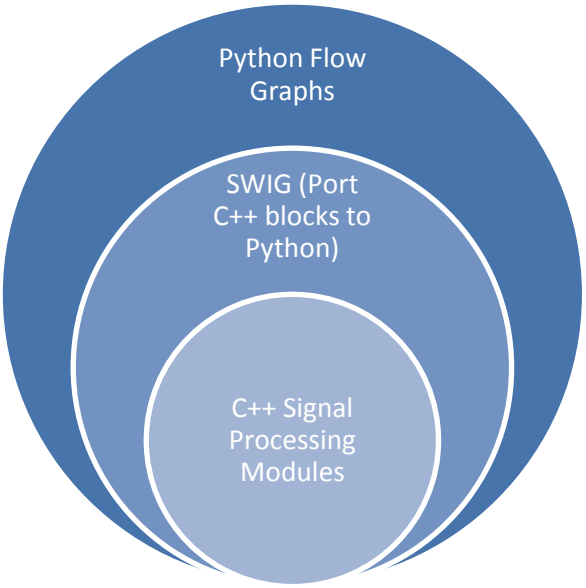

Figure 3.1 GNU Radio software architecture

In this project the RF hardware of GNU Radio Architecture is USRP B210, which will be presented in chapter 3.3. This board is connected with computer via a USB 3.0, and the GNU Radio, it gives a robust Application Programming Interface (API) that can control the USRP hardware device. Figure 3.2 shows the hardware architecture of GNU Radio.

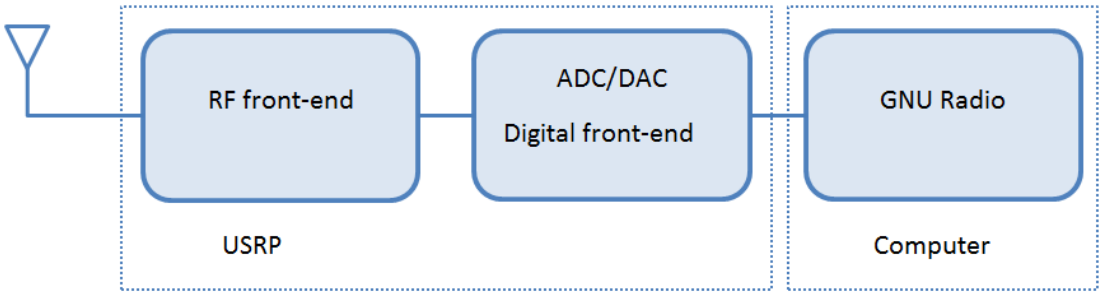

Figure 3.2 GNU Radio hardware architecture

3.2.2 GNU Radio Companion

GNU Radio Companion (GRC) is a graphical tool to create signal flow graphs and generate flow graph source code, which is used to develop GNU Radio applications. GRC is a front-end for GNU Radio libraries to implement signal processing.

In this project we choose the GRC as our experiment platform, which can easily implement a Cognitive Radio system without a complicated hardware. It provides many different kinds of blocks and visualization data sinks, including scope displays, FFT displays and symbol constellation diagrams. These are all useful for debugging radio applications.

GRC is a combination of both Python and C++ to construct a software systems. Users can also connect blocks and generate a signal flow graph using a graphical connecting interface in GRC. Most of the blocks in GRC are implemented by C++ programming language and connected by Python. Although, the visual programming is less flexible and powerful than programing by Python directly, it is more convenient and intuitive than programming by Python for most users. The figure 下方 illustrates a simple flow graph in the GRC which shows a simple structure of FM receiver.

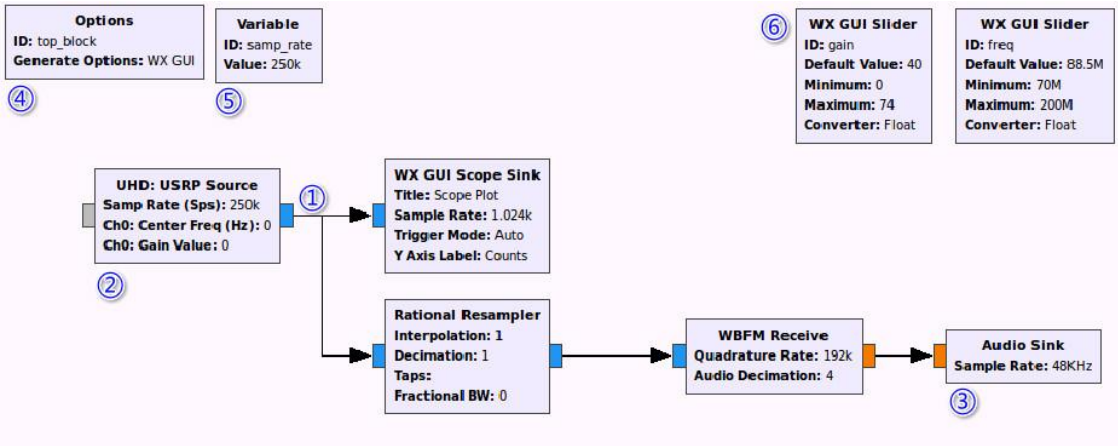

Figure 3.3 GNU Radio flow graph

In the centre of Figure 3.3, It can be seen that a flow graph that is connected by different blocks which will be introduced in section 3.2.4. These blocks are connected together at port (number ①), the first block has no input port, and it produces samples. If one block like this, without input ports, it is called a source (number ②). The final block, without outputs, is called a sink (number ③). The different colour of a port indicates the type of samples flowing through the port which is shown in Figure 3.4. Therefore, in this flow graph above, the blue port means complex float 32 samples flowing through the port and the orange one means float 32.

On the top left of Figure 3.3, the block *Options* (number ④) is used to set global value parameters. After executing the flowgraph, it will produce a python file. The name of the file is the same as *ID* in the *Options* block. The default name of the python file is *top_block*. The next is a *Variable* block (number ⑤) which contains an arbitrary Python expression. We can refer to it in another block by its ID. In this figure we set the *samp_rate* variable as 250 KHz. So we can use the variable *samp_rate* in other blocks other than use 250 KHz, which make it more convenient to change a variable.

On the top right of the figure, there are two *WX GUI Sliders* (number ⑥) which is the same as *Variable* blocks, but the variable can be changed in a range from Minimum to Maximum. And the default value of the variable can be set as *Default Value* in the blocks. In the first *WX GUI Slider*, we set the variable gain as 40 for default which can be changed from 0 to 74.

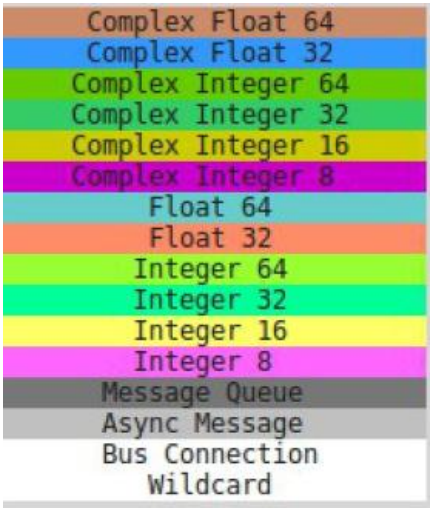

Figure 3.4 Colour mapping of different type in GNU Radio

3.2.3 Installation

GNU Radio can be installed on different operation system, including Linux, Windows and Mac OS X. In this project we install GNU Radio in Linux Ubuntu, so the recommended way to install it is via the gnuradio package from distribution's standard repositories.

After installing GNU Radio, the GRC should also be installed, which is bundled with gnuradio, following the next steps can install the GRC:

LINUX 3.1

```
$ apt-get install gnuradio
$ cmake ..
$ make
$ sudo make install
```

3.2.4 Executing a flow graph

In this chapter, there are some basic introductions about the flow graphs of GNU Radio. They are graphs that all data flows through it. Most GNU Radio applications include nothing other than a flow graph [54]. There are some basic terminologies about GNU Radio, which are shown in the table below.

Table 3.1 Terminology of GNU Radio

Name	Function
Block	A processing unit with inputs and outputs
Port	Input or output of a block
Source	Produce data, without input source
Sink	Consume data, without output ports
Flow graph	A number of blocks and connections
Item	Data unit. For example: baseband sample, vector...
Stream	The flow of consecutive items
IO signature	The description of input and output ports of blocks

1) Blocks

The boxes in a graph are blocks. The data flows along them. Besides, all the signal processing steps are done in these blocks. In most cases, each block only does one job which makes GNU Radio more flexible and modular. We can see from Figure 3.3,

there are several blocks, data flows from left hand to right hand in this scenario. The blocks are connected at ports and each output can connect to more than one input. The outputs of blocks called items, which can be different type. The most common types are real, complex, integer and vector.

There are several different types of blocks in gnuradio framework:

- Synchronous Blocks (1:1)
- Decimation Blocks (N:1)
- Interpolation Blocks (1:M)
- General Blocks (N:M)

The synchronous blocks are the most common blocks which allow users to create blocks with equal consuming and producing number of items per port. This kind of blocks may have multiple input and output ports. If only having output, it called a source. However, if only having input, it called a sink.

The number of input items of decimation block is N times of the output items. While the interpolation blocks is just the reverse of the previous one. The M in Interpolation Blocks means the output items is M times of input items. The general blocks have no relation between the input and output number of items.

2) Streams and Messages

Most of the blocks are using a synchronous method to passing the items, they simply continue working when there are items fed into inputs. However, when dealing with packets, this is not enough. There must be a method to identify the packet boundaries, such as which item is the first byte of the packet and how many byte it contains.

Therefore, GNU Radio provides two different ways to implement this:

- Tagged stream blocks
- Message passing blocks

The first one is like the regular streaming block; it is used to find PDU (Protocol Data Unit) boundaries. A tag is added to a stream with metadata and is related to a

particular item in a stream. A tag has four members (offset, key, value and srcid). This is convenient to connect blocks that know about PDUs with that don't know about PDUs.

There are some limitations of the tagged stream. For example, it is only accessible inside a work function and only flows at one direction. The benefit is that it is synchronous with the data. The message passing is an asynchronous method which can downstream to communicate back to blocks upstream. Besides, it allows us to communicate back and forth between external application and GNU Radio. The interface of message passing heavily relies on Polymorphic Types (PMTs). This might be a preferred method on a MAC (Media Access Control) layer.

3.2.5 Structure of a GNU Radio Out-of-Tree module

Sometimes the existing blocks cannot satisfy most users' demand, an Out-of-Tree (OOT) module needs to be created, which is a GNU Radio component that doesn't stay in the GNU Radio source tree. In most cases, if you want to implement your own functions or blocks, an OOT module is what you need to create. There is a useful module editing tool called *gr_modtool* that has done most monotonous work for users, such as makefile editing and boilerplate. The aim of this *gr_modtool* script is to help the developer to make more efforts on the DSP coding not editing makefiles.

The GNU Radio module is composed of different subdirectories: *cmake, examples, app, lib, docs, include, swig, python and grc.* There is a *Makefile.am* file in each folder, since auto tools are built in advance, in order to compile C++ code into python code automatically. The part below lists the functionality of each folder.

- The ***app/*** folder includes complete applications. When the module is built, they will be built at the same time;
- The ***cmake/*** folder contains files for *cmake* to include the libraries of GNU Radio;
- The ***docs/*** sub-direction consists of some instruction files to extract documents from Python files and C++ files. It also makes sure the availability of the document in python as *docstrings*;

- The ***lib/*** sub-direction consists of .cc files and .h files (source files) to build blocks. It also needs to be added by users when implementing a new module. Any source code written in C++ is put in this folder. However, the header files are also put at *include/* folder when they need to be exported, otherwise, to put in the *lib/* folder when they are related to compiling progress, such as *_impl.h* files;
- The ***swig/*** folder consist of files which are used to build python interface using SWIG tool for C++ program;
- The ***python/*** folder instores all the python files, which includes unit test files and python files for installing python modules;
- The ***grc/*** subdirectory includes the XML descriptions of the blocks which makes the blocks available in GRC.

3.3 Hardware – USRP

This part provides an introduction of the hardware platform Universal software radio peripheral (USRP) used for Software Defined Radio. It is one of the most common forms of hardware to be used to build a SDR system with GNU Radio software. There are several designs available for electronics enthusiasts to assemble their own SDR hardware. For example, Ettus Research USRP, Great Scott Gadgets HackRF and Firewaves' UmTRX are all compatible with GNU Radio. While others like Nuand's bladeRF and OpenBTS's Range Network SDR1 are using their own software. The table 下方 shows the difference of SDR platforms available from different companies.

Table 3.2 The comparison of different companies' SDR platform

Name		Frequency range	Sampling rate	Host interface	price
USRP	B210	70MHz - 6GHz	56Msps	USB 3.0	£890
	X310	DC - 6GHz	200Msps	Gigabit Ethernet, 10 Gigabit Ethernet, PCIe	£3870
HackRF		1MHz - 6GHz	8Msps - 20Msps	USB 2.0	$299
UmTRX v2.2		300MHz - 3.8GHz	13 Msps x2	Gigabit Ethernet	$950
BladeRF		300MHz - 3.8GHz	80 ksps - 40 Msps	USB 3.0	$420
Range Network SDR1		850MHz - 1.9GHz	8Msps	USB 2.0	

From the 上方 table, we can conclude that the USRP B210 is the most suitable one for a wide frequency range and relatively high sampling rate. For our experimental work we choose the platform from Ettus Research USRP. Ettus Research is a National Instrument (NI) company supplying software defined radio platforms, including the USRP family of products. The USRP products can work for RF applications from DC to 6GHz, including MIMO system, spectrum monitoring, radio networking and satellite navigation. In the following part, we will introduce two different USRP platforms: B210 and X310.

3.3.1 B210

The USRP B210 is a fully integrated USRP platform covering RF frequency from 70MHz to 6GHz which can be connected with host PC by USB 3.0. This platform enables experimentation with a wide range of signals including FM, cellular, Wi-Fi, and more.

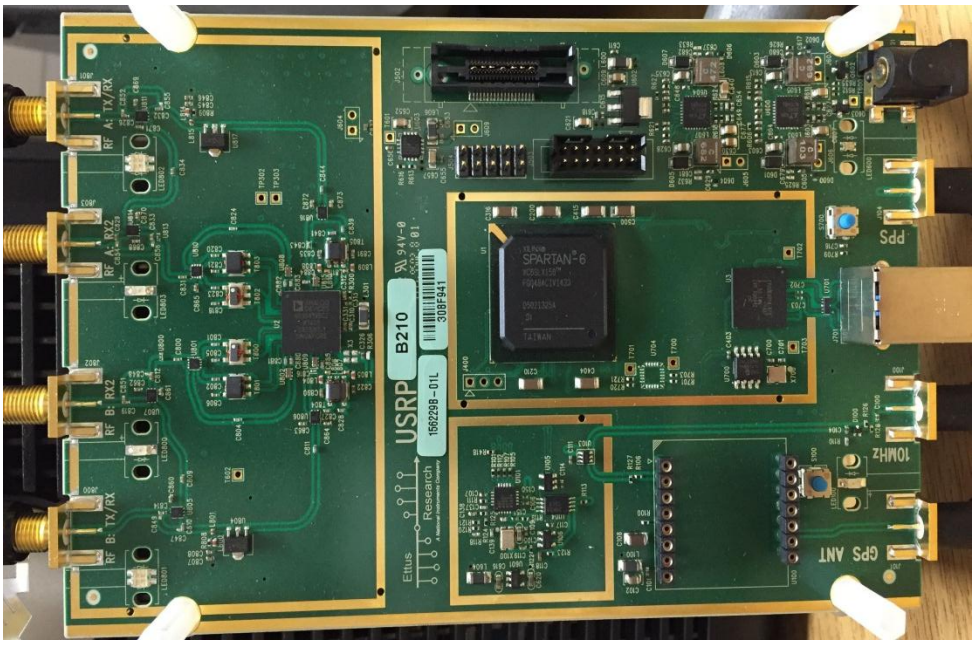

Figure 3.5 USRP B210 board

Figure 3.5 Shows the USRP B210 board, which integrated RF frontend with the new Analog Devices AD9361. It is capable if streaming up to 56 MHz of real-time RF bandwidth. On board signal processing and control of the AD9361 is performed by a Spartan6 XC6SLX150 FPGA connected to a host PC using USB 3.0. The USRP B210 real time throughput is 61.44MS/s, providing up to 56 MHz of instantaneous RF bandwidth in SISO.

The RF frontend of B210 is integrated on the board with 2 TX and 2 RX which can work half or full duplex. 2×2 MIMO is easy to realize when both transmit and receive are in use. For this case, the two receive frontends share the RX LO (local oscillator), and the two transmit frontends share the TX LO. Each LO is capable to be tuned from 50MHz to 60MHz.

Analog gain control is available for all the frontends. The gain of RX can be up to 73 dB and the TX can reach up to 89.5dB.

The frontend bandwidth (analog bandwidth) is seamlessly adjustable between 20 KHz and 56 MHz. If the current master clock rate is less than the bandwidth which set by users, the UHD will automatically configure the analog filter to avoid aliasing or out-of-band emissions.

3.3.2 X310

The Ettus Research USRP X310 is a high-performance SDR platform. It combines two extended bandwidth daughterboard slots which cover from DC to 6GHz. This board also contains multiple high-speed interference options, including PCIe (Peripheral Component Interconnect Express), Dual 1/10 GigE and a Kintex-7 FPGA with large programmable system integration.

1) RF daughter board

X310 board has two daughter board slots, in our project we choose UBX-160 which has a frequency range from 10 to 6GHz with up to 160 MHz bandwidth. It is a flexible transceiver and compatible with X310 Series board.

2) Host Interface

The USRP X310 provide three different interface options including 1 Gigabit Ethernet (1 Gigabit), 10 Gigabit Ethernet (10 Gigabit), and PCI-Express (PCIe). We choose 10 GigE in order to achieve the maximum throughput.

3) Assemble steps for X310 board

- Open the USRP X310 board;
- Align the 8 screw holes on the daughter board with the USRP Motherboard standoffs, then press the daughter board and put on screws;
- Connect the bulkhead cables and make sure connecting the TX/RX (RX2) on daughter board with TX/RX (RX2) on front panel of the X310, as the figure 下方;

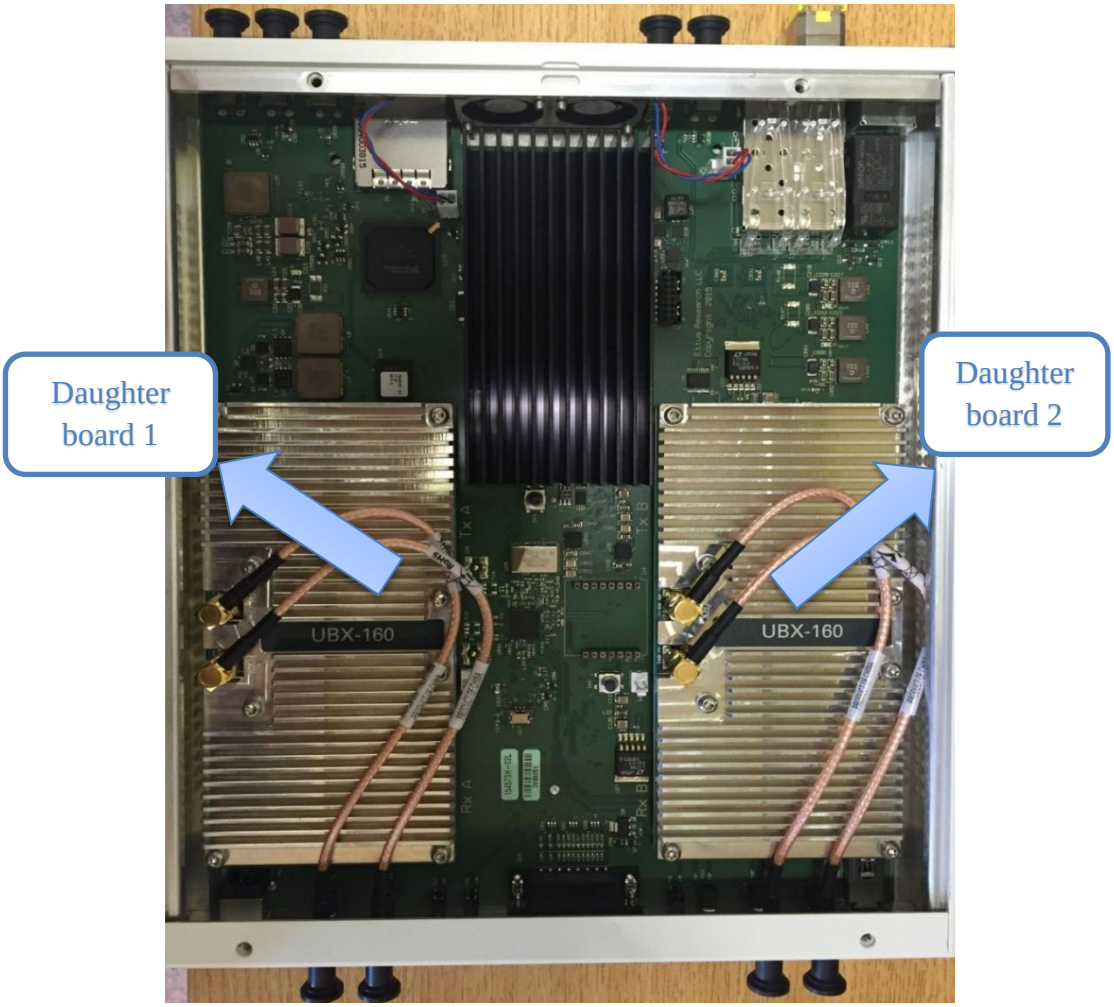

Figure 3.6 USRP X310 board

- Connect the SFP 1GigE adapter into USPR SFP port 0 as shown in figure 下方.

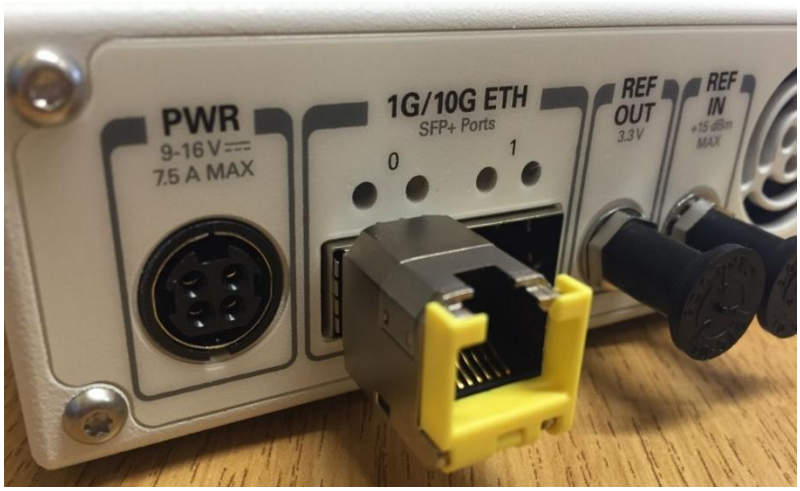

Figure 3.7 10 GigE adapter

4) UHD Installation

When using USRP X310 device, a version of UHD that supports the X310 family needs to be installed. In this project we use Linux system, the following steps show the building and installing of UHD from source. During this part, attention need be paid for make the UHD version compatible with GNU Radio companion.

- Install the dependencies on Linux: sudo apt-get install libboost-all-dev libusb-1.0-0-dev python-mako doxygen python-docutils cmake build-essential
- Getting the source code from Git repository: git clone git://github.com/EttusResearch/uhd.git
- Generate Makefiles with Cmake:
- cd uhd/host
- mkdir build
- cd build
- cmake ../
- Build and install:
- make
- make test
- sudo make install
- Set up the library path: sudo ldconfig

3.3.3 Compare B210 with X310

In this part we compare the different features between B210 and X310, as shown in the table below.

Table 3.3 Comparison between B210 and X310

Features	B210	X310
RF Coverage	70MHz-6GHz	DC-6GHz
Bandwidth	56MHz in 1×1	120MHz
	30.72MJHz in 2×2	
Interface	USB 3.0	Dual SFP(+) ports for 1/10 Giga bit
		PCIe x4
FPGA	Xilinx Spartan 6 XC6SLX150	Xilinx Kintex-7 XC7K410T
DAC/DAC Sample Rate	64.44MS/s	200MS/s
ADC Resolution	12	14
DAC Resolution	12	16
DC Input	6V	12V
Dimensions	9.7×15.5×1.5cm	27.7×21.8×3.9cm
Weight	350g	1.7kg
Price	890GBP	3880GBP

From the table, it is obviously that the features of X310 are better than B210 from bandwidth to ADC sample rate, but the B210 is more portable with integrated RF board and with small size. And the B210 is really a low-cost SDR device to do some basic experiments.

3.3.4 Bandwidth of USRP

When using the USRP board, we need to know the exact bandwidth of it, which will have benefits on better utilisation of the USRP board. Before introducing the bandwidth of USRP board, the architecture of USRP needs to be understood. In many cases, the RF frontend part translate the RF domain signals to baseband or IF signals. Then the signals are sampled by ADCs (Analog to Digital Converters) which are locked into an FPGA. The FPGA provides digital down conversion which known as decimation. After this step, samples are streamed into a host computer through the host interface. The reverse path is transmitting chain. The general USRP architecture is shown in Figure 3.8.

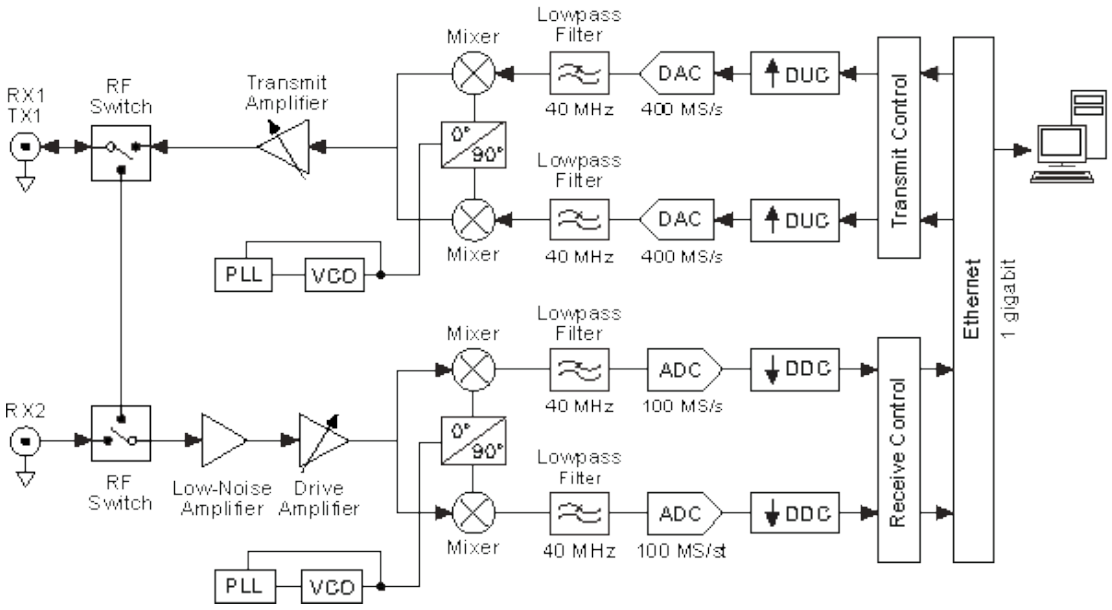

Figure 3.8 The general USRP architecture [55]

The bandwidth of USRP varies in different part of the signal chain. There are three bandwidth types including the analog bandwidth, the FPGA processing bandwidth and host bandwidth. And the system bandwidth is decided by the minimum of the three.

1) Analog bandwidth

The analog bandwidth is the 3dB bandwidth of the RF port and baseband/IF interface of an RF channel which is set by baseband/IF filters on the daughterboard. These

filters are used for avoid aliasing under the given ADC/DAC sample rates in motherboard.

2) FPGA processing bandwidth

The FPGA processing bandwidth is the sample rate of the ADCs and DACs on the USRP motherboard. This is the maximum digital bandwidth of the USRP-based system. For example, the FPGA of the URSP B210 processing samples at 61.44Ms/S from both ADCs and DACs.

3) Host bandwidth

The host bandwidth is defined by the host interface between the host PC and the FPGA of a USRP device. There are various interface options for the USRP devices. For example, the B210 uses USB 3.0 interface to transport stream in half duplex mode at 61.44 MS/s. Which means the interface can only stream in one direction for transmit or receive.

3.4 Summary

In this chapter, we introduced the software and hardware platform to implement spectrum sensing. For the software platform we choose GNU Radio and give a brief introduction of it. While for the hardware platform, we choose USRP B210. This chapter also provides different choices of SDR hardware platforms to choose from and compare the difference between two kinds of USRP toolkit (B210 and X310). Finally, we explain and define the bandwidth of a USRP board.

4 Dynamic Spectrum Access

4.1 Introduction

Dynamic Spectrum Access (DSA) is a novel spectrum management paradigm which gives secondary users an access to the abundant spectrum holes or white space in the licensed spectrum bands. It is also a promising technology to relieve the scarcity of spectrum and increase spectrum efficiency [5]. The key enabling technology of dynamic spectrum access is Cognitive Radio (CR) technology, which provides the capability to share the wireless channel with licensed users in an opportunistic manner. This capability can be realized by spectrum management, which includes four main categories: spectrum sensing, spectrum decision, spectrum sharing, and spectrum mobility [56]. Figure 4.1 shows the spectrum holes concept and how Dynamic spectrum Access works.

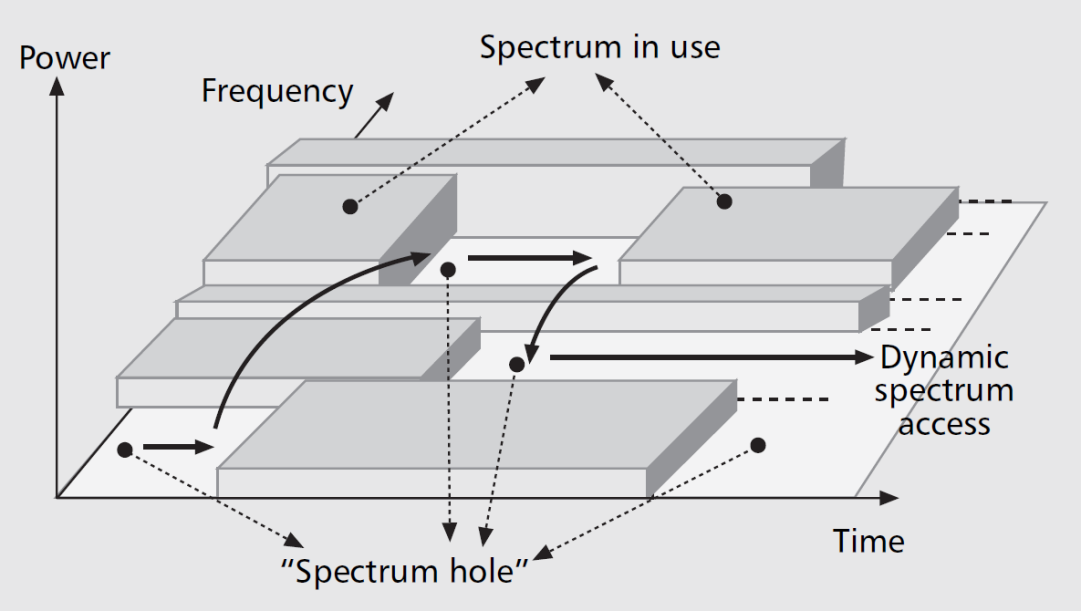

Figure 4.1 Dynamic Spectrum Access [56]

When the number of SUs is more than one, cooperative sensing can improve spectrum sensing reliability [57]. There are two different decision types of cooperative sensing: soft fusion decision and hard fusion decision [58]. The first one requires all SUs to send all their sensing data which consumes high network overhead. However, the second one only requires 1bit decision from a SU which needs low network overhead.

The hard fusion decision includes OR-rule and AND-rule. The probability of detection for OR-rule can be written as follows [57]:

$$P_{d,coop(OR)} = 1 - \prod_{i=0}^{N}(1 - P_{d,i}) \tag{4.1}$$

Where $P_{d,i}$ is the probability of detection for the i-th SU, and N is the number of cooperating SUs. Figure 4.2 illustrates the cooperative sensing model. A group of spatially distributed sensors observe phenomenon Hi through observations y_i, and then report their observations u_i to a central processor known as a Fusion Centre (FC) [59]. The FC combines the local decisions u_i by data fusion and makes cooperative decision u.

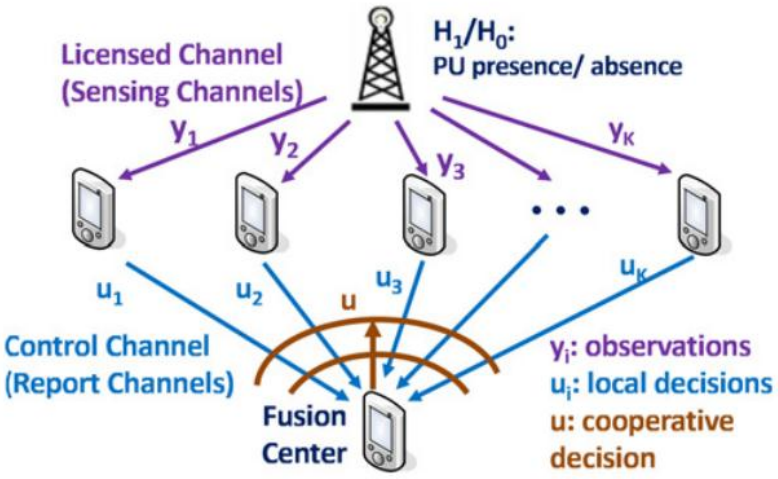

Figure 4.2 Cooperative sensing model [60]

Another important part of the DSA is spectrum mobility which will have an impact on the detection performance and how SUs collaborate in cooperative sensing. As shown in Figure 4.3, when CR1 moves from location A to location B, it creates the spatial diversity in the observations of PU1 by mobility. On the other hand, if CR1 continues to move to location C and CR3 moves from location D to E, CR1 and CR3 may be correlated for being close to each other. Therefore, mobility may improve the achievable cooperative gain [60].

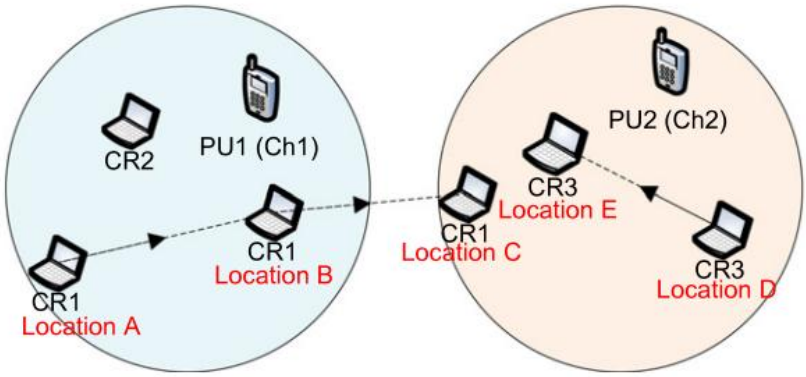

Figure 4.3 The mobility of SUs [60]

In this thesis, we are mainly concerned about the spectrum sensing part which is the basic part of Dynamic Spectrum Sensing. We conducted several experiments about spectrum sensing based on energy detection, sequential energy detection and maximum-minimum eigenvalue.

4.2 System model

The system considered in this project is composed of a single primary user and a single secondary user. The secondary user applies different spectrum sensing methods to detect whether the primary user is using the transmit channels. Assuming the traffic pattern of primary user is changing slowly; the second user will have enough time to finish the entire detection process before the primary user is changing its transmission state.

Detection of primary's presence can be treated as a binary hypothesis testing problem [61], where the state of primary user is expressed as:

$$H_0 : y[n] = w[n]$$

$$H_1 : y[n] = S[n] + w[n] \tag{4.2}$$

$$n = 1, \dots, N$$

Where $S[n]$ represents the digitally modulated signal of primary user, $w[n]$ corresponds to the AWGN with zero mean and variance σ^2, $y[n]$ represents the

sampled signal of the primary signal. Then $y[n]$ is received by second user, and N is the length of the observation interval to carry out the detection process.

4.3 Spectrum sensing

The objective of spectrum sensing is to decide between null hypothesis H_0 and occupied hypothesis H_1 according to the received signal $y[n]$. This can be done using a statistic test $\Lambda(y)$ of the received signal $y[n]$, and comparing it with a predetermined threshold η. The aim of this comparison is to know accurately whether the channel is available.

$$\begin{cases} \Lambda(y) > \eta & H_1 \\ \Lambda(y) < \eta & H_0 \end{cases} \tag{4.3}$$

The performance of detection is characterized by two probabilities:

- Probability of detection (P_d): Probability of an existing signal occupying the channel (H_1) and second user detects it correctly.

$$P_d = \mathrm{P}_r(\Lambda(y) > \eta | H_1) \tag{4.4}$$

- Probability of false alarm (P_{fa}): Probability of no primary signal occupying the channel (H_0) but second user wrongly detects as signal existing.

$$P_{fa} = \mathrm{P}_r(\Lambda(y) > \eta | H_0) \tag{4.5}$$

The performance of a signal detector is qualified by Receiver Operating Characteristics curve (RCO), which shows the relationship among P_{fa}, P_d, and SNR. In spectrum sensing, there are two options [62]:

a) The SNR keeps constant, while the threshold value η varies on the RCO curve.
b) For a given P_{fa}, plotting the relation between P_{fa} and P_d.

The spectrum sensing techniques are a recent development. There are different approaches recommended for detecting signal presence in transmissions system. The different methodologies can be classified from whether knowing about the character of primary signal, as shown in Figure 4.4.

- Knowledge-aware technique: it requires both primary user signal and noise variance information;
- Semi-blind technique: it needs only a few parameters, for example the variance of additive noise or the fundamental cyclic frequency.
- Blind technique: it exploits the received signals without any PU signal information or noise information.

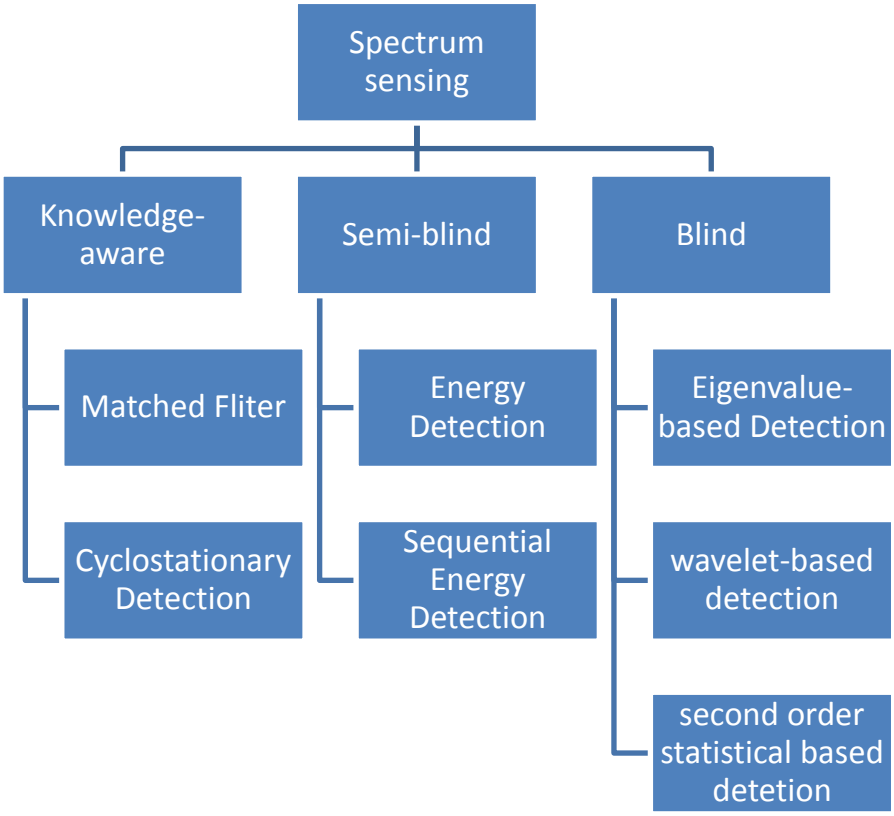

Figure 4.4 Classification of Spectrum Sensing

4.3.1 Matched Filter Detection

The matched filter detection based sensing uses a linear filter which is used in digital signal processing. Obviously, for matched filter based spectrum sensing needs a completely knowledge of primary user signal, for example, modulation format, waveform, carry frequency, pulse shape, etc. The matched filter detection is a transmit signal $s(t)$ defined over $0 \leq t \leq T$ maximizing the SNR at the output of filter. The matched filter can be described as follows:

$$h(t) = \begin{cases} s(T-t) & 0 \le t \le T \\ 0 & else \end{cases} \tag{4.6}$$

Equation (4.6) depicts the block diagram of matched filter based detection of primary user. After passing the matched filter, the test statistic $\xi(nT)$ is required to be compared with the threshold η in every $t = nT$ period to decide whether there is a primary user existing or not. The detector can be defined as:

$$d(nT) = \begin{cases} 0 & \xi(nT) \le \eta \\ 1 & \xi(nT) \ge \eta \end{cases} \tag{4.7}$$

Figure 4.5 Block diagram of matched filter detection

The matched filter detection can get a better detection probability compared with blind spectrum sensing techniques. However, the shortage is as mentioned before, it requires complete signal information of primary users, which is difficult to get in most practical scenarios.

4.3.2 Energy Detection

The energy detection technique is the most prevalent and simple way to make spectrum sensing. It uses the energy level of interest band to decide if a primary user is existing or not. And energy level is compared with a detection threshold which can be measured before comparison. The process of energy detection is illustrated in Figure 4.6.

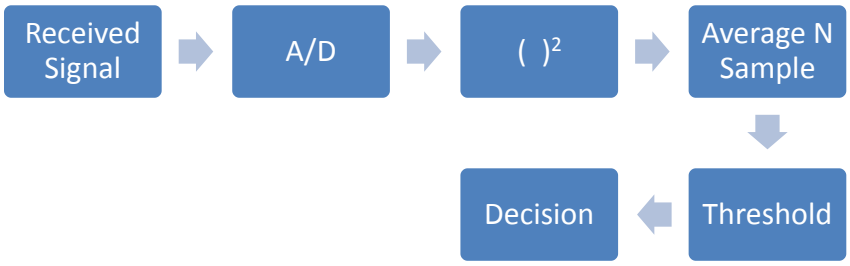

Figure 4.6 General structure of energy detection

The decision statistics for energy detection is [63]:

$$T = \sum_{n=1}^{N} (Y[n])^2 \tag{4.8}$$

When there is no primary signal, the decision statistic has a central chi square distribution with N degrees of freedom. While the primary signal is present, the decision statistic has a non-central chi square distribution with the same degrees of freedom [61]. So the generic decision rule D_N can be modified as following case:

$$D_N = \begin{cases} P_N \leq \eta & H_0 \\ P_N \geq \eta & H_1 \end{cases} \tag{4.9}$$

Considering the number of samples N used for energy detection is large enough (N > 500), the distribution of test statistic can be seen as Gaussian. The mean and variance of it are stated in (4.10). The variance of noise process is ${\sigma_w}^2$ and the power of primary signal is ${\sigma_s}^2$.

$$\begin{aligned} T &\sim N(N{\sigma_w}^2, 2N{\sigma_w}^4) \quad H_0 \\ T &\sim N\left(N({\sigma_w}^2 + {\sigma_s}^2), 2(N{\sigma_w}^2 + {\sigma_s}^2)^2\right) \quad H_1 \end{aligned} \tag{4.10}$$

Following [64], there are two types of detection errors: the first is H_0 be falsely rejected, which is called false alarm, the second is misdetection when H_1 is falsely accepted. The performance if this technique can be measured by these two probabilities. In this thesis we consider the probability of detection (P_d) instead of misdetection probability (P_{md}).The definition of P_{fa} and P_d are illustrated in (4.11) and (4.12).

Supposing $\Lambda(y)$ can be approximately considered as Gaussian distribution, the P_{fa} and P_d can be calculated by:

$$P_{fa} = Q\left(\frac{\eta - N{\sigma_w}^2}{\sqrt{2N{\sigma_w}^4}}\right) \tag{4.11}$$

$$P_d = Q\left(\frac{\eta - N{\sigma_w}^2}{\sqrt{2N(N{\sigma_w}^2 + {\sigma_s}^2)^2}}\right) \tag{4.12}$$

Where $Q(\,)$ represents the Gaussian Q- function. There is a trade-off between P_{fa} and P_d, as an improvement on P_d will increase P_{fa}, and vice versa. The Neyman-Pearson criterion creates a bond on the P_{fa} and maximize the P_d [64]. It is called Constant False Alarm Rate (CFAR).From Equation (4.11) the threshold can be written as:

$$\eta = {\sigma_w}^2 \left(Q^{-1}(P_{fa})\sqrt{2N} + N\right) \tag{4.13}$$

4.3.3 Sequential Energy Detection

In traditional energy detection, each second user makes decision only depending on one single threshold, in which the decision H_0 and H_1 represent the absence and presence of primary users respectively. In practise, the detector is periodically sensing the channel during data transmission to identify whether the primary signal is sent. In this situation, the sensing time should also be minimized because the SU must detect the PU signal rapidly and then abandon the channel. Otherwise, it will interfere with the PU.

To reduce the sensing time, two thresholds based method can be used which called Sequential Energy Detection (SED) [65]. It can also mitigate the spectrum sensing failure problem by concerning the detected energy which falls in between the threshold η_1 and η_2. The energy distribution of signal and noise about primary users is shown in Figure 4.7.

In this figure, we can see there is a confused region which makes it difficult to distinguish H_1 and H_0. Therefore, it is hard to make a single threshold to detect whether there is a primary user existing. To overcome this problem, we need to use the double-threshold detection.

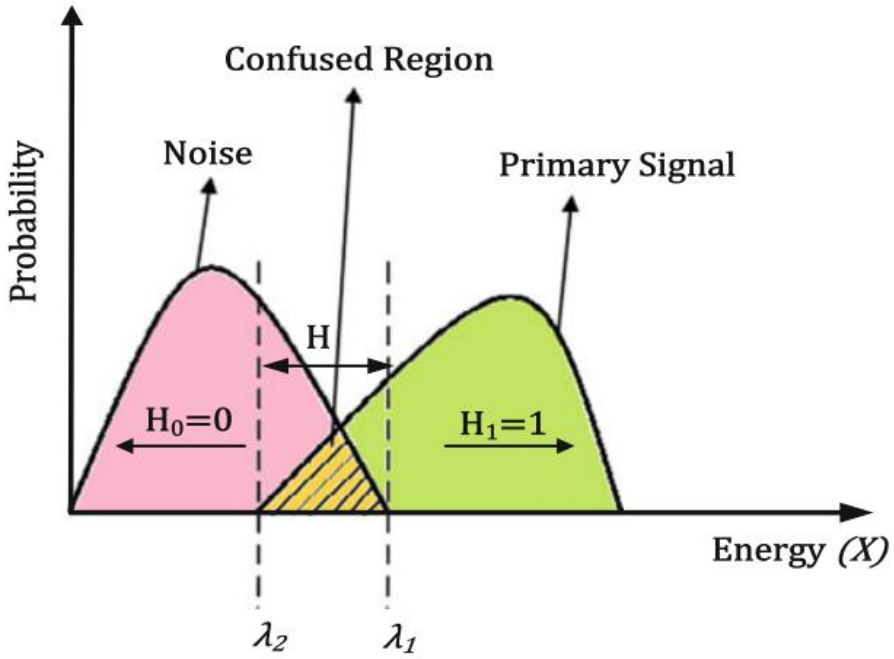

Figure 4.7 Energy distribution of signal and noise about primary users [66]

The detailed sequential energy spectrum sensing scheme is as follows:

- There are two thresholds in this method η_1 andη_2, which are used for making decision. If the signal is present or not in the sensing frequency band. The decision can be made by the following rule:

$$D_N^i = \begin{cases} P_N^i \leq \eta_1 & H_0 \\ P_N^i \in (\eta_1, \eta_2) & continue \\ P_N^i \geq \eta_2 & H_1 \end{cases} \quad (4.14)$$

  By using this method, the energy detector can collect less signal samples in a period $N_S < N$ and then make decisions. If the detected energy P_N^i is larger than η_2, it means the primary user is presence. If the received power is lower than η_1, the decision of no primary user can be made. If the calculated value is falling between these two thresholds, the local decision is not made as the secondary user is uncertain about its observation, this observation result will be sent to fusion centre directly.

- Then, the local decision or observation result is sent to the fusion centre.

- At the fusion centre, all the observation results (O_i) are aggregated and compared to the fusion threshold η_f to make a fusion decision. The fusion threshold can be calculated by the conventional energy detector algorithm in Equation (4.14).

$$fusion\ decision = \begin{cases} \eta_1 & o_i < \eta_f \\ \eta_2 & otherwise \end{cases} \quad (4.15)$$

- Finally, all the local decision (D_N^i) and the fusion decision are combined using the OR-rule to make a final decision.

$$final\ decision = \begin{cases} \eta_1 & if\ all\ D_N^i\ and\ fusion\ decision\ are\ \eta_1 \\ \eta_2 & otherwise \end{cases} \quad (4.16)$$

4.3.4 Eigenvalue-based Detection

Eigenvalue-based detection is proposed based on the eigenvalues of the covariance matrix of the received signal. In energy detection, it is important to obtain the threshold accurately, which needs to estimate the noise power. In practice, it is very difficult to obtain the accurate value due to the noise power uncertainty. The eigenvalue-based method is used to overcome this shortage.

In this part we discuss two spectrum sensing methods based on the distribution of eigenvalue in large dimensional random matrix theory [67]. According to Equation (4.17), under H_1 the received signal can be expressed as:

$$y[n] = \sum_{k=0}^{N_h} h[k]s[n-k] + w[n] \quad (4.17)$$

Where N_h is the channel filter length. At the second user's receiver, the received samples can be expressed as a matrix Y, which can be spilt into M vectors with each of length N_s.

$$Y = \begin{bmatrix} y_1[1] & y_1[2] & \cdots & y_1[N_S] \\ y_2[1] & y_2[2] & \cdots & y_2[N_S] \\ \vdots & \vdots & \ddots & \vdots \\ y_M[1] & y_M[2] & \cdots & y_M[N_S] \end{bmatrix} \tag{4.18}$$

When the primary user is absent (under H_0), all the received samples are uncorrelated no matter in which kind of fading channel model. Theoretically, the non-diagonal element of the received covariance matrix is zero, while the diagonal elements include the noise variance. Therefore, the covariance matrix of the receive signal converges to a true covariance matrix.

$$\frac{1}{N_s} YY^* \rightarrow \sigma^2 \mathbf{I}_M, \ \ when\ M, N_s \rightarrow \infty \tag{4.19}$$

As we have assumed before in system model:

- The noise is white Gaussian noise.
- The noise and transmitted signal are uncorrelated.

Let $\mathbf{R}(N_s)$ be the received signal's covariance matrix

$$\mathbf{R}(N_s) = \frac{1}{N_s} \sum_{n=L}^{L-1+N_s} Y(n) Y^*(n) \tag{4.20}$$

Where N_s is the number of collected samples. If $N_s$is large enough, the Equation (4.20) can be shown as

$$\mathbf{R}(N_s) \approx \mathbf{R} = \mathbf{E}\left[\frac{1}{N_s} YY^*\right] = \mathbf{H}\mathbf{R_s}\mathbf{H}^* + \sigma^2 \mathbf{I}_M \tag{4.21}$$

Where $\mathbf{R_s}$ is the covariance matrix of transmitted signal. $\sigma^2$is the noise variance. And $\mathbf{I}_M$is the identity matrix of order MN_s.

Let λ_{min} and λ_{max} be the minimum and maximum eigenvalue of **R**, and suppose that, ρ_{min} and $\rho_{max}$are the minimum and maximum eigenvalue of $\mathbf{H}\mathbf{R_s}\mathbf{H}^*$,then

$$\lambda_{max} = \rho_{max} + \sigma^2$$
$$\lambda_{min} = \rho_{min} + \sigma^2 \tag{4.22}$$

Obviously, only if $\mathbf{HR_sH}^* = \delta\mathbf{I}_M$ then $\rho_{min} = \rho_{max}$, where δ is a positive integer. In practice, due to dispersive channel and/or oversampling and/or correlation among the signal samples, it is very unlikely $\mathbf{HR_sH}^* = \delta\mathbf{I}_M$, when signals are present [23]. So we can define whether the primary signal is present by the following equation:

$$\begin{cases} \frac{\lambda_{max}}{\lambda_{min}} = 1 \;\; H_0 \\ \frac{\lambda_{max}}{\lambda_{min}} > 1 \;\; H_1 \end{cases} \tag{4.23}$$

There are two eigenvalue based methods in literature.

I. Maximum-Minimum Eigenvalue (MME) detection method

- The steps of this algorithm are as following:
  - Calculate the received signal covariance matrix $\mathbf{R}(N_s)$, as shown in equation 5-11.
  - Compute the minimum (λ_{min}) and maximum (λ_{max}) eigenvalue of the covariance matrix $\mathbf{R}(N_s)$.
  - Make decision if signal existing:

$$D = \begin{cases} H_1 & if \frac{\lambda_{max}}{\lambda_{min}} \geq \eta_{MME} \\ H_0 & otherwise \end{cases} \tag{4.24}$$

- Where η_{MME} is the threshold of the MME detection method, Which is computed from the following equation.

$$\eta_{MME} = \frac{(\sqrt{N_s} + \sqrt{MN_s})^2}{(\sqrt{N_s} - \sqrt{MN_s})^2} \times \left(1 + \frac{\left(\sqrt{N_s} + \sqrt{MN_s}\right)^{-\frac{2}{3}}}{\left(MN_s{}^2\right)} \times F_1^{-1}(1 - P_{fa})\right) \tag{4.25}$$

- Where F_1 presents the cumulative distribution function (CDF) of the Tracy-Widom distribution of order 1. If the signal is complex, the CDF is F_2 of order 2. It is shown that when there is no primary signal at the receiver, the received signal's covariance matrix can be defined as a special Wishart random matrix. There is joint expression about the probability density function of this matrix, but it is too complicated. As shown in [23], the distribution of complex Wishart matrix's largest eigenvalue converges to the Tracy-Widom limiting law.

II. Energy with Minimum Eigenvalue (EME) detection method

- The steps of this algorithm are as following:
  - Calculate the received signal covariance matrix $\mathbf{R}(N_s)$, as shown in equation 5-11.
  - Calculate the average power of the received samples

$$\Gamma(N_s) = \frac{1}{MN_s}\sum_{i=1}^{M}\sum_{n=0}^{N_S-1}|y_i[n]|^2 \tag{4.26}$$

  - Obtain the minimum eigenvalue of $\mathbf{R}(N_s)$.
  - Make decision by

$$\frac{\Gamma(N_s)}{\lambda_{min}} > \eta_{EME} \tag{4.27}$$

Where η_{EME} is the threshold of the EME method.

$$\eta_{EME} = \left(\sqrt{\frac{2}{MN_s}}Q^{-1}(P_{fa}) + 1\right)\frac{N_s}{\delta^2} \tag{4.28}$$

Where $Q(.)$ is the Gaussian Q-function, and P_{fa} is the probability of false alarm.

In the above discussion, it is obviously that the threshold η_{EME} does not only depend on the probability of false alarm, but also the noise variance, the number of vector and the number of samples per vector.

4.4 Summary

In this chapter, we first introduced the system model of spectrum sensing system. Then referred to different kinds of spectrum sensing methods according to different classifications: knowledge-aware, semi-blind and blind spectrum sensing methods. The implementation steps are also presented for the experiments which will be shown in next chapter.

5 Implementation of DSA with USRP B210 and GRC

5.1 Introduction

In this chapter, the main components of the test bed which are used to implement the spectrum sensing are presented in detail. The test bed can be divided into two parts, software and hardware. The hardware part is used to implement the RF part and sampling process, while the physical layer of the spectrum sensing system is in software which runs in a computer.

The software part of the test bed is based on GNU Radio Companion (version 3.7.4), and, all the blind spectrum sensing algorithms i.e. energy detection, sequential energy detection and eigenvalue based detection are implemented in GNU Radio Companion. For the hardware part, we use one USRP B210 board by Ettus Research as the radio frequency interface and two wide band antennas to perform the spectrum sensing test. The transmitter antenna and the receiver antenna are the same antennas (ImmersionRC 5.8G); the frequency band of the antennas is 2.36 – 2.44GHz and 5.75 - 6.02GHz. All the signal processing takes place on a Core i3 desktop computer. This computer hosts a SDR Live USB which is a bootable USB flash driver with Ubuntu Linux 14.10, GNU Radio, Universal Hardware Driver (UHD) and other relative software installed on it. When implementing signal processing, the GNU Radio Companion uses UHD to send or receive signals to or from the USRP B210 platform via USB 3.0 link. The basic spectrum sensing scheme is shown in Figure 5.1.

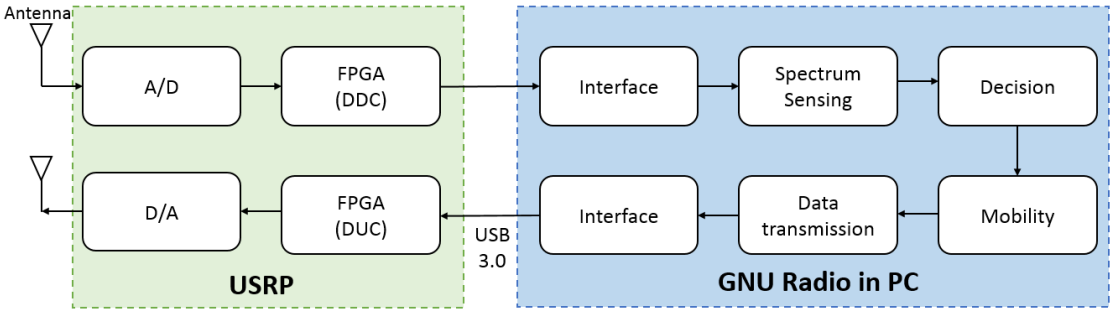

Figure 5.1 A basic Spectrum Sensing scheme

In this project, two channels of the USRP B210 board have been utilized: one TX channel (*RF A: TX/RX*) is connected with an antenna to generate the PU signal, while

another RX (*RF A: RX2*) channel is connected with an antenna for spectrum sensing purpose and used as the SU. The whole system setup is shown in Figure 5.2.

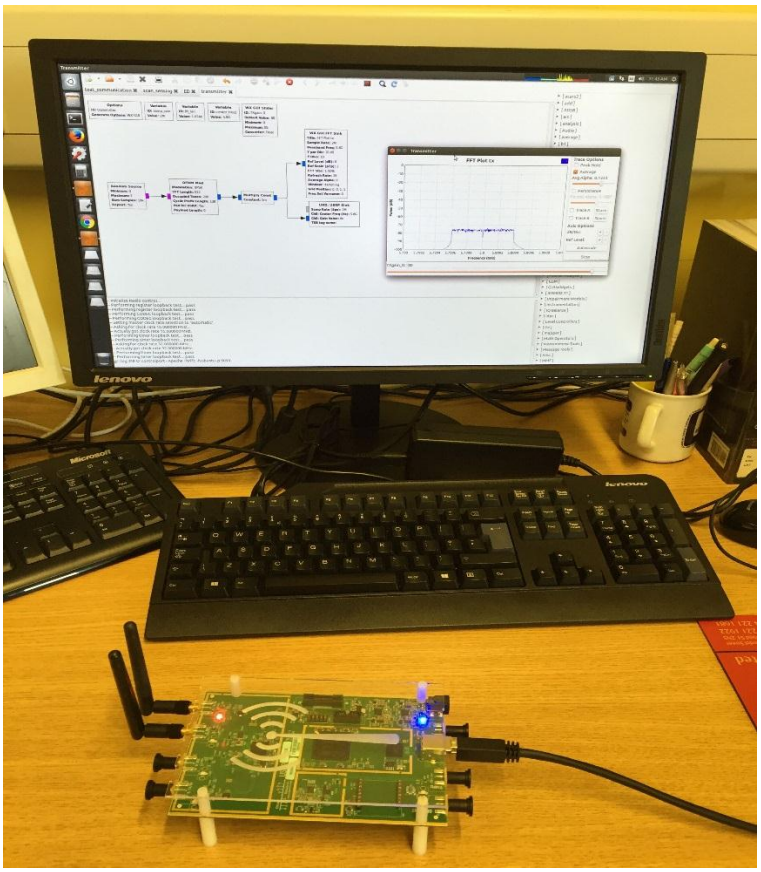

Figure 5.2 The experimental setup

5.2 Spectrum sensing experimental setup

5.2.1 Transmitter side (PU)

At the transmitter side, two types of signals are generated, narrowband QPSK signal and wideband multicarrier signal based on OFDM. In the first case, the PU signal has been created and modulated before sending to the USRP board. The bandwidth of the QPSK signal is narrow about 144 KHz. However, the bandwidth of the multicarrier signal is wider than the previous one, as can be seen from Figure 5.3, in the OFDM modulation block (number ①), there are N_{ofdm} = 512 subcarriers with 128 Cycic Prefix length, and the bandwidth is set as 1MHz. which is half of the samp_rate (number ②). As it has been mentioned before, all the baseband signal processing is implemented in GNU Radio environment on computer, especially in the GNU Radio

Companion – a graphical tool, where the whole system is built by blocks. Figure 5.3 illustrates the OFDM transmitter side; it is a screen shot from GRC.

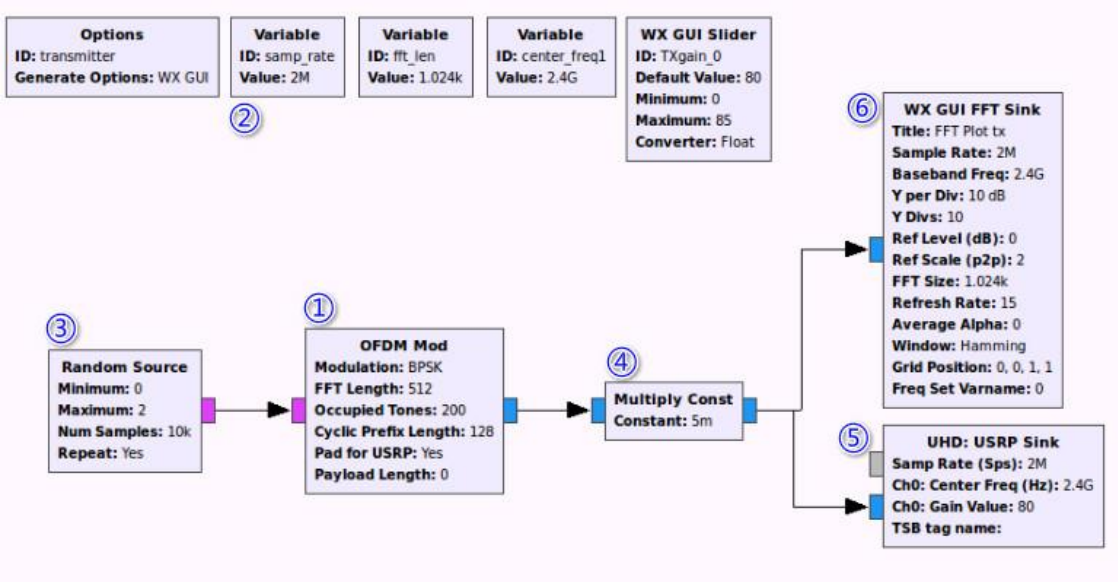

Figure 5.3 Flowgraph of the OFDM transmitter in GRC

In Figure 5.3, there is a *Random Source* (number ③) used as the signal source which generates repeated random data. These data is then mapped to BPSK symbols and are subjected to OFDM modulation, both of them are done in *OFDM Mod* block (number ①). From this block, it can be seen that only 200 subcarriers (*Occupied Tones*) from 512 have been occupied; and the length of the Cycic Prefix is set as 25% of the FFT length which equals to 128. Finally, the signal is multiplied by a constant in *Multiply Const* (number ④) to change the power of the signal, before sent to the *UHD: USRP Sink* (number ⑤) that is responsible to send signal from GRC to USRP board. There is also a *WX GUI FFT Sink* (number ⑥) been used that can display the receive signal in the frequency domain on a computer. One can see from the variable box, the sample rate is set to 2MHz and the centre frequency of the transmit signal is 2.4GHz in channel 0. This frequency is chosen in one of the free ISM (Industrial Scientific Medical) Band frequency range; we can easily change the centre frequency of the transmitted signal in the FR coverage range of both antennas and USRP B210 board. In this scenario, we can change the centre frequency from 2.36 to 2.44GHz. As shown in Figure 5.4, we can set the centre frequency as *center_freq1* in *UHD: USRP Sink*

block which can transmit the baseband signal. The *center_freq1* is a variable name that is set as 2.4GHz as shown in Figure 5.3.

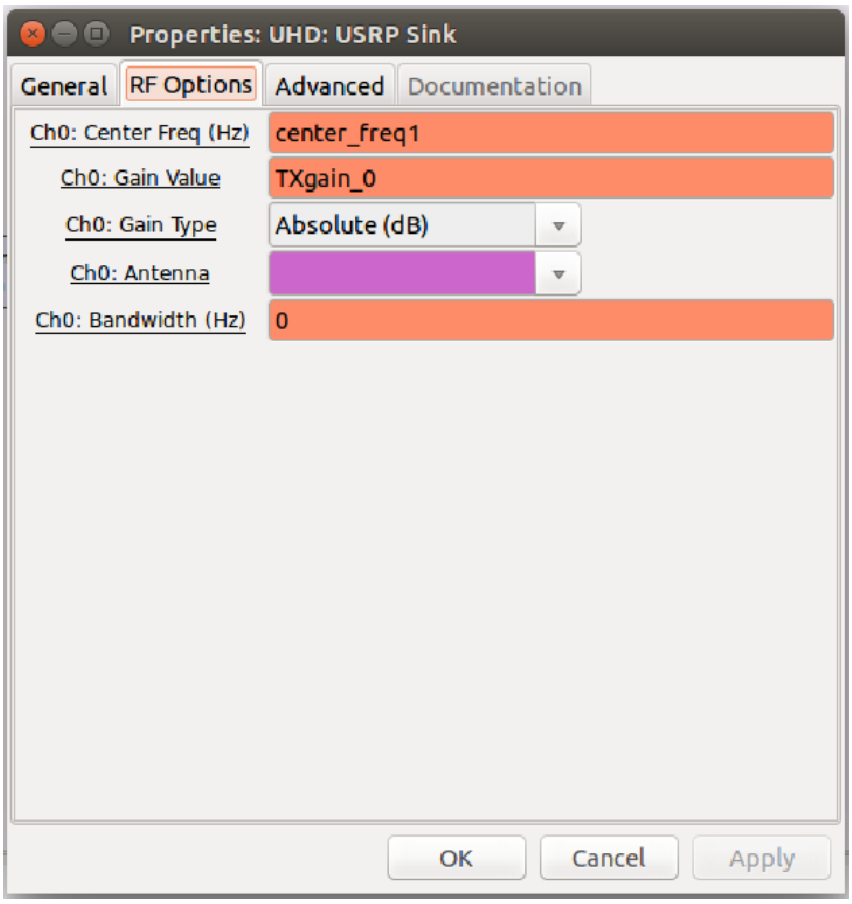

Figure 5.4 UHD: USRP Sink properties setting

Figure 5.5 illustrates the graphical result in the *UHD: USRP Sink* in frequency domain. On the right hand of the figure, there are some options to set the performance of the signal. In most cases, we will choose *Average* in Trace Options and *Autoscale* to display the result signal. When Averaging enabled, there is an effect of smoothing the noise floor, which makes us figure the signal more easily. In this figure, the energy of the signal is around -75dB and the noise is about -105dB. The bandwidth of the signal is about 1MHz.

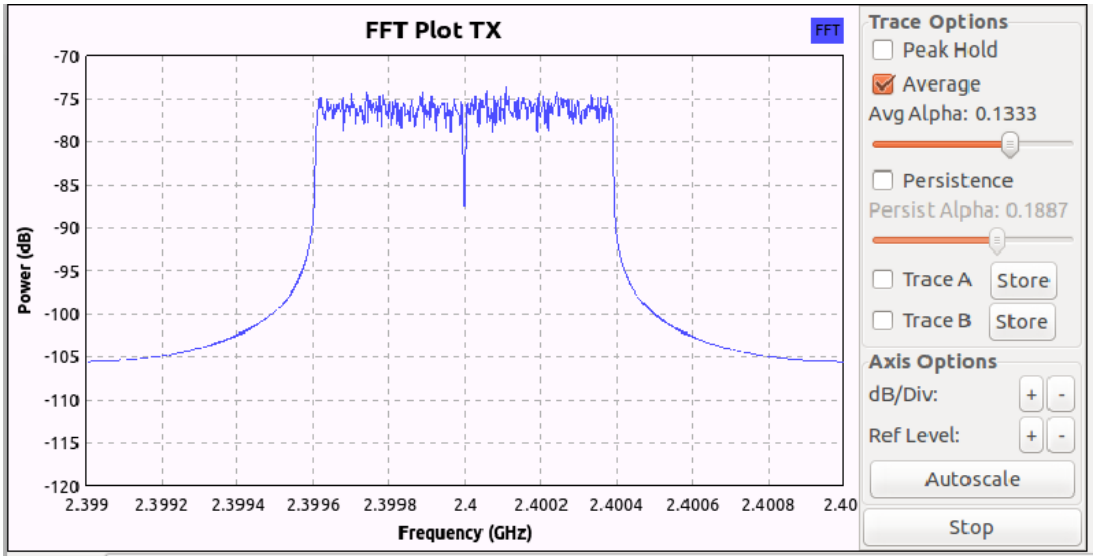

Figure 5.5 The OFDM transmitter signal in GRC

5.2.2 Receiver side (SU)

As mentioned in the introduction of this chapter, three spectrum sensing methods have been implemented. The first one is the conventional algorithm energy detection, the second one is an improvement method of the first one which uses the sequential feature of the signal, and the final one is the eigenvalue based method to detect the received signal. Analogously to the transmitter part, all the baseband receiving signal processing is implemented on the computer in GRC. The flowgraph of the receiver part is illustrated in Figure 5.6. The receiver signal in *UHD: USRP Source* displayed in the *FFT Sink* is shown in Figure 5.7. Form this figure; we can see that signal power is nearly -62 dB and the noise Power is about -90dB. Compared with Figure 5.5, the noise power increases from -105dB to -90dB, which is caused by the noise in the transmit channel. The signal power also increases 13dB (from -75dB to -62dB); because we set the gains in the *UHD: USRP Source* and *UHD: USRP Sink* as 40 and 60 respectively.

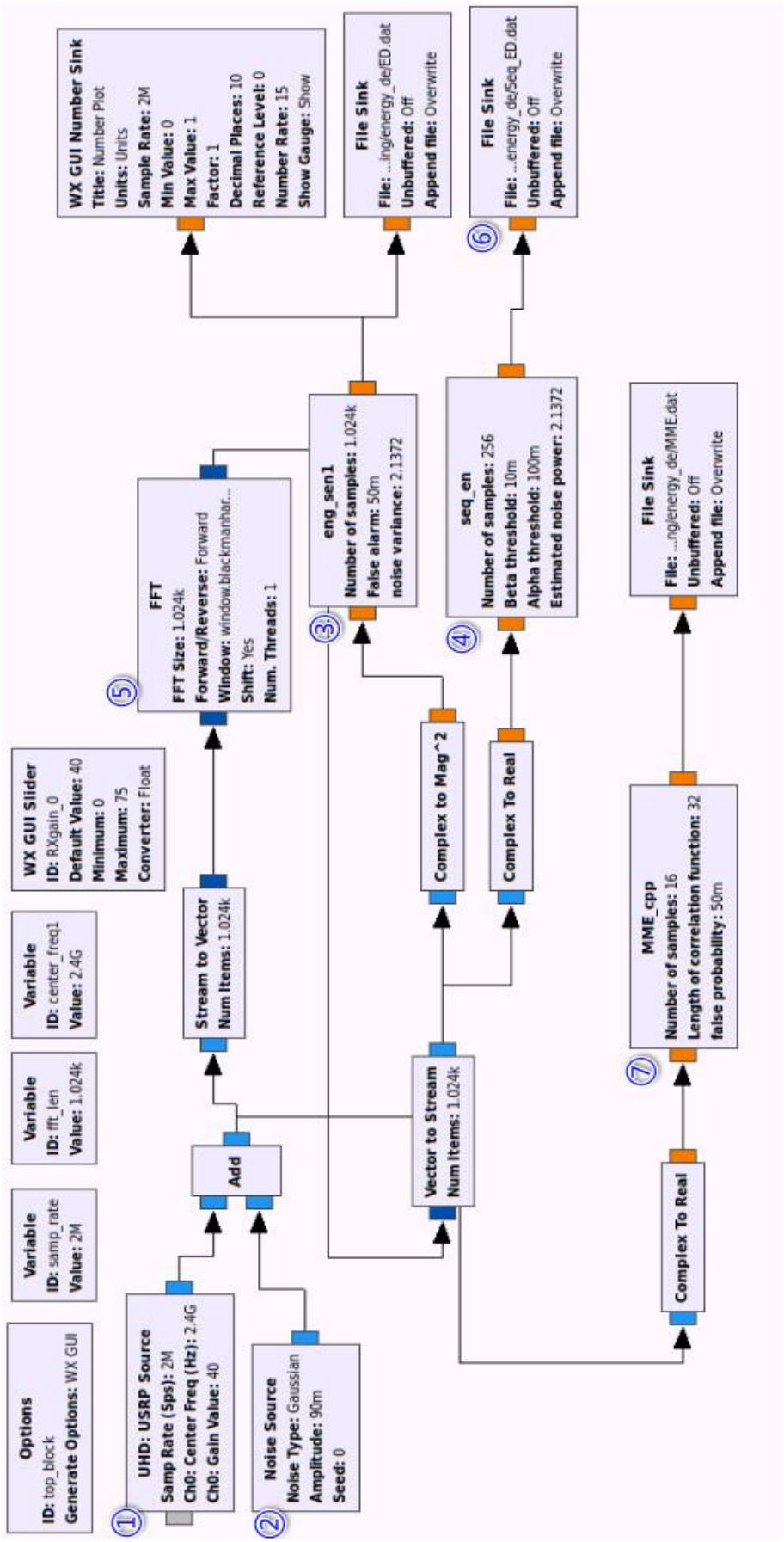

Figure 5.6 Flowgraph of the Spectrum Sensing Receiver in GRC

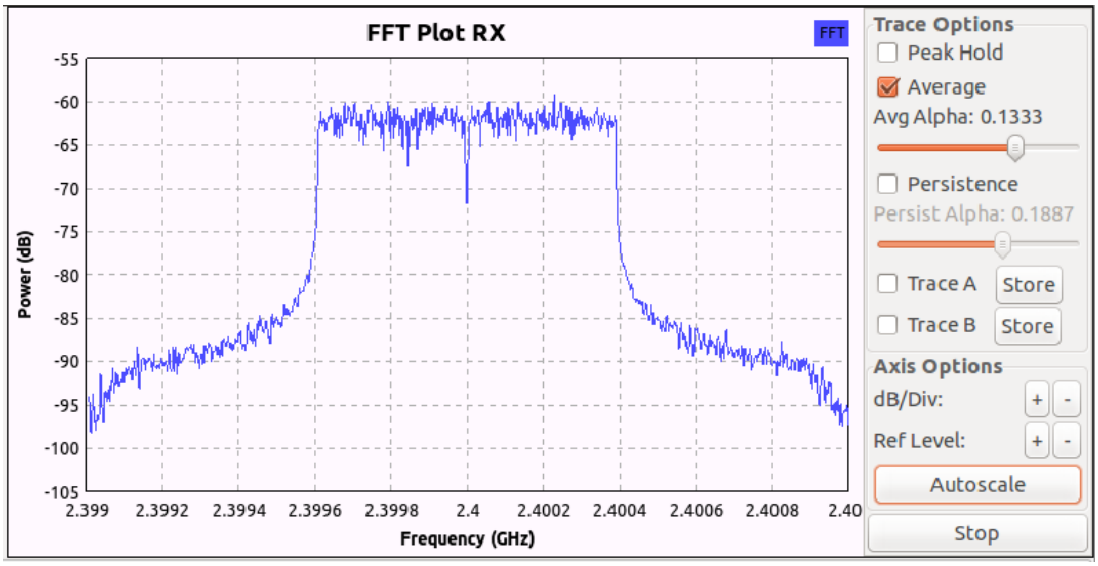

Figure 5.7 The OFDM receiver signal in GRC

We can observe that the *UHD: USRP Source* block (number ①) is responsible for transferring signal from RF spectrum to computer processing. It is shown in the block; the centre frequency of the receiver is the same as transmitter at 2.4GHz and the present bandwidth equal to half of the sample rate (1MHz).

In order to test the performance of spectrum sensing algorithms under different noise environment (SNR), a *Noise Source* block (number ②) is added to the signal source as shown in Figure 5.6. Before implementing spectrum sensing, the signal is divided into three parallel paths. The upper two paths are energy detection (number ③) and sequential energy detection (number ④), they both need to use *FFT* block (number ⑤) to transfer signal from time domain to frequency domain. For the sequential energy detection, it makes decision every 256 samples and then sends the decision values to a *File Sink* (number ⑥) to store the results. In the last path, it devotes for the eigenvalue based spectrum sensing algorithm. After converting the complex signal to real type, the signal is sent to MME blocks (number ⑦) to realize the functionality which has been referred in the previous chapter (Chapter 4.3.4).

5.3 Implementation spectrum sensing modules with GNU Radio platform

5.3.1 The structure of an OOT module

In chapter 3.2.5, there is a brief introduction about how to build an OOT module in GRC and the structure of the OOT module. While in this section, some detailed information about spectrum sensing modules will be described. To build a spectrum sensing module in GRC, we need to understand the structure of the GNU Radio OOT module. In this project, each spectrum sensing method needs to construct a new block to implement its functionality. There are four files in the module required to be changed, for simplicity taking the MME module as an example:

- The header files of the block that is defined in *gr-MME/include/MME_cpp.h and gr-MME/lib/MME_cpp_impl.h*, in which all the private and public variables will be declared. The prefix '*d_*' will be added on all the private. And if there is any private variable that need to be changed outside the block, the set and get functions will be declared.
- The source file which is defined in *gr-MME/lib/MME_cpp_impl.cc* and implement the actual code for the maximum and minimum eigenvalue spectrum sensing function. It defines a *make function* to the public class, which is shown as follows:

PYTHON 5.1

```
MME_cpp::sptr
    MME_cpp::make(int Ns, int M, float Pfa)
    {
      return gnuradio::get_initial_sptr
        (new MME_cpp_impl(Ns, M, Pfa));
    }
```

- All the main signal processing is implemented in the *work( )* function which operates the input streams to get output streams. The *work( )* function will be executed, while this data comes into a input buffer appropriately.
- The XML file needs to be modified which located in *gr-MME/grc/MME_MME_cpp.xml* path, if there is any changeable variable in the block, the *callback* need to be added into the block, as following:

HTML 5.1

```
<callback>set_Ns($Ns)</callback>
<callback>set_M($M)</callback>
<callback>set_Pfa($Pfa)</callback>
```

5.3.2 Building, installing and debugging the block

The following scripts are the steps of creating an OOT module by using gr_modtool.

LINUX 5.1

```
$ gr_modtool newmod introduction
gr-introduction$ ls
apps  cmake  CMakeLists.txt  docs  examples  grc  include
lib  python  swig
$ gr_modtool add -t sync -l cpp
GNU Radio module name identified: multiply
Language: C++
Enter valid argument list, including default arguments:
Add Python QA code? [Y/n]
Add C++ QA code? [y/N]
```

The first line means using gr_modtool to build a new module called introduction. Then go into the introduction file; there are several files in it as mentioned in section 3.2.5. After this, we need to add block to this module by *add –t sync –l cpp; -t* sync means the type of the block is synchronic; *-l cpp* means the program language is C++. And the block name is called multiply. This is a simple multiply block so we don't need to add arguments.

Before we start writing our C++ code, the test code needs to be added at first for a good behaviour. Then we can modify the *_impl.cc* and *.h* files to implement the multiply function. To build and install the new block we need run the following scripts. After installing the module, we need to make it available in GRC by modifying the XML file which is located in *grc* subdirectory.

LINUX 5.2

```
$ mkdir build      # We're currently in the module's top
directory
$ cd build/
$ cmake ../        # Tell CMake that all its config files
are one dir up
$ make             # And start building (should work after
the previous section)
```

We run *cmake* in a separate directory called *build*, where all the compiling is done in here. In reality, once we start to create GNU Radio modules, we will probably get stuck by some errors sooner or later. Thus, we need one useful debugging tool which can help us to inspect where an error might occur during a run-time. It is usually that the flowgraph runs without crashing. However, the final result is not correct. There are three easily methods in GNU Radio for users to debug their blocks:

- Quality Assurance (QA) codes

  It is a simple and obvious method to debug the blocks. When writing a new block, we should add QA code before other implementation codes. We can test as many options as possible which might cause troubles. One needs to make sure each individual block passes the QA test. If the block you are testing is failing, there are some suggestions we can follow. First of all, using the command *ctest –V* other than the command *make test*, in order to get more information from the output. Secondly, if necessary, we can add additional *print* statements in the C++ code in our blocks to show intermediary states until obtain the results we need.
- graphical sinks in GRC

  It is another simple way that everyone can follow. Using GRC can simply attach graphical sinks to our blocks. There are two different kinds of graphical sinks: QT GUI Widgets and WX GUI Widgets where you can choose Oscilloscope, number sink and FFT plot. The number sink is simply display the value of the input data. We can choose appropriate sinks to attach to our

blocks, according to the data type that you have. It is easily to remove or disable them in the flow graph.

- Storing data into file sinks after blocks

  For a more detailed analysis of the received data, we can choose some off-line analysis tools except GNU Radio, for example, Matlab, Octave or SciPy (with Matploylib). The most common way is to add the file sink to the blocks which you expect is making troubles. When we run the flow graph, all the data will be loaded into the file sink. After this we can use these tools to analyse receiving data.

5.4 Implementation of SNR estimation

In this project, the signal-to-noise ratio (SNR) value is considered as a function of the power level of input signal and the gain of receiving signal. The transmitting power can be adjusted from the transmitter gain in *UHD: USRP Sink* block and the value of constant in *Multiply Const* block. In order to ensure the detection performance at different SNR levels properly, in this thesis, a quantitative approach was employed to estimate the SNR by the following steps.

- Turn off the transmitter, then measure the power of the received signal in receiver and set this power to the noise power (${\sigma_w}^2$).
- Turn on the transmitter, then measure the power of the received signal and set this power to ${\sigma_s}^2$. For a better estimation, the measurements need to be repeated for several times to get an average number.
- After obtaining the noise power and the received power, the SNR can be calculated as the equation (5.1):

$$SNR_{dB} = 10\ \log 10\left(\frac{{\sigma_s}^2 - {\sigma_w}^2}{{\sigma_w}^2}\right) \tag{5.1}$$

After these steps, some system variables must stay the same to make sure the different results are got from the same situation, such as the gain of the transmitter and the receiver and the value of the multiply constant which is used to change the power of signal.

5.5 Summary

This chapter first presents an example of spectrum sensing test bed with USRP and GRC, especially the structure of spectrum sensing system with transmitter side and receiver side. Besides, a realization of spectrum sensing module has been analysed which includes the structure of the module and how to code and debug it. In the next chapter, the performance of the spectrum sensing system in different configurations, including DQPSK and OFDM with different spectrum sensing algorithm will be presented. These results will be compared at different SNR environment to demonstrate the functionalities of each method.

6 Experimental results

In the last chapter, the implementation of spectrum sensing was presented. In this chapter, a number of experimental results will be presented. The tests are performed on locally generated signals (DQPSK and OFDM) from the USRP B210 board. At first, we investigate the performance of the ED by comparing the implemented experimental value to the theoretical ED performance value. And then we use different modulation signals as PUs to compare the P_d of them. Finally the three spectrum sensing methods (ED, SED and MME) are implemented to compare the performance in different SNR values.

6.1 The Performance of Energy Detection in different configurations

As commented on in chapter 4, the theoretical basis of ED was illustrated in a detailed characterization. In this part of the work, we evaluate the performance of ED according to SNR and the number of samples. The results presented show the Receiver Operation Characteristics (ROC) curve of the secondary user for different SNR values and the number of samples, and the P_d as a function of them both. The latter results will help us to identify the optimal number of samples required to obtain a particular P_d when the P_{fa} is set to a given value. The results obtained for the ROC of the secondary user will help us as a reference for comparison with other spectrum sensing methods.

In this chapter we will refer to the result of the implemented spectrum sensing system in test bed as *implemented value*, while the result of Energy Detection calculated from equations which are plotted in MATLAB will be referred to as *theoretical value*. The theoretical value of ED can be calculated according to equation (4.11) and (4.12). While for the implemented value, after estimating the SNR of secondary user according to section 5.4, we measured the P_d as described in the equation (6.1):

$$P_d = \frac{number\ of\ detections}{number\ of\ observations} \tag{6.1}$$

6.1.1 OFDM signal as PU

In this section, we set P_{fa} = 0.05 for both theoretical value and implemented value tests. For the implemented value test, the parameters of the system are set as follows. The signal centre frequency is 2.4GHz and the sample frequency is 2MHz. The primary signal used is an OFDM modulated signal, BPSK is the constellation mode, the transmitted subcarriers number (occupied tone) is 200, and the FFT size is 1024. The number of average samples used for ED depends on the value of N. Furthermore, for a more accurate result of SNR estimation, high value of gain is used in this project, the receiver gain is 40 and the transmitter gain is 80.

Firstly, we measured how P_d scales as the SNR increases and compared the implemented value with theoretical value. For a better comparison, we set the number of samples N=1024 in both tests. As the theoretical value test, the primary signal power is varied from -75dBm to -59dBm, and the variance of noise is set $\sigma_w{}^2$ = -55dBm. Thus SNR values can be evaluated and vary from -4dB to -20dB. Figure 6.1 shows the detection probability for both the theoretical value and the implemented value. We can see from this figure that the probability of detection for the ED implemented value test is similar to the theoretical value. Both of the probabilities of detection in the two experiments increase from 0.1 to 1 while SNR values increases from -22dB to 0dB. Also in this figure we observe that for the SNR above -5dB, the P_d for implemented test reaches its peak; this means the secondary users can certainly detect the primary users. While, when the SNR is below -11dB, the P_d is lower than 0.5, which means ED is not useful when the SNR value is too small.

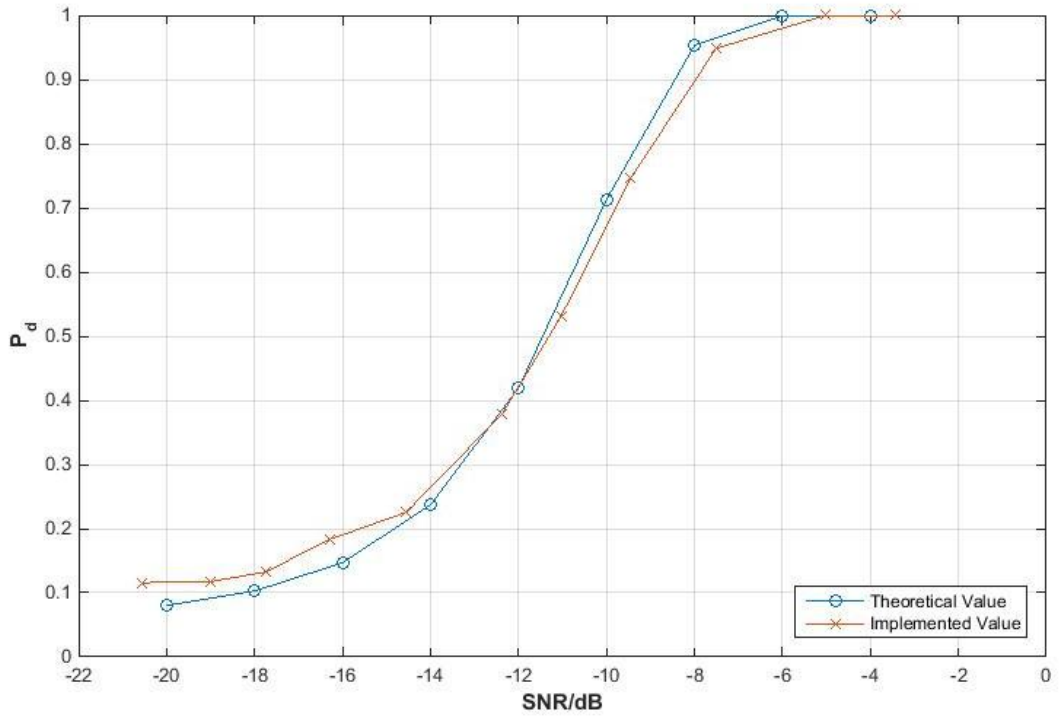

Figure 6.1 Probability of Detection as a function of SNR for both theoretical value and implemented value

To test the impact of the number of samples, we choose the SNR values at – 5 dB and – 11 dB and vary the number of samples from 900 to 1200. Figure 6.2 shows the P_d as a function of the number of samples. It can be seen that when the SNR values equal to – 5 dB, the P_d reaches the peak when the number of sample is larger than 1025. However, if SNR = - 11 dB, the P_d increases from about 0 to 1 when the number of samples vary between 900 and 1200. Also in this figure we observe that for N below 1000 when SNR = - 11 dB, the P_d is less than 0.4. Based on the result of Figure 6.2, we set N = 1024 to obtain the ROC curve for different SNR values and P_{fa}. This ROC curve is shown in Figure 6.3.

From Figure 6.3, it is possible to observe the trade-off between P_d and $P_{fa.}$ Thus, this curve is helpful for us to determine the detection threshold value for each pair (P_d and P_{fa}) we want to achieve. As observed from this figure, ED detector shows a poor performance when the SNR is relatively low. In this figure, there are five discrete SNRs. Under each SNR, we test the P_d of ED in the different P_{fa}. For example, if we set the P_{fa} = 0.1, and the P_d > 0.5, the ED detector can only detect signals with SNR equal or greater than -12dB.

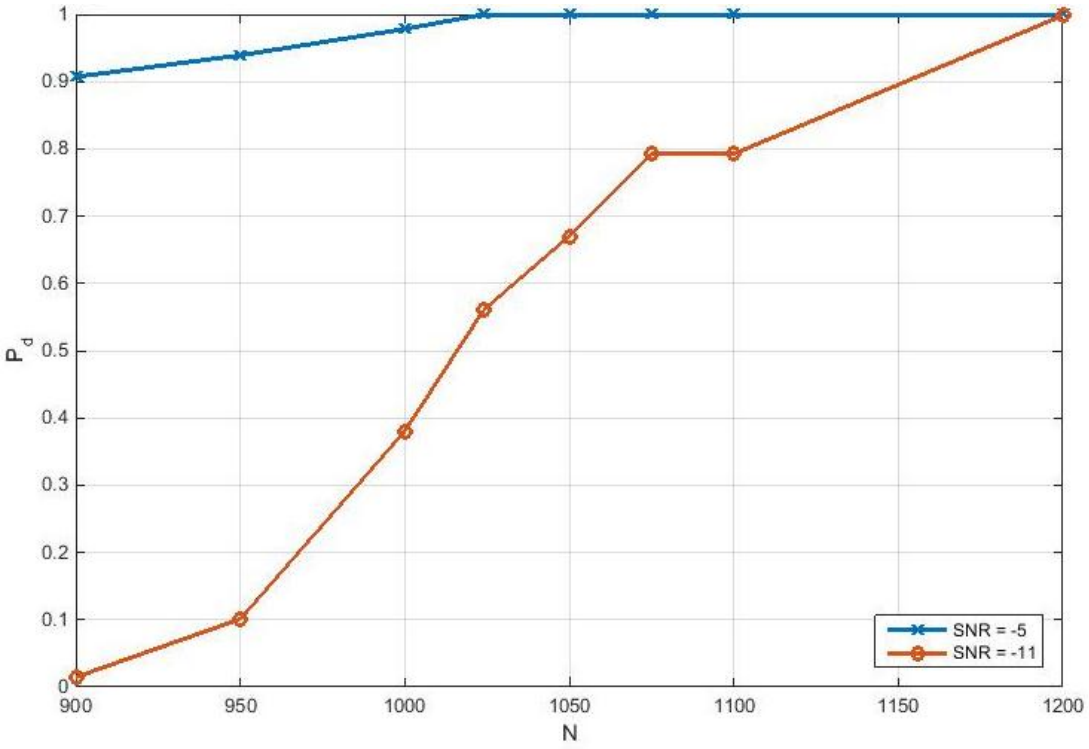

Figure 6.2 Probability of Detection as a function of the number of samples for P_{fa} = 0.05

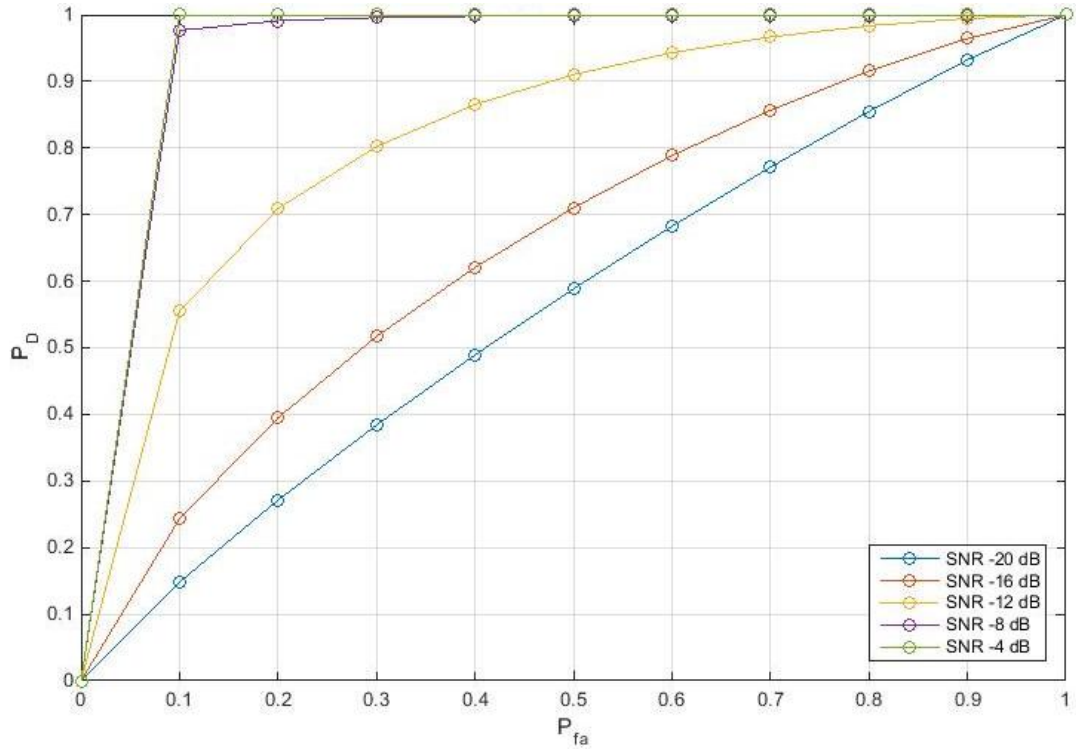

Figure 6.3 ROC curve for ED as a function of P_{fa}

6.1.2 DQPSK signal as PU

Here, we compare the performance of differential quadrature phase shift keying (DQPSK) modulated signal as PU signal with OFDM modulated signal. In this experiment, we set the central carrier frequency as 2.4GHz and the sample frequency as 2MHz, the P_{fa} is approximately equal to 0.05 for both PU signals. The number of received samples used by ED is 1024. We compute the P_d for the ED detectors at different values of the estimated SNR. Figure 6.4 shows the P_d of the ED detector for DQPSK and OFDM modulated signal as PU signals. These results are obtained through experiments as a function of SNRs. Further analysis shows that the performances of DQPSK as PU signals are significantly similar with OFDM other than some small fluctuation. The probability of detection of them are increased from more than 0.1 to 1 while the SNR growth from -20dB to 0dB.

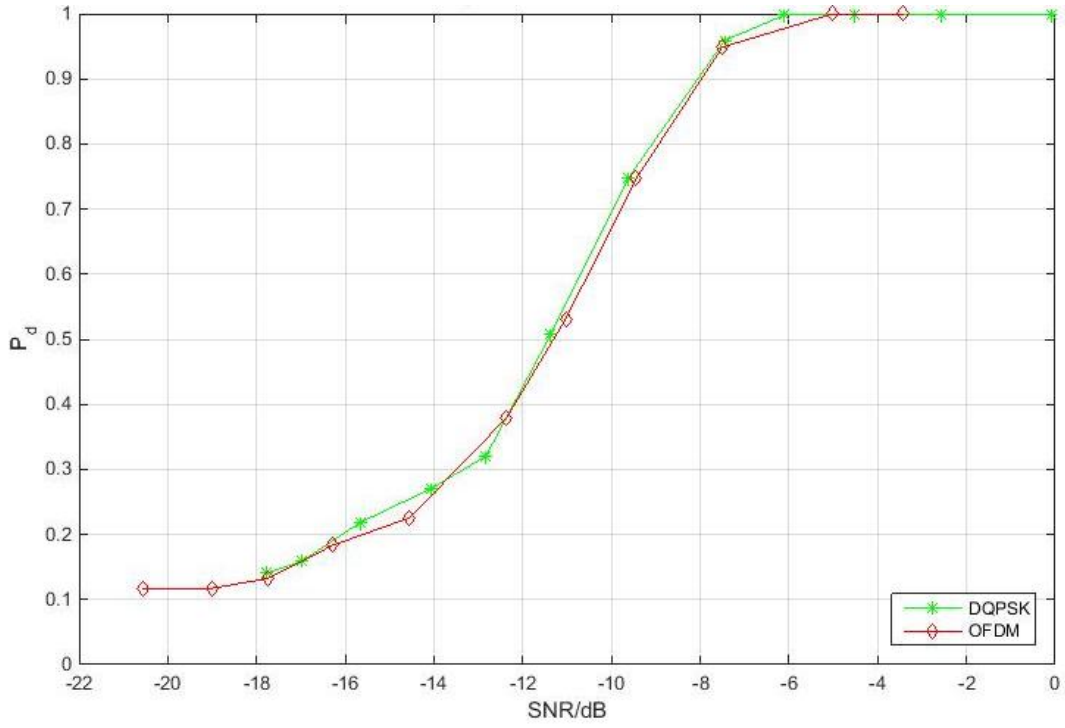

Figure 6.4 Probability of detection for two different PU signals (DQPSK and OFDM)

6.2 Frequency scanner in wide bandwidth

When implementing spectrum sensing, the test bed needs to scan a wide bandwidth which users are interested in. However, as it was mentioned in Table 3.3, the instantaneous bandwidth in 1×1 channels of B210 is only 56MHz, and, in practical

the sample rate of B210 is also limited. So there is a great demand to set up a frequency scanner to overcome the hardware's limitation.

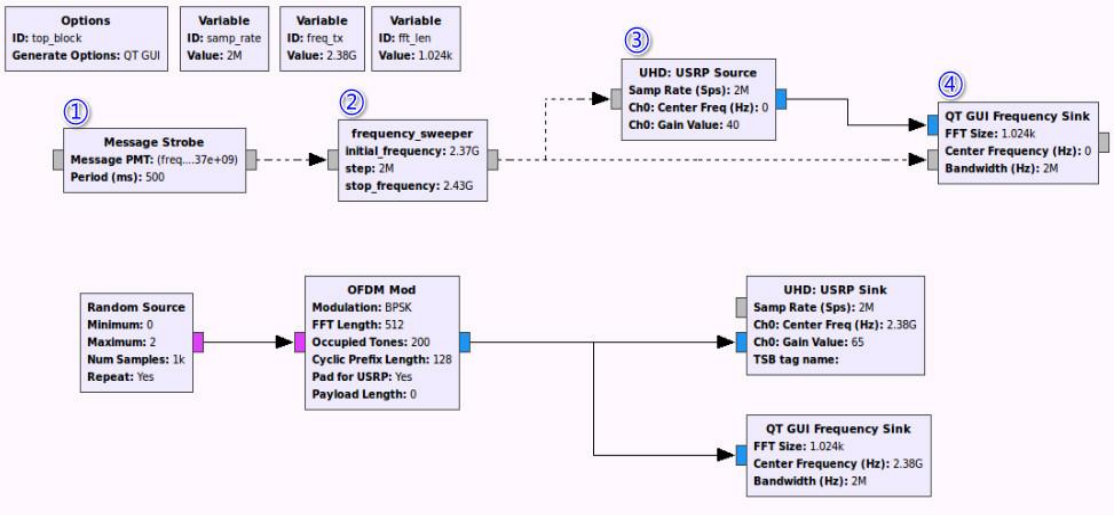

Figure 6.5 Frequency Scanner in GRC

In Figure 6.5, the frequency scanner is implemented in GRC, which can set the sweeping start frequency and stop frequency as user's requirement in a permanent time period. The start and stop frequency can be set in *frequency_sweeper* block (number ②) at *initial_frequency* and *stop_frequency* variables. The time period is set in the *Message Strobe* block (number ①) in *Period (ms)* variable. The first block in the scanner is *Message strobe* which uses Polymorphic Types (PMTs) as a message passing interface to control the message. After this block, the message goes into *frequency_sweeper* that can implement the change of the centre frequency from the start frequency to the stop frequency with same steps. The output message can work as a command to control the *UHD: USRP Source* (number ③) to produce signal as we set in *frequency_sweeper*. When sweeping at the stop frequency, it can hop to start point automatically. It also can set the scan step of each spectrum hopping. And the sampling time spent on each step can be set in Message Strobe from *Period* value. After receiving the command information from *frequency_sweeper* block, the *UHD: USRP Source* block produces the controlled signal and then send it to *QT GUI Frequency Sink* (number ④) to display. In this experiment, we set the start frequency as 2.37GHz, the stop frequency as 2.43GHz, and sweeping step as 2MHz. The

hopping period is equal to 5s which means the frequency will change in every 5 seconds.

The outcome of the Frequency Scanner is illustrated in the following three pictures. The first one is the transmitted signal from the transmitter; the centre frequency of this signal is 2.4GHz. The second figure shows there is no detected signal at the centre frequency of 2.37GHz with 2MHz bandwidth. However, when scanning at 2.4GHz as the centre frequency, the output of the frequency scanner is the same like the transmitter although some noises are added. These results illustrate the Spectrum Scanner can sweep the frequency from the start frequency to the stop frequency.

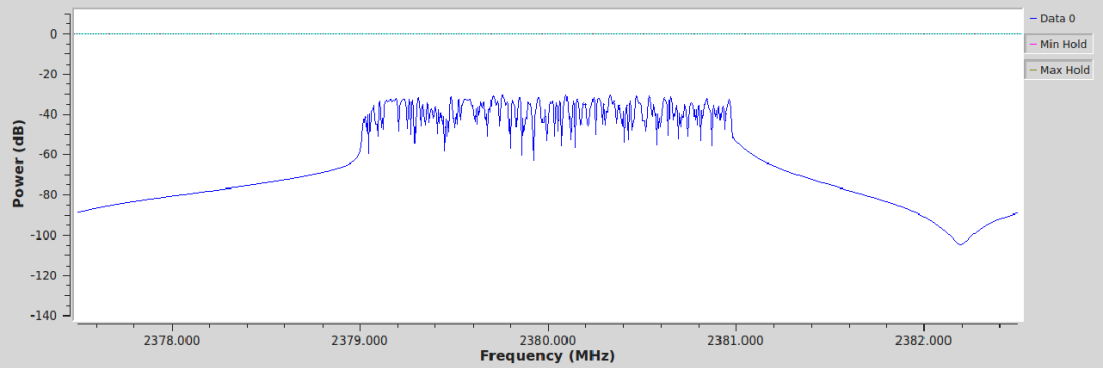

Figure 6.6 The transmitter signal for the Frequency Scanner

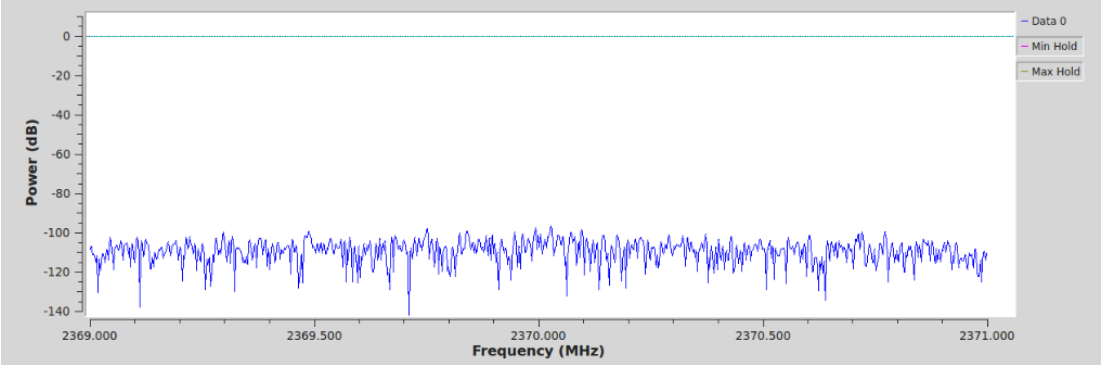

Figure 6.7 The result of the Frequency Scanner at 2.37 GHz

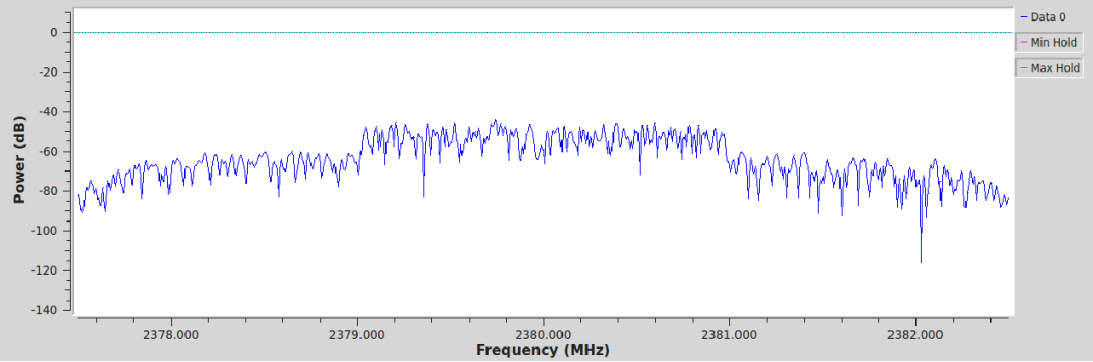

Figure 6.8 The result of the Frequency Scanner at 2.38 GHz

The *Message Strobe* block in the frequency scanner is used to make a message *msg* to send as a PMT in every *period* millisecond. In Figure 6.5, the Message PMT is set as the start frequency of the scanner as 2.37GHz. In the *frequency_sweeper* block, it is not the same as the spectrum sensing blocks designed as before; there is a message hander function (shown below) in the block to control the message instead of input/output stream flows. This function will do all the signal processing as the work function in stream blocks. The message hander in the *frequency_sweeper* block is written in python as following. The *message_port_pub* is a function to post message to the output port, there are two variables about this function. The first one is the output port; the second one is the message needs to be sent. In this block we set the output port name as *out*, which can be seen in Figure 6.5.

PYTHON 6.1

```
def handler(self, pdu):
      self.message_port_pub(pmt.intern("out"),
pmt.cons(pmt.intern("freq"), pmt.to_pmt(self.freq)))

      if self.freq < self.stop_freq:
          self.freq += self.step
      else:
          self.freq = self.freq1
```

6.3 The performance of Energy Detection in wide bandwidth

After implementing a spectrum scanner, we can connect it to the ED detector to find out the performance of this detector when it is used in wide bandwidth spectrum sensing. As shown in Figure 6.9, the spectrum scanner is connected with an ED block (*eng_sen1*) which makes the ED detector can scan a wide bandwidth spectrum. In this experiment, SU scans the frequency from 2.37GHz to 2.43GHz with each step of 2MHz. The hopping period is 0.5s. While the centre frequency of PU signal is 2.38GHz.

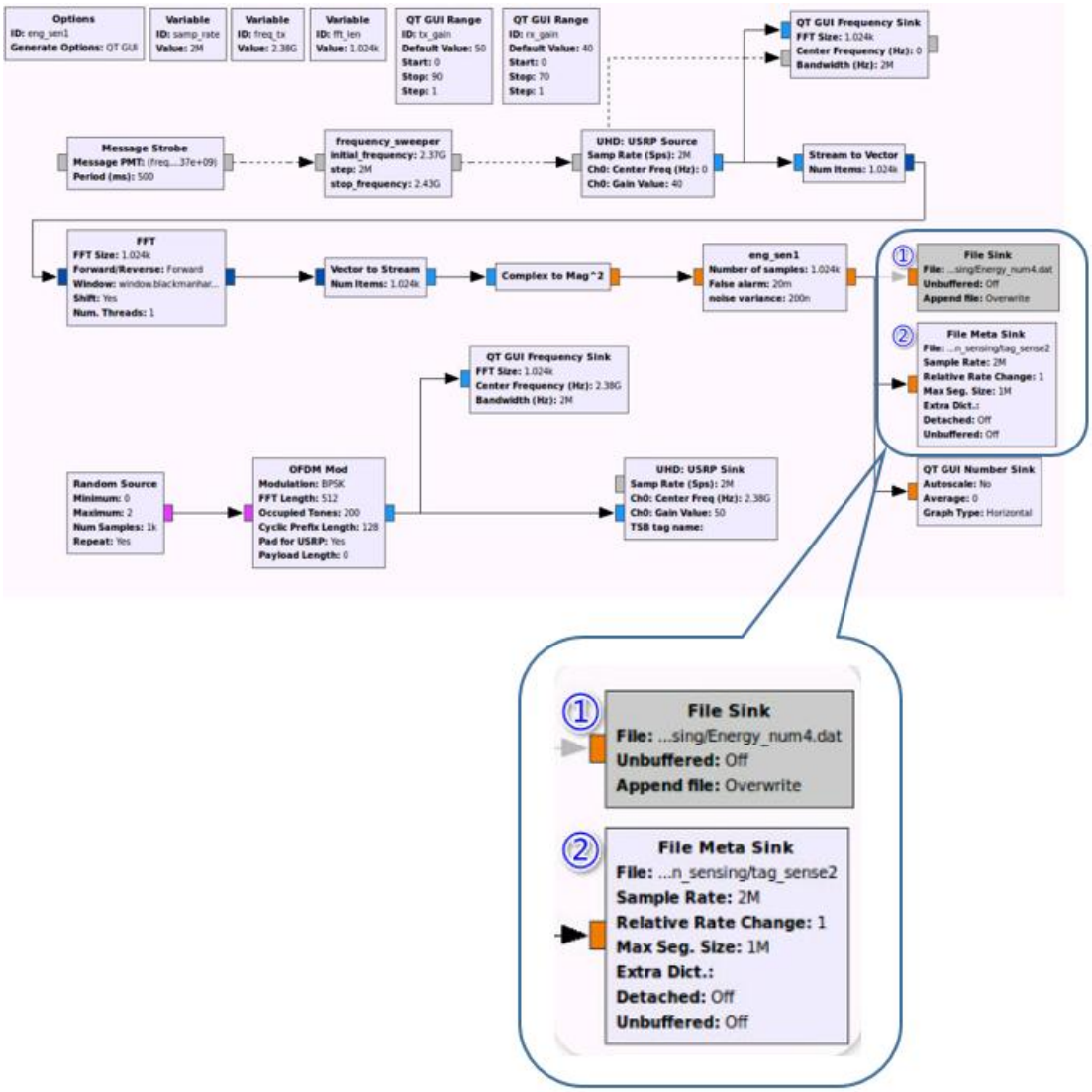

Figure 6.9 Spectrum sensing of ED in wide bandwidth

When we connect a *File Sink* (number ①) block to the ED block output port, the signal can be stored in the binary file called *Energy_num4.dat* in this experiment. As we explained before, we can see the I/O stream of ED block is float number, because the I/O ports are both in orange colour. So we read the binary file in Matlab as float type, and plot the ED detector outcome in Figure 6.10. This figure illustrates that the ED detector can detect the PU signal accurately. When the ED detector sweeps at 2.38GHz, the outcome become 1, this means there is a PU signal in this frequency band.

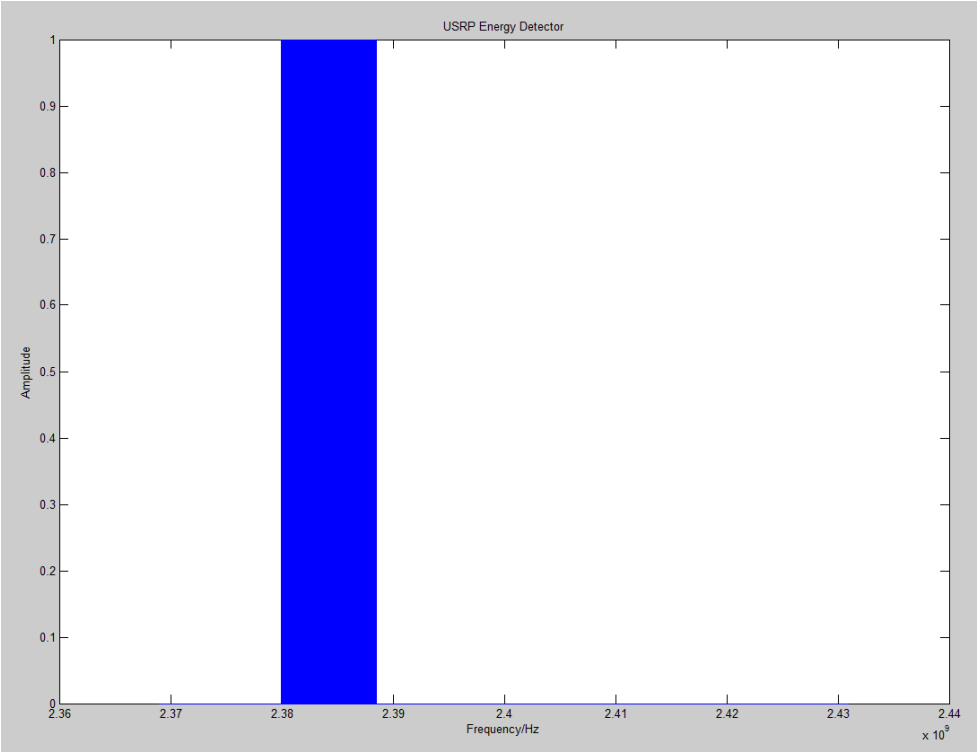

Figure 6.10 The spectrum sensing outcome of ED in wide bandwidth

An important point to be note here is that the *File Block* can only store the received signal as raw and binary files from the input port; we cannot know when the system starts working. Therefore, *File Meta Data* (number ②) file is used in this experiment to handle the file delicately. It is similar with *File sink* Block, but the most important is that it can store extra headers in the file which called metadata files

Metadata files have some extra information in the form of headers which carry the metadata about the samples in the file [68]. If there is any change in the system state such as the sample rate or whether the receiver's frequency is changed or not, they

will be shown in the files. The header structure of the metadata files is described in the table 下方.

Table 6.1 The information in the static portion of the header file

Number	Static portion	Contained information
1	Version	(char) version number
2	Rx_rate	(double) sample rate of the stream
3	Rx_time	(pmt::pmt_t pair - (uint64_t, double)) Time stamp (format from UHD)
4	Size	(int) item size in bytes - reflects vector length
5	Type	(int) data type
6	Cplx	(bool) true if data is complex
7	Strt	(uint64_t) start of data relative to current header
8	Bytes	(uint64_t) size of following data segment in bytes

We can open the metadata file using the following command:

LINUX 6.1

```
$ python gr_read_file_metadata tag_sens
```

The *gr_read_file_metadata* is an installed python program which is used to read out all header information from a metadata file. The *tag_sens* is the metadata file's name

which we use to store the header file information. The opened metadata file in Linux is shown in Figure 6.11. In header 0, we can figure out that the receiver time is shown as *Second: 1.587612.* According this, we can see that the receiver start receiving signal at the time 1.587612s.

Sometime we can get extra header information in the received file. In this experiment, we received *rx_freq* as the extra header, which shows the centre frequency of the receiver. When we use USRP board to receive signal, we can get an extra header called *rx_freq*. In header 0, the received frequency is 2.37e9Hz which is the start frequency of our experiment. According to this frequency, the hopping period can be verified by adding the time periods together between two different *rx_freqs*.

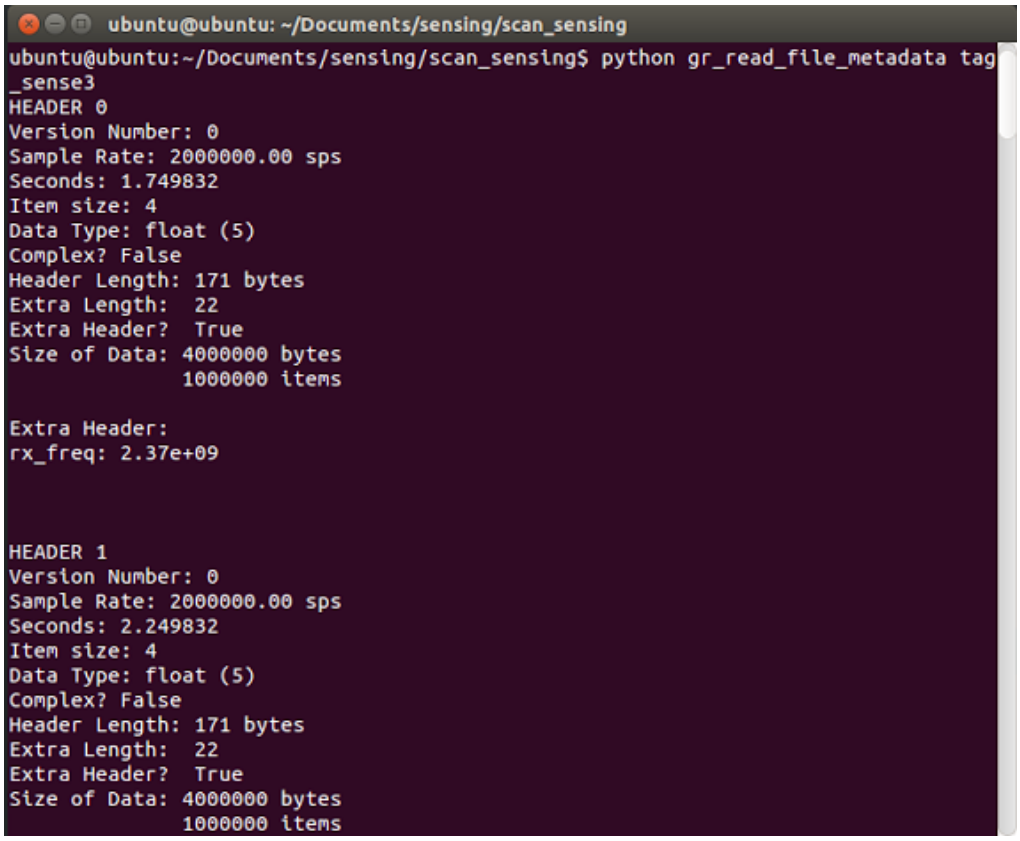

Figure 6.11 The inner content of metadata file

6.4 Comparison of different spectrum sensing methods performance

In this section, we test the performance of the three algorithms for OFDM as PU signal. The central carrier frequency is set to 2.4GHz. We compute the correct

detection probability (P_d) for the three detectors at different values of estimated SNR. For comparison, the P_{fa} is set to 0.05 in all of the detectors.

6.4.1 Sequential Energy Detection

For the ED detector, all the parameters are set as the same with section 6.1.1. While for the SED detector, the number of samples equals to 256, number of slots is 5, both of these two parameters are decided by the maximum sensing time. As we can see from the flow graph 下方, in the SED detector block *sed_spp* (number ②), there is a parameter called *factor* which is set to 0.8 in this scenario. This is a determined factor that can determine the double thresholds in SED detector. The relation between threshold and the variable factor is shown in equation (6.2)

$$\begin{cases} \eta_1 = (1-r) \times \eta \\ \eta_2 = (1+r) \times \eta \end{cases} \tag{6.2}$$

Whereas, r is the determined factor of SED detector, η is the traditional ED threshold calculated by equation (4.13).

The noise variance in SED detector is decided by the added noise in *Noise Sourse* block (number ①), in Figure 6.12, the amplitude of the added Guassian Noise in *Noise Source* block is 0.04, the noise variance is estimated as 0.4222. There are some blocks in gray, it means they are discharged. This funtion makes it more conveniet when you want to disable some blocks rather than delete it. When you want to reuse it, you can charge it easily. In this figure, the two *File Sinks* are discharged, this means that when starting running the flowgraph, the stream data will not forward to the *File Sinks.*

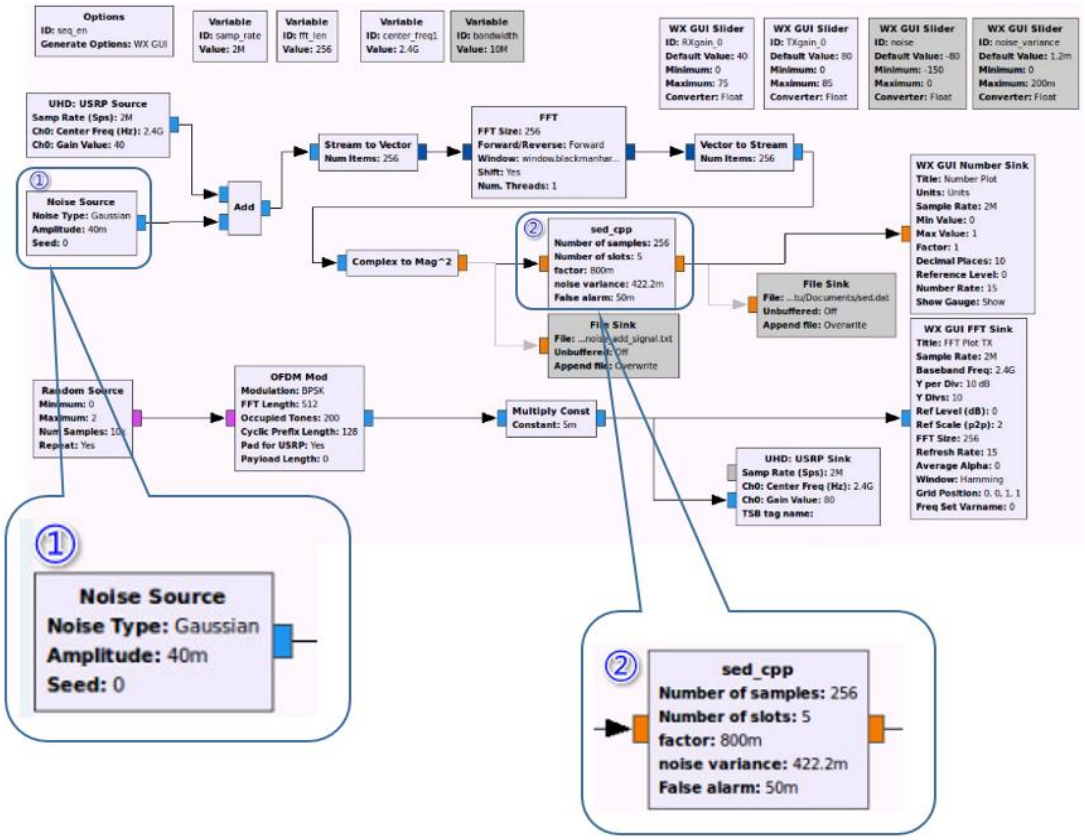

Figure 6.12 Flow graph of SED detector in the GNU Radio

6.4.2 Maximum-Minimum Eigenvalue

When implementing spectrum sensing by MME detector, we can follow the introduction of section 4.3.4. The Figure 6.13 is the flow graph of MME detector in GNU Radio. This kind of detector does not need a FFT block to transfer the input signal from time domain to frequency domain. According the MME detector block *MME_cpp*, the number of samples is 1024, the length of the correlation function is 16 and the probability of false (P_{fa}) is 0.05. In order to see the output of the MME block, a QT GUI Number Sink and a File Sink are connected to it to measure the P_d of this method.

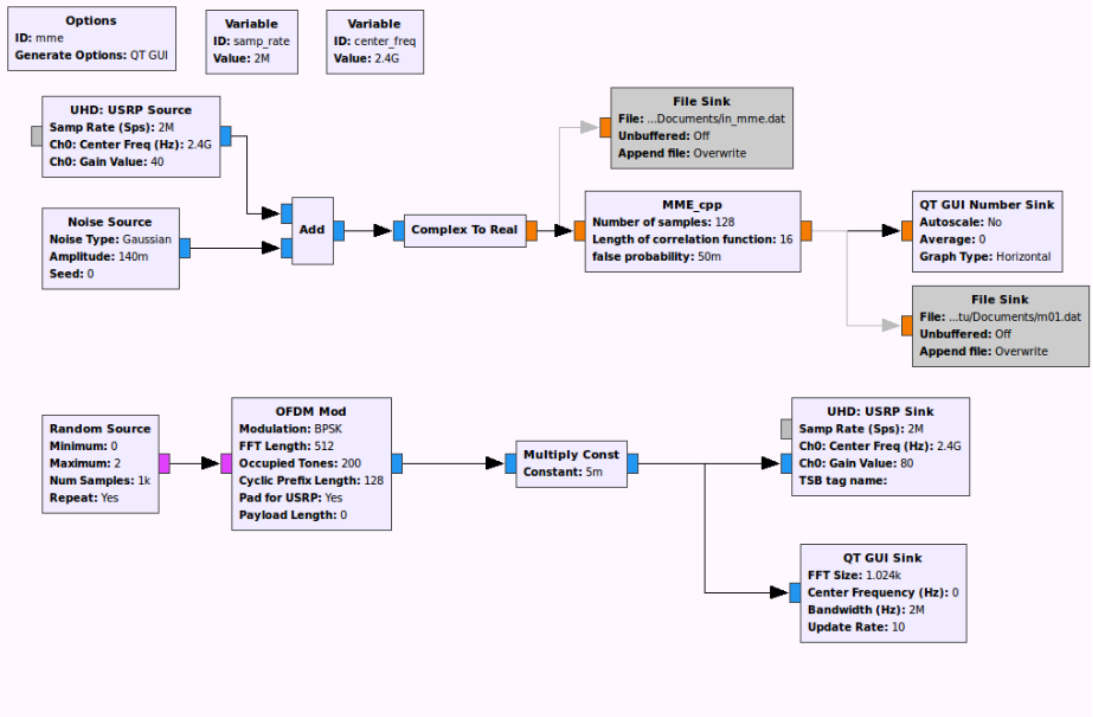

Figure 6.13 Flow graph of MME detector in the GNU Radio

6.4.3 Comparison among three different spectrum sensing methods

In this part, the primary signal is an OFDM signal; Figure 6.14 shows the detection probability achieved by the secondary user using ED, SED and MME, while the P_{fa} is kept below 0.05. The graph reveals that for the ED there has been a steady decline in the probability of detection when SNR decrease from -5dB to -20dB. As for SED, the outcome is better than ED when SNR is below -8dB. When the SNR falls to -18dB, the detection probability is 0.15 higher than ED.

Comparing the detection probability of MME with ED and MME, the MME detector also has a sharp fall trend while SNR decreases from -2dB to -8dB. However, when the SNR lower than -8dB, the detection probability shows a slight decrease to about 0.49. We can conclude from the figure that the MME outperforms ED and SED for a relatively low SNR under -11dB and -15dB respectively.

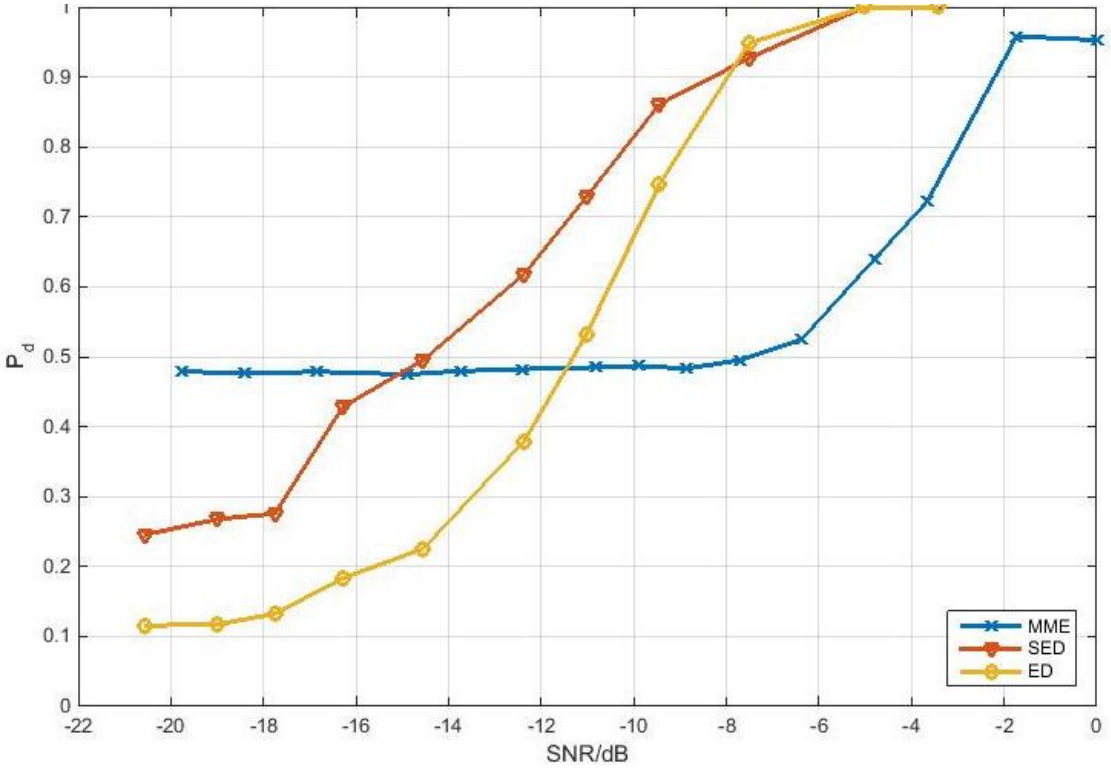

Figure 6.14 RCO curve comparison for ED, SED and MME

6.5 Summary

In this chapter, the performances of some spectrum sensing detector in different sensing methods (ED, SED and MME) are presented, using two constellation modulation modes (OFDM and DQPSK). To satisfy the practical needs, the results of ED in wide bandwidth also been implemented. The comparison confirms that both OFDM and DQPSK signal as PUs can get the similar results for ED detector. While for the three different sensing methods, at higher SNR, the ED and SED show better performance. Whereas, the MME has a better performance of detection probability at lower SNR. In the next chapter, we will implement the spectrum sensing in the GTEM Cell when adding electromagnetic interference.

7 Electromagnetic Interference Estimation Using a GTEM Cell

In the previous chapter, the performance of three different spectrum sensing schemes (ED, SED and MME) have been presented. In this chapter, we will identify potential source of interference by using a Gigahertz Transverse Electromagnetic (GTEM) Cell [69]. The chapter first details the function of the GTEM Cell. How to use GTEM Cell to set up the interference test environment is then presented. Finally, the implementation and results of interference estimation are discussed.

7.1 Introduction

The GTEM Cell is a high frequency version of the TEM Cell which is widely used alternative facility for Electromagnetic Compatibility (EMC) testing [70]. The intended use of the GTEM Cell is for radiated immunity testing and radiated emissions testing. In this chapter, we will use the radiated immunity testing to test how the system works when adding AWGN to the spectrum sensing test bed. Radiated immunity testing is conducted to ascertain if the Equipment under Test (EUT) will respond to radiated energy in the electromagnetic ambient in a deleterious manner. The GTEM Cell provides an ideal facility for the accomplishment of such tests in a laboratory environment. Immunity testing is typically performed using either of the following two techniques: substitution method and direct method [71]. In this project, we use the direct method to conduct this experiment.

- Substitution method

  This method uses the principle of calibrating the test volume. An E-field probe is positioned in the centre of the test plane and the input power is increased until the required test field is measured. The input power is then recorded as a function of frequency to create an empty volume calibration file. For the test, the field probe is replaced by the EUT and the forward power values of the calibration file are replayed while the device is monitored.

- Direct method

In this method the field probe and EUT are both placed in the GTEM Cell and the forward power increased until the required field is measured. This is repeated for each frequency as the EUT is monitored.

Figure 7.1 is the GTEM Cell which is used for the interference estimation. The model of this GTEM Cell is ETS Lindgren 5407 which has a frequency range from 9 KHz to 5GHz and outer cell dimension is $4.0 \times 2.2 \times 2.1m$ [69]. The GTEM Cell has the advantages over the anechoic chamber for immunity tests, for example, it is cheaper with less space and needs less amplification to generate certain field strength than in an anechoic chamber.

Figure 7.1 The GTEM Cell used for interference testing [69]

7.2 Experimental results

7.2.1 Interference estimation test bed setup

The following figure is the setup for conducting radiated immunity testing using GTEM Cell and USRP B210 board. The test can be completely automated since the receiving signal can be captured by the RX antenna on the USRP B210 board. The receiving signal then will be shown on the control PC. The signal generator is used for producing Additive White Gaussian Noise (AWGN) which is connected to the GTEM Cell input port. Application of the AWGN to the GTEM Cell input port produces the interference signal between the septum and the floor of the GTEM Cell. In the GTEM Cell, a pyramidal radio absorbing material is used for high frequency matched

termination and a resistive panel for low frequency matched termination to minimise reflections back towards the device under test (DUT). In this experiment, the DUT is the USRP B210 board and two connected antennas, which is located in the approximate centre of the test volume. The EUT performance is monitored via a USB 3.0 cable to a control computer. Once the setup is completed, the signal generator is turned over the test while monitoring the performance of the EUT for response to the applied noise signal. The level of the test noise signal can be adjusted by controlling the signal generator output.

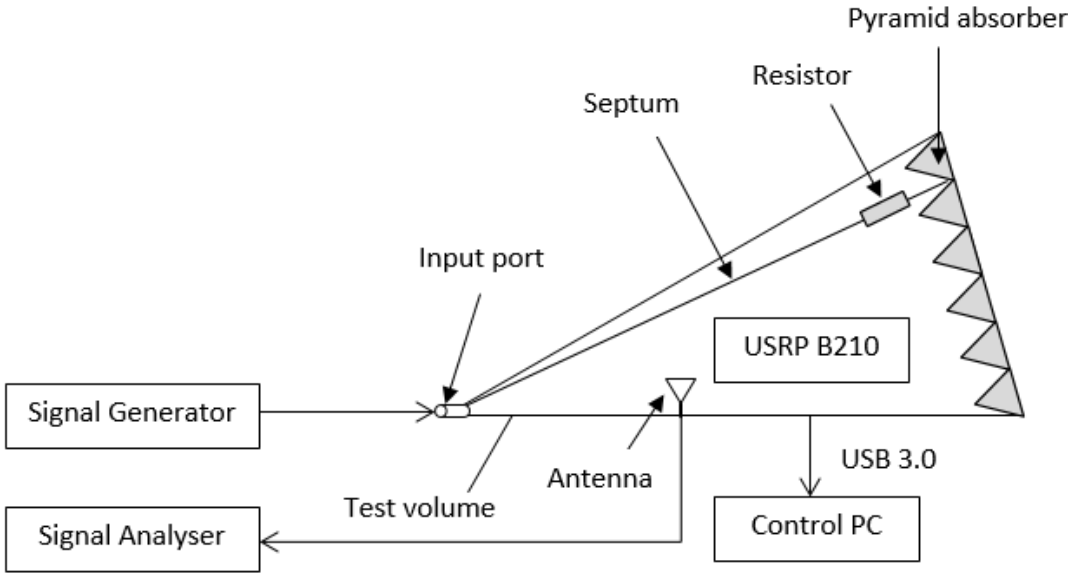

Figure 7.2 Interference testing using the GTEM Cell

7.2.2 Communication in the GTEM Cell

The spectrum sensing experiment results under simulation AWGN has been presented in chapter 6. This section presents how the spectrum sensing system works under the real noise in GTEM Cell. As it was mentioned in last part, we set up the communication system in laboratory as Figure 7.3. We put the USRP B210 platform (number ①) like Figure 7.4 in the GTEM Cell (number ②) and close the door before starting experiment. In order to make the comparison be more fairly for this spectrum sensing system, the place of the USRP B210 platform where it locates keeps the same. Then, we need to connect the USRP B210 platform with a computer which runs the GRC (number ③) software. For the interference part, a vector signal generator

(number ④) is used, which aims to generate the AWGN in GTEM Cell. In order to verify which kind of noise is added in the GTEM Cell, we use an extra antenna to receive the signal in the GTEM Cell which is connected to a signal analyser (number ⑤).

Figure 7.3 Spectrum sensing system setup in laboratory

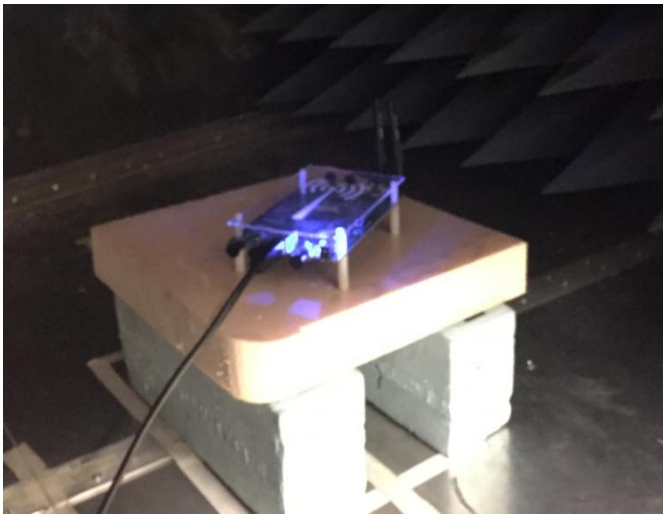

Figure 7.4 USRP board in the GTEM Cell

Before exploring the spectrum sensing system, we need to make sure the electromagnetic environment in the GTEM Cell without the USRP B210 board working in it. The environment can be captured using the signal analyser which connected with an antenna. Figure 7.5 presents the environment in GTEM Cell when there is no extra noise or signal adding to it, we can see the noise is very low at -80dBm. When adding extra noise the receiving signal shows on the signal analyser is Figure 7.6. The centre frequency of the extra noise is 2.4GHz and the amplitude is 20dBm. When receiving and displaying it as Figure 7.6, the amplitude of the signal is -40dBm.

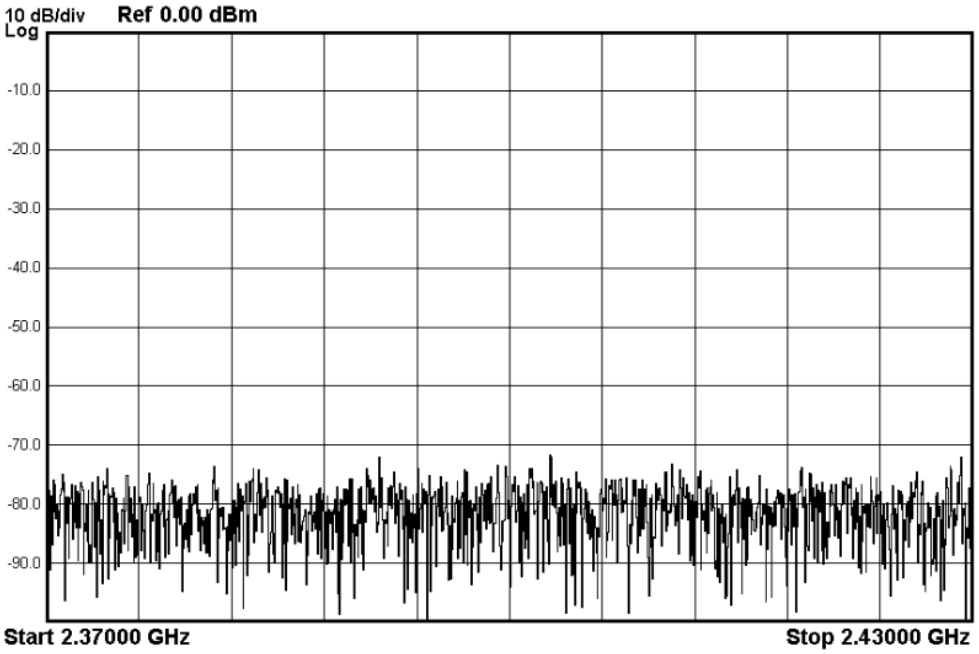

Figure 7.5 The GTEM Cell environment without extra noise

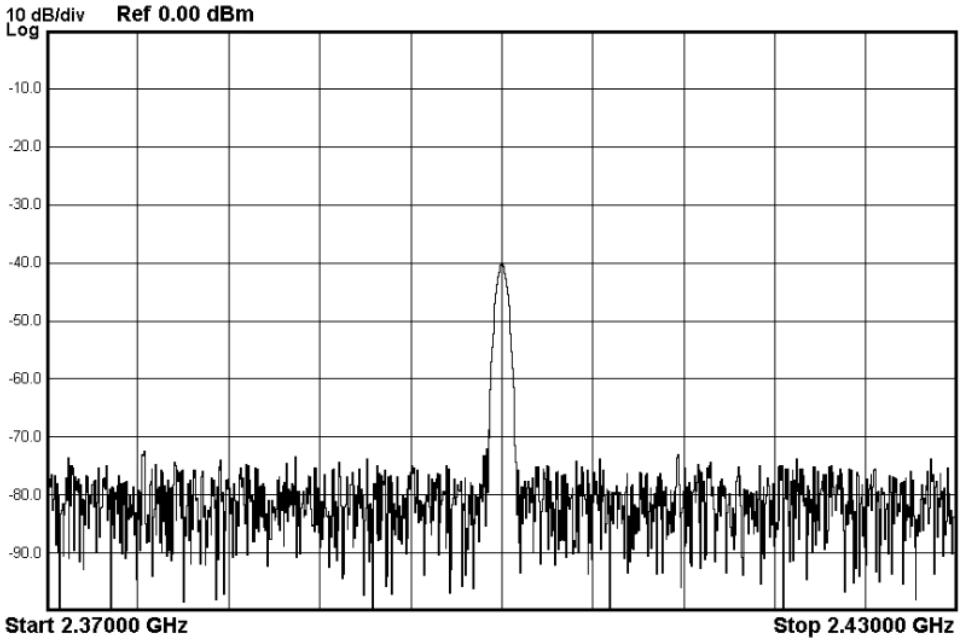

Figure 7.6 The GTEM Cell environment with extra noise

After setting up the system, the receiving signal at RX antenna is recorded while only using TX antenna to transmitting signal. The transmitting and receiving signal are shown in Figure 7.7 and Figure 7.8. According to these figures, we can see there is still some noise in the GTEM Cell which is maybe caused by the transceiver channel in the USRP B210 board.

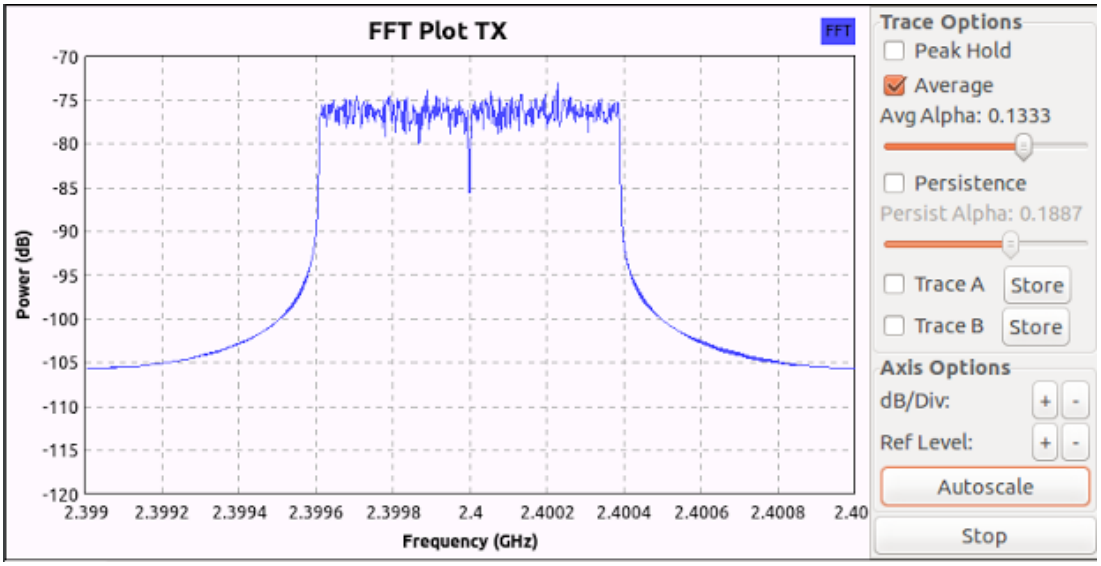

Figure 7.7 Transmitter signal in the GTEM Cell

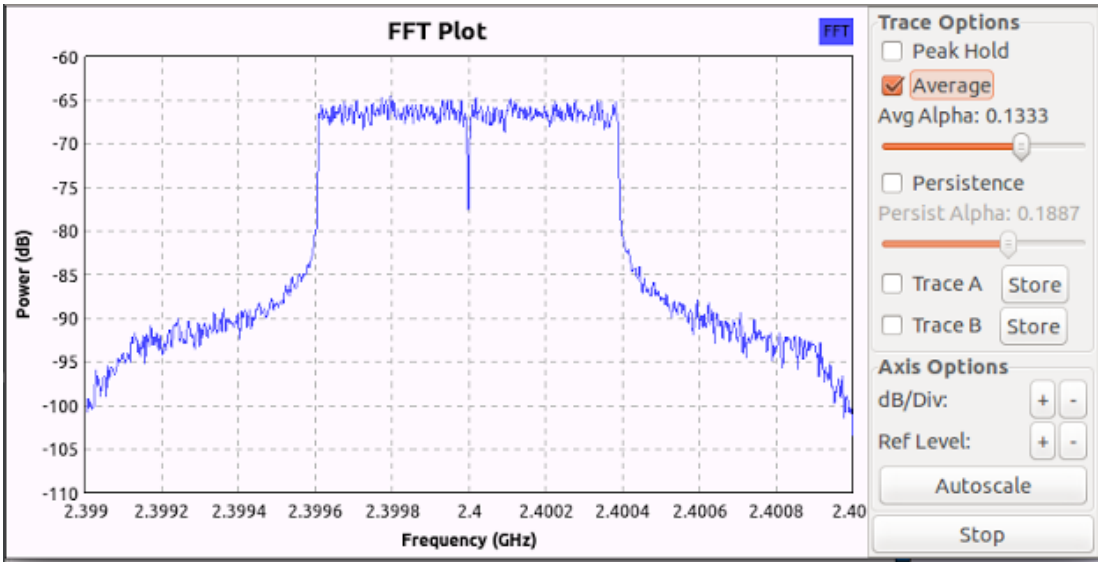

Figure 7.8 Receiving signal in the GTEM Cell without extra noise

The Figure 7.9 shows that when there is a noise added from signal generator with the amplitude of 20dBm the receiving signal presented in GRC. The noise power increases from -95dB to -68dB when comparing Figure 7.8 and Figure 7.9. This is due to the extra added noise in the GTEM Cell. After the noise added in the system, it becomes more difficult for the spectrum sensing system to detect the signal, especial for the Energy Detection which is totally depend on the signal energy. When the noise energy increased, the ED detector can detect the signal accurately only by increasing the detection threshold which has been described in section 4.3.2.

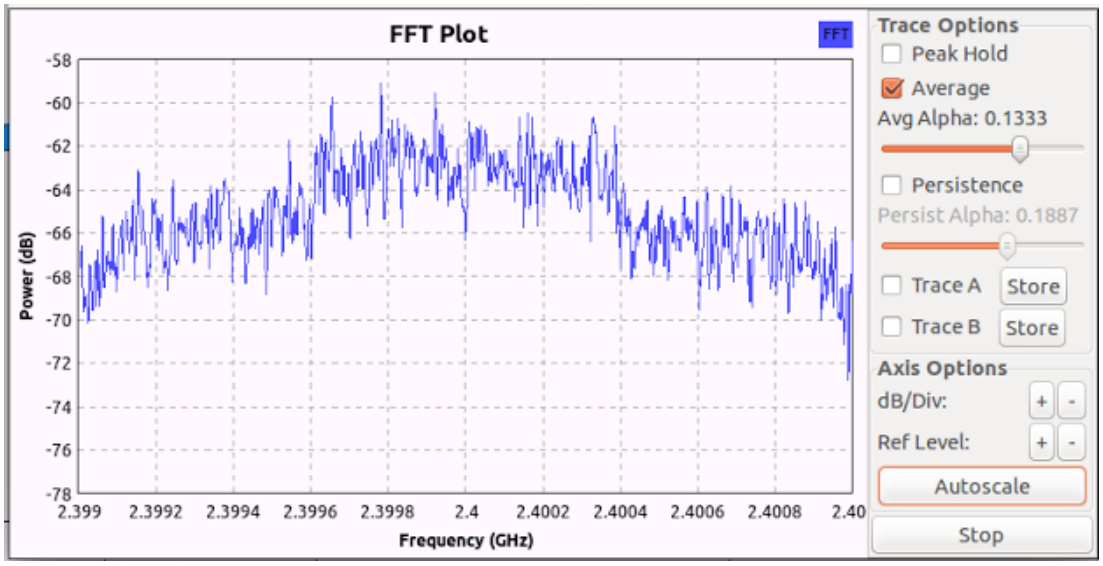

Figure 7.9 Receiving signal in the GTEM Cell with extra noise added

7.2.3 Implementing wide bandwidth sensing in the GTEM Cell

After making sure the spectrum sensing system can work in the GTEM Cell, the spectrum sensing experiment will be implemented in it. The first part is to implementing the wide bandwidth sensing in the GTEM Cell. Figure 7.10 depicts the transmitter signal in GTEM Cell for wide bandwidth sensing; this signal will not change after adding noise to the system. Figure 7.11 illustrates the wide bandwidth spectrum sensing system has the ability to figure out the useful receiving signal at the same frequency with transmitter signal. Figure 7.12 shows when the wide bandwidth sensing system scans the spectrum out of the centre frequency of transmitter signal, there is no signal output in the receiving part. In this scenario, the centre frequency of the receiving signal is 2.406GHz which is different from the transmitter signal centre frequency; the output the receiving signal shows there is no signal in this frequency band.

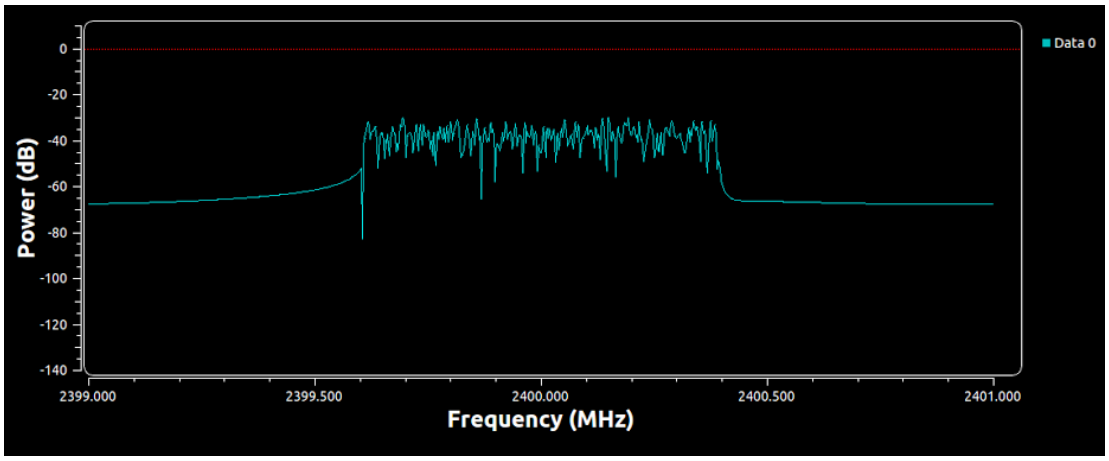

Figure 7.10 Transmitter signal in the GTEM Cell for wide bandwidth sensing

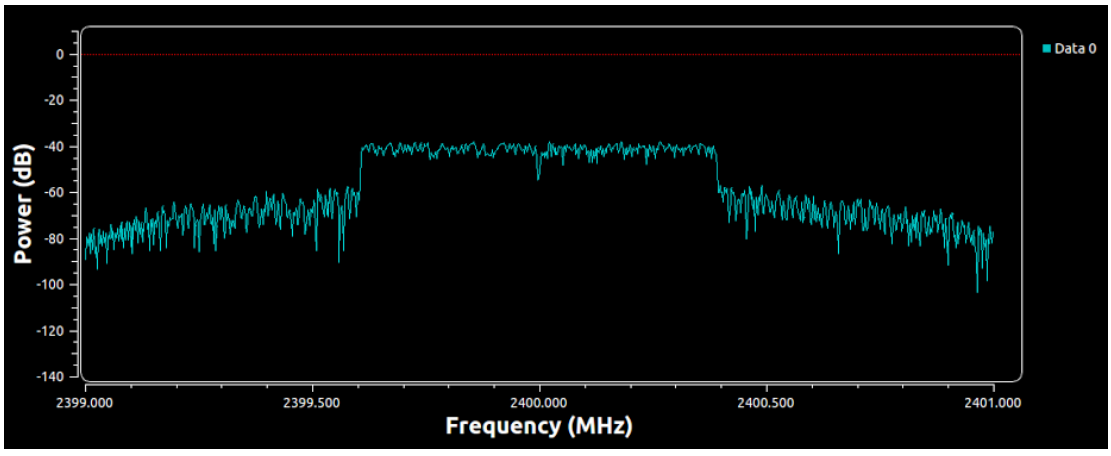

Figure 7.11 Receiving signal in the GTEM Cell at 2.4 GHz without extra noise for wide bandwidth sensing

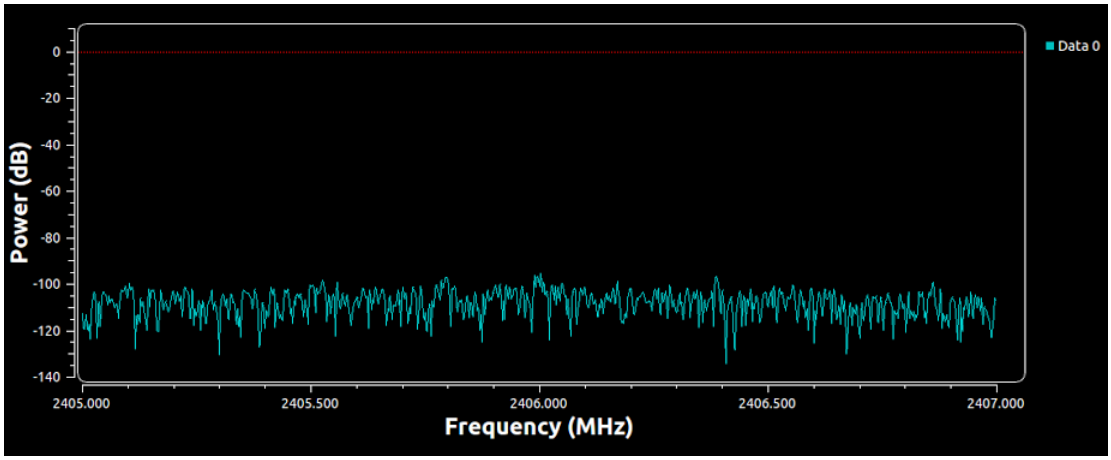

Figure 7.12 Receiving signal in the GTEM Cell at 2.406GHz without extra noise for wide bandwidth sensing

The receiving signal in GTEM Cell with extra noise added for wide bandwidth sensing is shown in Figure 7.13 and Figure 7.14. In the first figure, the centre frequency of the receiving signal is 2.4GHz which is the same with the transmitter signal. In this experiment, 0.86dBm noise is added from the signal generator. The noise power is about -60dB that is higher than it in Figure 7.11 which shows only around -80dB. This means when there is limited interference added in the system it can still distinguish the signal from noise. The second one illustrates when the centre frequency of receiving signal is different from that of transmitter signal, the output shows no signal in this frequency band. While comparing Figure 7.12 and Figure 7.14, we can see after the extra noise added in the system that the noise rises from – 120 dB to – 70 dB.

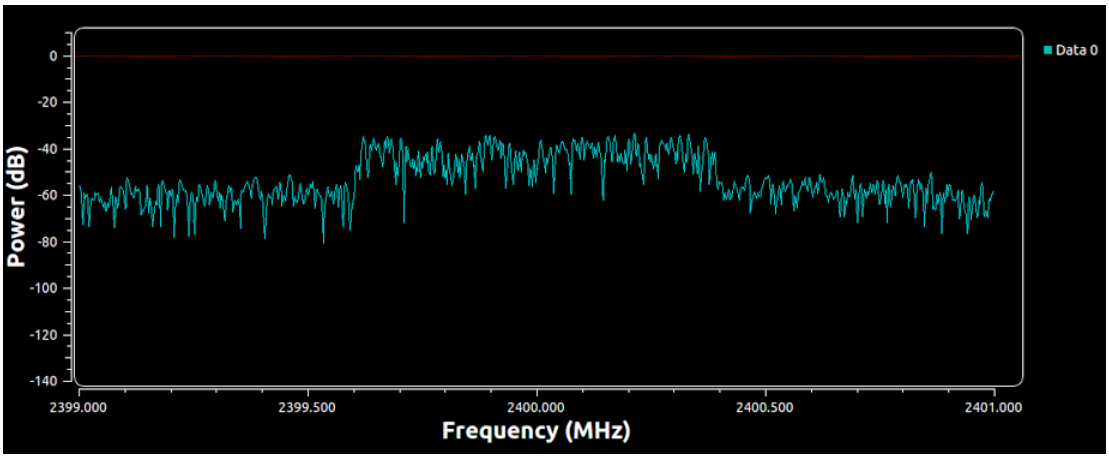

Figure 7.13 Receiving signal in the GTEM Cell at 2.4 GHz with noise added for wide bandwidth sensing

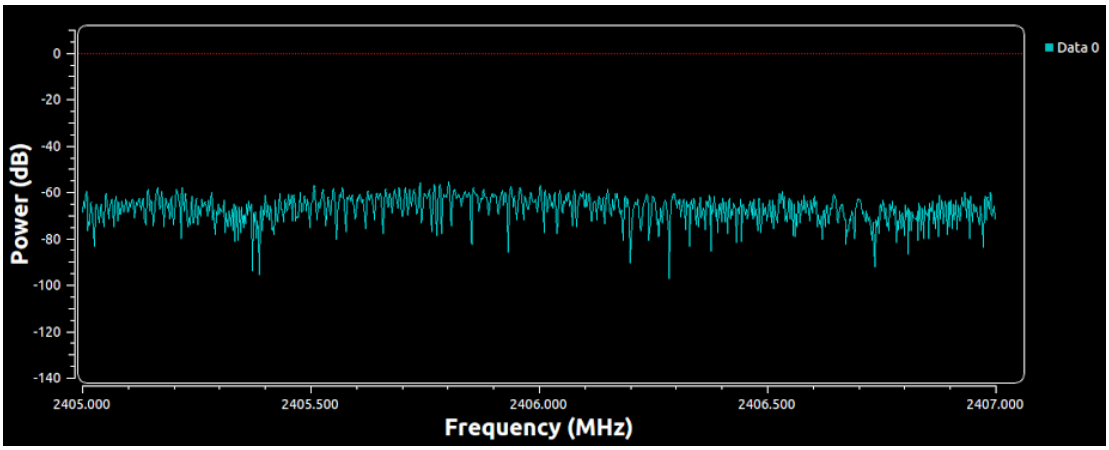

Figure 7.14 Receiving signal in the GTEM Cell at 2.406 GHz with noise added for wide bandwidth sensing

7.2.4 Implementing Energy Detection in the GTEM Cell

We have illustrated the spectrum sensing system can work in the presence of interference (AWGN). In this part, the comparison of simulated noise added in the system and real noise in GTEM Cell will be presented.

Figure 7.15 shows that the performance of spectrum sensing system under real noise in GTEM Cell is worse than it under simulation AWGN at the SNR higher than -13dB. However, when the system under lower SNR (below -13dB), the detection probability is higher in GTEM Cell than under simulation white Gaussian noise.

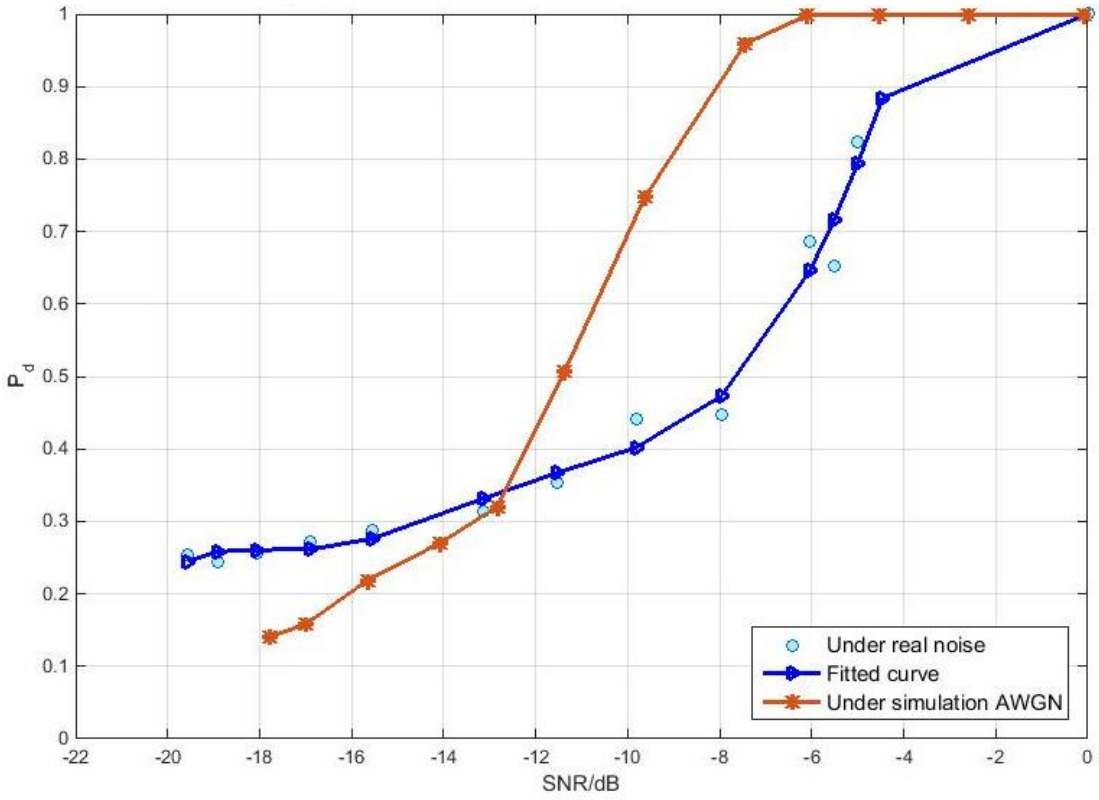

Figure 7.15 Probability of energy detection under different noise

7.3 Summary

In this chapter, we have implemented the interference experiment using GTEM Cell for the spectrum sensing system. The system setup in GTEM Cell is presented at first part; we put the USRP B210 board in the GTEM Cell and add extra interference to the system by signal generator. The performance of basic communication is illustrated by presenting the transmitting signal and receiving signal in GRC. While in the presence of interference, the wide bandwidth sensing can also figure out the useful signal. Finally, we compare the performance of the spectrum sensing system in the presence of real noise in GTEM Cell and under simulated noise from the probability of detection. It shows the spectrum sensing system works better than predicted when evaluated in the GTEM Cell subjected to AWGN.

8 Conclusion and Future Work

The spectrum sensing technique is one of the most important parts of Dynamic Spectrum Access which is widely used in wireless communication. In order to understand the spectrum sensing system more, both the wide bandwidth spectrum sensing and the electromagnetic interference sensing are required. Implementation of the wireless communication system by Software Defined Radio (SDR) is a prevalent method with more flexible utilities. It is a promising technique with less requirements of hardware than traditional hardware devices. In this project, a spectrum sensing system based on the GNU Radio Companion (GRC) software and the USRP B210 hardware is implemented to evaluate the performance of different spectrum sensing schemes, such as, different modulation constellations and different spectrum sensing methods. The experiment also compares the performance of different spectrum sensing methods and provides the performance of the spectrum sensing system under real interference.

8.1 Research contributions

In this project, our research outcomes can be concluded as follows:

- A comprehensive literature review of Cognitive Radio and Dynamic Spectrum Access has been provided. We also give a review of wireless communication and OFDM technique which are related to the implementation of spectrum sensing system. The literature review shows that there are little researches of wide bandwidth spectrum sensing system implemented in hardware; this is also the motivation of the thesis.
- A spectrum sensing test bed has been successfully developed based on the GRC software and the Ettus USRP B210 hardware. The wireless communication between two antennas in the Ettus USRP B210 platform has been also realized and documented in this thesis.
- The experimental performance of spectrum sensing system based on Ettus USRP B210 has also been presented with different modulation modes and signal to noise ratio (SNR) estimation. These performances demonstrate that there is no significant difference to choose different modulation schemes when implementing spectrum sensing.

- A number of spectrum sensing modules have been built using Out-of-Tree (OOT) tool in GNU Radio. The performances of these different modules have been compared and analysed under different SNR values. In this thesis we use the probability of detection as the standard to compare these spectrum sensing schemes.
- Furthermore, we have evaluated the performance of the spectrum sensing schemes under electromagnetic interference in the Gigahertz Transverse Electromagnetic (GTEM) Cell devices. The purpose of this experiment is to test the performance of the spectrum sensing schemes under the real noise. We also make a comparison of the Energy Detection scheme between under real noise and simulation Additive White Gaussian Noise (AWGN) in GRC software.

8.2 Future works

The successfully developed spectrum sensing test bed in this thesis gives a fundamental implementation for Dynamic Spectrum Access.

- For the Energy Detection (ED) of spectrum sensing method, a dynamical and adaptive threshold can be used for detection instead of static one. Considering present conditions of noise level would increase detection probability for the same SNR value as the traditional ED.
- Apply a hybrid algorithm which can combine different spectrum sensing schemes together considering energy-efficiency of the system. Such as, for higher SNR values to choose the ED detector while for lower SNR values to choose the MME detector or other high performance detector.
- Implement the spectrum sensing system on the USRP X310 board which has a better performance than the USRP B210 with higher bandwidth, sample rate and transmitting speed of the interface. These better properties can also improve the experimental results and decrease the system detection time.
- Implement some advanced spectrum sensing systems, such as cooperating sensing to increase the accuracy of spectrum sensing system.

8.3 Conclusion

In this thesis, we have set up a spectrum sensing system with the simulated noise in GRC and the real noise in GTEM Cell. This system is running in GNU Radio software toolkit and USRP B210 hardware board. In addition, the spectrum sensing system have been analysed in different signal modulation constellation schemes and the performances of the wide bandwidth spectrum sensing have been implemented.

To complete the aim of the project, we first set up the basic communication system using GNU Radio and USRP B210 with one transmitter and one receiver. After setting up this communication system, we described and analysed how to build a spectrum sensing module with GNU Radio platform. Based on these, we build our own spectrum sensing modules, as well as the SNR estimation in the spectrum sensing system. The first new Out-of-Tree module is energy detection which has been evaluated under two different modulation schemes: OFDM and DQPSK. Considering the practical use of the spectrum sensing system and the bandwidth limitation of USRP B210 board, we created a frequency scanner module to control the centre frequency of the spectrum sensing system to implement the wide bandwidth spectrum sensing.

To compare the performance of different spectrum sensing algorithms, we have created other two different spectrum sensing modules to improve the performance of the system, including sequential energy detection and maximum-minimum eigenvalue detection. The performance of each spectrum sensing module has been evaluated with different SNR values.

After the previous experiment, we intended to identify potential sources of the interference in reality. The performance of the spectrum sensing system under real noise has been implemented in the GTEM Cell. Before adding noise to the system, the basic communication of the system has been implemented. The comparison is also presented between the real noise in the GTEM Cell and the simulated noise in GNU Radio. The results show that the spectrum sensing system can work in the real noise and also has a better performance than under simulated noise in a relatively low SNR value.

In conclusion, SDR techniques with USRP B210 board and GNU Radio platform can be used to implement spectrum sensing systems to map the use of the limited spectrum. Future research could focus on improving the performance of the spectrum sensing system and dynamic spectrum sensing on both software and hardware.

References

[1] M. Subhedar and G. Birajdar, "Spectrum sensing techniques in cognitive radio networks: a survey," *International Journal of Next-Generation Networks*, vol. 3, pp. 37-51, 2011.

[2] V. C. Gungor and D. Şahin, "Cognitive radio networks for smart grid applications: A promising technology to overcome spectrum inefficiency," *Vehicular Technology Magazine, IEEE*, vol. 7, pp. 41-46, 2012.

[3] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum sensing for cognitive radio: State-of-the-art and recent advances," *Signal Processing Magazine, IEEE*, vol. 29, pp. 101-116, 2012.

[4] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio networks," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, 2005, pp. 137-143.

[5] M. Song, C. Xin, Y. Zhao, and X. Cheng, "Dynamic spectrum access: from cognitive radio to network radio," *Wireless Communications, IEEE*, vol. 19, pp. 23-29, 2012.

[6] S. Rakesh, U. Shetty, E. Engineer, T. Technetronic, and V. Bhavya, "Implementation of customized cellular network Using USRP," 2015.

[7] J. Hillenbrand, T. A. Weiss, and F. K. Jondral, "Calculation of detection and false alarm probabilities in spectrum pooling systems," *Communications Letters, IEEE*, vol. 9, pp. 349-351, 2005.

[8] Ofcom. (2016, 7th April). *UK spectrum map*. Available: http://www.ofcom.org.uk/static/spectrum/map.html

[9] J. Mitola III and G. Q. Maguire Jr, "Cognitive radio: making software radios more personal," *Personal Communications, IEEE*, vol. 6, pp. 13-18, 1999.

[10] J. Mitola, "Cognitive radio: Model-based competence for software radios," 1999.

[11] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *Selected Areas in Communications, IEEE Journal on*, vol. 23, pp. 201-220, 2005.

[12] P. Kolodzy and I. Avoidance, "Spectrum policy task force," *Federal Commun. Comm., Washington, DC, Rep. ET Docket*, 2002.

[13] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer networks*, vol. 50, pp. 2127-2159, 2006.

[14] M. Nekovee, "Dynamic spectrum access—concepts and future architectures," *BT Technology Journal*, vol. 24, pp. 111-116, 2006.

[15] P. P. Bhattacharya, R. Khandelwal, R. Gera, and A. Agarwal, "Smart radio spectrum management for cognitive radio," *arXiv preprint arXiv:1109.0257*, 2011.

[16] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *Signal Processing Magazine, IEEE*, vol. 24, pp. 79-89, 2007.

[17] G. S. Uyanik, O. Cepheli, G. K. Kurt, and S. Oktug, "Implementation and performance evaluation of dynamic spectrum access using software defined radios," in *Communications and Networking (BlackSeaCom), 2013 First International Black Sea Conference on*, 2013, pp. 157-161.

[18] (2016, 18th August). *Universal Software Radio Peripheral*. Available: https://en.wikipedia.org/wiki/Universal_Software_Radio_Peripheral

[19] Z. Yan, Z. Ma, H. Cao, G. Li, and W. Wang, "Spectrum sensing, access and coexistence testbed for cognitive radio using USRP," in *Circuits and Systems for Communications, 2008. ICCSC 2008. 4th IEEE International Conference on*, 2008, pp. 270-274.

[20] C. Weber and G. Hildebrandt, "Evaluation of blind sensing algorithms in the 2.4 GHz ISM-band on GNU radio and USRP2," in *Wireless Communication Systems (ISWCS), 2012 International Symposium on*, 2012, pp. 551-555.

[21] E. Rebeiz, P. Urriza, and D. Cabric, "Experimental analysis of cyclostationary detectors under cyclic frequency offsets," in *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*, 2012, pp. 1031-1035.

[22] A. Mate, K.-H. Lee, and I.-T. Lu, "Spectrum sensing based on time covariance matrix using GNU radio and USRP for cognitive radio," in *Systems, Applications and Technology Conference (LISAT), 2011 IEEE Long Island*, 2011, pp. 1-6.

[23] Y. Zeng and Y.-C. Liang, "Maximum-minimum eigenvalue detection for cognitive radio," in *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, 2007, pp. 1-5.

[24] A. Nafkha, B. Aziz, M. Naoues, and A. Kliks, "Cyclostationarity-based versus eigenvalues-based algorithms for spectrum sensing in cognitive radio systems: Experimental evaluation using GNU radio and USRP," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on*, 2015, pp. 310-315.

[25] K. Kim, Y. Xin, and S. Rangarajan, "Energy detection based spectrum sensing for cognitive radio: An experimental study," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, 2010, pp. 1-5.

[26] X. Shi and R. De Francisco, "Adaptive spectrum sensing for cognitive radios: an experimental approach," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, 2011, pp. 1408-1413.

[27] R. Budihal and H. Jamadagni, "Spectrum sensing performance characterization on ANRC's Hybrid Cognitive Radio Testbed," in *Advanced Networks and Telecommunication Systems (ANTS), 2011 IEEE 5th International Conference on*, 2011, pp. 1-6.

[28] N. M. Anas, H. Mohamad, and M. Tahir, "Cognitive Radio test bed experimentation using USRP and Matlab®/Simulink®," in *Computer Applications and Industrial Electronics (ISCAIE), 2012 IEEE Symposium on*, 2012, pp. 229-232.

[29] B. Jayawickrama, E. Dutkiewicz, and G. Fang, "Spectrum sensing error optimisation in cognitive radio networks," in *Communications and Information Technologies (ISCIT), 2012 International Symposium on*, 2012, pp. 787-792.

[30] S. Syed-Yusof, K. Khairul Rashid, N. Abdul Latiff, N. Fisal, M. A. Sarijari, R. A. Rashid, *et al.*, "TDMA-based cooperative sensing using SDR platform for cognitive radio," in *Communications (APCC), 2012 18th Asia-Pacific Conference on*, 2012, pp. 278-283.

[31] D. Tse and P. Viswanath, *Fundamentals of wireless communication*: Cambridge university press, 2005.

[32] H. Schulze and C. Lüders, *Theory and applications of OFDM and CDMA: Wideband wireless communications*: John Wiley & Sons, 2005.

[33] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of communication systems: modeling, methodology and techniques*: Springer Science & Business Media, 2006.

[34] A. Graham, N. C. Kirkman, and P. M. Paul, *Mobile radio network design in the VHF and UHF bands: a practical approach*: John Wiley & Sons, 2007.

[35] S. S. Haykin, M. Moher, and D. Koilpillai, *Modern wireless communications*: Pearson Education India, 2011.

[36] A. K. William Sethares. (18th April). *Modelling Corruption*. Available: http://cnx.org/contents/K-YPqibx@4/Modelling-Corruption

[37] U. Madhow, *Fundamentals of digital communication*: Cambridge University Press, 2008.

[38] D. Agrawal and Q.-A. Zeng, *Introduction to wireless and mobile systems*: Cengage Learning, 2015.

[39] A. Doukas and G. Kalivas, "Rician K factor estimation for wireless communication systems," in *Wireless and Mobile Communications, 2006. ICWMC'06. International Conference on*, 2006, pp. 69-69.

[40] S. B. Weinstein and P. M. Ebert, "Data transmission by frequency-division multiplexing using the discrete Fourier transform," *Communication Technology, IEEE Transactions on*, vol. 19, pp. 628-634, 1971.

[41] T. Shintaku, A. Kishida, M. Iwabuchi, T. Onizawa, and T. Sakata, "Experimental evaluation of a grouping method employing network allocation vector based on IEEE802. 11 wireless LAN," in *Microwave Conference (APMC), 2014 Asia-Pacific*, 2014, pp. 576-578.

[42] Y. Rahmatallah and S. Mohan, "Peak-to-average power ratio reduction in OFDM systems: A survey and taxonomy," *Communications Surveys & Tutorials, IEEE*, vol. 15, pp. 1567-1592, 2013.

[43] Y. Hou and T. Hase, "New OFDM system without guard interval," 2009.

[44] X. Gao, X. Wang, Y. Zou, and P. Ho, "An efficient OFDM with adaptive guard interval for amplify and forward relay systems," in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, 2013, pp. 1-5.

[45] B. Bhattacharyya, I. S. Misra, and S. K. Sanyal, "Novel Cycic Prefix Selection to Improve Spectral Efficiency and Signal Strength in OFDM Systems," *Int. J. on Recent Trends in Engineering and Technology*, vol. 8, 2013.

[46] R. Kristam and B. Doss, "A NOVEL APPROACH TO REDUCE PAPR IN OFDM SYSTEM USING DHT PRECODNG FOR M-QAM," 2014.

[47] A. K. Jaiswal, C. Kshyap, and S. R. Gupta, "Partial Transmit Sequence used in OFDM to Increase PAPR and Analysis of Phase Sequences and Data Blocks," *International Journal of Computer Applications*, vol. 65, 2013.

[48] Y.-H. Jan, "Low Complexity Channel Estimation in Fast Time-Varying Channels for OFDM Systems," *Journal of Internet Technology*, vol. 17, p. 2, 2016.

[49] C. Qi, G. Yue, L. Wu, and A. Nallanathan, "Pilot design for sparse channel estimation in OFDM-based cognitive radio systems," *Vehicular Technology, IEEE Transactions on*, vol. 63, pp. 982-987, 2014.

[50] S. I. Park, H. M. Kim, Y. Wu, L. Zhang, N. Hur, and J. Kim, "Robust synchronization for the OFDM-based Cloud Transmission system," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2013 IEEE International Symposium on*, 2013, pp. 1-3.

[51] W.-L. Chin, C.-W. Kao, H.-H. Chen, and T.-L. Liao, "Iterative synchronization-assisted detection of OFDM signals in cognitive radio systems," *Vehicular Technology, IEEE Transactions on*, vol. 63, pp. 1633-1644, 2014.

[52] S. Baumgartner, Y. El Hajj Shehadeh, and G. Hirtz, "A modified symbol timing and frequency synchronization method based on cyclic prefix for OFDM systems," in *Radioelektronika (RADIOELEKTRONIKA), 2015 25th International Conference*, 2015, pp. 331-334.

[53] J. C. Eric Blossom, Martin Braun, Matt Ettus, Nathan West, Tom Rondeau. (2016, 18th April). *Welcome to GNU Radio*. Available: http://gnuradio.org/redmine/projects/gnuradio/wiki/WikiStart

[54] (2016, 12th May). *Core concepts of GNU Radio*. Available: http://gnuradio.org/redmine/projects/gnuradio/wiki/TutorialsCoreConcepts

[55] (2016, 17th August). *About USRP Bandwidths and Sampling Rates*. Available: https://kb.ettus.com/About_USRP_Bandwidths_and_Sampling_Rates

[56] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *Communications Magazine, IEEE*, vol. 46, pp. 40-48, 2008.

[57] E. Peh and Y.-C. Liang, "Optimization for cooperative sensing in cognitive radio networks," in *2007 IEEE Wireless Communications and Networking Conference*, 2007, pp. 27-32.

[58] S. Syed-Yusof, K. K. Rashid, N. A. Latiff, N. Fisal, M. Sarijari, R. Rashid, *et al.*, "TDMA-based cooperative sensing using SDR platform for cognitive radio," in *2012 18th Asia-Pacific Conference on Communications (APCC)*, 2012, pp. 278-283.

[59] R. Viswanathan and P. K. Varshney, "Distributed detection with multiple sensors I. Fundamentals," *Proceedings of the IEEE*, vol. 85, pp. 54-63, 1997.

[60] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical communication*, vol. 4, pp. 40-62, 2011.

[61] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, vol. 55, pp. 523-531, 1967.

[62] A. Fernandez Eduardo and R. G. Gonzalez Caballero, "Experimental evaluation of performance for spectrum sensing: matched filter vs energy detector," in *Communications and Computing (COLCOM), 2015 IEEE Colombian Conference on*, 2015, pp. 1-6.

[63] D. M. M. Plata and Á. G. A. Reátiga, "Evaluation of energy detection for spectrum sensing based on the dynamic selection of detection-threshold," *Procedia Engineering*, vol. 35, pp. 135-143, 2012.

[64] H. V. Poor, *An introduction to signal detection and estimation*: Springer Science & Business Media, 2013.

[65] N. Kundargi and A. Tewfik, "Hierarchical sequential detection in the context of dynamic spectrum access for cognitive radios," in *Electronics, Circuits and*

Systems, 2007. ICECS 2007. 14th IEEE International Conference on, 2007, pp. 514-517.

[66] A. Bagwari and G. S. Tomar, "Cooperative spectrum sensing with adaptive double-threshold based energy detector in cognitive radio networks," *Wireless personal communications*, vol. 73, pp. 1005-1019, 2013.

[67] L. S. Cardoso, M. Debbah, P. Bianchi, and J. Najim, "Cooperative spectrum sensing using random matrix theory," in *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, 2008, pp. 334-338.

[68] (2016, 2nd, June). *Metadata Information*. Available: http://gnuradio.org/doc/doxygen/page_metadata.html

[69] (2016, July 7). *5407 GTEM! Test Cell*. Available: http://www.ets-lindgren.com/5407

[70] A. Nothofer, M. Alexander, D. Bozec, A. Marvin, and L. McCormack, *The use of GTEM cells for EMC measurements*: National Physical Laboratory, 2003.

[71] O. Manual, "Gigahertz Transverse Electromagnetic (GTEM!™) Cell," 2007.