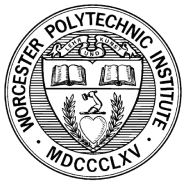
Primary User Emulation Detection in Cognitive Radio Networks

Di Pu



Department of Electrical & Computer Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts, USA

April 2013

APPROVED:

Professor Alexander M. Wyglinski, Primary Advisor

Professor Kaveh Pahlavan

Professor Andrew G. Klein

Professor Weichao Wang

A dissertation submitted to Worcester Polytechnic Institute in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

Cognitive radios (CRs) have been proposed as a promising solution for improving spectrum utilization via opportunistic spectrum sharing. In a CR network environment, primary (licensed) users have priority over secondary (unlicensed) users when accessing the wireless channel. Thus, if a malicious secondary user exploits this spectrum access etiquette by mimicking the spectral characteristics of a primary user, it can gain priority access to a wireless channel over other secondary users. This scenario is referred to in the literature as primary user emulation (PUE).

This dissertation first covers three approaches for detecting primary user emulation attacks in cognitive radio networks, which can be classified in two categories. The first category is based on cyclostationary features, which employs a cyclostationary calculation to represent the modulation features of the user signals. The calculation results are then fed into an artificial neural network for classification. The second category is based on video processing method of action recognition in frequency domain, which includes two approaches. Both of them analyze the FFT sequences of wireless transmissions operating across a cognitive radio network environment, as well as classify their actions in the frequency domain. The first approach employs a covariance descriptor of motion-related features in the frequency domain, which is then fed into an artificial neural network for classification. The second approach is built upon the first approach, but employs a relational database system to record the motion-related feature vectors of primary users on this frequency band. When a certain transmission does not have a match record in the database, a covariance descriptor will be calculated and fed into an artificial neural network for classification.

This dissertation is completed by a novel PUE detection approach which employs a distributed sensor network, where each sensor node works as an independent PUE detector. The emphasis of this work is how these nodes collaborate to obtain the final detection results for the whole network.

All these proposed approaches have been validated via computer simulations as well as by experimental hardware implementations using the Universal Software Radio Peripheral (USRP) software-defined radio (SDR) platform.

Acknowledgments

First and foremost, I want to thank my advisor Professor Alexander M. Wyglinski, who has brought me to the world of cognitive radio, accompanied me to overcome the difficulties along the way, and encouraged me to pursue perfection. Whenever I am overwhelmed or need help, Professor Wyglinski is the person who came along and directed me to handle these. He not only teaches me how to identify, formulate, and solve problems, but also instructs me on writing papers, giving good presentations, and impressing the audience. I have no doubt that the skills I have learned from him will stay with me throughout my life and help to maximize the chance of success in my career. Special thanks go to Professor Wyglinski for supporting me during my job searching and career decision. My gratitude towards him is beyond words.

The financial support provided by The MathWorks, Natick, MA, USA and by WPI Backlin Fund is duly acknowledged.

I am grateful to my dissertation committee: Professor Kaveh Pahlavan, Professor Andrew G. Klein and Professor Weichao Wang. This dissertation would not be possible without their critical questions and suggestions particularly during my area exam. I also would like to thank Mike McLernon from The MathWorks, for his great guidance on my internship at The MathWorks, as well as the contributions to our collaborated publication and the textbook I have published on SDR education.

The excellent courses in WPI ECE department have shaped my knowledge base and trained me to be a well-rounded engineer. WPI ECE also has the most impressive crew of technical and administrative staffs that I have ever seen. I definitely appreciate the professional solutions from Mr. Bob Brown to solve my computer-related problems and the quick responses from Mrs. Colleen Sweeney to solve my other issues. It makes me better utilize my time on research. I would like to thank my fellow graduate students at Wireless Innovation Laboratory (WI Lab): Srikanth Pagadarai, Si Chen, Jingkai Su, Zoe Fu, Sean Rocke, Travis Collins and Le Wang for making my time in the lab such an enjoyable and memorable experience. Thanks to all of you for making my time at WPI special.

Last but not least, I am grateful to my family, my parents and Di. I am not sure if I can achieve what I have achieved without their whole-hearted support and encouragement. They are the people who care me; who give me strength to take on any challenges that are ahead of me; and who always stand by me and make everything in my life simpler.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Wireless is Important

Modern society depends on wireless spectrum in order to properly function. Financial transactions, health services, national defense, security surveillance, and entertainment activities are just several of the numerous applications where reliable access to wireless spectrum is essential. Fig. 1.1 displays worldwide wireless modules revenues since 2007, and a forecast for the coming years in millions of dollars, which shows that wireless communication keeps growing fast nowadays.

However, given the rapid growth of the wireless sector, there has been an increasing strain on the availability of wireless spectrum to continue enabling these applications and services. This is partially due to the static, inflexible nature of wireless spectrum assignments defined by legacy regulatory guidelines and processes. Given that most of the wireless spectrum ranging between 0 Hz and 3 GHz already allocated via static assignments to a range of governmental, corporate, and academic entities [10], and that there exists numerous instances where multiple spectrum assignments have been made for several frequency bands, this assignment situation has resulted in fierce competition for the use of wireless spectrum. This is especially true in frequency bands located below 3 GHz, which is considered to be "prime" spectral real estate.

Conversely, a large portion of the assigned spectrum has been observed during several spectrum measurement campaigns [11, 12] to be sparsely and sporadically utilized. In particular, spectrum occupancy by licensed transmissions are often concentrated across specific frequency ranges while a significant amount of the spectrum remains either underutilized

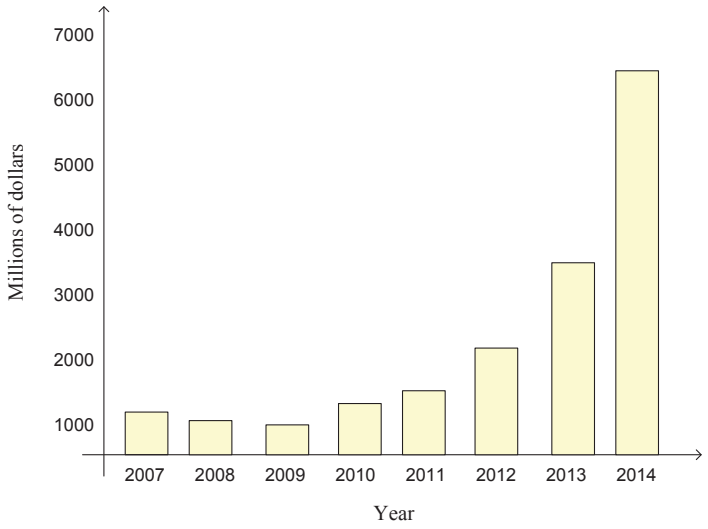


Fig. 1.1 Worldwide wireless modules revenues in millions of dollars [1].

or completely unoccupied. Consequently, it appears that the spectrum scarcity issue described above can be considered to be *artificially generated* due to the legacy regulatory and licensing processes as well as the inefficient utilization of assigned spectrum by the licensed transmissions.

To remedy this spectrum scarcity issue, a new approach for spectrum licensing is needed. This approach should be capable of providing wireless access to unlicensed applications and users (*i.e.*, secondary users) by allowing them to temporarily borrow unoccupied licensed spectrum, while simultaneously guaranteeing the rights of incumbent licensed users (*i.e.*, primary users), who possess a substantially higher priority or legacy rights across their assigned portion of wireless spectrum such that more efficient utilization of spectral resources can be achieved [13]. This new approach is called *dynamic spectrum access* [14].

1.2 Recent Legislative Developments

Dynamic spectrum access (DSA) is an approach for increasing spectrum efficiency via the real-time adjustment of radio resources via a combination of local spectrum sensing, probing, and autonomous establishment of local wireless connectivity among cognitive radio (CR) nodes and networks. During this process, *spectrum sensing* is employed for the purpose of identifying unoccupied licensed spectrum, *i.e.*, spectral "white spaces". Once

these white spaces have been identified, CR nodes opportunistically utilize these white spaces by wirelessly operating across them while simultaneously not causing interference to the primary users.

The US Federal Communications Commission (FCC) encouraged the application of the DSA technology to the secondary use of underutilized television spectrum, such as in ad hoc, short range wireless local area network (WLAN) in spectrum that is allocated to another primary purpose such as broadcast television. Owing to the increasing evidence of the under-utilization of wireless spectrum as demonstrated by [15, 16] and several others, an important policy step taken by the FCC is the enabling of cognitive access by secondary devices in TV broadcast spectrum [17]. White spaces are those unused TV band frequencies that were freed up by the switchover to digital television in the US. The FCC made these frequencies available on an unlicensed basis under its Part 15 rules in 2008 [18].

FCC endorsement of cognitive radio in secondary markets in the USA offered opportunities for improved spectrum utilization. In addition, the National Institute of Information and Communications Technology (NICT) Yokosuka, Japan have for characterized SDR and cognitive radio from technical [19, 20] and regulatory [21] perspectives. Ofcom, the regulatory body of the UK remains appropriately skeptical of the economics of dynamic spectrum [22]. On the other hand, the Commission for Communications Regulation (COMREG), Ireland, imposes constraints [23] but also encourages innovation such as by allocating over 100 MHz of spectrum for experiments and demonstrations during IEEE DySPAN 2007 in Dublin. Guatemala [24] employs Titulos de Usurfrucato de Frecuencias (TUF), specifying spectrum use parameters in great detail, which establishes a strong reference point for the liberalization of spectrum allocation towards dynamics [25]. In Europe, countries including Austria, Sweden, and the UK apparently have sanctioned de facto transfers of spectrum rights among spectrum licensees, while a recent EU Framework Directive empowers all EC countries to introduce secondary trading of spectrum usage rights [26].

Besides the policy and rule development discussed above, there exist several companies providing services and products to facilitate DSA. For example, a US company called Shared Spectrum Company (SSC) [27] was founded in 2000 to develop technology that increases the efficient use of RF spectrum resources. During that same year, SSC became the first company to file comments at the FCC proposing the shared use of "white spaces" in the television band for broadband Internet access. SSC focuses on the research and development of dynamic spectrum access technology for the U.S. Department of Defense. Another US

company called Spectrum Bridge, Inc. (SBI) [28] was founded in 2007, which features in developing TV White Space (TVWS) ecosystem and was certified as the first TVWS Database Administrator [29] in the United States. The platform ensures that there is no interference caused to protected users or devices by unlicensed secondary users. This marks the first time the U.S. has initiated a solution to allocate and manage wireless frequencies. In addition, a UK company called Neul [3] launches a custom designed, FCC certifiable TV white space radio system called NeulNET [2], which is the world's first dedicated TV white space 'network in a box', as shown in Fig. 1.2. With NeulNET, it is now possible to operate your own white space network that delivers up to 16Mb/s over 10Km range with excellent in building penetration.

Fig. 1.2 A custom designed, FCC certifiable TV white space radio system called NeulNET [2], developed by Neul [3] (from [2]).

1.3 Motivation

One of the major technical challenges regarding spectrum sensing is the problem of accurately distinguishing primary user signals from secondary user signals. In cognitive radio networks, primary users possess the priority to access the channel, while secondary users must always relinquish access to the channel over to the primary user and ensure that no interference is generated. Consequently, if a primary user begins to transmit across a frequency band occupied by a secondary user, the secondary user is required to leave that specific spectral band immediately. Conversely, when there is no primary user activity present within a frequency range, all the secondary users possess equal opportunity to the unoccupied frequency channel. Based on this principle, there exists the potential for malicious secondary users to mimic the spectral characteristics of the primary users in order to gain priority access to the wireless channels occupied by other secondary users. This

scenario is referred to in the literature as *primary user emulation* (PUE) [30–32].

In order to overcome the problem of PUE, the FCC currently employs a centralized control approach, as shown in Fig. 1.3. For a certain area, there is a fixed master device or base station, which connects to the Internet and communicates with Spectrum Bridge's white-space database [29]. It then connects with personal and portable devices or fixed device clients in homes and businesses. In order to utilize the unlicensed spectrum band, a TV band device (TVBD) must make a request to the database via the base station. The database provides available channel data in accordance with a set of rules, such as those defined by the FCC. After a device receives a channel map, final channel selections are made by the device. This decision is based on rules, radio technology and the offered channel map. In all cases, each device negotiates for an available channel.

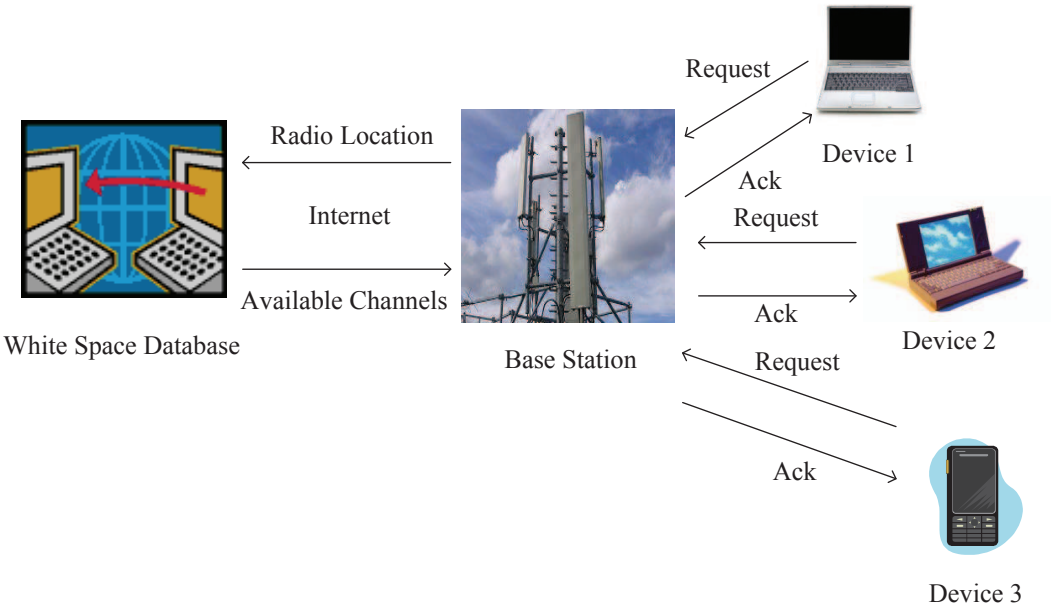


Fig. 1.3 A centralized control approach currently employed by FCC to avoid the problem of PUE, where a TV band device (TVBD) must go through a request and acknowledgement process in order to utilize the unlicensed spectrum band.

However, there are two major problems related to this approach. First of all, in some cases, this type of centralized control is not feasible, such as an emergency/disaster relief situation [33–35], or a military application [36–38], where there is no Internet or base

station infrastructure available. Secondly, this type of centralized control is not efficient enough. The request and acknowledgement process incurs much overhead to the network, and the TVBD may have to wait a long time before it receives an acknowledgement from the basestation. Therefore, it is necessary to propose techniques for the detection of PUE attacks that do not rely on the white-space database. Using these techniques, a contention-based cognitive radio ad hoc network [39, 40] can be constructed, where every network device is free to access the network any time it needs to send the data, without the request and acknowledgement step, as shown in Fig. 1.4. In this network, each device is capable of spectrum sensing, which is enabled by cognitive radio techniques. For a secondary user device, as long as it detects a "white space" in the spectrum, it can access the network and begin transmission. In this process, if it detects a primary user showing up on its spectral band, it needs to leave that spectral band immediately. This network does not have any base station or centralized control. Instead, there is a PUE detector which monitors each user's behavior. If it detects a primary user emulator in this network, some actions will be taken.

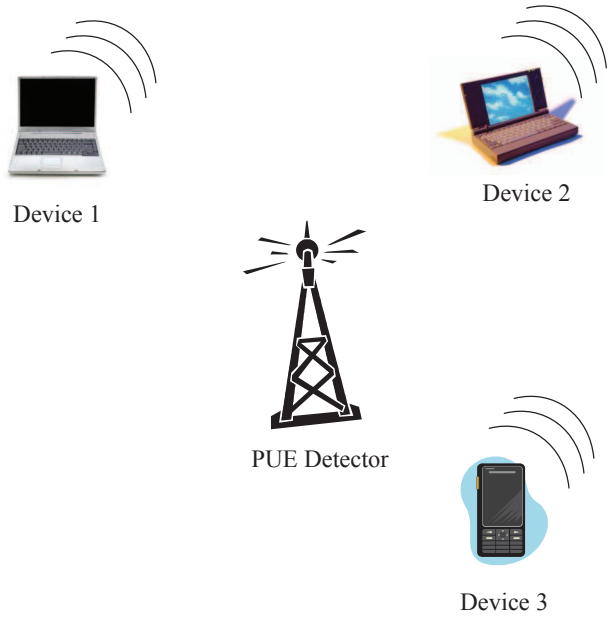


Fig. 1.4 A contention based dynamic spectrum access network that consists of devices with spectrum sensing capability. There is a PUE detector monitoring each user's behavior. No centralized control is needed for this network.

1.4 Current State-of-the-Art

Currently, there are several proposed techniques for the detection of PUE attacks that do not rely on the white-space database, which can be classified into the following categories.

- *Energy Detection*: To differentiate between the two kinds of signals, existing spectrum sensing techniques based on energy detectors implicitly assume a "naive" transmitter verification scheme [41, 42]. When energy detection is employed, a secondary user will be able to recognize the signal characteristics of the other secondary users but it will not be able to recognize the primary user signals. Thus, when a secondary user detects a signal that it can readily identify, it assumes that the signal is that of another secondary user. Otherwise, if the secondary user cannot identify the detected signal, it assumes that the incepted signal belongs to a primary user.

- *Feature Detection*: In feature detection methods, such as [43–46], secondary users attempt to find a specific feature of a captured signal, for example, a pilot, a synchronization word, or correlation. Devices capable of performing these detection techniques are able to recognize the intrinsic characteristics of the primary user signals, thus enabling them to distinguish these signals from those belonging to the secondary users.

- *Analytical Model-based Detection*: This type of detection methods are based on the analytical models of the cognitive radio networks proposed by the authors. In [47–54], the authors present a Neyman-Pearson composite hypothesis test (NPCHT) and a Wald's sequential probability ratio test (WSPRT) to detect primary user emulation attacks (PUEA) in fading wireless channels in the presence of multiple randomly located malicious users. In [55, 56], a passive anti-PUE approach, similar to the random frequency hopping in traditional anti-jamming schemes, is proposed and called dogfight in spectrum. In this scheme, the defenders randomly choose channels to sense and avoid the PUE attack.

- *Localization-based Detection*: One proposed transmitter verification scheme, called LocDef (localization-based defense [57]), is designed to verify whether an incepted signal belongs to an incumbent licensed transmitter by estimating its location and

observing its signal characteristics. In order to estimate the location of the signal transmitter, LocDef employs a non-interactive localization scheme.

- *Signature-based Detection*: [58] proposes an approach that integrates cryptographic signatures with wireless link signatures to distinguish a primary users signal from an attacker's signal. Besides, it employs a helper node to authenticate signals from its associated primary user.

1.5 Research Contributions

Given the published solutions currently available in the open literature, there still exists several technical challenges associated with enabling primary user emulation detection in cognitive radio networks, namely:

- Simple energy detector-base schemes possess a significant probability of missed detection.
- Signature-based detection and most of the feature detection methods require special hardware and software.
- Analytical model-based detection approach works well for a specific network model, but it may not work well for the other models.
- Localization-based detection can only be employed for stationary primary transmitters with known coordinates.

To resolve these issues, in this dissertation, I propose three PUE detection approaches as shown in Fig. 1.5, which possess the following novel contributions to the research community:

Cyclostationary has been widely used in signal classification, since different modulation schemes correspond to different cyclostationary features. Considering that in most situations, PU and SU use different modulation schemes, I suggest that cyclostationary is also a viable component for PUE detection and prove it.

Action recognition is a problem in the context of video processing, which is used to recognize different types of human actions. However, in this dissertation, I propose to use the framework of action recognition on FFT sequences, in order to recognize different

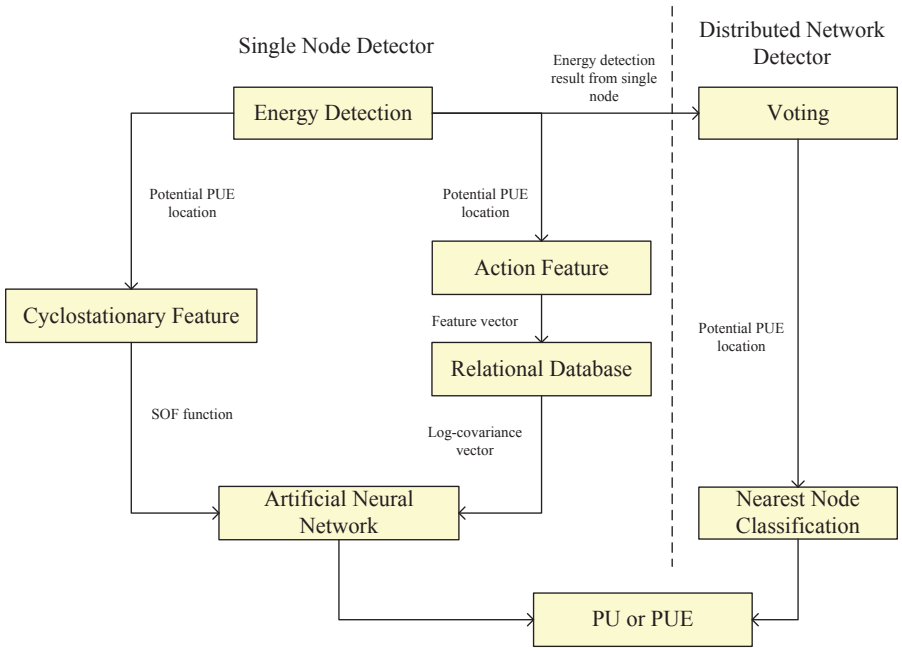


Fig. 1.5 Three novel PUE detection approaches proposed in this dissertation.

types of frequency "actions". In most of the literature, people pay more attention to the detection accuracy, *i.e.*, probability of detection. However, in this dissertation, I also consider the efficiency of detection, so I propose to use a database to assist the action recognition method. In this way, it is more likely that the approach can be applied in real-time applications.

Most of the PUE detection approaches proposed so far employs single node detector, but the third approach I propose is based on a distributed sensor network. A voting algorithm is used to improve the performance of energy detection, and the classification is conducted by the nearest node to improve the efficiency of the detector.

In addition, this dissertation features a real hardware implementation due to my expertise and experience in SDR technology and SDR education. I implement the proposed approaches on the USRP platform, so that the approaches can be tested in real-world wireless environment.

Considering these three approaches, they share several common benefits as following:

- No special hardware or software required in order to operate (the proposed approach

can be employed without significant structural and functional modifications).

- Can be used to verify mobile transmitters possessing unknown coordinates.
- Robustness in the presence of noise.

1.5.1 Resulting Peer-reviewed Publications

The list of publications both resulting from the work submitted in this dissertation and other related results not submitted in this dissertation are as follows:

Doctoral Publications

1. D. Pu, and A. M. Wyglinski, *Digital Communication Systems Engineering with Software-Defined Radio* Artech House, January 2013.
2. D. Pu, Y. Shi, A. Ilyashenko, and A. M. Wyglinski, "Detecting Primary User Emulation Attack in Cognitive Radio Networks," in *IEEE Global Communications Conference* (GLOBECOM) 2011.
3. D. Pu, and A. M. Wyglinski, "Primary User Emulation Detection Using Frequency Domain Action Recognition," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing* (PacRim) 2011.
4. A. M. Wyglinski, D. Pu, and D. Cullen, "Digital Communication Systems Education via Software-Defined Radio Experimentation," in *American Society for Engineering Education* (ASEE) 2011.
5. D. Pu, and A. M. Wyglinski, "Primary User Emulation Detection Using Database Assisted Frequency Domain Action Recognition," submitted to *IEEE Transactions on Vehicular Technology*, April 2013.
6. D. Pu, and A. M. Wyglinski, "Primary User Emulation Detection Using Distributed Sensor Network," submitted to *IET Communications*, April 2013.
7. D. Pu, and A. M. Wyglinski, "Overview of Primary User Emulation Detection in Cognitive Radio Network," journal in preparation.

Pre-Doctoral Publications

1. D. Pu, A. M. Wyglinski, and M. McLernon, "An Analysis of Frequency Rendezvous for Decentralized Dynamic Spectrum Access," *IEEE Transactions on Vehicular Technology - Special Issue on "Achievements and the Road Ahead: The First Decade of Cognitive Radio"*, vol. 59, no. 4, pp. 1652–1658, May 2010.

2. M. J. Leferman, D. Pu, and A. M. Wyglinski, *Cognitive Radio Communications and Networks: Principles and Practice* (A. M. Wyglinski, M. Nekovee, and Y. T. Hou, Eds.), Ch. 17. GNU Radio for Cognitive Radio Experimentation. Academic Press, 2009.

3. D. Pu, A. M. Wyglinski, and M. McLernon, "A Frequency Rendezvous Approach for Decentralized Dynamic Spectrum Access Networks," in *4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications* (CROWNCOM) 2009.

4. Z. Li, D. Pu, W. Wang, and A. M. Wyglinski, "Forced Collision: Detecting Wormhole Attacks with Physical Layer Network Coding," *Elsevier Tsinghua Science and Technology, special issue on Wireless Mobile Computing and Networking*, vol. 16, no. 5, pp. 505–519, October 2011.

5. W. Wang, D. Pu, and A. M. Wyglinski, "Detecting Sybil Nodes in Wireless Networks with Physical Layer Network Coding," in *40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (DSN) 2010.

6. Z. Li, D. Pu, W. Wang, and A. M. Wyglinski, "Node Localization in Wireless Networks Through Physical Layer Network Coding," in *IEEE Global Communications Conference* (GLOBECOM) 2010.

1.6 Dissertation Organization

As shown in Fig. 1.6, the remainder of the dissertation is organized as follows: In Chapter 2, the background of cognitive radios and how it relates to the dynamic spectum access and spectrum sensing will be discussed. Then, in Section 2.3, the concept of primary user emulation attack will be introduced. Also to be reviewed are all the techniques employed in

primary user emulation detection approaches. In Section 2.4, we introduce the basic idea of the signal detection and classification using artificial neural networks. In Section 2.5, an overview of the covariance descriptor for action representation is given. In Section 2.6, a relational database model and its design are introduced. This chapter ends with a review of the well-known SDR hardware platforms, with a focus on USRP2 platform, and some of the available SDR software architectures.

Chapter 4 and Chapter 5 dive into two specific categories for PUE detection. Chapter 4 proposes the first category based on cyclostationary features. It employs a cyclostationary calculation to represent the modulation features of the user signals, which are then fed into an artificial neural network for classification. In Chapter 5, we study the second category based on video processing method of action recognition in frequency domain, which includes two approaches. Both of them analyze the FFT sequences of wireless transmissions operating across a cognitive radio network environment, as well as classify their actions in the frequency domain. Built upon these two chapters, Chapter 6 proposes a third PUE detection approach which employs the distributed sensor network.

Finally, Chapter 7 concludes the dissertation. It contains concluding remarks and a list of anticipated future tasks in relation to the progress of the PUE detection described in this dissertation.

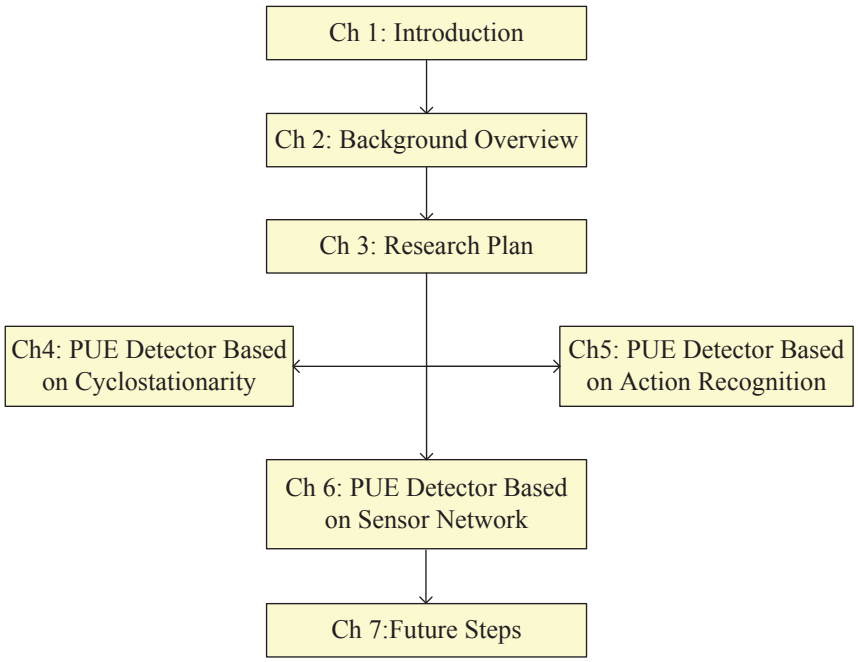


Fig. 1.6 Dissertation Organization.

Chapter 2

An Overview of Wireless Access

This chapter provides an overview on several subjects that are relevant to this dissertation, namely the background of cognitive radios, dynamic spectum access and spectrum sensing. It also reviews all the techniques that will be employed in the primary user emulation detection approaches. The goal of this chapter is to bridge the problem of PUE detection introduced in Chapter 1 and the approaches proposed in next chapter.

2.1 Cognitive Radio and Dynamic Spectrum Access

In Chapter 1, cognitive radio and dynamic spectrum access have already been introduced in the context of primary user emulation. Since these two subjects are very important for all the discussions in the dissertation, I will elaborate on them in this section.

2.1.1 Cognitive Radio

Cognitive Radio (CR) was formally introduced to the radio community in 1999 by Joseph Mitola and Gerald Q. Maguire, Jr. in [59] as an extension of an SDR, which served to improve the overall performance of the radio in relation to its interaction with the spectrum using a cognition cycle, as shown in Fig. 2.1. In [60], Mitola describes that a CR "is a goal-driven framework in which the radio autonomously observes the radio environment, infers context, assesses alternatives, generates plans, supervises multimedia services, and learns from its mistakes." While other definitions have been developed from research groups across the SDR community, the two components that are most often considered core features of the

CR involve awareness of the RF environment and adaptation and/or learning algorithms to improve the performance of the radio.

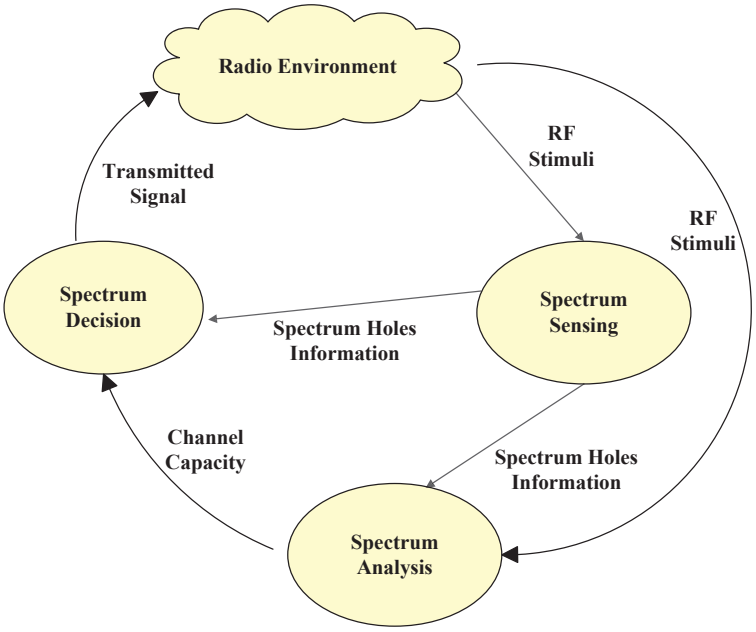


Fig. 2.1 Cognitive cycle (from [4]).

A *Cognitive Radio* (CR) is an *Software Defined Radio* that additionally senses its environment, tracks changes, and reacts upon its findings. A CR is an autonomous unit in a communications environment that frequently exchanges information with the networks it is able to access as well as with other CRs. From our point of view, a CR is a refined SDR [61].

2.1.2 Why Dynamic Spectrum Access?

Today's wireless networks are regulated by a fixed spectrum assignment policy, *i.e.* the spectrum is regulated by governmental agencies and is assigned to license holders or services on a long term basis for large geographical regions. Although the fixed spectrum assignment policy has generally worked well in the past, there is a dramatic increase in the access to the limited spectrum for mobile services in recent years. Consequently, this increase is straining the effectiveness of the traditional spectrum policies [62].

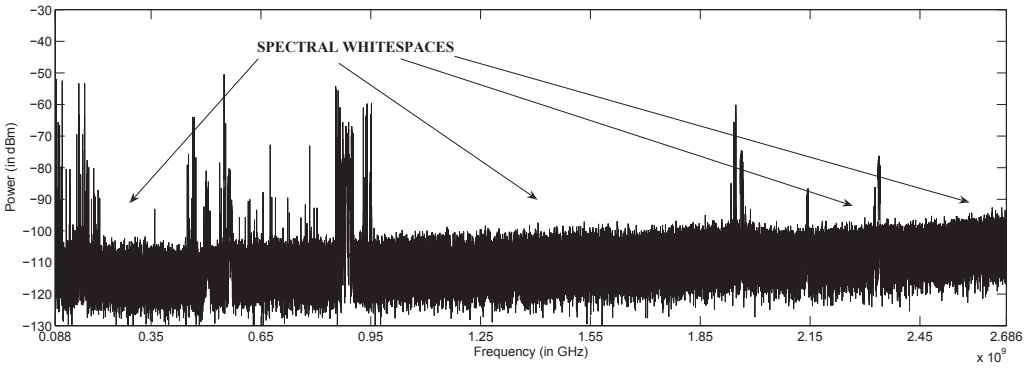


Fig. 2.2 A power spectral density snapshot of wireless spectrum ranging from 88 MHz to 2686 MHz measured on July 11, 2008, in Worcester, MA, USA at coordinates 42 °16′8″N, 71 °48′14″W (from [5]).

It is commonly believed that there is a crisis of spectrum availability at frequencies that can be economically used for wireless communications. This misconception is strengthened by a look at the FCC frequency chart [10], which shows multiple allocations over all of the frequency bands; which is a situation essentially also true worldwide. This has resulted in fierce competition for use of spectra, especially in the bands below 3 GHz. On the other hand, a large portion of the assigned spectrum is used sporadically as illustrated in Fig. 2.2, where the signal strength distribution over a large portion of the wireless spectrum is shown. The spectrum usage is concentrated on certain portions of the spectrum while a significant amount of the spectrum remains unutilized. This appears to be a contradiction to the concern of spectrum shortage since in fact we have an abundant amount of spectrum, and the spectrum shortage is partially an artifact of the regulatory and licensing process.

It is this discrepancy between FCC allocations and actual usage, which indicates that a new approach to spectrum licensing is needed. This approach should provide the incentives and efficiency of unlicensed usage to other spectral bands, while accommodating the present users who have higher priority or legacy rights (*primary users*) and enabling future systems a more flexible spectrum access [13]. This new approach is called *dynamic spectrum access*.

Dynamic spectrum access is the process of increasing spectrum efficiency via the real-time adjustment of radio resources, *i.e.* via a process of local spectrum sensing, probing, and the autonomous establishment of local wireless connections among cognitive nodes and networks. As originally proposed in [60], cognitive radio envisioned real time spectrum auc-

tions among diverse constituencies, using for one purpose, such as cellular radio, spectrum allocated and in use for another purpose such as public safety, and conversely, in order to multiply both the number of radio access points for public safety and to more efficiency use public safety spectrum commercially during peak periods.

2.1.3 Dynamic Spectrum Access Models

Standing for the opposite of the current static spectrum management policy, the term dynamic spectrum access has broad connotations that encompass various approaches to spectrum reform. The diverse ideas presented at the first IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN) suggest the extent of this term. As illustrated in Fig. 2.3, dynamic spectrum access strategies can be broadly categorized under three models [6].

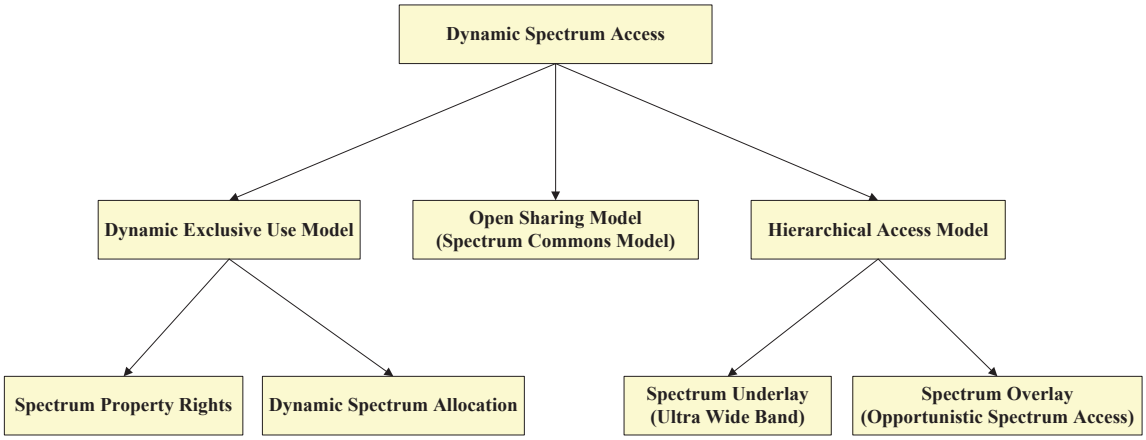


Fig. 2.3 Three models of dynamic spectrum access strategies (from [6]).

Dynamic Exclusive Use Model

This model maintains the basic structure of the current spectrum regulation policy: Spectrum bands are licensed to services for exclusive use. The main idea is to introduce flexibility to improve spectrum efficiency. Two approaches have been proposed under this model: Spectrum property rights and dynamic spectrum allocation. The former approach allows licensees to sell and trade spectrum and to freely choose technology. Economy and market will thus play a more important role in driving toward the most profitable use of

this limited resource. Note that even though licensees have the right to lease or share the spectrum for profit, such sharing is not mandated by the regulation policy.

The second approach, dynamic spectrum allocation, was brought forth by the European DRiVE project. It aims to improve spectrum efficiency through dynamic specturm assignment by exploiting the spatial and temporal traffic statistics of different services. In other words, in a given region and at a given time, spectrum is allocated to services for exclusive use. This allocation, however, varies at a much faster scale than the current policy.

Based on an exclusive use model, these approaches cannot eliminate white space in spectrum resulting from the bursty nature of wireless traffic.

Open Sharing Model

Also referred to as spectrum commons, this model employs open sharing among peer users as the basis for managing a spectral region. Advocates of this model draw support from the phenomenal success of wireless services operating in the unlicensed industrial, scientific, and medical (ISM) radio band (*e.g.*, WiFi). Centralized and distributed spectrum sharing strategies have been initially investigated to address technological challenges under this spectrum management model.

Hierarchical Access Model

This model adopts a hierarchical access structure with primary and secondary users. The basic idea is to open licensed spectrum to secondary users while limiting the interference perceived by primary users (licensees). Two approaches to spectrum sharing between primary and secondary users have been considered: Spectrum underlay and spectrum overlay.

The underlay approach imposes severe constraints on the transmission power of secondary users so that they operate below the noise floor of primary users. By spreading transmitted signals over a wide frequency band (UWB), secondary users can potentially achieve short-range high data rate with extremely low transmission power. Based on a worst-case assumption that primary users transmit all the time, this approach does not rely on detection and exploitation of spectrum white space.

Spectrum overlay was first envisioned by Mitola under the term spectrum pooling and then investigated by the DARPA Next Generation (XG) program under the term oppor-

tunistic spectrum access. Differing from spectrum underlay, this approach does not necessarily impose severe restrictions on the transmission power of secondary users, but rather on when and where they may transmit. It directly targets at spatial and temporal spectrum white space by allowing secondary users to identify and exploit local and instantaneous spectrum availability in a nonintrusive manner.

Compared to the dynamic exclusive use and open sharing models, this hierarchical model is perhaps the most compatible with the current spectrum management policies and legacy wireless systems. Furthermore, the underlay and overlay approaches can be employed simultaneously to further improve spectrum efficiency.

2.2 Spectrum Sensing

With advanced digital communication systems such as *cognitive radio* [59, 63] being used in a growing number of wireless applications, the topic of spectrum sensing has become increasingly important. This section will introduce the concept, the fundamental principles, and the practical applications of spectrum sensing.

In recent years, a large portion of the assigned spectrum has been observed to be sparsely and sporadically utilized during several spectrum measurement campaigns [11, 12], as illustrated in Fig. 2.2. In particular, spectrum occupancy by licensed transmissions are often concentrated across specific frequency ranges while a significant amount of the spectrum remains either underutilized or completely unoccupied. Therefore, *dynamic spectrum access* [6, 64] has been proposed for increasing spectrum efficiency via the real-time adjustment of radio resources using a combination of local spectrum sensing, probing, and autonomous establishment of local wireless connectivity among cognitive radio (CR) nodes and networks. During this process, *spectrum sensing* is employed for the purpose of identifying unoccupied licensed spectrum, *i.e.*, spectral "white spaces". Once these white spaces have been identified, secondary users (SU) opportunistically utilize these spectral white spaces by wirelessly operating across them while simultaneously not causing harmful interference to the primary users (PU) [1]. Currently, there exists several techniques for spectrum sensing. This chapter will emphasize on two of them, namely, the *energy detection* and

[1] Primary users are licensed users who are assigned with certain channels, and secondary users are unlicensed users who are allowed to use the channels assigned to a primary user only when they do not cause any harmful interference to the primary user [58].

cyclostationary feature detection.

2.2.1 Power Spectral Density

Power spectral density is a crucial concept in spectrum sensing. To analyze a signal in the frequency domain, the power spectral density (PSD), $S_{XX}(f)$, is often used to characterize the signal, which is obtained by taking the Fourier transform of the autocorrelation $R_{XX}(\tau)$ of the WSS random process $X(t)$. The PSD and the autocorrelation of a function are mathematically related by the *Einstein-Wiener-Khinchin* (EWK) relations [65], namely:

$$S_{XX}(f) = \int_{-\infty}^{\infty} R_{XX}(\tau) e^{-j2\pi f\tau}\, d\tau, \tag{2.1}$$

$$R_{XX}(f) = \int_{-\infty}^{\infty} S_{XX}(\tau) e^{+j2\pi f\tau}\, df. \tag{2.2}$$

A very powerful consequence of the EWK relations is its usefulness when attempting to determine the autocorrelation function or PSD of a WSS random process that is the output of a linear time-invariant (LTI) system whose input is also a WSS random process. Specifically, suppose we denote $H(f)$ as the frequency response of an LTI system $h(t)$. We can then relate the power spectral density of input and output random processes by the following equation:

$$S_{YY}(f) = |H(f)|^2 S_{XX}(f), \tag{2.3}$$

where $S_{XX}(f)$ is the PSD of input random process and $S_{YY}(f)$ is the PSD of output random process, as illustrated in Fig. 2.4.

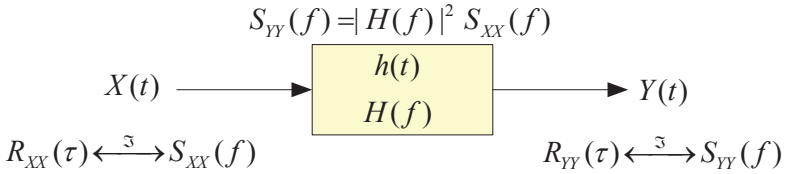
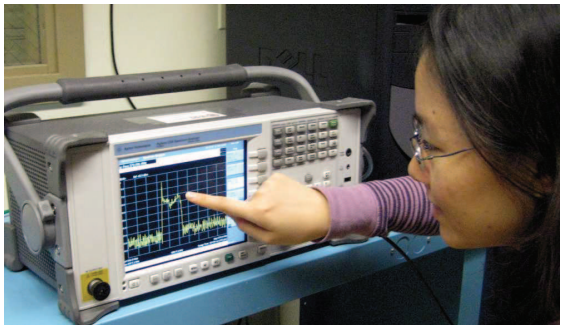


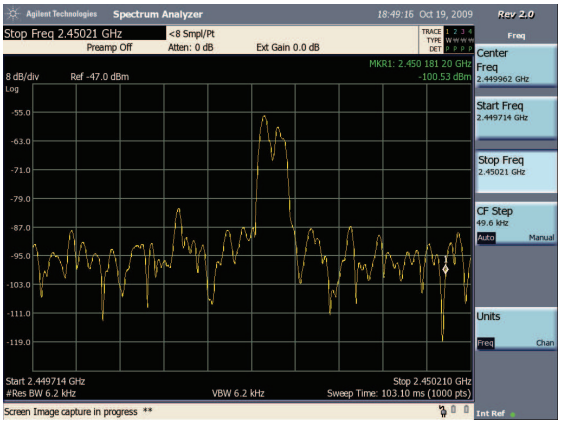
Fig. 2.4 An example of how the an LTI system $h(t)$ can transform the PSD between the WSS random process input $X(t)$ and the WSS random process output $Y(t)$.

2.2.2 Practical Issues of Collecting Spectral Data

Although spectrum sensing is a rather intuitive process, its implementation possesses several engineering trade-offs that can potentially impact the performance of the overall operation the results obtained. One of the key considerations when designing a spectrum sensing system is how to collect and store spectrum measurement data via a spectrum sweep. A spectrum analyzer can be employed to provide a nearly instantaneous snapshot of radio frequency bandwidth via the sampling of intercepted signals at a specified rate. For example, Fig. 2.5(a) shows that Agilent CSA-N1996A spectrum analyzer is used to take spectrum measurements and Fig. 2.5(b) shows the PSD of a pulse shaped QPSK signal collected by the spectrum analyzer.



(a) Using Agilent CSA-N1996A spectrum analyzer to take spectrum measurements (from [66]).



(b) Power spectral density of a pulse shaped QPSK signal, collected by an Agilent Technologies spectrum analyzer [67].

Fig. 2.5 A spectrum analyzer is employed to provide a snapshot of radio frequency bandwidth.

There are several practical issues when parameterizing a spectrum sweep, such as the *sweep time* and *sweep resolution*, *i.e.*, the speed and detail at which the spectrum sweeps are obtained. Higher resolution sweeps result in longer sweep time, but provide a more accurate measurement of the spectrum. The sweeps used later in this chapter are the average of thousands of spectrum sweeps across a single bandwidth. Note that for the same measurement equipment, the speed at which spectrum measurements can be obtained

varies from seconds to hours and days depending on the choice of several sweep parameters. However, the selection of these sweep parameters is heavily dependent on what signals are being observed, what sort of characteristics are being sought after in the spectrum measurements, and how the spectrum measurement information will be post-processed afterwards. As a result, Table 2.1 summarizes several parameters used to define spectrum measurement processes, and the aspects of measurement they mainly effect. Then in the following subsection, we will understand how the choice of these parameters can directly impact the outcome of the measurements obtained.

Table 2.1 Five parameters used to define spectrum measurement processes.

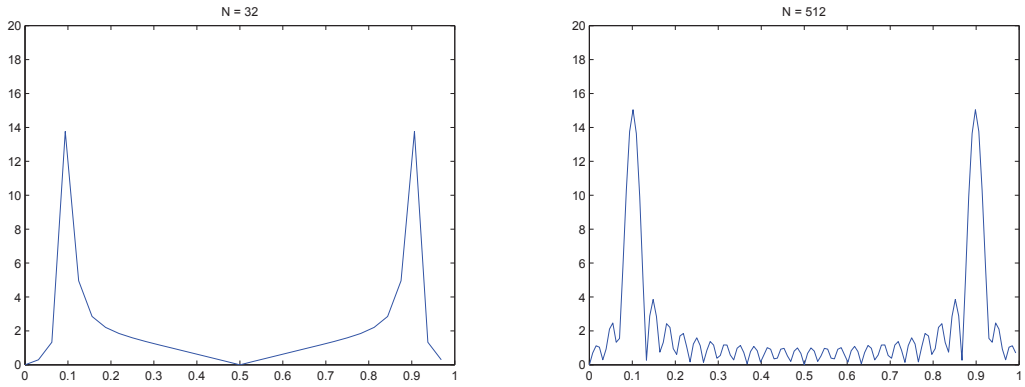
Parameter	*Related Equation*	*Main Impacted Aspect*
Sweep Resolution (δ)	$\delta = \frac{B_1}{N_{FFT}}$	accuracy
Window Size (W)	$N_{\text{step}} = \frac{B}{W}$	time
Dwell Time (t_{dwell})	$t_{\text{dwell}} = \frac{t_{\text{sweep}}}{N_{\text{sweep}}}$	time
Averaging Sweeps	$\lim_{N\to\infty}\{E[n(t)]\} \to 0$	accuracy
Sweep Time (t_{sweep})	$t_{\text{sweep}} = t_{\text{dwell}} \times N_{\text{sweep}}$	time

Sweep Resolution

In order to provide a snapshot of the radio frequency bandwidth, a spectrum analyzer samples the intercepted signals at a specified rate. Considering a unit time frame, each sample in this frame displays a degree of freedom. Given a larger number of samples in this frame, more information of the intercepted signals can be represented. Thus, for an FFT-based spectrum analyzer, given the bandwidth of one single sweep, the sweep resolution is decided by the number of FFT points, which can be expressed as:

$$\delta = \frac{B_1}{N_{FFT}}, \tag{2.4}$$

where B_1 is the bandwidth of one single sweep, and N_{FFT} is the number of FFT points. Based on (2.4), we observe that a larger number of FFT points will result in a higher sweep resolution. For example, in Fig. 2.6, there are two FFT plots of the same sine wave. Fig. 2.6(a) is generated using a 32-point FFT and Fig. 2.6(b) is a 512-point FFT. Comparing these two plots, we find that 512-point FFT provides a higher resolution, as well as a more accurate description of the spectrum than the 32-point FFT.



(a) A 32-point FFT plot. (b) A 512-point FFT plot.

Fig. 2.6 Two FFT plots of the same sine wave.

However, in order to handle the additional FFT samples, a spectrum analyzer that possesses a stronger computational processing ability is required, as well as a longer sensing time. Thus, there is an upper bound on the sweep resolution due to the limits of computational processing ability. On the other hand, a smaller FFT size may incur time-domain aliasing due to the undersampling in the frequency domain. Therefore, the input spectrum data is usually zero-padded to be consistent with the larger FFT size that provides the required resolution.

Window Size

The bandwidth of the spectrum we would like to analyze is usually larger than the bandwidth that one single sweep can support, so the concept of "window" needs to be introduced here. For each single sweep, the spectrum analyzer sweeps the size of a window, and then some post-processing approaches are employed to integrate these FFT results from the windows.

The bandwidth of the spectrum we would like to analyze can be calculated by:

$$B = f_{stop} - f_{start}, \tag{2.5}$$

where f_{stop} and f_{start} are two input parameters to the spectrum analyzer, which specify the stop frequency and the start frequency. The total number of the sweeps to cover the whole bandwidth is referred to as frequency steps, N_{step}, defined as:

$$N_{\text{step}} = \frac{B}{W}, \tag{2.6}$$

where B is the total bandwidth, and W is the window size. For example, in Fig. 2.7, the whole bandwidth is four times the window size, so the number of frequency steps is four.

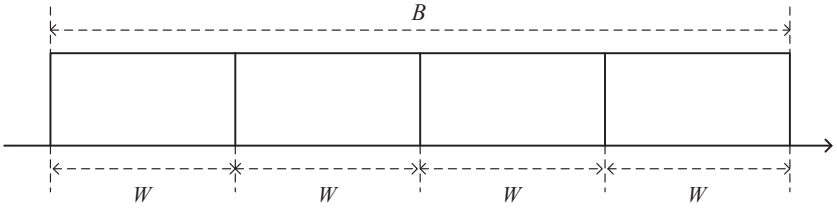


Fig. 2.7 A simple example showing the relation between the total bandwidth B and the window size W.

Equation (2.6) implies that the larger the window size, the fewer frequency steps are required. However, according to (2.4), for a fixed FFT size the sweep resolution would become lower. In addition, the larger window size implies a higher sampling frequency that the processing computer must be able to keep up with. Thus, although a larger window size can effectively reduce the number of sweeps, it may also incur some other issues that need to be taken into account.

Dwell Time

Given the window size, the dwell time is the amount of time that the spectrum analyzer spends in each "window" or sub-band. This is one of the specifications provided by most commercial spectrum analyzers, since it is more likely to generate an accurate spectrum sensing result if enough time is spent in each sub-band. The dwell time can be adjusted by changing the number of samples per FFT point input parameter to the spectrum. If this input value is set as N, then N samples will be collected in order to calculate the FFT for each frequency point. It usually holds that the larger the dwell time, the larger the overall

sweep time will be. These two factors can approximately be related by:

$$t_{\text{dwell}} = \frac{t_{\text{sweep}}}{N_{\text{sweep}}}, \tag{2.7}$$

where t_{dwell} is the dwell time, t_{sweep} is the overall sweep time, and N_{sweep} is the total number of the sweeps.

Effect of Averaging Sweeps

When analyzing electro-magnetic spectrum, we usually sweep the spectrum several times and then take the average of these values. This is due to the fact that the averaging process can smooth out the spectrum. Since we usualy assume the noise $n(t)$ to be an additive white Gaussian process with zero mean, as the number of samples approaches infinity, we will have:

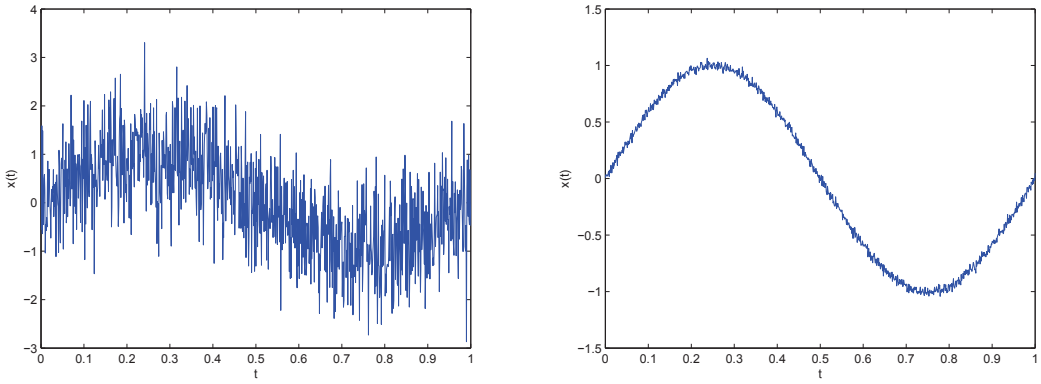
$$\lim_{N\to\infty} \{E[n(t)]\} \to 0, \tag{2.8}$$

where $E[\cdot]$ is the expectation operator, and N is the number of sweeps. For example, Fig. 2.8(a) shows a single sine wave whose amplitude is 1 with the added white Gaussian noise. Fig. 2.8(b) shows the result by accumulating 1000 sine waves in Fig. 2.8(a) and taking their average. Comparing these two plots, we find that the effect of additive white Gaussian noise can be greatly eliminated by the averaging process.

In real-world applications, as the number of sweeps used in the averaging process increases, the noise level will be better captured. This can be used to estimate the noise statistics and would be very useful when determining the energy threshold for the SDR implementation of an energy detector.

Sweep Time

Given that the dwell time is the time spent in each sub-band, we define the sweep time as the time spent in the entire bandwidth of interest. It is determined by several factors, including the dwell time, window size, and the number of sweeps. For the first factor, increasing the dwell time usually increases the sweep time. For the second factor, increasing the window size decreases the number of frequency steps according to (2.6), and hence decreases the sweep time. For a fixed window size, although increasing the FFT size



(a) A sine wave whose amplitude is 1 with the added white Gaussian noise.

(b) Accumulating 1000 sine waves and taking their average.

Fig. 2.8 The sine wave with and without the averaging process.

will result in a finer frequency resolution, the number of the frequency steps remains the same. Therefore, without accounting for the computational processing ability, the sweep time would be the same. However, in a real-world situation, with a larger FFT size, the resulting higher frequency resolution increases the computational processing time and hence the total sweep time. For the last factor, it is obvious that a larger number of the sweeps will result in a longer sweep time.

In conclusion, when collecting spectral data, we should consider all of the issues introduced above, in order to generate an accurate result, while at the same time be time-efficient.

2.2.3 Hypothesis Testing

As mentioned at the beginning of Section 2.2, in dynamic spectrum access networks, spectrum sensing is employed for the purpose of identifying unoccupied licensed spectrum, which is equivalent to detecting the frequency locations of the primary signals. Therefore, spectrum sensing can be interpreted as a signal detection problem. Since most signal detection problems can be formulated in the framework of an M-ary hypothesis test, where we have an observation (possibly a vector or function) upon which we wish to decide among M possible statistical situations describing the observations [68]. According to this criterion, the spectrum sensing performs a binary hypothesis testing in order to decide

whether or not there are primary signals in a particular channel. The two hypotheses are denoted as follows:

$$\begin{aligned} \mathcal{H}_0: &\quad \text{no primary signals,} \\ \mathcal{H}_1: &\quad \text{primary signals exist,} \end{aligned} \tag{2.9}$$

where $\mathcal{H}_0$ is usually referred to as null hypothesis, and $\mathcal{H}_1$ is usually called alternative hypothesis.

For a null hypothesis, since there are no primary signals present, the received signal is just the noise in the RF environment. On the other hand, for the alternative hypothesis, the received signal would be the superposition of the noise and the primary signals. Thus, the two hypotheses in (2.9) can be represented as:

$$\begin{aligned} \mathcal{H}_0: &\quad x[k] = n[k], \\ \mathcal{H}_1: &\quad x[k] = s[k] + n[k], \end{aligned} \tag{2.10}$$

for $k = 1, .., N$, where N is the number of received signals, $x[k]$ is the received signal, $n[k]$ is the noise in the RF environment, and $s[k]$ is the primary signal. Consequently, the spectrum sensing can be considered as such a detection problem, that based on the observation x, we need to decide among two possible statistical situations describing the observation, which can be expressed as:

$$\delta(x) = \begin{cases} 1 & x \in \Gamma_1, \\ 0 & x \in \Gamma_1^c. \end{cases} \tag{2.11}$$

When the observation x falls inside the region Γ_1, we will choose $\mathcal{H}_1$. However, if the observation falls outside the region Γ_1, we will choose $\mathcal{H}_0$. Therefore, (2.11) is known as *decision rule*, which is a function that maps an observation to an appropriate hypothesis [68]. In the context of spectrum sensing, different spectral detectors and classifiers are actualy the implementations of different decision rules. In Section 2.2.4, two decision rules will be introduced.

Regardless of the precise signal model or detector used, sensing errors are inevitable due to

additive noise, limited observations, and the inherent randomness of the observed data [69]. In testing $\mathcal{H}_0$ versus $\mathcal{H}_1$ in (2.9), there are two types of errors that can be made, namely $\mathcal{H}_0$ can be falsely rejected or $\mathcal{H}_1$ can be falsely rejected [68]. In the first hypothesis, there are actually no primary signals in the channel, but the testing detects an occupied channel, so this type of error is called a *false alarm* or *Type I error*. In the second hypothesis, there actually exist primary signals in the channel, but the testing detects only a vacant channel. Thus, we refer to this type of error as a *missed detection* or *Type II error*. Consequently, a false alarm may lead to a potentially wasted opportunity for the SU to transmit while a missed detection could potentially lead to a collision with the PU [69].

Given these two types of errors, the performance of a detector can be characterized by two parameters, namely, the *probability of false alarm* (P_F), and the *probability of missed detection* (P_M) [70], which correspond to Type I and Type II errors respectively, and thus can be defined as:

$$P_F = P\{\text{Decide } \mathcal{H}_1|\mathcal{H}_0\}, \tag{2.12}$$

and

$$P_M = P\{\text{Decide } \mathcal{H}_0|\mathcal{H}_1\}. \tag{2.13}$$

Note that based on P_M, another frequently used parameter is the *probability of detection*, which can be derived as follows:

$$P_D = 1 - P_M = P\{\text{Decide } \mathcal{H}_1|\mathcal{H}_1\}, \tag{2.14}$$

which characterizes the detector's ability to identify the primary signals in the channel, so P_D is usually referred to as the power of the detector.

As for detectors, we would like their probability of false alarm as low as possible, and at the same time, their probability of detection as high as possible. However, in real-world situation, this is not achievable, because these two parameters are constraining each other. To show their relationship, a plot called *receiver operating characteristic* (ROC) is usually employed [71], as shown in Fig. 2.9, where its x-axis is the probability of false alarm, and its y-axis is the probability of detection. From this plot, we observe that as P_D increases, the P_F is also increasing. There does not exist such an optimal point that reaches the highest

P_D and the lowest P_F. Therefore, the detection problem is also a trade off, which depends on how the Type I and Type II errors should be balanced.

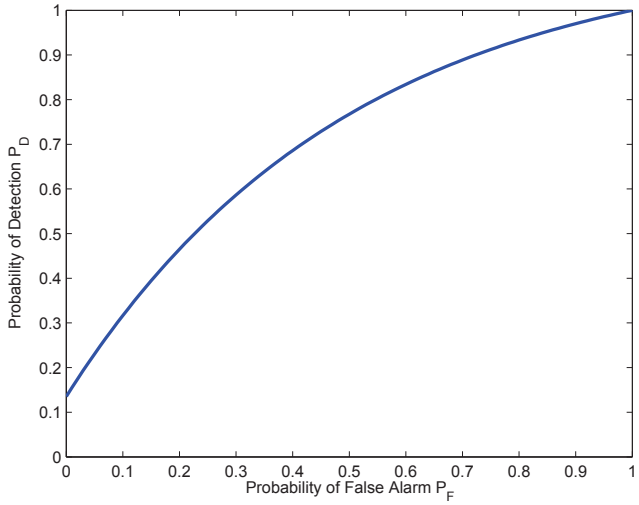


Fig. 2.9 A typical receiver operating characteristic (ROC), where the x-axis is the probability of false alarm (P_F), and the y-axis is the probability of detection (P_D).

2.2.4 Spectral Detectors and Classifiers

The spectral detectors and classifiers are implementations of the different decision rules. For example, the energy detector in Section 2.2.4 assumes that the received signal will have more energy when the channel is occupied than when it is vacant, thus the two decision regions in (2.11) are divided by an energy threshold. The cyclostationary feature detector in Section 2.2.4 is not only a detector, but also a classifier that recognizes the cyclic features of different modulation schemes in advance, so the decision regions are divided according to the cyclic features of the modulation schemes.

Energy Detector

Energy detection uses the energy spectra of the received signal in order to identify the frequency locations of the transmitted signal. This detection approach relies only on the

energy present in the channel, and no phase information is required. Since the energy of a signal $x(t)$ is defined as:

$$E = \int_{-\infty}^{\infty} |x(t)|^2 dt, \tag{2.15}$$

a decision statistic for energy detector can be:

$$D = \sum_{k=1}^{N} (x[k])^2, \tag{2.16}$$

where $x[k]$ is the received signal, and N is the total number of the received signals. The underlying assumption is that with the presence of a signal in the channel, there would be significantly more energy than if there was no signal present. Therefore, energy detection involves the application of a *threshold* T in the frequency domain, which is used to decide whether a transmission is present at a specific frequency, as shown in Fig. 2.10. Any portion of the frequency band where the energy exceeds the threshold is considered to be occupied by a transmission, namely, the decision rule is as follows:

$$\delta(D) = \begin{cases} 1 & D > T, \\ 0 & D < T. \end{cases} \tag{2.17}$$

where D is the decision statistic calculated by (2.16) and T is the pre-defined energy threshold. Therefore, in Fig. 2.10, any portion of the frequency band where the energy exceeds -100 dBm is considered to be an occupied channel.

Since different transmitters employ different signal power levels and transmission ranges, one of the major concerns of energy detection is the selection of an appropriate threshold. A threshold that works for one transmission may not be appropriate for another. Fig. 2.11 shows two typical detection errors caused by inappropriate energy detection threshold. In Fig. 2.11(a), the threshold is too low, so some noise is considered as primary signals, resulting in Type I error, false alarm. While in Fig. 2.11(b), the threshold is too high, so some primary signals are ignored, incurring the Type II error, missed detection.

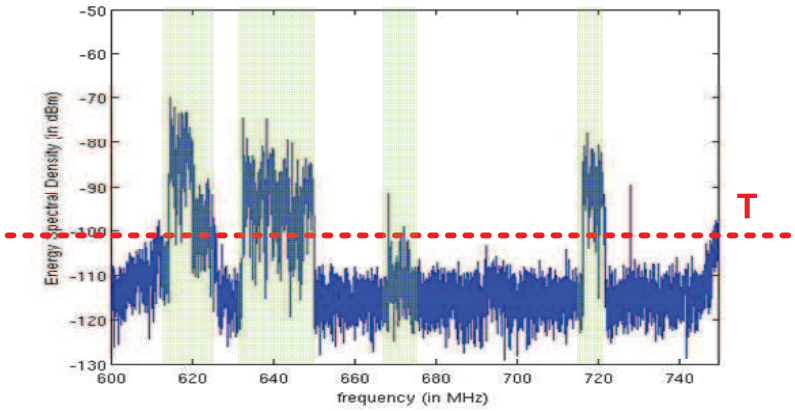
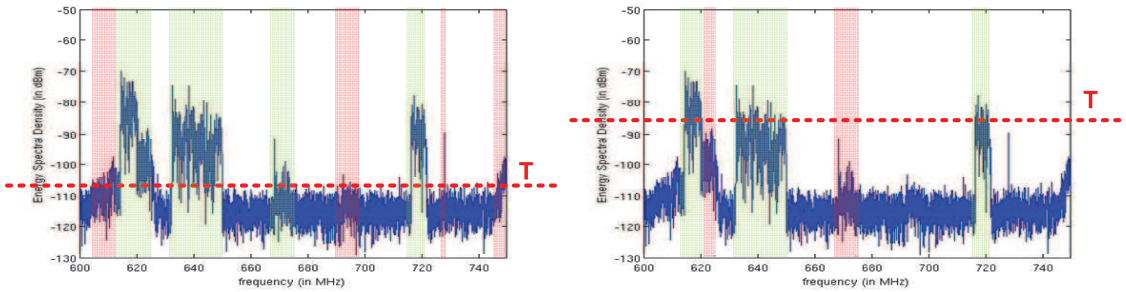


Fig. 2.10 Energy detection threshold level, denoted as T. Any portion of the frequency band where the energy exceeds the threshold is considered to be occupied by a transmission.



(a) Low energy detection threshold level yields Type I error, *false alarm*.

(b) High energy detection threshold level yields Type II error, *missed detection*.

Fig. 2.11 Inappropriate energy detection threshold levels.

Cyclostationary Feature Detector

Modulation recognition and signal classification has been a subject of considerable research for over two decades. Classification schemes can generally be separated into one of two broad categories: likelihood-based (LB) approaches and feature-based (FB) approaches [72]. Cyclostationary feature detection is an FB technique based on the fact that communications signals are not accurately described as a stationary process, but rather more appropriately modeled as a cyclostationary process [73].

A cyclostationary signal possesses statistics that vary periodically with time. By definition,

a signal $x(t)$ is wide-sense cyclostationary if its mean and autocorrelation are periodic [74], namely:

$$M_x(t) = M_x(t + T_0), \tag{2.18}$$

and

$$R_x(t, \tau) = R_x(t + T_0, \tau), \tag{2.19}$$

where $M_x(t)$ is the mean value of the signal $x(t)$, and $R_x(t, \tau)$ is the autocorrelation function of the signal $x(t)$. These periodicities occur for signals possessing well-defined characteristics due to several processes such as sampling, scanning, modulating, multiplexing, and coding, which can be exploited to determine the modulation scheme of the unknown signal [73].

The periodic nature of the signal allows it to be expressed as a Fourier series [74, 75]:

$$R_x(t, \tau) = E\{x(t + \frac{\tau}{2})x^*(t - \frac{\tau}{2})\} = \sum_{\{\alpha\}} R_x^\alpha(\tau)e^{j2\pi\alpha t}, \tag{2.20}$$

where $E\{.\}$ is the expectation operator, $\{\alpha\}$ is the set of Fourier components, and $R_x^\alpha(\tau)$ is the cyclic autocorrelation function (CAF) given by:

$$R_x^\alpha(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} R_x(t, \tau)e^{-j2\pi\alpha t}. \tag{2.21}$$

Alternatively, in the case when $R_x(t, \tau)$ is periodic in t with period T_0, (2.21) can be expressed as:

$$R_x^\alpha(\tau) = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} R_x(t, \tau)e^{-j2\pi\alpha t}. \tag{2.22}$$

Consequently, the Fourier transform of the CAF, which is referred to as the spectral correlation function (SCF), is given by:

$$S_x^\alpha(f) = \int_{-\infty}^{\infty} R_x^\alpha(\tau)e^{-j2\pi f\tau} d\tau, \tag{2.23}$$

which can be shown to be equivalent, assuming cycloergodicity, to the following expres-

sion [74]:

$$S_X^{\alpha}(f) = \lim_{T\to\infty} \lim_{\Delta t\to\infty} \frac{1}{\Delta t} \int_{-\Delta t/2}^{\Delta t/2} \frac{1}{T} X_T(t, f + \frac{\alpha}{2}) X_T^*(t, f - \frac{\alpha}{2}) dt, \tag{2.24}$$

where $X_T(t, f)$ is the time varying Fourier transform defined as:

$$X_T(t, f) = \int_{t-T/2}^{t+T/2} x(u) e^{j2\pi f u} du. \tag{2.25}$$

A significant advantage of the SCF is its lack of sensitivity to additive white noise, since modulated information is a cyclostationary process, while noise is not. In other words, the spectral components of white noise are uncorrelated, so it does not contribute to the resulting SCF for any value of $\alpha \neq 0$. This property is especially useful when the noise power exceeds the signal power, which would make the signal undetectable when using a simple energy detector. As a result, cyclic detectors can successfully operate even in low SNR environments.

To derive a normalized version of the SCF, the spectral coherence function (SOF) is given by:

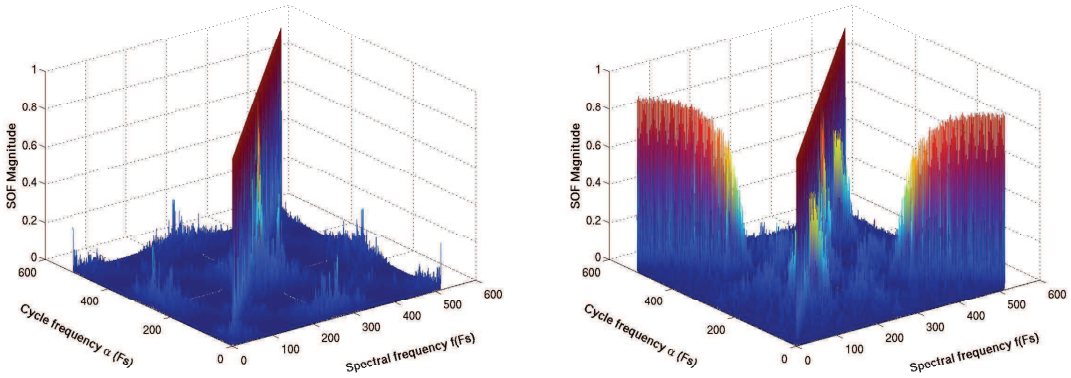
$$C_X^{\alpha}(f) = \frac{S_X^{\alpha}(f)}{[S_X^0(f + \alpha/2) \times S_X^0(f - \alpha/2)]^{1/2}}. \tag{2.26}$$

Therefore, the three important functions derived in cyclostationary feature detector can be summarized as in Table 2.2.

Table 2.2 Three important functions derived in cyclostationary feature detector.

Function Name	*Characteristic*
Cyclic Autocorrelation Function (CAF) $R_x^{\alpha}(\tau)$	Fourier series of autocorrelation
Spectral Correlation Function (SCF) $S_x^{\alpha}(f)$	Fourier transform of the CAF
Spectral Coherence Function (SOF) $C_X^{\alpha}(f)$	normalized version of the SCF

The magnitude of the SOF varies from 0 to 1 and represents strength of second order periodicity within the signal. The SOF contains the spectral features of interest. These features are non-zero frequency components of the signal at various cyclic frequencies. All modulation schemes contain a range of spectral components at different cyclic frequencies,



(a) SOF of QPSK signal in an AWGN channel at 10 dB SNR.

(b) SOF of 4PAM signal in an AWGN channel at 10 dB SNR.

Fig. 2.12 Distinctive cyclic features of different modulation schemes.

thus distinguishing them from other modulation schemes, *i.e.*, the spectral components form a spectral fingerprint for the specific modulation scheme. The SOFs of two typical modulation schemes, QPSK and 4PAM, are shown in Fig. 2.12(a) and Fig. 2.12(b). Notice how the SOF for each modulation scheme generates a highly distinct spectral image. These distinctions allow signals to be classified from cyclic analysis. In general, the plot of SOF is a three-dimensional figure, where the x-axis represents the cyclic frequency α, the y-axis represents the spectral frequency f, and the z-axis represents the corresponding magnitude of the SOF for each (α, f) pair. Note that when α does not equal zero, the SOF values are approximately zero.

Cyclostationary feature detector can be implemented via FFTs. Knowledge of the noise variance is not required to set the detection threshold. Hence, the detector does not suffer from the "SNR wall" problem of the energy detector. However, the performance of the detector degrades in the presence of timing and frequency jitters (which smear out the spectral lines), and RF non-linearities (which induce spurious peaks). Representative papers that consider the approach are [43, 76, 77].

2.3 Primary User Emulation

One of the major technical challenges regarding spectrum sensing is the problem of accurately distinguishing primary user signals from secondary user signals. In cognitive radio networks, primary users possess the priority to access the channel, while secondary users must always relinquish access to the channel over to the primary user and ensure that no interference is generated. Consequently, if a primary user begins to transmit across a frequency band occupied by a secondary user, the secondary user is required to leave that specific spectral band immediately. Conversely, when there is no primary user activity present within a frequency range, all the secondary users possess equal opportunity to the unoccupied frequency channel. Based on this principle, there exists the potential for malicious secondary users to mimic the spectral characteristics of the primary users in order to gain priority access to the wireless channels occupied by other secondary users. This scenario is referred to in the literature as *primary user emulation* (PUE) [30–32].

2.3.1 An PUE Example

Fig. 2.13 shows a simple but classic example of primary user emulation attack in a dynamic spectrum access network. In this network, there are three normal secondary users, named $D1$, $D2$ and $D3$. They are communicating with each other using the "white space" channels. $D1$ and $D2$ are using Channel 1, $D2$ and $D3$ are using Channel 2, and $D1$ and $D3$ are using Channel 3. At this time, a malicious secondary user, *i.e.*, a primary user emulator appears on Channel 3. Since this malicious secondary user mimics the spectral characteristics of the primary users, $D1$ and $D3$ think that there is a primary user transmitting on this channel. According to the criteria of dynamic spectrum access network, $D1$ and $D3$ have to leave Channel 3 immediately. However, the other two channels are both occupied by the other users right now, so they cannot find any other available channels to continue their communication, making the connection terminated.

2.3.2 Impact of PUE on DSA Networks

Based on the discussion above, there are several outcomes that can be incurred by PUE attacks in a dynamic spectrum access network:

- Unstable Connections: Based on the example presented in Section 2.3.1, one of the

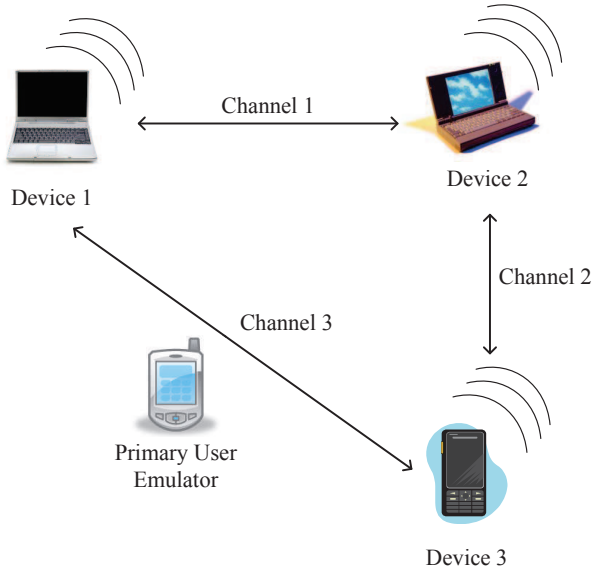


Fig. 2.13 A simple example of primary user emulation attack in a dynamic spectrum access network, where normal secondary users $D1$ and $D3$ relinquish their access to Channel 3 over to the primary user emulator.

most direct outcomes is unstable connections between secondary users. If a network is frequently attacked by a primary user emulator, the secondary users in this network always have to leave their current channels and seek new channels. However, it is very likely that other channels are also occupied, so their connections have to be terminated.

- Spectrum Under-utilization: The original purpose of dynamic spectrum access is to address the problem of spectrum scarcity caused by FCC's fixed spectrum allocation. In DSA, the secondary users can temporarily borrow unoccupied licensed spectrum. However, if there are severl primary user emulators in the network, it is possible that all the available licensed channels are occupied by them, so the normal SU cannot find any channels to borrow. If it is the case, then the problem of spectrum scarcity is not solved at all.

- Denial of Service: In the current setting of FCC's DSA approach presented in Section 1.3, when a secondary user wants to transmit some data, it has to go through a request and acknowledgement process. However, if all the channels are occupied

by the primary user emulators, the normal SU cannot even find a channel to send a request, so their service will be denied.

- Interferance with Primary Users: Although the PUE attacks are solely aimed at secondary users, and the primary user emulators are supposed to obey the rule that they will not cause any interferences with the primary users. However, in a dynamic spectrum access network, the PU and SU exit in the same network, so any user's activates would have some impact on the others. Especially since primary user emulators mimic the spectral characteristics of the primary users, their transmission power is usually higher than that of the normal secondary users, so it can cause an interference with the primary users.

It is noted that PUE is different from traditional jamming in wireless networks. The malicious users do not aim to cause significant interference to the secondary users. The objective of the malicious users is to cause the secondary users to vacate the spectrum by having them believe that primary transmission is in progress. Thus, when PUE is successfully detected, the secondary users do not suffer degradation in the quality of their communication due to the transmission from the malicious users. [54]

2.4 Artificial Neural Network

Artificial neural networks (ANNs) can generally be thought of as an approximation or fitting function [78]. The robustness and generalization capabilities of ANN models make them good candidates for signal detection and classification.

2.4.1 Artificial Neural Networks

Artificial neural networks attempt to simulate the way that neurons in a brain work. A neural network consists of a set of elements called *neurons* or *nodes*, that are connected to each other. Each connection has a weight or strength value associated with it, and these values determine the state of the neural network [79]. An artificial neural network can be set up by three steps [78]:

1. **Input-Output Definition** Define a set of inputs x and a set of outputs y. This definition is critical in the overall model accuracy as the output y may not be sensitive to some input parameters defined in the vector x.

2. **Training ANN Models** Accurate data samples (x_p, d_p), where p is the pth sample in a data set, are employed for training the ANN. To this end, several training algorithms are available, including quasi-Newton (QN), quasi-Newton MLP (QNM), conjugate gradient algorithm, conjugate gradient quasi-Newton algorithm (CQN) and the conjugate gradient quasi-Newton MLP algorithm (CQNM).

3. **Validation and Testing ANN Models** To verify the accuracy of the trained ANN, the model is tested against a set of new data samples.

After these three steps, we can proceed to classify the signals using this artificial neural network.

In this dissertation, a multi-layer perceptron (MLP) neural network is employed. This type of network consists of a series of nodes arranged into layers as shown in Fig. 2.14. Each node is connected to all the nodes in the layer before it, and all the nodes in the layer after it. Each connection has a weight value associated with it. The nodes in the first layer are called input nodes, while the nodes in the last layer are referred to as the output nodes. Note that all the other nodes are called hidden nodes. The activation of the hidden nodes (H_j) is calculated as follows:

$$H_j = f(\sum_{i=1}^{ni} w_{1ij} I_i), \tag{2.27}$$

where I_i is the activation of the ith input node, ni is the total number of input nodes, w_{1ij} is the weight number associated with the connection between hidden node i and input node j, and f is a function that smooths the resultant activation and bounds it between -1 and 1. In this dissertation, we assume that $f(x) = \tanh(x)$.

Once the activation of the hidden nodes have been calculated, the activation of the output nodes are calculated by:

$$O_j = f(\sum_{i=1}^{nh} w_{2ij} H_i), \tag{2.28}$$

where nh is the number of hidden nodes, and w_{2ij} is the weight factor associated with the connection between the ith hidden node and the jth output node.

The process of training a neural network refers to choosing the appropriate weight factors in such a way that a specific input gives rise to the desired output. Consequently,

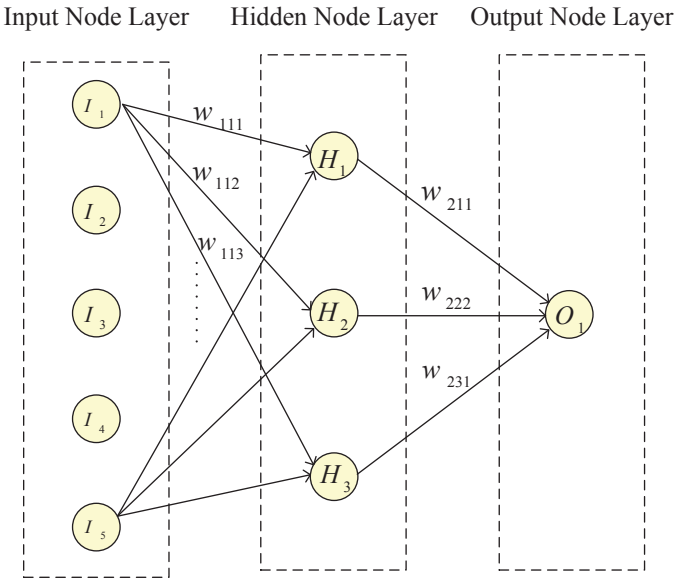


Fig. 2.14 The structure of the nodes in a Multi Layer Perceptron (MLP) neural network. There are 5 input nodes, one layer of 3 hidden nodes, and 1 output node in this network.

given several sets of data and the desired output for each set, the training of the neural network can be rephrased as the minimizing of the error, which is given by:

$$E(I_1, I_2, \ldots, I_{ni}) = \sum_{i=1}^{no} (O_i - d_i), \tag{2.29}$$

where no is the number of output nodes, and d_i is the desired value of the ith output. In this dissertation, the training algorithm employed to minimize the error is the back propagation (BP) algorithm [80]. For this algorithm, the learning rate η controls how quickly the algorithm converges. A higher learning rate corresponds to a quicker convergence. However, if the learning rate is too high, then the algorithm will oscillate and not converge at all. To prevent oscillations, the momentum ζ forces the algorithm to take into account its movement from the previous iteration. By doing so, the system will tend to avoid local minima or saddle points, and approach the global minimum. The learning rate and momentum are used in the following equation to determine the change of weight value:

$$\Delta_i = \eta \times \delta_i + \zeta \times \Delta_{i-1}, \tag{2.30}$$

where δ_i is the amount that the weight has to change to approach the minimum of the error, and Δ_i is the amount that the weight value was actually changed during the ith iteration.

2.4.2 Artificial Neural Networks in Signal Classification

Creating a linear algorithm to classify signals in real time is a difficult problem due to the unique features of each signal that need to be programmed in, and the ambient noise level that needs to be considered. However, this problem can be solved by using an artificial neural network approach due to the following reasons: First, an artificial neural network can simply be trained to recognize the unique features of the signal by giving it sample signal data, and telling it what signal type to associate with each set of data [79]. Second, artificial neural networks have been shown to have a high noise tolerance if the data to be classified is not very similar [81]. Third, artificial neural networks are typically able to run fast enough for real time applications, since the output activation defined in [82] is a very simple expression.

To actually use an artificial neural network to classify a signal, the following steps need to be taken: First, the signal must be clearly intercepted. Then, signal statistics need to be computed. For example, in the first approach of this dissertation, the cycle frequency profile of the SOF is used, which is defined as:

$$\text{profile}(\alpha) = \max_{\text{f}} \left[\text{C}_{\text{X}}^{\alpha}(\text{f})\right], \tag{2.31}$$

where $\text{C}_{\text{X}}^{\alpha}$ is the SOF of the signal, as defined in (2.26). This profile reduces the two dimensional SOF data to one dimension, thus can be processed in real time. Next, this data is fed into a system of artificial neural networks, each of which is trained to identify a given signal. In the end, the network with the largest output activation is found, and the identity of the signal is known.

Additionally, a reliability parameter, similar to the one discussed in [82], can be used to detect whether the artificial neural network has failed. This parameter is defined as half of the difference between the largest and second largest output activations:

$$\chi = \frac{O_{\text{Largest}} - O_{\text{2ndLargest}}}{2}, \tag{2.32}$$

where O_{Largest} is the value of the largest output activation, and $O_{\text{2ndLargest}}$ is the value of the second largest output activation. Therefore, if one artificial neural network had an activation of 1, which implies a perfect match, and all the others had an activation of -1, which implies that there is no match, then the reliability is 1. On the other hand, if two of the artificial neural networks had an activation of 1, then the reliability would be 0. If the reliability is close to 0, then there is a good chance that the artificial neural network has incorrectly classified the signal, and the classification should be disregarded. By discarding suspect classifications, the percentage of correct classifications can be improved.

2.5 Action Recognition in Video

Action recognition in video stands for the problem of recognizing different types of human actions, such as walking, running, jumping, waving, etc., from video footage. For example, given a human action sequence shown in Fig. 2.15, how can the computer tell which action it belongs to?

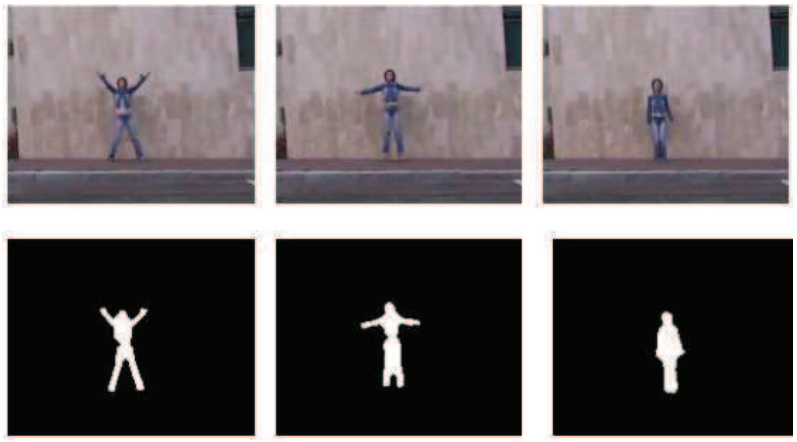


Fig. 2.15 Example of a human action sequence: Three frames from a "jumping-jack" action sequence (top row) and corresponding silhouettes (bottom row) from the Weizmann Human Action Database (from [7]).

A variety of methods have been explored to obtain different features to represent actions. The goal of these features is to extract significant and discriminative information for action classification. Although action characteristics are implicitly embedded within raw image sequences, it is difficult to classify actions by using raw images that may have very different

chromatic, photometric and textural properties even if they consist of the same type of action. It is therefore natural to utilize features that more explicitly describe actions, *e.g.*, motion-related features that are usually chromatic, photometric and textural invariant [83, 84]. In this section, we present the three-step action representation framework adopted from [83] and [84]. We follow the mathematical expressions and notations from [84].

2.5.1 Descriptor for Action Representation

Given an action segment, [83] and [84] first extract a rich collection of 12-dimensional feature vectors that describe the property of motion. Components of the feature vector consist of different motion characteristics. Therefore, action is embedded within the feature space. The second step is a dimension reduction step: [83] and [84] obtain a simplified action representation by fusing the collection of feature vectors into a 12×12 empirical covariance matrix, which can be viewed as a compact feature descriptor in a much lower dimensional space. Covariance matrices lie in a Riemannian manifold, not a vector space; thus, they are not directly amenable to the artifical neural network. In order to use the covariance matrices with the artifical neural network, a third step of the matrix logarithm is necessary. The resulting log-covariance matrix lies in the vector space of symmetric matrices. These three steps are elaborated as following:

Feature Vector Construction

Let $I(x, y, t)$ denote the raw image sequence and let $\mathbf{u}(x, y, t)$ represent the corresponding flow vector (u, v) at each pixel position (x, y, t). Let A denote the set of coordinates of all pixels which belong to an action segment which is W pixels wide, H pixels tall, and N frames long, *i.e.*, $A := \{(x, y, t)^T : x \in [1, W], y \in [1, H], t \in [1, N]\}$. Based on $I(x, y, t)$ and $\mathbf{u}(x, y, t)$, the feature vector $\mathbf{f}(x, y, t)$ is defined in [84] as:

$$\mathbf{f}(x, y, t) := [x, y, t, I_t, u, v, u_t, v_t, Div, Vor, Gten, Sten]^T, \tag{2.33}$$

where $(x, y, t) \in A$. I_t is the first order partial derivative of $I(x, y, t)$ with respect to t, *i.e.*, $I_t = \frac{\partial I(x,y,t)}{\partial t}$. u and v are optical flow components, and u_t and v_t are respectively the first order partial derivative of $u(x, y, t)$ and $v(x, y, t)$ with respect to t. Div is the spatial

divergence of a flow field and is defined at each pixel position as:

$$Div(x,y,t) = \frac{\partial u(x,y,t)}{\partial x} + \frac{\partial v(x,y,t)}{\partial y}. \tag{2.34}$$

Divergence can capture the amount of local expansion in the fluid, which can indicate action differences. Vor is the vorticity of a flow field and is defined as:

$$Vor(x,y,t) = \frac{\partial v(x,y,t)}{\partial x} - \frac{\partial u(x,y,t)}{\partial y}. \tag{2.35}$$

In fluid dynamics, vorticity is used to measure local spin around the axis perpendicular to the plane of the flow field; thus, it is useful to highlight local circular motion of the moving object. Before explaining $Gten$ and $Sten$, two matrices need to be defined, called gradient tensor of optical flow $\nabla\mathbf{u}(x,y,t)$ and rate of strain tensor $S(x,y,t)$:

$$\nabla\mathbf{u}(x,y,t) = \begin{pmatrix} \frac{\partial u(x,y,t)}{\partial x} & \frac{\partial u(x,y,t)}{\partial y} \\ \frac{\partial v(x,y,t)}{\partial x} & \frac{\partial v(x,y,t)}{\partial y} \end{pmatrix}, \tag{2.36}$$

$$S(x,y,t) = \frac{1}{2}(\nabla\mathbf{u}(x,y,t) + \nabla\mathbf{u}(x,y,t)^T). \tag{2.37}$$

$Gten$ and $Sten$ are tensor invariants that remain constant no matter what coordinate system they are referenced in. They can be written as follows:

$$Gten(x,y,t) = \frac{1}{2}(tr(\nabla\mathbf{u}(x,y,t))^2 - tr(\nabla\mathbf{u}(x,y,t)^2)), \tag{2.38}$$

$$Sten(x,y,t) = \frac{1}{2}(tr(S(x,y,t))^2 - tr(S(x,y,t)^2)), \tag{2.39}$$

where tr stands for the trace operation.

So far, a feature vector at each pixel is represented by (2.33). However, only some of these feature vectors are related to the action of a moving object while the remaining feature vectors just indicate background characteristics (that have no contribution or negative impact on action classification). Therefore, we want to include only those feature vectors whose corresponding pixels lie within the moving object. In this dissertation, following [84], we determine the approximate locations of moving pixels by thresholding the smoothed temporal gradients I_t. Only pixels with I_t greater than some threshold are claimed as

moving pixels. As for the database, only the feature vectors of primary users' moving pixels will be included.

Covariance Descriptor of Feature Vectors

The collection of feature vectors from each moving pixel position provides a dense representation of the action in a segment. Although these dense feature vectors consist of a great amount of motion characteristics, they are inconvenient in action classification because they lie in a very high dimensional space. Recently, [85] and [86] proposed to compare two sets of feature samples by computing and comparing their empirical covariance matrices. The covariance matrix of feature vectors enables efficient fusion of different types of features and its dimensionality is small. Moreover, since feature properties are captured in a single covariance matrix, feature sets of different sizes can be easily and efficiently compared. Therefore, the action properties of an action segment are captured by a 12×12 covariance matrix C_S.

Let S denote the set of coordinates that are related to a moving object, and $|S|$ denote the number of elements in S. Then C_S is defined in [84] as:

$$C_S = \frac{1}{|S|} \sum_{(x,y,t)\in S} (\mathbf{f}(x,y,t) - \mu_S)(\mathbf{f}(x,y,t) - \mu_S)^T, \tag{2.40}$$

where

$$\mu_S = \frac{1}{|S|} \sum_{(x,y,t)\in S} \mathbf{f}(x,y,t) \tag{2.41}$$

is the mean feature vector. Since C_S is a 12×12 symmetric matrix, only its $(12^2+12)/2 = 78$ entries are independent thus affording a low-dimensional representation of all feature samples, independently of their number.

Log-covariance Descriptor of Feature Vectors

The covariance matrix C_S computed in (2.40) does not lie in a vector space because -C_S is not a covariance matrix. [87] proposed the use of the matrix logarithm to map the manifold of covariance matrices into the vector space of symmetric matrices. This property is important since only features in the vector space are amenable to the artificial neural network. The log-covariance matrix L_S of a covariance matrix C_S is computed in [84] as

follows:

Suppose that the eigen-decomposition of C_S is given by $C_S = VDV^T$, where the columns of V are orthonormal eigenvectors and D is the diagonal matrix of eigenvalues. Then the log-covariance matrix is

$$L_S = \log(C_S) = V\tilde{D}V^T, \tag{2.42}$$

where $\tilde{D}$ is a diagonal matrix obtained from D by replacing D's diagonal entries by their natural logarithms.

Since matrix L_S has only 78 independent entries, we scan its upper triangular part and form a vector $\mathbf{l}_S$ to be used as an action descriptor in the following artificial neural network.

2.6 Relational Database

Database systems are important data management tools, which store, organize and let users to query stored data in a structural manner. Databases can store information about people, products, orders, or anything else. In the last few decades, they have been widely used in people's daily life to serve various data management goals. Most of the database systems use relational model, and thus being called relational databases [88]. In this section, we will first discuss the general database, and then focus on the relational database and its design.

2.6.1 Database

Database is a very large, integrated collection of data, which models real-world applications. Usually data is too large to fit into main memory, and often used by many users.

Many databases start as a list in a word-processing program or spreadsheet. As the list grows bigger, redundancies and inconsistencies begin to appear in the data. The data becomes hard to understand in list form, and there are limited ways of searching or pulling subsets of data out for review. Once these problems start to appear, it's a good idea to transfer the data to a database created by a database management system (DBMS), such as MySQL.

Nowadays, the most frequent applications of DBMS include the following: E-commerce like Amazon.com, airlines and travel services, scientific data such as biology, oceanography,

etc., spatial data such as maps, travel networks, and digital libraries.

2.6.2 Relational Database

Relational database was proposed by Edgar Codd (of IBM Research) around 1969 [89]. It has since become the dominant database model for commercial applications (in comparison with other database models such as hierarchical, network and object models). Today, there are many commercial Relational Database Management System (RDBMS), such as Oracle, IBM DB2 and Microsoft SQL Server. There are also many free and open-source RDBMS, such as MySQL, mSQL (mini-SQL) and the embedded JavaDB (Apache Derby).

A relational database organizes data in tables (or relations). A table refers to a two dimensional representation of your data using columns and rows. A row is also called a record (or tuple). A column is also called a field (or attribute). In other words, in relational databases, data are structured as individual records, each representing a multi-dimensional data item. Individual records with same attributes are collected together to form tables, which are the basic accessable units in the databases. Each database table is given a unique name. Without a unique name, the DBMS would get very confused. Each column in the table is also given a unique name. For example, in our approach, a record regarding a feature vector should have 12 columns (attributes), namely, $x, y, t, I_t, u, v, u_t, v_t, Div, Vor, Gten, Sten$, as shown in Fig. 2.16. However, it doesn't mean that each column you name has to be unique within the entire database. It only has to be unique within the table you have created. Also notice that the names should not include any spaces. When naming tables and columns, it is recommended to keep it simple with only letters and numbers, since spaces and symbols can be illegal characters.

x	y	t	I_t	u	v	u_t	v_t	Div	Vor	$Gten$	$Sten$

Fig. 2.16 A record regarding a feature vector has 12 columns.

As shown in Fig. 2.16, a database table is similar to a spreadsheet. However, the relationships that can be created among the tables enable a relational database to efficiently store huge amount of data, and effectively retrieve selected data. A language called SQL (Structured Query Language) [90] was developed to work with relational databases in a

way that relational databases usually use standarded SQL language to access the data in the databases by submitting queries. The SQL langauge is a very powerful language, which can express complicated logics [91].

Organizing data into tables brings many benefits for data management. Here are several most important ones: First of all, each table is a good encapsulation for information in one aspect, such that users can easily locate the information in a certain aspect by querying the corresponding table. For our database, if we want the feature vectors of a certain primary user, we only need to look into one corresponding table. Second, it breaks the potentially large piece of information into relatively smaller and independent pieces, and thus effectively reduce the amount of redundant informaiton stored and the disk I/O cost in most of the cases. Therefore, in our approach, as long as we have located a record in a table, we do not need to read the remaining tables, which lowers the disk I/O cost. Third, one can easily add new tuples to a database without disturbing the existing parts of it. For example, if our system needs to incorporate a new primary user, it can easily create a new table including the feature vectors of this new user, without changing anything else in the database.

2.6.3 Relational Database Design

There are many requirements for a well-designed database. Among them, the most important ones are eliminating data redundancy and ensuring data integrity and accuracy. To meet this end, there are several important constraints to take into account when designing a relational database.

Primary Key

To better coordinate the relationships among tuples in a single table, relational model introduces the constraints on the table. The most important and widely used constraint is key. In the relational model, a table cannot contain duplicate rows, because that would create ambiguities in retrieval. To ensure uniqueness, each table should have a column (or a set of columns), called primary key, that uniquely identifies every records of the table. In our approach, given a pixel location and a time instant, we can find a unique feature vector in the table, so (x, y, t) is a key for the table. In other words, if a (x, y, t) value is provided, we can uniquely identify a record in this table. The key constraint in the table

mainly helps to prevent the tuple duplication in the table, so that we will not waste any time on scanning the duplicate informaiton.

Foreign Key

A database consisting of independent and unrelated tables serves little purpose. The power of relational database lies in the relationship that can be defined between tables. The most crucial aspect in designing a relational database is to identify the relationships among tables, where the concept of the foreign key serves an important role in the cross-table reference. It works together with the concept of primary key, which is a selected key from all candidate keys in a table. The idea is that the foreign key for a (referencing) table A is a set of attributes that match with the primary key in the another (referenced) table B. Then, the values of the foreign key attributes for any tuple in the referencing table A must match the values of one tuple in referenced table B on these same attributes. For example, if we have a table A recording ECE students' scores for ECE 4305 and a table B recording all ECE students' basic profile, and we put a foreign key constraint on the ID numnber attribute of A referencing the ID numnber attribute of B, then table A can only store tuple with a ID number that exist in table B. If one tries to insert a new tuple to A which has a ID number that does not exist in B, the insertion will be rejectd because of foreign key violation. Or, if one tried to delete a tuple in B with its ID number referenced by tuple(s) in A, the deletion will be rejected as well due to foreign key violation. By imposing the foreign key constraints, one can better coordinate the data in different tables, and avoid cross-table inconsistency.

Index

Besides storing data in tables, organizing data using index is another very important aspect of database design. This is because the purposes of database systems are not only storing data but also letting users to retrieve their data in an efficient way. Therefore, you can create index on selected column(s) to facilitate data searching and retrieval.

An index is a structured file that speeds up data access for SELECT, but may slow down INSERT, UPDATE, and DELETE. Without an index structure, to process a SELECT query with a matching criterion, the database engine needs to compare every record in the table. On the contrary, a specialized index (*e.g.*, in B+ tree structure) could reach the

record without comparing every record. For example, one can leave the records in a table recording ECE students' scores in ECE 4305 unorganized, or use a index structure to index the scores in a ranked order. Then if we need to find the student with the highest score, for the first case, we have to scan the whole table to figure it out, while in the second case, we can simply grab the first tuple in the score index. However, the index needs to be rebuilt whenever a record is changed, which results in overhead associated with using indexes. There are many different index structures available in relational database. Among them the tree-based structures, such as B+ tree and R-tree, are the most widely used ones [92].

Index can be defined on a single column, a set of columns (called concatenated index), or part of a column (called partial index). You can even build more than one index on a table.

2.7 SDR Technology

Given the brief overview of the background of cognitive radio and spectrum sensing, as well as a tutorial of various techniques that will be employed in the PUE detection approaches, we will now focus our attention on implementing an SDR simulation system, which enables us to test our approaches in real-world environment. Note that when designing a complete SDR system from scratch, it is very important to have both a hardware platform that is both sufficiently programmable and computationally powerful, as well as a software architecture that can allow a communication system designer to implement a wide range of different transceiver realizations. In this section, we will first study some of the well-known SDR hardware platforms before talking about some of the available SDR software architectures.

2.7.1 Hardware Platforms

Exploration into advanced wireless communication and networking techniques require highly flexible hardware platforms. As a result, SDR is very well suited due to its rapidly reconfigurable attributes, which allows for controlled yet realistic experimentation. Thus, the use of real-time test bed operations enables a large set of experiments for various receiver settings, transmission scenarios, and network configurations. Furthermore, SDR hardware provides an excellent alternative for comprehensive evaluation of communication systems operating within a networked environment, whereas Monte Carlo simulations can

be computationally exhaustive and are only as accurate as the devised computer model. In this section, we will study several well-known SDR hardware platforms used by the wireless community for research and experimentation.

One of the most well-known of all SDR hardware platforms is the *Universal Software Radio Peripheral* (USRP) concept that was introduced by Matt Ettus, founder and president of Ettus Research LLC, which is considered to be a relatively inexpensive hardware for enabling SDR design and development [93]. All the baseband digital communication algorithms and digital signal processing are conducted on a computer workstation "host", where the USRP platform acts as a radio peripheral allowing for over-the-air transmissions and the `libusrp` library file defines the interface between the USRP platform and the host computer workstation. Note that the USRP design is open source, which allows for user customization and fabrication. Furthermore, USRP platform design is modular in terms of the supported RF front-ends, referred to as *daughtercards*. We will now talk about two types of USRP platforms: the *USRP1* and *USRP2*.

The *Universal Software Radio Peripheral – Version 1* (USRP1) was designed and manufactured by Ettus Research LLC for a variety of different communities interested in an inexpensive SDR platform. The USRP1 consists of a USB interface between host computer workstation and USRP1 platform, which resulted in a data bottleneck due to the low data rates supported by the USB connection. The USRP1 supports up to two RF transceiver daughtercards, possesses an Altera Cyclone EP1C12Q240C8 FPGA for performing sampling and filtering, contains four high-speed analog-to-digital converters, each capable of 64 MS/s at a resolution of 12 bits, with an 85 dB SFDR (AD9862), and contains four high-speed digital-to-analog converters, each capable of 128 MS/s at a resolution of 14 bita, with 83 dB SFDR (AD9862).

Following the success of the USRP1, Ettus Research LLC officially released the *Universal Software Radio Peripheral – Version 2* (USRP2) platform in September 2008, as shown in Fig. 2.17(a). The USRP2 platform provides a more capable SDR device for enabling digital communication system design and implementation. The USRP2 features include a Gigabit ethernet interface between host computer workstation and USRP2 platform, supports only one RF transceiver daughtercard, possesses a Xilinx Spartan 3-2000 FPGA for performing sampling and filtering, contains two 100 MS/s, 14 bits, analog-to-digital converters (LTC2284), with a 72.4 dB SNR and 85 dB SFDR for signals at the Nyquist frequency, contains two 400 MS/s, 16 bits, digital-to-analog converters (AD9777), with a

160 MS/s without interpolation, and up to 400 MS/s with 8x interpolation, and is MIMO-capable for supporting the processing of digital communication system designs employing multiple antennas

(a) Front view of a Universal Software Radio Peripheral – Version 2 (USRP2) software-defined radio platform by Ettus Research LLC.

(b) Front view of a Kansas University Agile Radio (KUAR) software-defined radio platform.

Fig. 2.17 Examples of software-defined radio platforms.

The radio frequency (RF) front-ends are usually very difficult to design and are often limited to a narrow range of transmission carrier frequencies. This is due to the fact that the properties of the RF circuit and its components change across different frequencies, and that the RF filters are constrained in the sweep frequency range. Consequently, in order to support a wide range of transmission carrier frequencies, both the USRP1 and USRP2 platforms can use an assortment of modular RF daughtercards, such as:

- *BasicTX*: A transmitter that supports carrier frequencies within 1-250 MHz.
- *BasicRX*: A receiver that supports carrier frequencies within 1-250 MHz.
- *RFX900*: A transceiver that supports carrier frequencies within 800-1000 MHz with a 200+mW output.
- *RFX2400*: A transceiver that supports carrier frequencies within 2.3-2.9 GHz with a 20+mW output.

- *XCVR2450*: A transceiver that supports carrier frequencies within two bands, namely, 2.4-2.5 GHz with an output of 100+mW and 4.9-5.85 GHz with an output of 50+mW. This is the daughtercard that will be used in the SDR experiments in Chapter 4 and 5, as shown in Fig. 2.18.

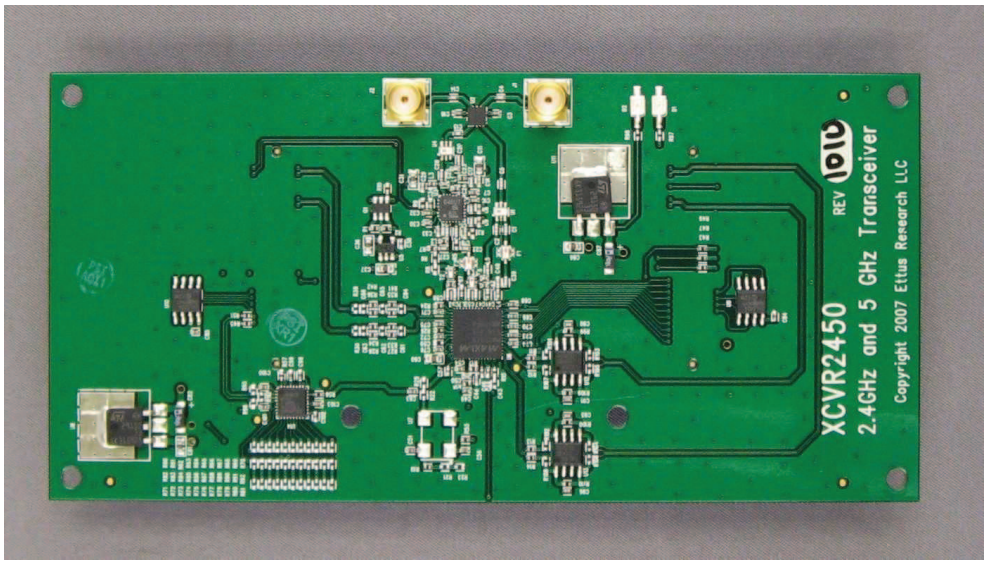


Fig. 2.18 A photograph of XCVR2450 RF transceiver daughterboard (from [8]), where a common local oscillator is used for both receive and transmit.

Another well-known SDR hardware platform was the *Kansas University Agile Radio* (KUAR), which is a small form factor SDR platform containing a Xilinx Virtex-II Pro FPGA board and a PCI Express 1.4 GHz Pentium-M microprocessor, as shown in Fig. 2.17(b) [94, 95]. For its size and capability, the KUAR was one of the leading SDR implementations in its day, incorporating substantial computational resources as well as wideband frequency operations. Around the same time period, the Berkeley BEE2 was designed as a powerful reconfigurable computing engine with five Xilinx Virtex-II Pro FPGAs on a custom-built emulation board [96]. The Berkeley Wireless Research Center (BWRC) cognitive radio test-bed hardware architecture consists of the BEE2, several reconfigurable 2.4 GHz radio modems, and fiber link interfaces for connections between the BEE2 and the radios modems. The software architecture consists of Simulink-based design flow and BEE2 specific operating system, which provides an integrated environment for implemen-

tation and simple data acquisition during experiments.

With respect to compact SDR platforms, Motorola developed and built a 10 MHz-4 GHz CMOS-based small form factor cognitive radio platform prototype [97]. Fundamentally flexible, with a low-power transceiver radio frequency integrated circuit (RFIC) at the core of this experimental platform, this prototype can receive and transmit signals of many wireless protocols, both standard and experimental. Carrier frequencies from 10 MHz to 4 GHz with channel bandwidths from 8 kHz to 20 MHz were supported. Similarly, the *Maynooth Adaptable Radio System* (MARS) is a custom-built small form factor SDR platform [98]. The MARS platform had the original objectives of being a personal computer connected radio front-end where all the signal processing is implemented on the computers general purpose processor. The MARS platform was designed to deliver performance equivalent to that of a future base station and the wireless communication standards in 1700 MHz to 2450 MHz frequency range. Furthermore, the communication standards GSM1800, PCS1900, IEEE 802.11b/g, UMTS (TDD and FDD) are also supported.

Rice University *Wireless Open Access Research Platform* (WARP) radios include a Xilinx Virtex-II Pro FPGA board as well as a MAX2829 transceiver [99], while the Lyrtech Small Form Factor SDR is developed by a company from the Canadian Province of Québec that leverages industrial collaborations between Texas Instruments and Xilinx in order to produce these high performance SDR platforms that consist of an array of different microprocessor technology [100]. Finally, Epiq Solutions recently released the MatchStiq SDR platform, which is a powerful yet very compact form factor SDR platform capable of being deployed in the field to perform a variety of wireless experiments, including their inclusion onboard vehicles such as automobiles and unmanned aerial vehicles [101].

2.7.2 Software Architecture

Given the programmable attributes of an SDR platform, it is vitally important to also develop an efficient and reliable software architecture that would operate on these platforms in order to perform various data transmission functions we expect from a wireless communications system. In this section, we will review some of the SDR software architectures currently available for use with a wide variety of SDR hardware platforms.

One of the first Simulink interfaces to the USRP2 platform was implemented as part of an MS thesis at WPI and generously sponsored by the MathWorks [9]. In this research

project, the focus was on creating a Simulink blockset capable of communicating with the USRP2 libraries, which can then allow for communications with the USRP2 platform itself. The resulting blocksets from this thesis research is shown in Fig. 2.19. By creating a Simulink interface to this SDR hardware, it is expected that the existing signal processing libraries provided by the MathWorks can be extensively leveraged in order to create actual digital communications systems capable of performing over-the-air data transmission with other USRP2 platforms. The architecture of the Simulink transmit and receiver blocks are

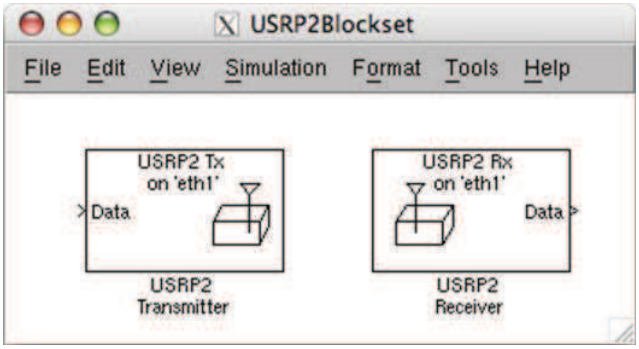
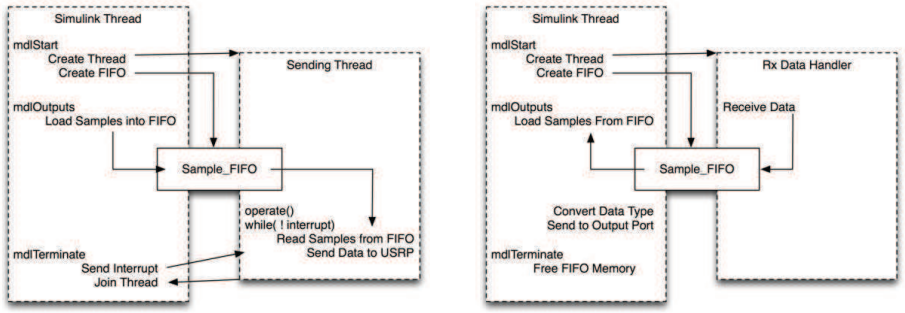


Fig. 2.19 The initial prototype Simulink transmitter and receiver interfaces for the USRP2 platform [9].

shown in Fig. 2.20. In Fig. 2.20(b), we observe how the Simulink transmitter block calls the functions implemented by the S-Function at different parts of the simulation. While the simulation is initializing, it calls on `mdlStart` such that a data handler object and first-in first-out (FIFO) register are created and a USRP2 object is instantiated. Once the USRP2 object has been created, the operating parameters set in the mask are passed down to the USRP2 hardware while the model is in the process of initializing. Furthermore, the data handler object loads data into the FIFO and while the simulation is running, Simulink is repeatedly calling `mdlOutputs` such that a frame of data is read from the FIFO, converted to a Simulink data type, and sent to the output port to be received in the simulation. Note that when the simulation has finished, the FIFO and data handler are deallocated. The transmitter shown in Fig. 2.20(a) possesses a similar mode of operation.

From the MS thesis and the development of the first Simulink prototype blockset interface with the USRP2 SDR platform, the MathWorks built upon the lessons learned from this experience and ultimately created the SDRu blockset, which can be downloaded from the MathWorks website and installed with MATLAB R2011a or later along with the Com-



(a) Initial prototype Simulink transmitter interface.

(b) Initial prototype Simulink receiver interface.

Fig. 2.20 Architecture of the initial prototype interfaces to the USRP2 platform [9].

munications Toolbox installed [102]. After several years of development, the SDRu blocks are at the core of numerous SDR implementations using Simulink and the USRP2, as well as educational activities such as those to be discussed later in this book.

Other SDR software architectures include the popular open-source GNU Radio software [103], which is a community based effort for devising an SDR software architecture capable of interfacing with any SDR hardware platform, especially the USRP family of products, and enable them to reliably and seamlessly communicate with other SDR platforms as well as conventional wireless systems. Given the large open-source community supporting the GNU Radio software architecture, several community members have posted their customized solutions that were not incorporated into GNU Radio directly on the Comprehensive GNU Radio Archive Network (CGRAN) website for download by the rest of the community [104]. Finally, another SDR software interface is the *Implementing Radio in Software* (IRIS) project [105], which is led by the researchers at the Centre for Telecommunications Value-Chain Research (CTVR) at Trinity College Dublin and used by many researchers across Europe and around the world.

2.8 Chapter Summary

This chapter summarizes the background knowledge related to this dissertation. The topics covered are cognitive radios, dynamic spectrum access, spectrum sensing, artificial neural network, action representation, relational database and SDR simulation tools. While

they are separated topics by themselves, primary user emulation detection possesses the potential to combine them, and it is the cross-discipline efforts that revolutionize the security of cognitive radio networks.

Chapter 3

Proposed Research Plan

For my PhD research, I will focus on the topic of PUE detection techniques that do not rely on the centralized control and the white-space database. The motivation of this topic has been introduced in Section 1.3. The goal of this topic is that by employing my proposed techniques, a contention based dynamic spectrum access network can be set up, as shown in Fig. 1.4.

Based on the discussion of current state of the art in Section 1.4, I decide to focus on the feature detection. As a starting point, I look at something simple, which is cyclostationary feature detection. As introduced in Section 2.2.4, different modulation schemes have different spectral features, which can be represented by the spectral coherence function (SOF). Assume the modulation scheme of the authentic primary user is known. If we calculate the SOF of the unknown user, and compare it with the SOF of the authentic primary user, we can find whether they are the same or not. If they are different, we can safely come to the conclusion that the unknown user is a primary user emulator.

However, it is possible that the authentic primary user and the primary user emulator use the same modulation type, where the first approach is not applicable. Therefore, we need to consider other features of users in the network. So for the next step, I will study the approach based on action recognition in video, as introduced in Section 2.5. But for the context of PUE detection, I will apply this approach to the frequency-domain FFT. Although the malicious secondary users can mimic the spectral characteristics of the primary users during the competition of gaining priority access to the wireless channels, they will not follow the spectral characteristics forever. As long as they successfully gain

the access, they will switch back to their normal behavior. This change can be considered as an "action" in frequency domain. If the action of a suspicious user is different from that of an authentic primary user, we can tell this suspicious user is a malicious secondary user. This method analyzes the FFT sequences of wireless transmissions operating across a cognitive radio network environment, and classifies their actions in the frequency domain.

For the previous two approaches, I assume that there is only one primary user in the system, so all the detection operations can be done in relatively short time. However, when there are several primary users in the system, it requires more computations when constructing the covariance descriptor of feature vectors, as well as testing with the artifical neural network, so it may not satisfy those scenarios that real-time processing is a priority. To resolve this new issue, I propose a PUE detection approach that builds upon the previous approach, and at the same time, introduces a relational database system in order to overcome the problem of intensive computation. This new approach records the feature vectors of primary users in the database system, then it monitors each users FFT sequence and compares the unknown users feature vectors with those in the database. In most applications, primary users possess routine wireless transmissions, so they have a limited number of feature vectors, which means the resulting database is stable and limited in size. In case that an unknown users feature vector does not have a match entity in the database, this approach will continue to recognize its action in the frequency domain using artifical neural network. This approach operates on intercepted signals and analyzes it in the frequency domain over a time interval. Besides the benefits of the previous two approaches, this new approach takes the stability of primary users into account and creates a database system, so it is can save some computations compared to the previous approach.

All the approaches proposed so far assume there is a single-node PUE detector in the system. However, in order to improve the efficiency and accuracy of the detection, we can actually employ a distributed sensor network as our detector. Each node of this sensor network employs one of the proposed approaches, and the final conclusion is based on the detection results from all the sensor nodes.

For my PhD study, in addition to the research discussed above, I have also been devoting a lot of time to SDR experiment and SDR education. Therefore, given all the proposed apporaches, initial design, implementation, and evaluation will first be conducted via computer simulation in MATLAB and Simulink. However, it is expected that the proposed apporaches will also be prototyped and evaluated in hardware. Specifically, I will be em-

ploying the Universal Software-Defined Radio Peripheral 2 (USRP2) by Ettus Research LLC as the development platform of choice for the SDR experiments, as introduced in Section 2.7.1. In this way, I can figure out how the proposed apporaches work in real-life wireless environment and multi-fading channel.

3.1 Chapter Summary

This chapter summarizes the proposed research plan for my PhD study. The research topic I will focus on is proposing PUE detection techniques that do not rely on the centralized control and the white-space database. More specifically, I will propose four different approaches based on feature detection. For each approach, I will evaluate it via both computer simulations and SDR hardware experiments.

Chapter 4

Proposed PUE Detector Based on Cyclostationarity[1]

The following two chapters propose three approaches for PUE detection, which can be divided into two categories. This chapter discusses the first category based on cyclostationary features. It employs a cyclostationary calculation to represent the modulation features of the user signals, which are then fed into an artificial neural network for classification.

4.1 Physical Layer Primary User Emulator

Before proposing a solution to the PUE detection, let us first study how a primary user emulator is formed from the physical layer perspective. In digital communication theory, it is sometimes more convenient to study a data transmission system in the frequency domain rather than in the time domain for several reasons, including the ability to obtain tractable mathematical analyses. To enable the translation between the time domain signal $x(t)$ and the frequency domain signal $X(f)$, the Fourier transform can be employed, which is defined as:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft}dt, \tag{4.1}$$

[1]This work has been published in parts in IEEE Global Telecommunications Conference [31].

and an inverse Fourier transform, defined as:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(f) e^{j2\pi ft} df. \tag{4.2}$$

With respect to random processes, which often describes most data transmissions and other signal sources, the Fourier transform and its inverse can be used in order to determine the power spectrum density. This is achieved by taking the Fourier transform of the correlation function of the random process $X(t)$, namely $R_{XX}(\tau)$, such that it yields the power spectral density $S_{XX}(f)$:

$$S_{XX}(f) = \int_{-\infty}^{+\infty} R_{XX}(\tau) e^{-j2\pi f\tau} d\tau. \tag{4.3}$$

Note that the power spectral density describes the frequency content of a signal and helps to identify periodicities. The power spectral density can be used to find the expectation of the positive frequency component of a random signal and then use it in order to estimate the signal's spectrum density.

From the aspect of power spectrum density, a physical layer approach to primary user emulation can be realized via the manipulation of the power spectral density characteristics of the signal, which can be achieved using the Einstein-Wiener-Khintchine (EWK) Theorem [106]:

$$S_{target}(f) = |H(f)|^2 S_{emulator}(f), \tag{4.4}$$

where $S_{emulator}(f)$ is the power spectrum density (PSD) of the malicious secondary user, and $S_{target}(f)$ is the PSD of an authentic primary user.

In theory, if an appropriate pulse shaping filter $h(t)$ is designed and applied to the malicious secondary user, its PSD characteristics can be made to appear similar to that of an authentic primary user. Consequently, (4.4) can be rewritten as:

$$H(f) = \sqrt{\frac{S_{target}(f)}{S_{emulator}(f)}}. \tag{4.5}$$

Using the resulting transfer function from (4.5), we can obtain the PUE pulse shaping

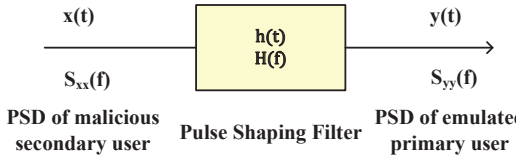


Fig. 4.1 Block diagram for the usage of pulse shaping filter in primary user emulation.

impulse response via the inverse Fourier transform using the expression:

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(f) e^{j2\pi ft} df. \tag{4.6}$$

Fig. 4.1 shows how a transfer function $H(f)$ based on the impulse response $h(t)$ can be designed such that by the EWK Theorem the input PSD $S_{XX}(f)$ of the malicious secondary user can be "shaped" into a new PSD, namely $S_{YY}(f)$, such that the latter ressembles the PSD of a known existing primary user transmission. Consequently, all the malicious user needs to know in order to preform a PUE attack is the power spectral density characteristics of the native primary user to a specific frequency range, as well as the necessary filtering operations to perform the power spectral density transformation using the EWK Theorem.

4.2 System Model

In this chapter, we consider a cognitive radio network as shown in Fig. 4.2. All the users, including the primary users, primary user emulators and secondary users are distributed in a circular grid with a PUE detector in the center. In order to avoid interference, we assume at each time, there is only one user transmitting in this network.

Let $x(t)$ denote the transmitted signal. If it is the authentic PU signal, $x(t)$=$s(t)$. If it is the PUE signal, $x(t)$=$s'(t)$. Considering the energy of the signal, since the PUE signal is very similar to the PU signal, we assume both $s(t)$ and $s'(t)$ are independently and identically distributed (iid) random processes with mean zero and variance σ_s^2, namely:

$$s(t), s'(t) \sim \mathcal{N}(0, \sigma_s^2), \tag{4.7}$$

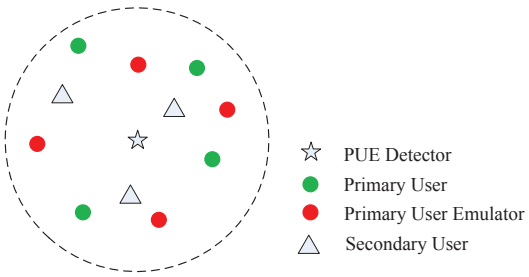


Fig. 4.2 A cognitive radio network in a circular grid.

where $\mathcal{N}(\cdot)$ denotes the normal distribution. Since the secondary users have a significant lower transmitted power than the primary users, we assume $x(t) = 0$ when the SU is transmitting.

Let $h(t)$ and $n(t)$ denote the impulse response and the noise of the channel between the transmitted signal and the PUE detector. We assume the channel is a slow flat fading channel during the observation process, so $h(t)$ becomes a constant gain h. $n(t)$ is the additive white Gaussian noise (AWGN) with mean zero and variance σ_n^2, namely:

$$n(t) \sim \mathcal{N}(0, \sigma_n^2). \tag{4.8}$$

Therefore, there are three possible received signals at the PUE detector:

$$y(t) = \begin{cases} n(t) & \text{SU}, \\ h \times s(t) + n(t) & \text{PU}, \\ h \times s'(t) + n(t) & \text{PUE}, \end{cases} \tag{4.9}$$

where $y(t)$ is the received signal at the PUE detector. The PUE detection algorithm presented in Section 4.3 will differentiate these three cases.

4.3 Proposed PUE Detection Algorithm

According to the system model, our proposed approach makes the following assumptions: (i) All the users, including the malicious users and primary users, are located within the same frequency band; (ii) For each period of time, there is only one user transmitting; (iii) The modulation scheme of primary users is known, and it is different from the other

users.

Fig. 4.3 provides a flow diagram of the proposed PUE detection algorithm, which is a two-step process.

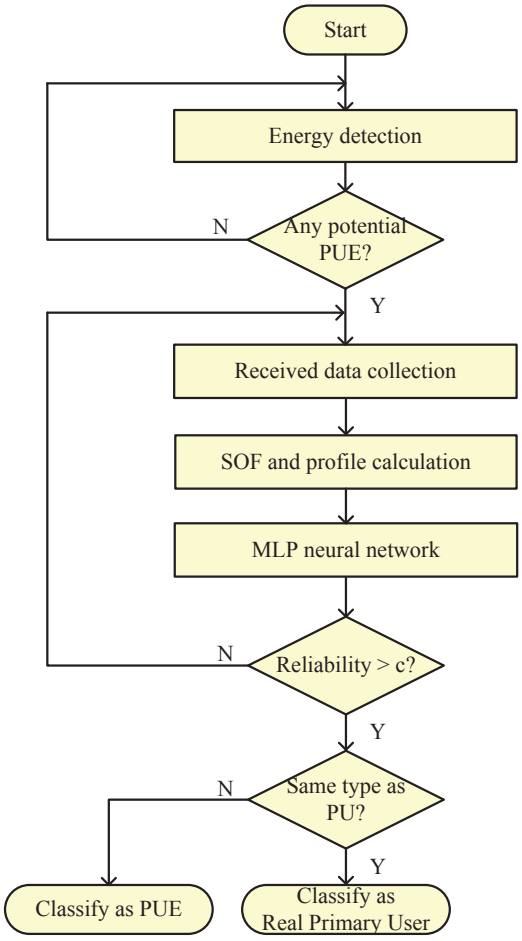


Fig. 4.3 Proposed PUE detection algorithm employing energy detection, cyclostationary calculation and artificial neural networks.

First, the algorithm is initialized using energy detection in order to determine the frequency location of the potential primary user, as introduced in Section 2.2.4. Based on the assumption that the transmission power of the PU and PUE is much higher than that of the SU, this step can record the frequency location of these two types of users. In other

words, this step is trying to differentiate the following two cases of the received signal:

$$y(t) = \begin{cases} n(t) & \text{SU}, \\ h \times x(t) + n(t) & \text{PU \& PUE}, \end{cases} \tag{4.10}$$

where $x(t) = s(t)$ or $s'(t)$. If $y(t)$ belongs to the second case, it will be recorded along the time on the receiver side. After a certain period of time T, this observation process is terminated and the saved signals are passed on to the cyclostationary classifier.

In the second step, the classifier calculates the SOF data and cycle frequency profile of the received signal, as introduced in Section 2.2.4, and uses an artificial neural network to classify the signal based on the profile, as introduced in Section 2.4.2. If the reliability χ of the testing result is less than a constant number c specified at the beginning, we need to collect some new received data and run the procedures above again. Otherwise, the artificial neural network will output the modulation scheme. Since the modulation scheme of primary users is known, we can readily identify whether the observed signal is from a real primary user or a malicious user.

4.3.1 Mathematical Analysis

In the first step, an energy detector is employed to record the frequency location of the potential primary users, which is essentially a hypothesis testing. The two hypotheses are denoted as follows:

$$\begin{aligned} \mathcal{H}_0 :&\quad \text{no potential primary signals}, \\ \mathcal{H}_1 :&\quad \text{potential primary signals exist}, \end{aligned} \tag{4.11}$$

where $\mathcal{H}_0$ is the null hypothesis. In our algorithm, these two hypotheses can be further represented as:

$$\begin{aligned} \mathcal{H}_0 :&\quad y[k] = n[k], \\ \mathcal{H}_1 :&\quad y[k] = h \times x[k] + n[k], \end{aligned} \tag{4.12}$$

for $k = 1, .., N$, where N is the number of received samples. Since both $n[k]$ and $x[k]$ are iid normal random variables, $y[k]$ has the following distribution:

$$y[k] \sim \begin{cases} \mathcal{N}(0, \sigma_0^2) & \mathcal{H}_0, \\ \mathcal{N}(0, \sigma_1^2) & \mathcal{H}_1, \end{cases} \tag{4.13}$$

where $\sigma_0^2 = \sigma_n^2$ and $\sigma_1^2 = h^2\sigma_s^2 + \sigma_n^2$.

Consequently, a decision statistic for energy detector can be defined as:

$$Y = \sum_{k=1}^{N} |y[k]|^2, \tag{4.14}$$

where $y[k]$ and N follow the definitions in (4.12). Under both hypotheses, the decision statistic Y is the sum of the squares of N mutually independent normal random variables. According to [107, 108], Y has the central chi-square distribution with $2N$ degrees of freedom, namely:

$$Y \sim \begin{cases} \chi_{2N}^2(\sigma_0^2) & \mathcal{H}_0, \\ \chi_{2N}^2(\sigma_1^2) & \mathcal{H}_1, \end{cases} \tag{4.15}$$

where χ^2 denotes the chi-square distribution.

Given the decision statistic Y and a threshold T, the performance of this energy detector can be characterized by two parameters, namely, the probability of false alarm (P_F), and the probability of detection (P_D), which can be defined as:

$$P_F = P\{\text{Decide } \mathcal{H}_1|\mathcal{H}_0\} = P(Y > T|\mathcal{H}_0) = P(Y > T|Y \sim \chi_{2N}^2(\sigma_0^2)), \tag{4.16}$$

and

$$P_D = P\{\text{Decide } \mathcal{H}_1|\mathcal{H}_1\} = P(Y > T|\mathcal{H}_1) = P(Y > T|Y \sim \chi_{2N}^2(\sigma_1^2)). \tag{4.17}$$

We have already known that the cumulative distribution function (CDF) of a standard central chi-square distribution is

$$F(x; k) = \frac{\gamma(k/2, x/2)}{\Gamma(k/2)}, \tag{4.18}$$

where k is the degree of freedom, $\gamma(\cdot,\cdot)$ is the lower incomplete gamma functions, and $\Gamma(\cdot)$ is the ordinary gamma function [108]. Since (4.16) and (4.17) are complement of the CDF defined in (4.18), they can be obtained by:

$$P_F = 1 - F(\frac{T}{\sigma_0^2}; 2N) = 1 - \frac{\gamma(N, \frac{T}{2\sigma_0^2})}{\Gamma(N)} = \frac{\Gamma(N, \frac{T}{2\sigma_0^2})}{\Gamma(N)}, \tag{4.19}$$

and

$$P_D = 1 - F(\frac{T}{\sigma_1^2}; 2N) = 1 - \frac{\gamma(N, \frac{T}{2\sigma_1^2})}{\Gamma(N)} = \frac{\Gamma(N, \frac{T}{2\sigma_1^2})}{\Gamma(N)}. \tag{4.20}$$

where $\Gamma(\cdot,\cdot)$ is the upper incomplete gamma function and $\Gamma(\cdot)$ is the ordinary gamma function.

Therefore, given a required probability of false alarm P_F, we can calculate the appropriate threshold T using (4.19):

$$T = 2\Gamma^{-1}(N, P_F\Gamma(N))\sigma_0^2, \tag{4.21}$$

where $\Gamma^{-1}(\cdot,\cdot)$ is the inverse incomplete Gamma function [109]. And then, plug this value into (4.20) to get the corresponding probability of detection:

$$P_D = \frac{\Gamma\left(N, \Gamma^{-1}(N, P_F\Gamma(N))\frac{\sigma_0^2}{\sigma_1^2}\right)}{\Gamma(N)}. \tag{4.22}$$

It is obvious that given a probability of false alarm P_F, the probability of detection P_D is determined by two factors, namely the number of samples N, and the ratio of σ_0^2 to σ_1^2.

According to the definition of signal-to-noise ratio, the SNR at the receiver is [110]:

$$\text{SNR} = \frac{h^2\sigma_s^2}{\sigma_n^2} = \frac{\sigma_1^2 - \sigma_0^2}{\sigma_0^2} = \frac{\sigma_1^2}{\sigma_0^2} - 1. \tag{4.23}$$

Therefore, the second factor can be represented in SNR.

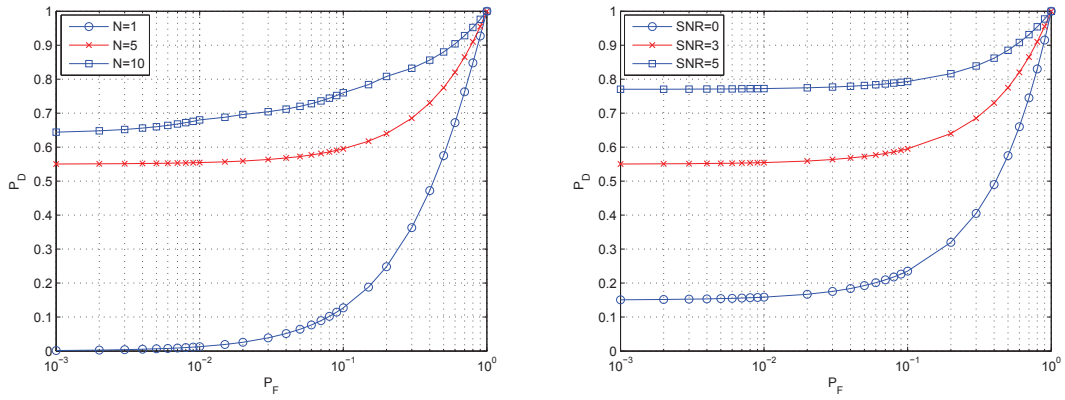
4.3.2 Numerical Results

This section provides the numerical results of the energy detector introduced in Section 4.3.1. The results are based on (4.22), presented in terms of the ROC curves (*i.e.*, P_D

versus P_F) for an AWGN fading channel.

Since the probability of detection P_D is determined by two factors, the ROC curves in Fig. 4.4 are plotted by varying these two factors. Specifically, the ROC curves in Fig. 4.4(a) are plotted by varying the number of samples N from 1 to 10 given SNR =3, and the ROC curves in Fig. 4.4(b) are plotted by varying the SNR value from 0 to 5 given N=5. Based on Fig. 4.4, we can come to the following conclusions:

- Probability of false alarm (P_F) and probability of detection (P_D) are tradeoff. We cannot optimize both of them at the same time.
- Given an AWGN channel or a fixed SNR value, the more we can tolerate the P_F, the higher P_D we can achieve.
- Given a P_F, we can improve the P_D by increasing the SNR value or increasing the number of collected samples.



(a) ROC curves by varying the number of samples N from 1 to 10 given SNR =3.

(b) ROC curves by varying the SNR value from 0 to 5 given N=5.

Fig. 4.4 Numerical results of the energy detector in terms of the ROC curves.

4.4 Experimental Setup & Results

Since the algorithm proposed in Section 4.3 is a two-step procedure, the performance of the PUE detector depends not only on the energy detector, but also on the cyclostationary classifier. Assume the performance of the energy detector is P_D given P_F, and the

probability of correct classification is P_C of the classifier. Since the energy detector and the classifier are two independent events, the final performance of the PUE detector will be:

$$P_{\text{detection}} = P_D \times P_C, \tag{4.24}$$

given P_F.

In this section, two different experiments are conducted in order to validate the performance of the proposed cyclostationary classifier. The first experiment uses a computer simulation based on Simulink, while the second experiment is based on a hardware implementation using the Universal Software Radio Peripheral (USRP) software-defined radio (SDR) platform. This section assumes that the energy detector has successfully identify a potential PUE given a probability of false alarm, so all the experiments will start from received data collection in Fig. 4.3. For simplicity, we assume that each user is constrained to one modulation scheme throughout the observation time window and the primary user uses a modulation scheme that possesses a significantly different characteristic relative to the other user signals present in the area.

4.4.1 Computer Simulation

Simulation Setup

In this part, a Simulink model is constructed in order to collect the received signals based on different modulation schemes. If there are N different modulation schemes, then this model needs to contain N branches, which share an identical structure. Fig. 4.5 shows the structure of one of the branches.

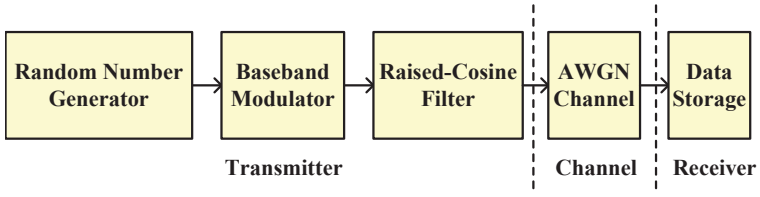


Fig. 4.5 The structure of one branch in the Simulink model. All the other branches have the identical structure.

In this model, there are three blocks on the transmitter side: random data is generated and modulated by one of several possible modulation schemes; a raised cosine filter is used

for pulse-shaping in order to minimize intersymbol interference (ISI). Then, an AWGN channel block is applied to emulate the transmission environment. For this block, the signal-to-noise ratio (SNR) can be specified to represent channels of different noise level. In the end, a sink block is used on the receiver side to save the received signals. These signals are stored away in a workspace for post processing, including the SOF and cycle frequency profile calculations, as introduced in Section 2.2.4.

In this dissertation, an multi-layer perceptron (MLP) neural network with 514 input nodes, one layer of 6 hidden nodes, and one output node is employed. $f(x) = \tanh(x)$ is selected as the activation function. For training, the back propagation algorithm is used with a fixed training constant of $\eta = 0.5$, and momentum constant $\zeta = 0.7$. The cycle frequency profile of the second order statistics is then fed into the system of artificial neural networks, and the system outputs a classification result along with a reliability parameter. If the reliability parameter is larger than 0.7, the classification result is accepted.

Simulation Results

As discussed in Section 2.2.4, different modulation schemes yield different SOF diagrams, which can be used as a criterion for signal classification. In our proposed algorithm, cycle frequency profile of the SOF is used to improve the efficiency by reducing the amount of data to be processed. As shown in Fig. 4.6, different modulation schemes also feature distinctive cycle frequency profiles, defined in (2.31).

Fig. 4.7 shows the percentage of correct classification with different SNR values ranging from -8 dB to 8 dB. For a specific SNR value, the "Initial Seed" parameter of the "Random Number Generator" block is changed to get different input data so that we can repeat the experiment many times and average the resulting percentages. Overall, higher SNR values yield better algorithm performance in terms of successfully classifying primary signals and PUE signals. This figure also compares the percentage of correct classification with and without a reliability check. Even when the channel possesses a substantial amount of noise, such as when SNR = -8 dB, the percentage of correct classification can still reach 72% with a reliability check. Also, for an SNR above -4 dB, we are able to get at least 95% of the signals to be classified correctly.

With respect to the time required to successfully classify a signal, although it takes some time for the artificial neural network to be trained with the back propagation algorithm,

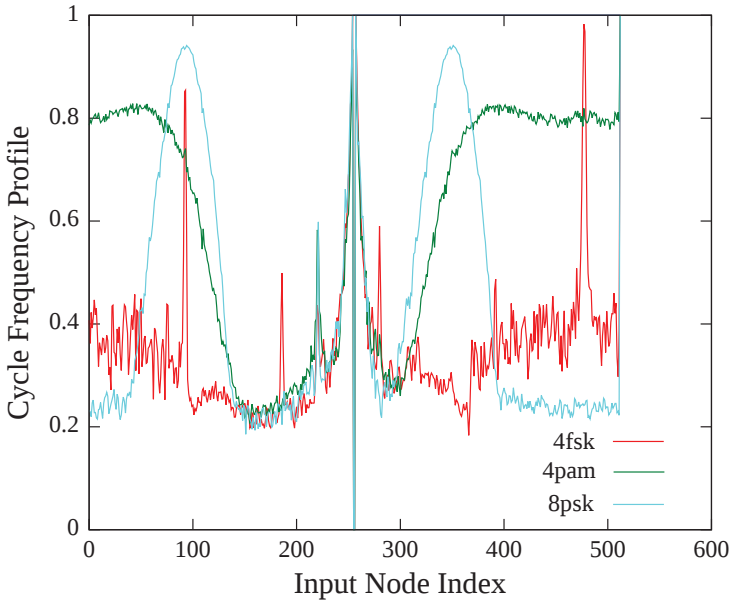


Fig. 4.6 The cycle frequency profile of the received signals. The x-axis represents the index of the input node, and the y-axis represents the node's cycle frequency profile value. It's obvious that different modulation schemes feature distinctive cycle frequency profiles.

this process can be conducted offline with some previously known training signals. Once the artificial neural network has been trained, it only needs to be evaluated rather than both trained and evaluated with any newly intercepted signals. Our simulations show that it takes 0.593 seconds to classify 60 signals during the testing stage on a 1.6 GHz processor, which make this proposed algorithm a viable candidate for operation in real-time situations.

4.4.2 Software-Defined Radio Experiment

The Simulink model in Section 4.4.1 was then used as a starting point for the design of a hardware implementation. This was achieved by initially changing the AWGN channel block with a real-life fading channel, and by using the Simulink USRP2 blocks available in the Communications System Toolbox. Consequently, our resulting Simulink design that operates on the USRP2 SDR platform is shown in Fig. 4.8.

The universal software radio peripheral - version 2 (USRP2) is a high-speed gigabit Ethernet-based SDR platform designed to allow general purpose computers to function as high bandwidth software radios. Consequently, it serves as a digital baseband and IF section

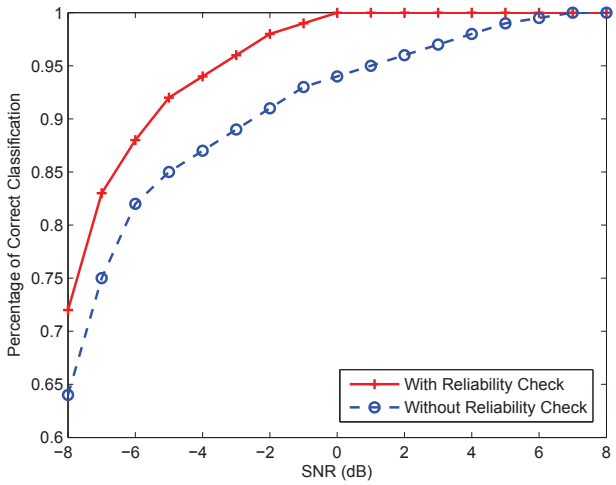


Fig. 4.7 The classification performance with and without a reliability check in computer simulations. The x-axis represents SNR value, and the y-axis represents the percentage of correct classification. For the reliability check, all classifications with a reliability parameter less than 0.7 are ignored.

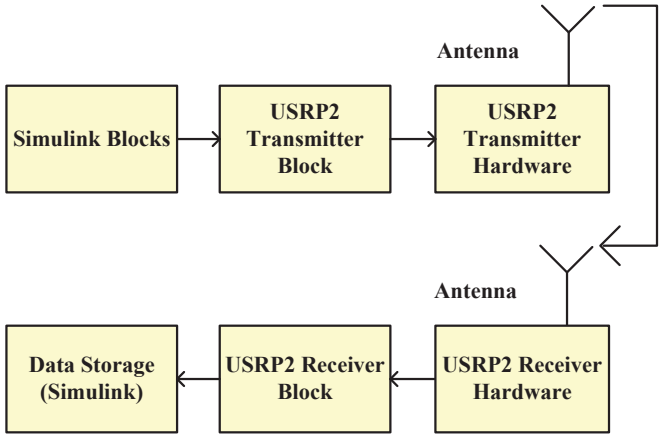


Fig. 4.8 The Simulink design that operates on the USRP2 SDR platform.

of a radio communication system [111]. In this dissertation, two USRP2 hardware equipped with the XCVR2450 daughterboard are employed [112]. One serves as a transmitter and the other as a receiver, as shown in Fig. 4.9 and Fig. 4.10. Both of them work across 2.4 GHz frequency band.

In order to incorporate the USRP2 hardware into the existing Simulink model, two

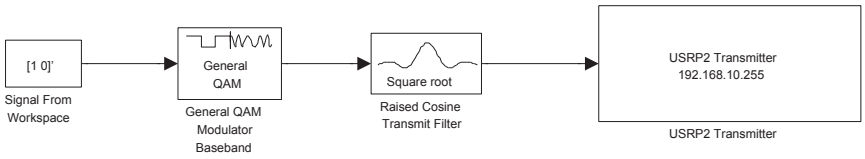


Fig. 4.9 The Simulink model for transmitter, which includes a USRP2 transmitter block and a USRP2 transmitter hardware.

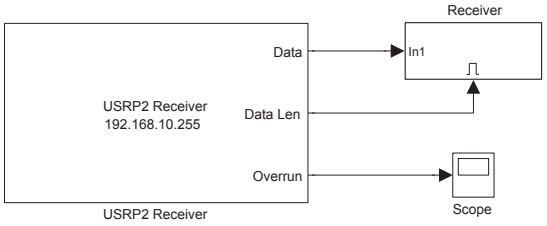


Fig. 4.10 The Simulink model for receiver, which includes a USRP2 receiver block and a USRP2 receiver hardware.

Simulink blocks called `USRP2 Transmitter` and `USRP2 Receiver` are used here as interfaces[2]. These two blocks were developed by The MathWorks and have been available since the R2010a release of MATLAB. With these two blocks, the USRP2 SDR platforms can be used in conjunction with the previous Simulink design environment.

The rest of the Simulink model remains the same, including the random number generator, baseband modulator, raised cosine filter as shown in Fig. 4.9, and data storage as shown in Fig. 4.10. The next steps, including cycle frequency profile calculation and artificial neural network, are operated as introduced in Section 4.4.1.

The percentage of correct classification with the hardware implementation is shown in Table 4.1. Note that even without the reliability check, the percentage of correct classification can be as high as 91.5%, which means that the proposed algorithm possesses the potential to be a viable PUE detector operating under real world conditions.

Table 4.1 Software-Defined Radio Experimental Results

With Reliability Check	98.3%
Without Reliability Check	91.5%

In terms of execution and convergence times, once the neural network has been trained,

[2]These two blocks were replaced by `SDRu Transmitter` and `SDRu Receiver` in the latter release, as shown in Section 6.3.1.

it only takes 0.698 seconds to classify 60 signals in the testing stage on a 1.6 GHz processor. Consequently, the proposed algorithm is a viable option for performing PUE detection in real time.

4.5 Chapter Summary

A novel algorithm for detecting non-intelligent primary user emulation attack has been presented in this chapter. This approach does not require any special hardware or software, and can be applied to mobile transmitters with unknown coordinates. Using USRP2 hardware experimentation, our work features an analysis in real-life channel with the effect of multipath fading and interferences. Both computer simulations and hardware implementations have shown that the proposed approach is feasible in real world conditions. The future work of this approach will be focus on other features of the users in the cognitive radio network.

Chapter 5

Proposed PUE Detector Based on Action Recognition[1]

In the previous chapter, the first category of PUE detection approach based on cyclostationary features has been discussed. However, it is possible that the authentic primary user and the primary user emulator use the same modulation type, where the first approach is not applicable. Therefore, we need to consider other features of users in the network. This chapter proposes the second category of approaches based on video processing method of action recognition in frequency domain, which explores the motion related features of the users.

Although the malicious secondary users can mimic the spectral characteristics of the primary users during the competition of gaining priority access to the wireless channels, they will not follow the spectral characteristics forever. As long as they successfully gain the access, they will switch back to their normal behavior. This change can be considered as an action in frequency domain. If the action of a suspicious user is different from that of an authentic primary user, we can tell this suspicious user is a malicious secondary user. This category analyzes the FFT sequences of wireless transmissions operating across a cognitive radio network environment, and classifies their actions in the frequency domain.

[1]This work has been published in parts in IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing [32].

5.1 Non-Database Approach

Since an FFT sequence can be considered as a video sequence, we adopt the action representation method proposed in [83] and [84], namely, we first extract optical flow features, such as velocity, velocity gradient, divergence, and vorticity, and we subsequently reduce the dimensionality of this feature set by computing an empirical covariance matrix. Since the resulting covariance descriptor belongs to a Riemannian manifold, we map it to a vector space by taking the matrix logarithm. However, unlike [83] and [84], we directly use the resulting log-covariance descriptor as the input to the artificial neural networks, because we only need to know whether a specific action belongs to the authentic primary user or not.

5.1.1 Proposed PUE Detection Algorithm

Without loss of generality, our proposed algorithm makes the following assumptions: (i) All the users, including the malicious users and primary users, are located within the same frequency band; (ii) Each user's transmission power is much higher than the ambient noise in the channel; (iii) The behavior of primary user is known, and it is different from the other users. Note that all the assumptions concerning modulation have been eliminated.

Fig. 5.1 provides a flow diagram of the proposed PUE detection algorithm. The structure of this algorithm is quite similar to the one proposed in Section 4.3, but now we feed the log-covariance descriptor into the artificial neural network, instead of the cycle frequency profile. First, the algorithm is initialized using energy detection in order to determine the frequency location of the potential PUE. Based on the assumption that the transmission power of all the PU and PUE is much higher than that of the SU, this step can record frequency location of the potential PUE. The energy detector employed in this step is the same as the one used in Section 4.3.1, so the numerical results can be adopted from Section 4.3.2. On the receiver side, the received signal of the potential PUE and its FFT plots are recorded along the time. After a certain period of time T, this observation process is terminated and the saved FFT plots are passed on to the classifier.

Consequently, the classifier calculates the log-covariance descriptor $\mathbf{l}_S$ of the FFT sequence, as introduced in Section 2.5.1, and uses an artificial neural network to classify the signal based on the action. If the reliability χ of the testing result is less than a constant number c specified at the beginning, we need to collect some new received data and run

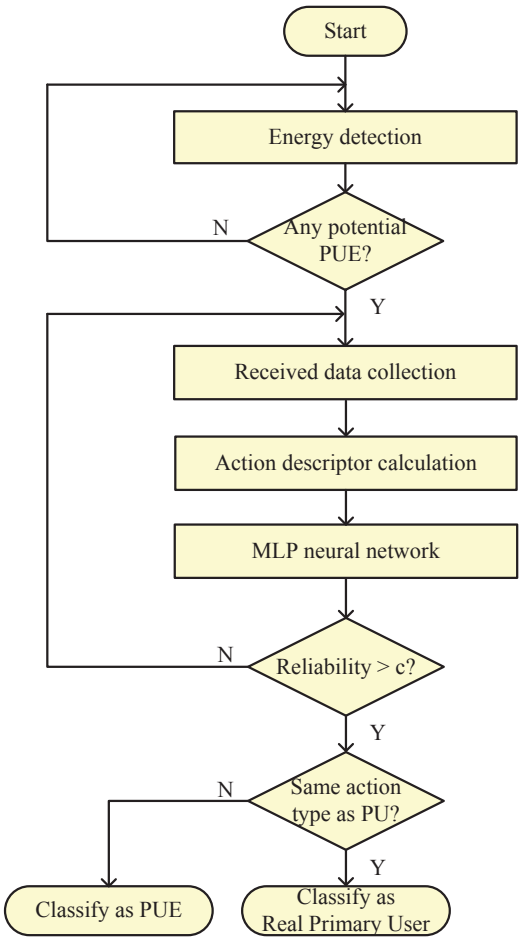


Fig. 5.1 Proposed PUE detection algorithm employing energy detection, action recognition and artificial neural networks.

the procedures above again. Otherwise, the neural network will output the classification result. Since the action type of primary users is known, we can readily identify whether the observed signal is from a real primary user or a malicious user.

5.1.2 Experimental Setup & Results

In this section, two different experiments are conducted in order to validate the performance of the proposed action recognition based classifier. The first experiment uses a computer simulation based on Simulink, while the second experiment is based on a hardware

implementation using the Universal Software Radio Peripheral (USRP) software-defined radio (SDR) platform. This section assumes that the energy detector has successfully identify a potential PUE given a probability of false alarm, so all the experiments will start from received data collection in Fig. 5.1. For simplicity, we assume that there is only one primary user in the system and the primary user's behavior is known in advance.

Computer Simulation

In this part, a Simulink model is constructed in order to collect the received signals and their FFT plots from different users. If there are N different users, then this model needs to contain N branches, which share an identical structure. Fig. 5.2 shows the structure of one of the branches.

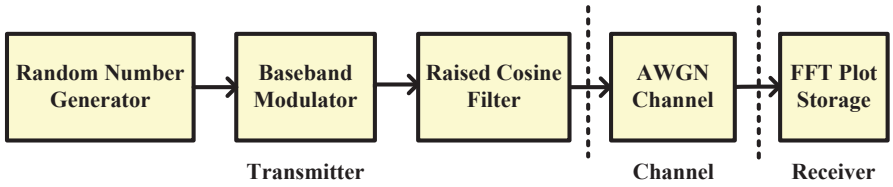


Fig. 5.2 The structure of one branch in the Simulink model. All the other branches have the identical structure.

In this model, there are three blocks on the transmitter side: random data is generated and modulated by same or different modulation schemes; a raised cosine filter is used for pulse-shaping in order to minimize intersymbol interference (ISI). Then, an AWGN channel block is applied to emulate the transmission environment. For this block, the signal-to-noise ratio (SNR) can be specified to represent channels of different noise level. In the end, a sink block is used on the receiver side to save the received signals and their FFT plots. These plots are stored away in a workspace for post processing, including the three steps introduced in Section 2.5.1.

In this dissertation, an multi-layer perceptron (MLP) neural network with 256 input nodes, one layer of 6 hidden nodes, and one output node is employed. $f(x) = \tanh(x)$ is selected as the activation function. For training, the back propagation algorithm is used with a fixed training constant of $\eta = 0.5$, and momentum constant $\zeta = 0.75$. The log-covariance descriptor vector is then fed into the system of artificial neural networks, and the system outputs a classification result along with a reliability parameter. If the

reliability parameter is larger than 0.75, the classification result is accepted.

With different SNR values ranging from -8 dB to 2 dB, Fig. 5.3 compares the percentage of correct classification using action recogniton-based method proposed in this chapter and the cyclostationary-based method proposed in the previous chapter. For a specific SNR value, the "Initial Seed" parameter of the "Random Number Generator" block is changed to get different input data so that we can repeat the experiment many times and average the resulting percentages. In both approaches, higher SNR values yield better algorithm performance in terms of successfully classifying primary signals and PUE signals. However, at each point, the action recogniton method yields a better performance than the cyclostationary method. As for the action recogniton-based method, even when the channel possesses a substantial amount of noise, such as when SNR = -8 dB, the percentage of correct classification can still reach 78% with a reliability check. Also, for an SNR above -5 dB, we are able to get at least 95% of the signals to be classified correctly.

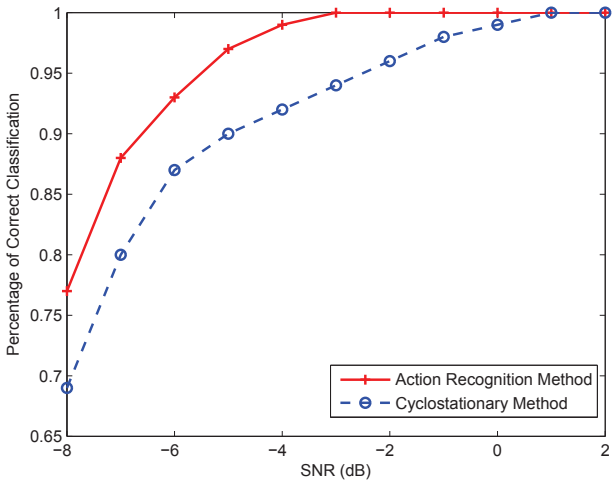


Fig. 5.3 The classification performance using action recogniton-based method and cyclostationary-based method in computer simulations. The x-axis represents SNR value, and the y-axis represents the percentage of correct classification. For the reliability check, all classifications with a reliability parameter less than 0.75 are ignored.

As mentioned in [72, 73], one of the limitations of cyclostationary-based classification is that it does not work well in the case when two users employing the modulation types that belong to the same modulation family, for example, QPSK and 8PSK. However, the

approach proposed in this chapter can solve this problem. As shown in Fig. 5.4, when one user employs QPSK and the other uses 8PSK, cyclostationary-based classification can only achieve approximately 50% percentage of correct classification, while action recogniton-based method is not affected at all.

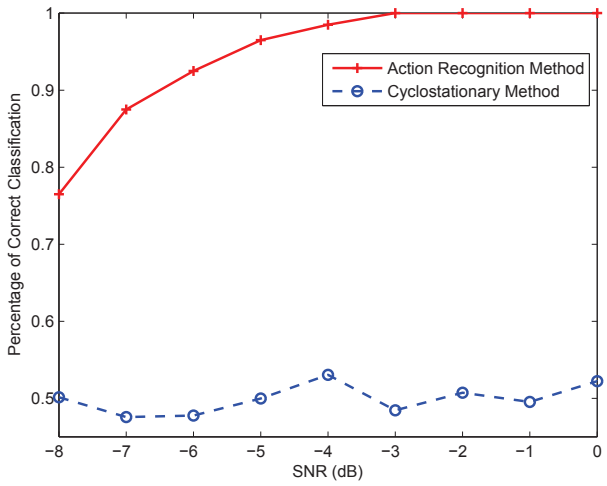


Fig. 5.4 The classification performance using action recogniton-based method and cyclostationary-based method in the case when one user employs QPSK and the other uses 8PSK. The x-axis represents SNR value, and the y-axis represents the percentage of correct classification. For the reliability check, all classifications with a reliability parameter less than 0.75 are ignored.

With respect to the time required to successfully classify a signal, although it takes some time for the artificial neural network to be trained with the back propagation algorithm, this process can be conducted offline with some previously known training signals. Once the artificial neural network has been trained, it only needs to be evaluated rather than both trained and evaluated with any newly intercepted signals. Our simulations show that it takes 0.632 seconds to classify 100 signals during the testing stage on a 1.6 GHz processor. However, it is also noticed that action descriptor vector calculation takes about the same time.

Software-Defined Radio Experiment

The Simulink model in Section 5.1.2 was then used as a starting point for the design of a hardware implementation. This was achieved by initially changing the AWGN channel

block with a real-life fading channel, and by using the Simulink USRP2 blocks available in the Communications System Toolbox. Consequently, our resulting Simulink design that operates on the USRP2 SDR platform can be found in Section 4.4.2. The transmitter design is shown in Fig. 4.9 and the receiver design is shown in Fig. 4.10. The next steps, including action descriptor vector calculation and artificial neural network, are operated as introduced in Section 5.1.2.

The percentage of correct classification with the hardware implementation is shown and compared in Table 5.1. Note that for the new approach proposed in this chapter, even without the reliability check, the percentage of correct classification can be as high as 93.8%, which means that the proposed algorithm possesses the potential to be a viable PUE detector operating under real world conditions.

Table 5.1 Software-Defined Radio Experimental Results

	Approach in Chapter 4	Proposed Approach
With Check	98.3%	99.2%
Without Check	91.5%	93.8%

In terms of execution times, once the neural network has been trained, it only takes 0.695 seconds to classify 100 signals in the testing stage on a 1.6 GHz processor. Considering this, the proposed algorithm is a viable option for performing PUE detection in real time.

5.2 Database Assisted Approach

In Section 5.1.2, when conducting the experiments, we assume that there is only one primary user in the system, so all the operations can be done in very short time. However, when there are several primary users in the system, it requires more computations when constructing the covariance descriptor of feature vectors, as well as testing with the artifical neural network, so it may not satisfy those scenarios that real-time processing is a priority. To resolve this new issue, in this section, we propose a PUE detection approach that builds upon the previous approach, and at the same time, introduces a relational database system in order to overcome the problem of intensive computation.

This new approach records the feature vectors of primary users in the database system, then it monitors each user's FFT sequence and compares the unknown users' feature vectors with those in the database. In most applications, primary users possess routine wireless

transmissions, so they have a limited number of feature vectors, which means the resulting database is stable and limited in size. In case that an unknwon user's feature vecotr has a match entity in the database, this approach will continue to double check its action in the frequency domain using artifical neural network. Otherwise, this unknown user will be classified as a PUE. Our approach operates on intercepted signals and analyzes it in the frequency domain over a time interval. Besides the benefits of our previous approach, our new approach takes the stability of primary users into account and creates a database system, so it is can save some computations compared to the previous approach.

5.2.1 Proposed PUE Detection Algorithm

Without loss of generality, our proposed algorithm makes the following assumptions: (i) All the users, including the malicious users and primary users, are located within the same frequency band; (ii) Each user's transmission power is much higher than the ambient noise in the channel; (iii) The actions and the corresponding feature vectors of primary users are known, and they are different from the other users.

Fig. 5.5 provides a flow diagram of the proposed PUE detection algorithm, which is built upon the one proposed in Section 5.1.1, but now we add a step of database search before feeding the log-covariance descriptor into the artificial neural network.

First, the algorithm is initialized using energy detection in order to determine the frequency location of the potential PUE. Based on the assumption that the transmission power of all the PU and PUE is much higher than that of the SU, this step can record frequency location of the potential PUE. The energy detector employed in this step is the same as the one used in Section 4.3.1, so the numerical results can be adopted from Section 4.3.2. On the receiver side, the received signal of the potential PUE and its FFT plots are recorded along the time. After a certain period of time T, this observation process is terminated and the saved FFT plots are passed on to the classifier.

Consequently, the classifier calculates the feature vectors of the moving pixels, as introduced in Section 2.5.1, and searches the database to see whether it can find a match record. If it does not, then this user is considered as a PUE. Otherwise, the classifier proceeds to calculate the log-covariance descriptor $\mathbf{l}_S$ of the FFT sequence, as introduced in Section 2.5.1, and uses an artifical neural network to classify the signal based on the action. If the reliability χ of the testing result is less than a constant number c specified at the beginning,

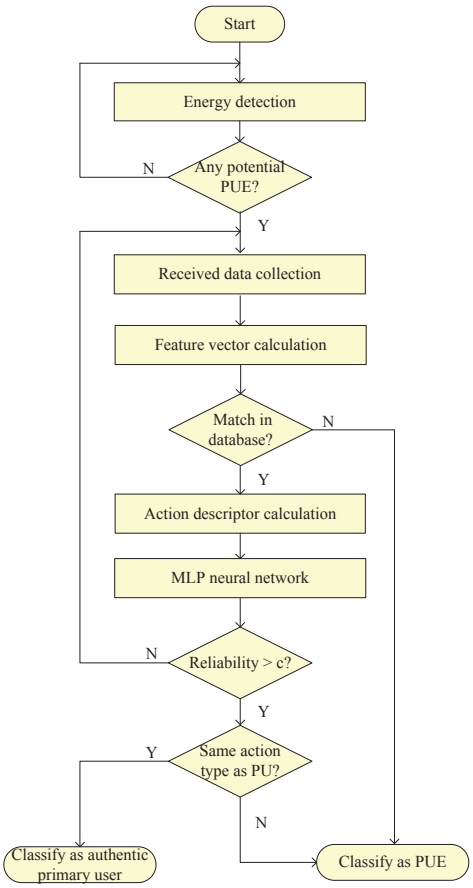


Fig. 5.5 Proposed PUE detection algorithm employing action recognition, relational database and artifical neural network.

we need to collect some new received data and run the procedures above again. Otherwise, the neural network will output the classification result. Since the action type of primary users is known, we can readily identify whether the observed signal is from a real primary user or a malicious user.

5.2.2 Experimental Setup & Results

In this section, two different experiments are conducted in order to validate the performance of the proposed database assisted classifier. The first experiment uses a computer simulation based on Simulink, while the second experiment is based on a hardware implementation using the Universal Software Radio Peripheral (USRP) software-defined radio

(SDR) platform. This section assumes that the energy detector has successfully identify a potential PUE given a probability of false alarm, so all the experiments will start from received data collection in Fig. 5.5. The database is created and updated using MATLAB. For simplicity, we assume that each user has one and only one action throughout the observation time window and the primary users' behavior is known in advance.

Computer Simulation

In this part, a Simulink model is constructed in order to collect the FFT plot of a user. If there are N different users, then we need to have N different parameter settings for this model. Fig. 5.6 shows the structure of this model.

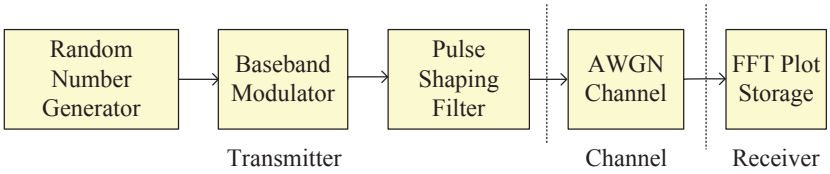


Fig. 5.6 The structure of the Simulink model used to collect the FFT plot of a user.

In this model, there are three blocks on the transmitter side: a random data number generator, a modulator, and a pulse-shaping filter. By picking up the different modulation type and filter type, or by setting the different parameters for these three blocks, we can express different users and get different FFT plots. Then, an AWGN channel block is applied to emulate the transmission environment. For this block, the signal-to-noise ratio (SNR) can be specified to represent channels of different noise level. In the end, a sink block is used on the receiver side to save the FFT plots. These plots are stored away in the workspace for post processing, including the three steps introduced in Section 2.5.1.

According to Section 2.6, a relational database is created to store the feature vectors of the primary users. Although there are several widely used database management tools, such as MySQL, Oracle and Access, in order to facilitate the connection with the Simulink model, we build this database using MATLAB. We can use the `read` function to search the database and the `write` function to update the database, if there exists new primary user.

In this section, an multi-layer perceptron (MLP) neural network with 256 input nodes, one layer of 6 hidden nodes, and one output node is employed. $f(x) = \tanh(x)$ is selected

as the activation function. For training, the back propagation algorithm is used with a fixed training constant of $\eta = 0.5$, and momentum constant $\zeta = 0.75$. The log-covariance descriptor vector is fed into the system of artificial neural networks, and the system outputs a classification result along with a reliability parameter. If the reliability parameter is larger than 0.75, the classification result is accepted.

As mentioned at the beginning of Section 5.2, one of the drawbacks of the frequency domain action recognition approach proposed in Section 5.1 is that it requires intensive computations when calculating the log-covariance descriptor vector and testing using the artificial neural network, becuase this approach solely relies on the artificial neural network. With the help of the database system, in many cases, we can get the result by just searching the database, and thus avoiding the calculation of the covariance matrix and the remaining steps. In Fig. 5.7, we compare the time to classify an unknown signal using the database-assisted approach proposed in this section and the non-database approach proposed in Section 5.1. This time does not include the training time of the artificial neural network, becuase this process can be conducted offline with some previously known training signals. Once the artificial neural network has been trained, it only needs to be evaluated rather than both trained and evaluated with any newly intercepted signals.

It's very obvious that for both approaches, the classification time is highly related to the number of primary users. When there are more primary users in the system, it costs more time to get the conclusion. However, it is noted that with a larger number of primary users, the classification time increases more dramatically for the non-database approach, which is approximately an exponential growth. While for the database-assisted approach, it is approximately a linear growth. It is because in the new approach, the classification time is dominated by the database searching time. According to the cost model introduced in [88], the database searching time is proportional to the number of tables to scan. Since each primary user has one corresponding table, when there are N primary users, there are N tables to scan. Based on this figure, we can pick up the appropriate approach depending on the number of primary users. If this number is small, the classification time is comparable for both approaches. However, if this number is large, the database-assisted approach is much more efficient than the non-database approach.

Besides efficiency, one other important performance metric is the percentage of correct classifications, because it shows whether an approach is reliable or not. With different SNR values ranging from -6 dB to 2 dB, assume there are 5 primary users in the system (equal

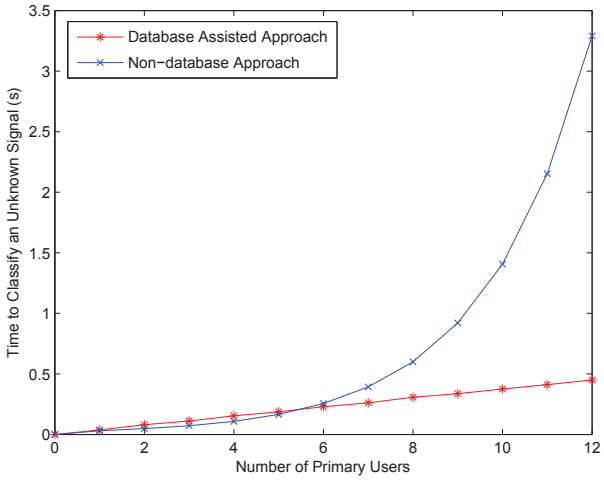


Fig. 5.7 Time to classify an unknown signal using the database-assisted approach and the non-database approach. The x-axis represents the number of primary users, and the y-axis represents the time to classify an unknown signal. For the reliability check, all classifications with a reliability parameter less than 0.75 are ignored.

classification time), Fig. 5.8 compares the percentage of correct classification using two different approaches. For a specific SNR value, we can change the parameter settings on the transmitter side to generate different FFT plots so that we can repeat the experiment many times and average the resulting percentages. In both approaches, higher SNR values yield better algorithm performance in terms of successfully classifying primary signals and PUE signals. Besides, both approaches have an identical performance for each SNR value. Even when the channel possesses a substantial amount of noise, such as when SNR = -6 dB, the percentage of correct classification can still reach 75% with a reliability check. Also, for an SNR above -2 dB, we are able to get at least 95% of the signals to be classified correctly. These results show that the two approaches are very reliable and robust.

Software-Defined Radio Experiment

The Simulink model in Fig. 5.6 was then used as a starting point for the design of a hardware implementation. This was achieved by initially changing the AWGN channel block with a real-life fading channel, and by using the Simulink SDRu blocks available in

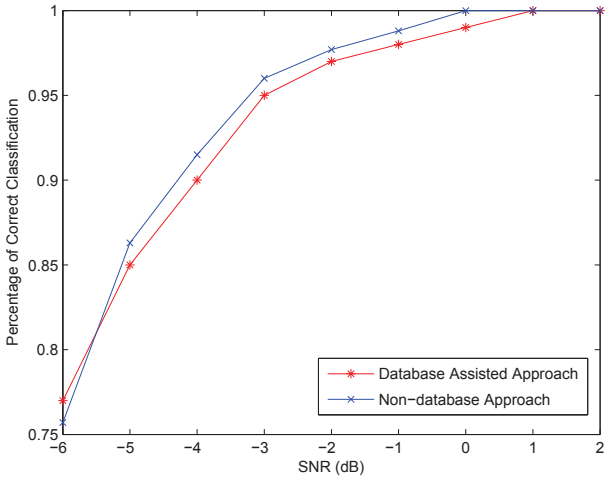


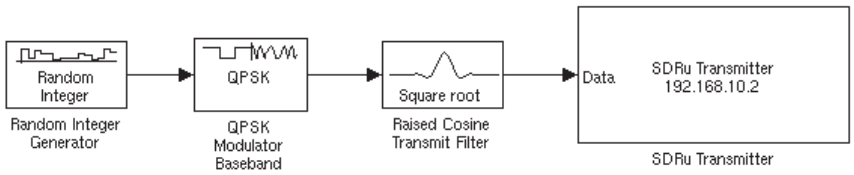
Fig. 5.8 The classification performance using the database-assisted approach and the non-database approach in computer simulations, assuming there are 5 primary users in the system. The x-axis represents SNR value, and the y-axis represents the percentage of correct classification. For the reliability check, all classifications with a reliability parameter less than 0.75 are ignored.

the Communications System Toolbox. Consequently, our resulting Simulink design that operates on the USRP SDR platform is shown in Fig. 5.9.

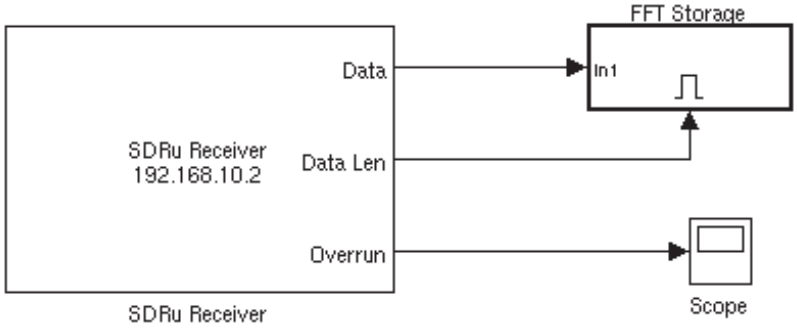
In order to incorporate the USRP2 hardware into the existing Simulink model, two Simulink blocks called `SDRu Transmitter` and `SDRu Receiver` are used here as interfaces. These two blocks are developed by The MathWorks and have been available since the R2011a release of MATLAB. With these two blocks, the USRP SDR platforms can be used in conjunction with the previous Simulink design environment.

The rest of the Simulink model remains the same, including the random number generator, baseband modulator, pulse shaping filter as shown in Fig. 5.9(a), and FFT storage as shown in Fig. 5.9(b). The next steps, including the database search, action descriptor vector calculation and artificial neural network, are the same as in Section 4.4.1.

Assume there are 5 primary users in the system, the percentage of correct classification with the hardware implementation is shown and compared in Table 5.2. Similar to the results derived from computer simulations, the two approaches have very close performance, and the new approach is slightly better than the previous approach. Note that for the



(a) The Simulink model for transmitter, which includes an SDRu Transmitter block.



(b) The Simulink model for receiver, which includes an SDRu Receiver block.

Fig. 5.9 The structure of hardware implementation framework.

new approach proposed in this section, even without the reliability check, the percentage of correct classification can be as high as 87.8%, which means that the majority of the classification results are correct, so the proposed algorithm possesses the potential to be a viable PUE detector operating under real world conditions.

Table 5.2 Software-Defined Radio Experimental Results

	Non-databse Approach	Databse-assisted Approach
With Check	91.5%	92.3%
Without Check	86.6%	87.8%

In terms of execution and convergence times, we do not take the training time for the artifical neural network into account. When the FFT plot of an unknown user is collected, it takes 0.35 s for the new approach and 0.45 s for the previous approach to output the result. Considering both the efficiency and the performance, the new approach proposed in this section is a good candidate for the real world implementation.

5.3 Chapter Summary

Two novel algorithms for detecting primary user emulation attack have been presented in this chapter. These two approaches are based on video processing method of action recognition in frequency domain, which explores the motion related features of the users. By introducing a relational database system, we can improve the time efficiency of the algorithm, especially in the case when there are multiple primary users. Both computer simulations and hardware implementations have shown that the proposed approaches are feasible in real world conditions. The future work of this chapter is to design a detection approach employing a distributed sensor network.

Chapter 6

Proposed PUE Detector Based on Distributed Sensor Network

All the PUE detection approaches proposed so far assume there is a single-node PUE detector in the network. However, in order to improve the efficiency and accuracy of the detection, we can actually employ a distributed sensor network as our detector, as shown in Fig. 6.1, where each sensor node works as an independent PUE detector. For an unknown user in this network, each sensor node makes its own decision as an energy detector and a classifier. Based upon this, a final detection result will be made.

When working as a classifier, each node of this sensor network can employ one of the three approaches proposed in Chapter 4 and 5. However, the real emphasis of this work is how these nodes collaborate to obtain the final detection results for the whole network.

6.1 System Model

In this chapter, we consider a cognitive radio network as shown in Fig. 6.2. All the users, including the primary users, primary user emulators and secondary users, as well as the PUE detectors are distributed in a circular grid. In order to avoid interference, we assume at each time, there is only one user transmitting in this network.

Similar to the model in Section 4.2, let $x(t)$ denote the transmitted signal. If it is the authentic PU signal, $x(t)=s(t)$. If it is the PUE signal, $x(t)=s'(t)$. Since the PUE signal is very similar to the PU signal, we assume both $s(t)$ and $s'(t)$ are independently and

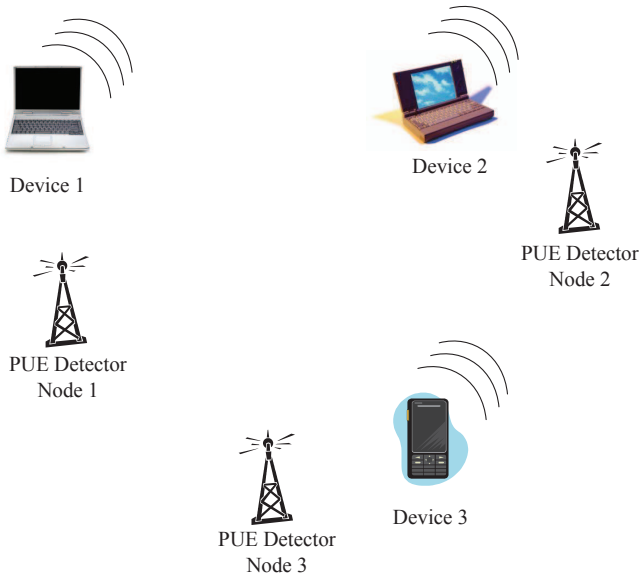


Fig. 6.1 A contention based dynamic spectrum access network that employs a three-node distributed sensor network as the PUE detector, where each sensor node works as an independent PUE detector. For an unknown user in this network, each sensor node makes its own decision and a final detection result will be made based upon this.

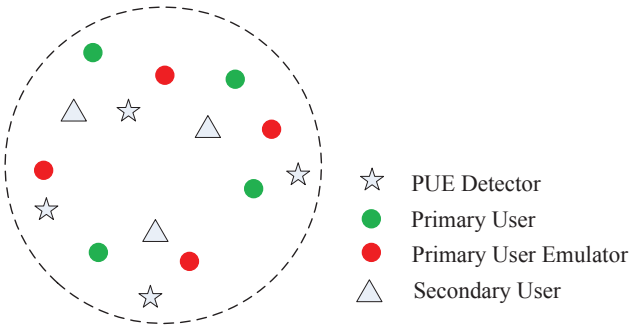


Fig. 6.2 A cognitive radio network in a circular grid.

identically distributed (iid) random processes with mean zero and variance σ_s^2, namely:

$$s(t), s'(t) \sim \mathcal{N}(0, \sigma_s^2), \tag{6.1}$$

where $\mathcal{N}(\cdot)$ denotes the normal distribution. Since the secondary users have a significant lower transmitted power than the primary users, we assume $x(t) = 0$ when the SU is transmitting.

Let $h_i(t)$ and $n_i(t)$ denote the impulse response and the noise of the channel between the transmitted signal and the ith PUE detector. We assume the channel is a slow flat fading channel during the observation process, so $h_i(t)$ becomes a constant gain h_i. $n_i(t)$ is the additive white Gaussian noise (AWGN) with mean zero and variance σ_n^2, namely:

$$n_i(t) \sim \mathcal{N}(0, \sigma_n^2). \tag{6.2}$$

Therefore, for the ith $(1 \leq i \leq M)$ PUE detector, there are three possible received signals:

$$y_i(t) = \begin{cases} n_i(t) & \text{SU}, \\ h_i \times s(t) + n_i(t) & \text{PU}, \\ h_i \times s'(t) + n_i(t) & \text{PUE}, \end{cases} \tag{6.3}$$

where $y_i(t)$ is the received signal at the ith PUE detector. The PUE detection algorithm presented in Section 6.2 will differentiate these three cases at each detector, and then combine their results into a final decision.

6.2 Proposed PUE Detection Algorithm

Fig. 6.3 provides a flow diagram of the proposed PUE detection algorithm, which looks similar to the previous two algorithms. However, in order to incorporate the sensor network, there is a round of voting in the whole process. Please note when working as a classifier in the second step, the nearest node can employ one of the three approaches proposed in Chapter 4 and 5. In Fig. 6.3, we show the database approach.

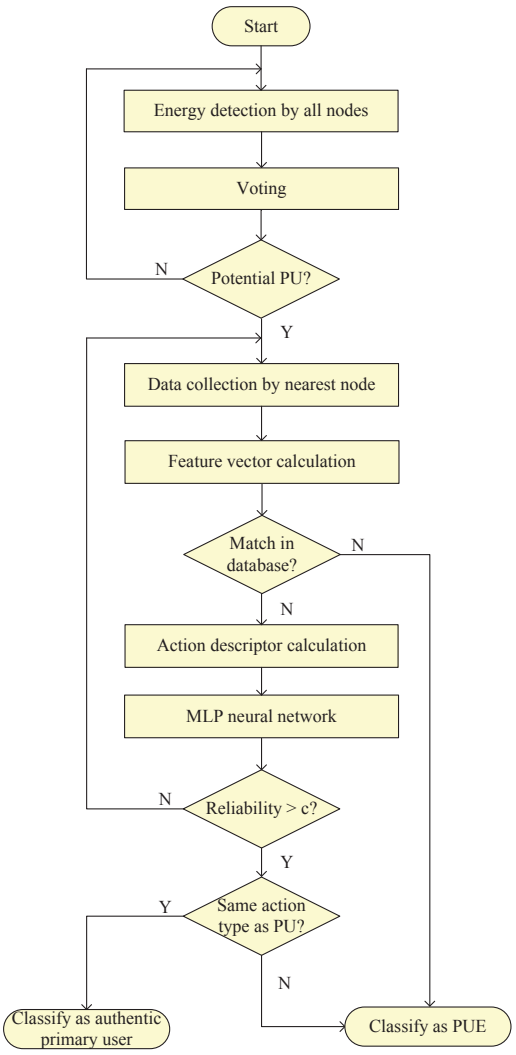


Fig. 6.3 Proposed PUE detection algorithm based on distributed sensor network.

According to the system model, our proposed approach makes the following assumptions: (i) All the users, including the malicious users and primary users, are located within the same frequency band; (ii) For each period of time, there is only one user transmitting;

First, the algorithm is initialized using energy detection in order to determine the frequency location of the potential primary user based on the assumption that the transmission power of the PU and PUE is much higher than that of the SU. For each given interval, all the detector nodes scan the same frequency bin and try to differentiate the following two cases of the received signal:

$$y_i(t) = \begin{cases} n_i(t) & \text{SU}, \\ h_i \times x(t) + n_i(t) & \text{PU \& PUE}, \end{cases} \tag{6.4}$$

where $y_i(t)$ is the received signal at the ith detector, $x(t) = s(t)$ or $s'(t)$.

In this step, each sensor node will make a detection concerning whether the received signal belongs to a potential PU or not. Suppose there are M sensor nodes in the network, and the detection result is either 1 (potential PU) or 0 (SU). For an unknown received signal, the result is:

$$r = \begin{cases} 1 & \sum_{i=1}^{M} r[i] \geq \frac{M}{2}, \\ 0 & \text{otherwise}, \end{cases} \tag{6.5}$$

where $r[i]$ is the detection result from the ith sensor node.

In other words, if majority of the sensor nodes decide this is a potential primary user, the algorithm will continue to the second step and this signal will be recorded along the time by its nearest node. After a certain period of time T, this observation process is terminated and the saved signals are passed on to the classifier.

This algorithm features the energy detection by all the nodes and data collection by only the nearest not. Thus, it greatly eliminates the overhead of long sensing time caused by energy detection and high computations due to feature calculations.

6.2.1 Mathematical Analysis

In this section, compared to the single-node case in Section 4.3.1, we will study how the sensor network will impact the performance of energy detector in terms of probability of false alarm and probability of detection.

For each detector node, the hypothesis testing still exists and is the same as the one in Section 4.3.1, so we can directly employ the conclusion from there. For each detector node, the probability of false alarm and probability of detection are as follows:

$$P_{F_i} = \frac{\Gamma(N, \frac{T_i}{2\sigma_{0i}^2})}{\Gamma(N)}, \tag{6.6}$$

and

$$P_{D_i} = \frac{\Gamma(N, \frac{T_i}{2\sigma_{1i}^2})}{\Gamma(N)}, \tag{6.7}$$

where $\sigma_{0i}^2 = \sigma_n^2$ and $\sigma_{1i}^2 = h_i^2\sigma_s^2 + \sigma_n^2$. In order to simplify our analysis, we assume that h_i is the same for all the channels. Therefore, the P_{F_i} and P_{D_i} for each detector are identical, and can be expressed as:

$$P_F = \frac{\Gamma(N, \frac{T}{2\sigma_0^2})}{\Gamma(N)}, \tag{6.8}$$

and

$$P_D = \frac{\Gamma(N, \frac{T}{2\sigma_1^2})}{\Gamma(N)}, \tag{6.9}$$

where $\sigma_0^2 = \sigma_n^2$ and $\sigma_1^2 = h^2\sigma_s^2 + \sigma_n^2$.

Since we use voting to determine the result of energy detection, the overall probability of false alarm and probability of detection can be calculated using the law of total probability:

$$Q_F = \sum_{i=\frac{M}{2}}^{M} \binom{M}{i} P_F^i (1 - P_F)^{M-i}, \tag{6.10}$$

and

$$Q_D = \sum_{i=\frac{M}{2}}^{M} \binom{M}{i} P_D^i (1 - P_D)^{M-i}, \tag{6.11}$$

where P_F is from (6.8) and P_D is from (6.9).

6.2.2 Numerical Results

This section provides the numerical results of the energy detector introduced in Section 6.2.1. The results are based on (6.10) and (6.11), presented in terms of the ROC curves

(*i.e.*, Q_D versus Q_F) for an AWGN fading channel.

For each sensor node, we can use the P_F and P_D data derived from Section 4.3.1. Since (6.8) and (6.9) are complement of the binomial cumulative distribution function [113, 114], there is a useful MATLAB function `binocdf` that we can use [1].

In order to have a quick evaluation of the sensor network, we do the following calculation based on one node in Fig 4.4(b), where $P_F = 0.1$ and P_D=0.8. Assume we use a sensor network of 4 nodes, then

$$Q_D = 1 - \texttt{binocdf(2,4,0.8)} = 0.82 > P_D, \tag{6.12}$$

and

$$Q_F = 1 - \texttt{binocdf(2,4,0.1)} = 0.0037 < P_F, \tag{6.13}$$

which means that by employing the sensor network, we can not only improve the overall probability of detection, but also decrease the overall probability of false alarm.

Based on the ROC curves of single node detector, the ROC curves of the distributed sensor network can be obtained by changing the number of sensor nodes M. The ROC curves in Fig. 6.4(a) are plotted by changing M from 4 to 10 when N=5, SNR=5, and the ROC curves in Fig. 6.4(b) are plotted by changing M from 4 to 10 when N=5, SNR=3. In these two figures, the ROC curves of the single node detector (M=1) are also provided for reference.

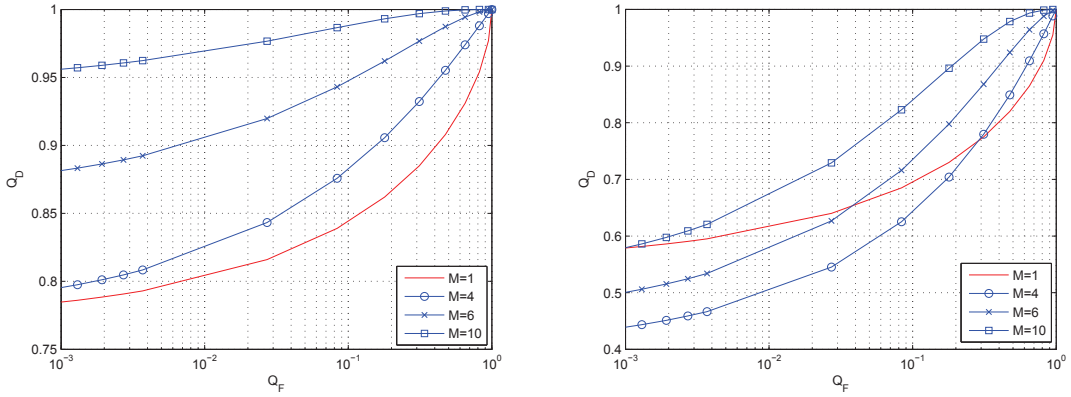
Based on Fig. 6.4, we can come to the following conclusions:

- Employing the distributed sensor network is an effective way of improving the performance of the energy detection if each sensor node of this network has a reliable performance, as shown in Fig. 6.4(a).

- If the sensor node does not have a good P_D, the overall performance of the sensor network can be even worse. For example, in Fig. 6.4(b), when M=4, the distributed sensor network has a lower P_D than the single node detector in the area of $Q_F <$ 0.3. Therefore, we need to make sure that each sensor node in this network has an acceptable performance.

[1] `Y = binocdf(X,N,P)` computes a binomial CDF at each of the values in X using the corresponding number of trials in N and probability of success for each trial in P [115].

- Given a probability of false alarm, the larger number of sensor nodes M will yield higher overall probability of detection.



(a) ROC curves by varying the number of sensor nodes M from 4 to 10 given N=5, SNR=5.

(b) ROC curves by varying the number of sensor nodes M from 4 to 10 given N=5, SNR=3.

Fig. 6.4 Numerical results of the energy detector in terms of the ROC curves.

6.3 Experimental Setup & Results

In this section, two different experiments are conducted in order to validate the performance of the classifier conducted by the nearest node. The first experiment uses a computer simulation based on Simulink, while the second experiment is based on a hardware implementation using the Universal Software Radio Peripheral (USRP) software-defined radio (SDR) platform. The classifier is tested in two aspects, accuracy and efficiency, with an emphasis on the impact of distance, *i.e.*, nearest node.

6.3.1 Path-loss Modeling

In most environment, it is observed that the radio signal strength falls as some power α of the distance, called the power-distance gradient or path-loss gradient [116]. Depending on the radio frequency, there are additional losses, and in general the relationship between the transmitted power P_t and the received power P_r in free space is given by:

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi d}\right)^2, \tag{6.14}$$

where G_t and G_r are the transmitter and receiver antenna gains, λ is the wavelength of the carrier, and d is the distance between the transmitter and receiver [117].

Since all the detector nodes are equipped with the same antenna, so G_r is the same. According to our assumptions, for each period of time, there is only one user transmitting, so G_t and λ are the same. Therefore, given the transmitted power, the only variable for received power is d. We can rewrite (6.14) in decibels (dB) as:

$$10\log(P_r) = 10\log(P_0) - 20\log(d), \tag{6.15}$$

where P_0 is the received power at the first meter (d=1), which applies to all the detector nodes. Therefore, there is a 20 dB per decade loss in signal strength as a function of distance in free space [118].

Based on the single node simulation results in Chapter 4 and 5, it is obvious that the performance of the classifier is highly related to the signal-to-noise ratio (SNR) on the receiver, namely, higher SNR value yields better classification performance. Broadly speaking, SNR is the ratio of the average signal power to the average noise power:

$$\text{SNR} = \frac{P_r}{P_{\text{noise}}}. \tag{6.16}$$

Since the average noise power P_{noise} is fixed for all the detector nodes, in order to get a higher SNR value, a larger P_r is required. According to (6.15), a minimum distance d will lead to a maximum P_r, thus the nearest node is picked up to perform the classification.

Computer Simulation

In this part, a Simulink model is constructed in order to collect the FFT plot of a user. If there are N different users, then we need to have N different parameter settings for this model. Fig. 6.5 shows the structure of this model.

In this model, there are three blocks on the transmitter side: a random number generator, a modulator, and a pulse-shaping filter. By picking up the different modulation type and filter type, or by setting the different parameters for these three blocks, we can express different users and get different FFT plots. Then, a free space path loss block [2] and an AWGN channel block are applied to emulate the transmission environment. Specifically,

[2]The free space path loss block belongs to the RF Impairments Library.

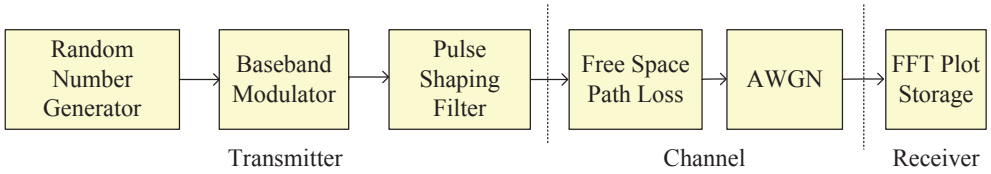


Fig. 6.5 The structure of the Simulink model used to collect the FFT plot of a user.

the free space path loss block is employed to simulate the loss of signal power due to the distance between transmitter and receiver [119] as expressed in (6.15). This block reduces the amplitude of the transmitted signal by an amount related to d. Then, the AWGN channel block represents the noise level by setting the variance of the white Gaussian noise. In the end, a sink block is used on the receiver side to save the FFT plots. These plots are stored away in the workspace for post processing, including the three steps introduced in Section 2.5.1.

For each sensor node, a relational database is created to store the feature vectors of the primary users that are close to this node. In most cases, the primary users are stationary in the system, so the database is quite stable. Compared to the approach in Section 5.2, there is a significant advantage brought by the distributed sensor network. With the single node detection approach, all the feature vectors exist in one large database. However, with the distributed sensor network, those feature vectors will be divided into smaller databases. Therefore, it will greatly reduce the time to search the database. Although there are several widely used database management tools, such as MySQL, Oracle and Access, in order to facilitate the connection with the Simulink model, we build this database using MATLAB. We can use the `read` function to search the database and the `write` function to update the database, if there exists new primary user.

In this section, an multi-layer perceptron (MLP) neural network with 256 input nodes, one layer of 6 hidden nodes, and one output node is employed. $f(x) = \tanh(x)$ is selected as the activation function. For training, the back propagation algorithm is used with a fixed training constant of $\eta = 0.5$, and momentum constant $\zeta = 0.75$. The log-covariance descriptor vector is fed into the system of artificial neural networks, and the system outputs a classification result along with a reliability parameter. If the reliability parameter is larger than 0.75, the classification result is accepted.

The most important performance metric of a classifier is the percentage of correct

classifications, which shows whether an approach is accurate or not. In most cases, we would like this percentage as high as possible. The first experiment will show how distance affects this percentage. In order to incorporate the variable of distance, in free space path loss block, we choose "Distance and Frequency" in the "Mode" field, and then specify the distance between transmitter and receiver. For a specific distance value, we can change the parameter settings on the transmitter side to generate different FFT plots so that we can repeat the experiment many times and average the resulting percentages. Fig. 6.6 shows the percentage of correct classifications given by different distances d. Based on the figure, with distance values ranging from 1m to 10m, the percentage of correct classifications drops dramatically from around 90% to 20%. More specifically, smaller distance value yields better algorithm performance in terms of successfully classifying primary signals and PUE signals. Therefore, in order to get the optimal classification performance, we need to pick up the nearest node to be the classifier.

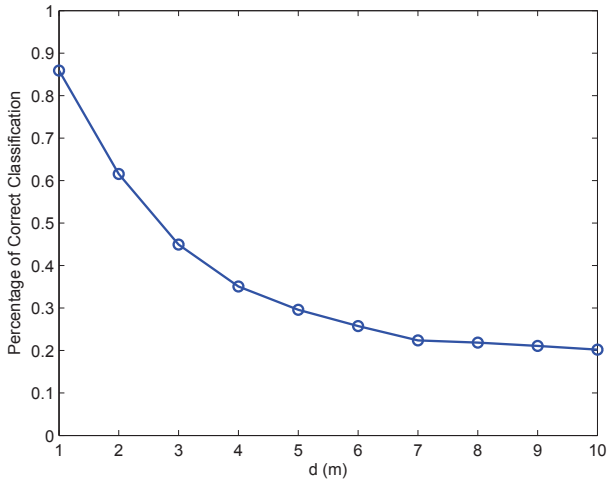


Fig. 6.6 The classification performance using the database-assisted approach in computer simulations. The x-axis represents distance value d, and the y-axis represents the percentage of correct classification. For the reliability check, all classifications with a reliability parameter less than 0.75 are ignored.

The second most important performance metric of a classifier is the time spent to classify the known signals, which shows whether an approach is efficient or not. In most cases, we would like this time as short as possible. The second experiment will show how distributed sensor network affects this time. As mentioned at the beginning of this section, one of the

advantages of the nearest node classification is that all the feature vectors are divided into smaller databases. In many cases, we can get the result by just searching one or several small databases, and thus avoiding going through all the feature vectors. In Fig. 6.7, we compare the time to classify an unknown signal using the distributed network detector proposed in this chapter and the single node detector proposed in Section 5.2. Assume the distributed network detector consists of two sensor nodes. This time does not include the training time of the artificial neural network, becuase this process can be conducted offline with some previously known training signals. Once the artificial neural network has been trained, it only needs to be evaluated rather than both trained and evaluated with any newly intercepted signals.

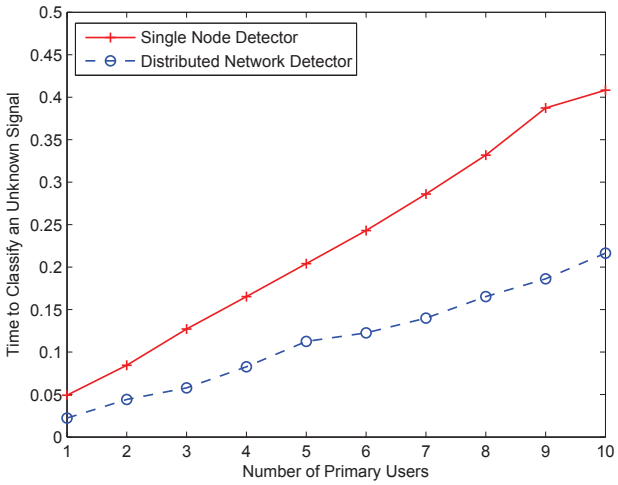


Fig. 6.7 Time to classify an unknown signal using the distributed network detector and the single node detector. The x-axis represents the number of primary users, and the y-axis represents the time to classify an unknown signal. For the reliability check, all classifications with a reliability parameter less than 0.75 are ignored.

It is very obvious that for both approaches, the classification time is highly related to the number of primary users. When there are more primary users in the system, it costs more time to get the conclusion. However, it is noted that given a fixed number of primary users, it always takes less time for the nearest node approach to classify. According to the cost model introduced in [88], the database searching time is proportional to the number of tables to scan. Since each primary user has one corresponding table, when

there are N primary users, there are N corresponding tables. Given an unknown user, the single detector approach in Section 5.2 has to scan all the tables to make the classification. However, the nearest node approach in this section only needs to scan the tables stored in the unknown user's nearest detector node to make the decision, thus saving a substantial amount of time.

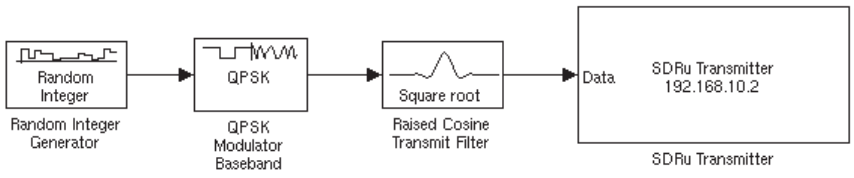
Software-Defined Radio Experiment

The Simulink model in Fig. 6.5 was then used as a starting point for the design of a hardware implementation. For simplicity, a small scale distributed sensor network is constructed in this section, which includes one unknown user as a transmitter and two sensor nodes as receviers. Based on Fig. 6.5, this was achieved by changing the AWGN channel block with a real-life fading channel, setting up the radios in different locations to represent the impact of the free space path loss block, and by using the Simulink SDRu blocks available in the Communications System Toolbox. Consequently, our resulting Simulink design that operates on the USRP SDR platform is shown in Fig. 6.8.

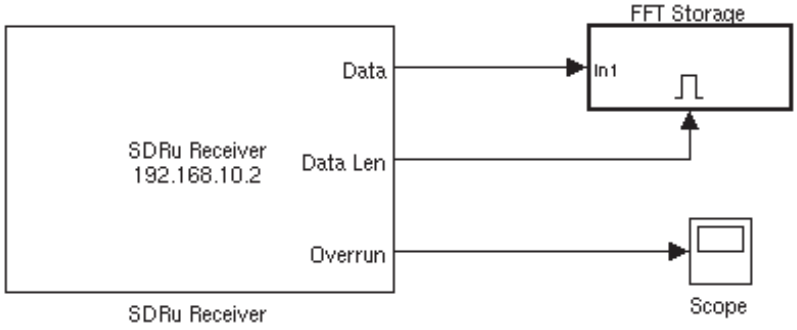
In order to incorporate the USRP2 hardware into the existing Simulink model, two Simulink blocks called `SDRu Transmitter` and `SDRu Receiver` are used here as interfaces. These two blocks are developed by The MathWorks and have been available since the R2011a release of MATLAB. With these two blocks, the USRP SDR platforms can be used in conjunction with the previous Simulink design environment.

The rest of the Simulink model remains the same, including the random number generator, baseband modulator, pulse shaping filter as shown in Fig. 6.8(a), and FFT storage as shown in Fig. 6.8(b). The next steps, including the database search, action descriptor vector calculation and artificial neural network, are the same as in Section 4.4.1, except that the feature vectors of the primary users are distributed in two databases.

Since there are two sensor nodes in the network, given an unknown user, there is one closer node, and one further node. In Table 6.1, the percentage of correct classifications of the two nodes is shown and compared with the hardware implementation. Similar to the results derived from computer simulations in Fig. 6.6, the closer node has a much better performance than the further node. Note that for the closer node, even without the reliability check, the percentage of correct detection can be as high as 85%, which means that the majority of the classification results are correct, so the proposed nearest



(a) The Simulink model for unknown user, which includes an SDRu Transmitter block.



(b) The Simulink model for sensor node, which includes an SDRu Receiver block.

Fig. 6.8 The structure of hardware implementation framework.

node classification possesses the potential to be a viable component of the PUE detector operating under real world conditions.

Table 6.1 Software-Defined Radio Experimental Results

	Closer Node	Further Node
With Check	90%	47%
Without Check	85%	43%

In terms of execution and convergence times, we do not take the training time for the artifical neural network into account. When the FFT plot of an unknown user is collected, it takes 0.2 s for the nearest node to output the result, which is faster than that of the single detector approach obtained in Section 5.2.2. Considering both the efficiency and the performance, the nearest node classification approach proposed in this section is a good candidate for the real world implementation.

6.4 Chapter Summary

A novel algorithm for detecting non-intelligent primary user emulation attack based on distributed sensor network has been presented in this chapter. This approach does not require any special hardware or software, and can be applied to mobile transmitters with unknown coordinates. Using USRP2 hardware experimentation, our work features an analysis in real-life channel with the effect of multipath fading and interferences. Numerical results, computer simulations and hardware implementations have shown that the proposed approach has a better performance than the single detector in terms of the accuracy and efficiency. The future work of this approach will be focus on the node selection of the distributed sensor network.

Chapter 7

Conclusions and Future Work

7.1 Completed Research Tasks

The following are the steps I have completed after the area exam. They are primarily concentrated on designing a primary user emulation detector using distributed sensor network. The others are based on the feedback from the committee in area exam.

- Design a primary user emulation detector using distributed sensor network. The algorithm proposed in Section 6.2 regarding integrating the results from all the sensor nodes is the focus of this work. In the phase of classification, the three PUE detection approaches proposed in Chapter 4 and 5 have been applied and verified respectively.
- Test the distributed sensor network detector using computer simulation and hardware implementation. The efficiency and accuracy of this detector have been compared with the single node detector in Chapter 4 and 5.
- Set up the system model for each case in order to conduct the mathematical analysis.
- Analyze the performance of energy detector in single node case and distributed sensor network case. It is completed by the mathematical analysis of hypothesis testing and its numerical results.
- Several more publications related to the work of PUE detection in cognitive radio network.

7.2 Future Work

Based on the work presented in this dissertation, there are several directions in which the results can be extended. The following is a short list.

- **Node Selection** The node selection for distributed sensor network plays a key role in determining the performance of the network because it can be utilized to improve accuracy and address the overhead issues [120]. The primary user emulation detector proposed in Section 6.2 employs all the nodes for energy detection and the nearest node for classification. It is possible that we can use a fixed set of nodes for both operations along with one round of voting for each operation. For example, we can assign the sensor nodes according to frequency bands. The number of sensor nodes assigned to each band is determined by the number of primary users registered within this band. If there are more primary users on a certain band, it is more likely that a primary user emulator will exist on this band, so more sensor nodes will help to make the correct decision. Given this type of node selection, the computations are distributed according to the probability of PUE attacks and the result of each round of voting will be:

$$r = \begin{cases} 1 & \sum_{i=m}^{n} r[i] \geq (n-m+1)/2, \\ 0 & \text{otherwise}, \end{cases} \tag{7.1}$$

  where $r[i]$ is the result provided by the ith node (either 0 or 1), and the mth to the nth ($m \leq n$) sensor nodes are allocated to this spectrum band. The number of sensor nodes from m to n is determined by:

$$\frac{n-m+1}{M} = p_{mn}, \tag{7.2}$$

  where p_{mn} is the percentage of primary users registered within this band, and M is the total number of sensor nodes.

- **Index of Database** Further improve the efficiency of database assisted PUE detection approach using index, as introduced in Section 2.6.3. A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes and increased storage space. Since the main operation on

database is read, and storage space is not a problem, index possesses the potential of further improving the efficiency.

- **Indoor and Outdoor** Comparative study of PUE detection in indoor environment and outdoor environment. So far, all the hardware implementation tests are conducted indoor. Since the communication channels outdoor are different from those indoor, and many practical applications happen outdoor, it is attractive to see how the approaches work outdoor.

References

[1] Sierra Wireless, "M2m wireless communications revenues and markets." [Online]: http://www.sierrawireless.com/Solutions/Newsletter_Market_Research.aspx.

[2] Neul, "Products." [Online]: http://www.neul.com/products.php.

[3] Neul, "Welcome to neul: World leaders in white space." [Online]: http://www.neul.com/.

[4] S. Haykin, "Cognitive radio: brain empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, pp. 201–220, February 2005.

[5] S. Pagadarai, R. Rajbanshi, and A. M. Wyglinski, *Cognitive Radio Communications and Networks: Principles and Practice*, ch. Agile Transmission Techniques. Elsevier, 2009.

[6] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Sginal Processing Magazine*, vol. 24, no. 3, pp. 79–89, 2007.

[7] K. Guo, P. Ishwar, and J. Konrad, "Action recognition in video by covariance matching of silhouette tunnels," in *Proceedings of Brazilian Symposium on Computer Graphics and Image Processing*, pp. 299–306, October 2009.

[8] Olifantasia, "Usrp prices and specifications." [Online]: http://www.olifantasia.com/drupal2/en/node/12.

[9] M. J. Leferman, "Rapid prototyping interface for software defined radio experimentation," Master's thesis, Worcester Polytechnic Institute, Worcester, MA, USA, Feb. 2010.

[10] Federal Communications Commission (FCC), "Spectrum Inventory Table 137 MHz to 100 GHz." [Online]: http://www.fcc.gov/oet/info/database/spectrum/.

[11] M. A. McHenry, P. A. Tenhula, D. McCloskey, D. A. Roberson, and C. S. Hood, "Chicago spectrum occupancy measurements analysis and a long-term studies proposal," in *Proceedings of Workshop on Technology and Policy for Accessing Spectrum*, (Boston, MA), August 2006.

[12] S. Pagadarai and A. M. Wyglinski, "A quantitative assessment of wireless spectrum measurements for dynamic spectrum access," in *Proceedings of the IEEE International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, (Hannover, Germany), June 2009.

[13] D. Cabric, S. Mishra, D. Willkomm, R. Brodersen, and A. Wolisz, "A cognitive radio approach for usage of virtual unlicensed spectrum," in *Proceedings of the 14th IST Mobile and Wireless Communications Summit*, June 2005.

[14] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Sginal Processing Magazine*, vol. 24, no. 3, pp. 79–89, 2007.

[15] M. A. McHenry, "Nsf spectrum occupancy measurements project summary," tech. rep., Shared Spectrum Company, August 2005.

[16] D. A. Roberson, C. S. Hood, J. L. LoCicero, and J. T. MacDonald, "Spectral occupancy and interference studies in support of cognitive radio technology deployment," in *Proceedings of IEEE Workshop*, (Boulder, CO, USA), March 2006.

[17] Federal Communications Commission, "In the matter of unlicensed operation in the tv broadcast bands, additional spectrum for unlicensed devices below 900 mhz and in the 3 ghz band.." [Online]: http://hraunfoss.fcc.gov/edocs public/attachmatch/FCC08260A1.pdf.

[18] Mobile Dev and Design, "Rural white-space network shows promise for broadband deployment." [Online]: http://mobiledevdesign.com/standards_regulations/rural-white-space-network-shows-promise-broadband-deployment-111309/.

[19] H. Harada, "Software defined radio prototype toward cognitive radio communication systems," in *Proceedings of the IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, 2005.

[20] H. Harada, "Regulatory perspective of japan," in *Proceedings of the VCE Regulatory Workshop*, (London, UK), Apr. 2007.

[21] H. Harada, "Advances in flexible radio technology to support cognitive radio," in *Proceedings of the VCE International Research Workshop on Intelligent Spectrum Usage for Personal Communications*, (London, UK), Apr. 2007.

[22] W. Webb, "IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks 2007 keynote presentation."

[23] K. E. Nolan, E. Ambrose, and D. O'Mahony, "Cognitive radio: Value creation and value migration," in *Proceedings of the SDR Forum Technical Conference 2006*, 2006.

[24] I. T. Union, "Radio-spectrum management for a converging world." [online]: http://www.itu.int/osg/spu/ni/spectrum/.

[25] R. Ercole, "Innovation, spectrum regulation, and dyspan technologies access to markets," in *Proceedings of the IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, 2005.

[26] L. Kovacs and A. Vidacs, "Spectrum auction and pricing in dynamic spectrum allocation networks," in *Proceedings of the IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, 2007.

[27] Shared Spectrum Company, "Shared spectrum company." [Online]: http://www.sharedspectrum.com/.

[28] Spectrum Bridge, "Spectrum bridge: Enabling universal spectrum access." [Online]: http://spectrumbridge.com/Home.aspx.

[29] Spectrum Bridge, "White space overview." [Online]: http://spectrumbridge.com/ProductsServices/WhiteSpacesSolutions/WhiteSpaceOverview.a

[30] Z. Jin, S. Anand, and K. P. Subbalakshmi, "Detecting primary user emulation attacks in dynamic spectrum access networks," in *Proceedings of the IEEE International Conference on Communications*, (Dresden, Germany), June 2009.

[31] D. Pu, Y. Shi, A. V. Ilyashenko, and A. M. Wyglinski, "Detecting primary user emulation attack in cognitive radio networks," in *Proceedings of the IEEE Global Telecommunications Conference*, December 2011.

[32] D. Pu and A. M. Wyglinski, "Primary user emulation detection using frequency domain action recognition," in *Proceedings of the 2011 IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing*, August 2011.

[33] F. Li and K. Wu, "Reliable, distributed and energy-efficient broadcasting in multihop mobile ad hoc networks," in *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*, (Tampa, FL), November 2002.

[34] R. Aldunate, S. F. Ochoa, F. Peña-Mora, and M. Nussbaum, "Robust mobile ad hoc space for collaboration to support disaster relief efforts involving critical physical infrastructure," *Journal of Computing in Civil Engineering*, 2006.

[35] Q. Liang, "Ad hoc wireless network trafficself-similarity and forecasting," *IEEE Communications Letters*, vol. 6, July 2002.

[36] K. Wongthavarawat and A. Ganz, "IEEE 802.16 based last mile broadband wireless military networks with quality of service support," in *Proceedings of the IEEE Military Communications Conference*, vol. 2, pp. 779–784, October 2003.

[37] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless Networks*, 2005.

[38] M. J. Zieniewicz, C. Douglas, D. C. Wong, and J. D. Flatt, "The evolution of army wearable computers," *IEEE Pervasive Computing*, October-December 2002.

[39] I. F. Akyildiz, W.-Y. Lee, and K. R. Chowdhury, "Crahns: Cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 810 – 836, 2009.

[40] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive mac for opportunistic spectrum access in ad hoc networks: A pomdp framework," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, pp. 589 – 600, 2007.

[41] K. Challapali, S. Mangold, and Z. Zhong, "Spectrum agile radio: Detecting spectrum opportunities," in *Proceedings of the 6th Annual International Symposium on Advanced Radio Technologies*, March 2004.

[42] M. P. Olivieri, G. Barnett, A. Lackpour, A. Davis, and P. Ngo, "A scalable dynamic spectrum allocation system with interference mitigation for teams of spectrally agile software defined radios," in *Proceedings of the IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, November 2005.

[43] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Proceedings of the Thirty-eight Asilomar Conference on Signals, Systems, and Computers*, November 2004.

[44] L. P. Goh, Z. Lei, and F. Chin, "Dvb detector for cognitive radio," in *Proceedings of the International Conference on Communications*, (Glasgow, Scotland), p. 64606465, June 2007.

[45] Y. Qi, T. Peng, W. Wang, and R. Qian, "Cyclostationarity-based spectrum sensing for wideband cognitive radio," in *Proceedings of the 2009 WRI International Conference on Communications and Mobile Computing*, (Washington, DC, USA), p. 107111, 2009.

[46] W. Xia, S. Wang, W. Liu, and W. Cheng, "Correlation-based spectrum sensing in cognitive radio," in *Proceedings of the 2009 ACM Workshop on Cognitive Radio Networks*, (New York, NY, USA), p. 6772, 2009.

[47] S. Anand, Z. Jin, and K. Subbalakshmi, "An analytical model for primary user emulation attacks in cognitive radio networks," in *Proceedings of IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, (Chicago, IL, USA), October 2008.

[48] Z. Jin, S. Anand, and K. Subbalakshmi, "Detecting primary user emulation attacks in dynamic spectrum access networks," in *Proceedings of IEEE International Conference on Communications*, (Dresden, Germany), June 2009.

[49] Z. Jin, S. Anand, and K. Subbalakshmi, "Mitigating primary user emulation attacks in dynamic spectrum access networks using hypothesis testing," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 2, pp. 74–85, 2009.

[50] Z. Jin, S. Anand, and K. Subbalakshmi, "Robust spectrum decision protocol against primary user emulation attacks in dynamic spectrum access networks," in *Proceedings of IEEE Global Telecommunications Conference*, (Miami, FL, USA), December 2010.

[51] Z. Jin, S. Anand, and K. Subbalakshmi, "Performance analysis of dynamic spectrum access networks under primary user emulation attacks," in *Proceedings of IEEE Global Telecommunications Conference*, (Miami, FL, USA), December 2010.

[52] Z. Jin, S. Anand, and K. Subbalakshmi, "Neat: A neighbor assisted spectrum decision protocol for resilience against primary user emulation attacks," tech. rep., Stevens Institute of Technology, December 2009.

[53] Z. Jin, S. Anand, and K. Subbalakshmi, "Impact of primary user emulation attacks on dynamic spectrum access networks," *IEEE Transactions on Communications*, vol. 60, no. 9, pp. 2635–2643, 2012.

[54] Z. Jin, *Primary user emulation attack in dynamic spectrum access networks: threats, mitigation and impact*. Licentiate dissertation, Stevens Institute of Technology, Hoboken, NJ, May 2012.

[55] H. Li and Z. Han, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part i: Known channel statistics," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3566–3577, 2010.

[56] H. Li and Z. Han, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part ii: Unknown channel statistics," *IEEE Transactions on Wireless Communications*, vol. 10, no. 1, pp. 274–283, 2011.

[57] R. Chen, J. M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Journal on Selected Areas in Communcations Special Issue on Cognitive Radio Theory and Applications*, 2008.

[58] P. N. Yao Liu and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *Proceedings of the IEEE Symposium on Security and Privacy*, (Oakland, CA), May 2010.

[59] J. M. III and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[60] J. Mitola III, *Cognitive Radio*. Licentiate dissertation, The Royal Institute of Technology, Stockholm, Sweden, Sept. 1999.

[61] F. K. Jondral, "Software-defined radio-basics and evolution to cognitive radio," *EURASIP Journal on Wireless Communications and Networking*, vol. 3, pp. 275–283, 2005.

[62] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Elsevier Computer Networks*, vol. 50, pp. 2127–2159, 2006.

[63] S. Haykin, "Cognitive radio: brain empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, pp. 201–220, February 2005.

[64] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Elsevier Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[65] H. A and S. L, "A theory of polyspectra for nonstationary stochastic processes," *IEEE. Transactions on Signal Processing*, vol. 51, pp. 1243 – 1252, May 2003.

[66] WPI Wireless Innovation Lab, "Welcome to squirrelweb." [Online]: http://www.spectrum.wpi.edu/.

[67] M. J. Leferman, "Rapid prototyping interface for software defined radio experimentation," 2010.

[68] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Springer, 2010.

[69] Q. Zhao and A. Swami, *Cognitive Radio Communications and Networks: Principles and Practice*, ch. Spectrum Sensing and Identification. Elsevier, 2009.

[70] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*, ch. Statistical Decision Theory I. Prentice Hall, 1998.

[71] K. S. Shanmugan and A. M. Breipohl, *Random Signals: Detection, Estimation and Data Analysis*, ch. Signal Detection. Wiley, 1988.

[72] E. C. Like, "Non-cooperative modulation recognition via exploitation of cyclic statistics," 2007.

[73] E. Like, V. D. Chakravarthy, P. Ratazzi, and Z. Wu, "Signal classification in fading channels using cyclic spectral analysis," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, 2009.

[74] W. A. Gardner, W. A. Brown, and C.-K. Chen, "Spectral correlation of modulated signals - part ii: digital modulation," *IEEE Transactions on Communications*, vol. 35, no. 6, pp. 595 – 601, 1987.

[75] W. A. Gardner, *Cyclostationarity in Communications and Signal Processing*. Piscataway, NJ, USA: IEEE Press, 1993.

[76] K. B. J.-S. U. C. S. K. Kim, I.A. Akbar and J. Reed, "Cyclostationary approaches to signal detection and classification in cognitive radio," in *Proceedings of IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, (Dublin, Ireland), 2007.

[77] K. N. P.D. Sutton and L. Doyle, "Cyclostationary signatures in practical cognitive radio applications," *IEEE Journal on Selected Areas in Communications*, 2008.

[78] D. T. Kvale, *Artificial Neural Network-Based Approaches for Modeling the Radiated Emissions from Printed Circuit Board Structures and Shields*. Licentiate dissertation, The University of Toledo, Toledo, OH, 2010.

[79] K. Gurney, *An Introduction to Neural Networks*, ch. Neural Networks - An Overview. CRC Press, 1997.

[80] M. M. Gupta, L. Jin, and N. Homma, *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*, ch. Multilayered Feedforward Neural Networks (MFNNs) and Backpropagation Learning Algorithms. Wiley, 2003.

[81] K. Matasuoka, "Noise injection into inputs in back-propagation learning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 436 – 440, 1992.

[82] A. Fehske, J. Gaeddert, and J. Reed, "A new approach to signal classification using spectral correlation and neural networks," in *Proceedings of the IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, November 2005.

[83] K. Guo, P. Ishwar, and J. Konrad, "Action recognition in video by sparse representation on covariance manifolds of silhouette tunnels," in *Proceedings of the International conference on Recognizing patterns in signals, speech, images, and videos*, August 2010.

[84] K. Guo, P. Ishwar, and J. Konrad, "Action recognition using sparse representation on covariance manifolds of optical flow," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, August 2010.

[85] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proceedings of the European Conference on Computer Vision*, May 2006.

[86] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1713–1727, October 2008.

[87] V. Arsigny, P. Pennec, and X. Ayache, "Log-euclidean metrics for fast and simple calculus on diffusion tensors," *Magnetic resonance in medicine*, vol. 56, no. 2, pp. 411–421, 2006.

[88] R. Ramakrishnan and J. Gehrke, *Database Management Systems*. McGraw-Hill Science/Engineering/Math, 3rd ed., 2002.

[89] J. L. Harrington, *Relational Database Design Clearly Explained*. Morgan Kaufmann, 2nd ed., 2002.

[90] J. L. Harrington, *SQL Clearly Explained*. Morgan Kaufmann, 3rd ed., 2010.

[91] L. Rockoff, *The Language of SQL: How to Access Data in Relational Databases*. Course Technology PTR, 1st ed., 2010.

[92] T. Lahdenmaki and M. Leach, *Relational Database Index Design and the Optimizers*. Wiley-Interscience, 2005.

[93] Ettus Research LLC, "Usrp networked series." `https://www.ettus.com/product/category/USRP_Networked_Series`.

[94] G. J. Minden, J. B. Evans, L. Searl, D. DePardo, V. R. Petty, R. Rajbanshi, T. Newman, Q. Chen, F. Weidling, J. Guffey, D. Datla, B. Barker, M. Peck, B. Cordill, A. M. Wyglinski, and A. Agah, "KUAR: A flexible software-defined radio development platform," in *Proceedings of the IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, (Dublin, Ireland), Apr. 2007.

[95] G. J. Minden, J. B. Evans, L. Searl, D. DePardo, V. R. Petty, R. Rajbanshi, T. Newman, Q. Chen, F. Weidling, J. Guffey, D. Datla, B. Barker, M. Peck, B. Cordill, and A. Agah, "Cognitive radio for dynamic spectrum access - an agile radio for wireless innovation," *IEEE Communications Magazine*, May 2007.

[96] C. Chang, J. Wawrzynek, and R. W. Brodersen, "BEE2: A high-end reconfigurable computing system," *IEEE Design and Test of Computers Magazine*, March/April 2005.

[97] Q. Shi, D. Taubenheim, S. Kyperountas, P. Gorday, and N. Correal, "Link maintenance protocol for cognitive radio system with OFDM PHY," in *Proceedings of the IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, (Dublin, Ireland), Apr. 2007.

[98] R. Farrell, M. Sanchez, and G. Corley, "Software-defined radio demonstrators: An example and future trends," *International Journal of Digital Multimedia Broadcasting*, 2009.

[99] Rice University WARP, "Warp: Wireless open-access research platform." `http://warp.rice.edu/`.

[100] Lyrtech RD Incorporated, "Lyrtech rd processing systems: Software-defined radios." `http://lyrtechrd.com/en/products/families/+processing-systems+software-defined-radio`.

[101] Epiq Solutions, "Matchstiq: Handheld reconfigurable rf transceiver." `http://epiqsolutions.com/matchstiq/`.

[102] The MathWorks, "USRP hardware support from MATLAB and simulink." `http://www.mathworks.com/discovery/sdr/usrp.html`.

[103] GNU Radio, "Welcome to GNU radio!." `http://gnuradio.org/`.

[104] George Nychis, "The comprehensive GNU radio archive network." `http://www.cgran.org/`.

[105] P. D. Sutton, J. Lotze, H. Lahlou, S. A. Fahmy, K. Nolan, B. Ozgul, T. W. Rondeau, J. Noguera, and L. E. Doyle, "Iris: An architecture for cognitive radio networking testbeds," *IEEE Communications Magazine*, Sept. 2010.

[106] J. G. Proakis, ed., *Digital Communications*, ch. Characterization of Communication Signals and Systems. McGraw-Hill, 2001.

[107] A. J. Hayter, *Probability and Statistics for Engineers and Scientists*. Duxbury Press, 4th ed., 2012.

[108] J. L. Devore and K. N. Berk, *Modern Mathematical Statistics with Applications*. Springer, 2nd ed., 2012.

[109] C. Chen, H. Cheng, and Y.-D. Yao, "Cooperative spectrum sensing in cognitive radio networks in the presence of the primary user emulation attack," *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, pp. 2135 – 2141, 2011.

[110] J.S.Chitode, *Principles Of Communication*. Technical Publications, 2009.

[111] M. Leferman, D. Pu, and A. M. Wyglinski, *Cognitive Radio Communications and Networks: Principles and Practice*, ch. GNU Radio for Cognitive Radio Experimentation. Elsevier, 2009.

[112] Ettus Research LLC, "USRP Family Products and Daughter Boards." [Online]: http://www.ettus.com/products.

[113] A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 4th ed., 2002.

[114] H. Stark and J. Woods, *Probability and Random Processes with Applications to Signal Processing*. Prentice-Hall, 3rd ed., 2002.

[115] T. MathWorks, "Binomial cumulative distribution function." [Online]: http://www.mathworks.com/help/stats/binocdf.html.

[116] K. Pahlavan and P. Krishnamurthy, *Principles of Wireless Networks*. Prentice Hall, 2002.

[117] K. Pahlavan and A. Levesque, *Wireless Information Networks*. John Wiley and Sons, 2nd ed., 2005.

[118] K. Pahlavan and P. Krishnamurthy, *Networking Fundamentals - Personal, Local and Wide Area Communications*. John Wiley and Sons, 2009.

[119] T. MathWorks, "Free space path loss." [Online]: http://www.mathworks.com/help/comm/ref/freespacepathloss.html.

[120] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communication*, vol. 4, no. 1, pp. 40 – 62, 2011.