January 2014

A Comparative Study Of Spectrum Sensing Methods For Cognitive Radio Systems

Mohammad Shakib Apu

Follow this and additional works at: https://commons.und.edu/theses

A COMPARATIVE STUDY OF SPECTRUM SENSING METHODS FOR COGNITIVE
RADIO SYSTEMS

by

Mohammad Shakib Apu
Bachelor of Science in Electrical and Electronic Engineering, Bangladesh University of
Engineering and Technology, 2008

A Thesis
Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfillment of the requirements
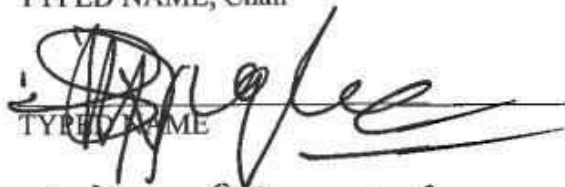
for the degree of

Master of Science

Grand Forks, North Dakota
December
2014

This thesis, submitted by Mohammad Shakib Apu in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

_____
TYPED NAME, Chair

_____
TYPED NAME

_____
TYPED NAME

This thesis meets the standards for appearance, conforms to the style and format requirements of the Graduate School of the University of North Dakota, and is hereby approved.

_____
Wayne Swisher,
Dean of the Graduate School

_____
December 11, 2014
Date

iii

PERMISSION

Title   A Comparative Study of Spectrum Sensing Methods for Cognitive
                 Radios Systems

Department   Electrical Engineering

Degree   Master of Science

In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, in his absence, by the chairperson of the department or the dean of the Graduate School. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my thesis.

Signature   Mohammad Shakib Apu

Date   12.11.2014

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

To my family

ABSTRACT

With the increase of portable devices utilization and ever-growing demand for greater data rates in wireless transmission, an increasing demand for spectrum channels was observed since last decade. Conventionally, licensed spectrum channels are assigned for comparatively long time spans to the license holders who may not over time continuously use these channels, which creates an under-utilized spectrum. The inefficient utilization of inadequate wireless spectrum resources has motivated researchers to look for advanced and innovative technologies that enable an efficient use of the spectrum resources in a smart and efficient manner.

The notion of Cognitive Radio technology was proposed to address the problem of spectrum inefficiency by using underutilized frequency bands in an opportunistic method. A cognitive radio system (CRS) is aware of its operational and geographical surroundings and is capable of dynamically and independently adjust its functioning. Thus, CRS functionality has to be addressed with smart sensing and intelligent decision making techniques. Therefore, spectrum sensing is one of the most essential CRS components. The few sensing techniques that have been proposed are complicated and come with the price of false detection under heavy noise and jamming scenarios. Other techniques that ensure better detection performance are very sophisticated and costly in terms of both processing and hardware.

The objective of the thesis is to study and understand the three of the most basic spectrum sensing techniques i.e. energy detection, correlation based sensing, and matched

filter sensing. Simulation platforms were developed for each of the three methods using GNU radio and python interpreted language. The simulated performances of the three methods have been analyzed through several test matrices and also were compared to observe and understand the corresponding strengths and weaknesses. These simulation results provide the understanding and base for the hardware implementation of spectrum sensing techniques and work towards a combined sensing approach with improved sensing performance with less complexity.

CHAPTER I

INTRODUCTION

1.1 Background and motivation

Wireless communications and the utilization of the radio frequency spectrum have witnessed a tremendous boom over the past decade. The multitude of different wireless devices and technologies, the dramatic increases in the number of wireless subscribers, the advent of new applications, and the continuing demand for higher data rates are all reasons for the radio frequency spectrum becoming more and more crowded. The technical innovations in wireless radio communications have made significant improvement over spectral efficiency and capacity. However, increase in the spectrum requirement is outpacing these advances and there is always extra spectrum required by the users.

Research executed by many administrations such as the Federal Communications Commission (FCC) suggests that the assumption of spectrum sufficiency is far from being the truth; there are unfilled spectrum bands as most of the allocated spectrum remains underutilized as shown in Figure 1. Several studies in various regions of the planet have sustained the fact that spectrum access is static which leads to some percentages of the spectrum to be overloaded and some other parts to be not used properly [1]. A research group at Kansas University discovered that in New York City the average U.S. spectrum

utilization is 5.2% while the maximum occupancy is 13.2% [2]. Hence, it can be said that the static spectrum access is not an effective way to manage the spectrum.
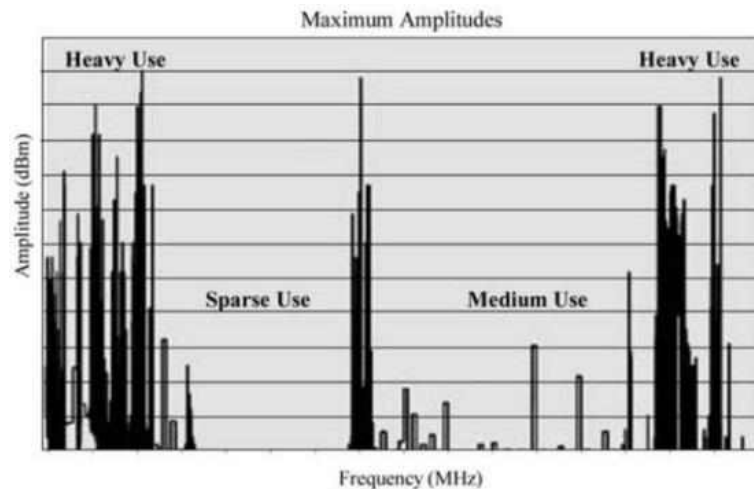
Figure 1. Spectrum utilization [3]

This development of the scarcity in radio spectrum calls for the systems and devices that are aware of their surrounding radio environment, hence, facilitating flexible, efficient, and reliable operation and utilization of the available spectral resources. Wireless communication systems must collect information about the radio spectrum in order to adapt their operation and behavior to provide a better match for the prevailing conditions. Thus, spectrum sensing is becoming increasingly important to modern and future wireless communication and radar systems for identifying underutilized spectrum and characterizing interference, and consequently, achieving reliable and efficient operation.

Cognitive radio (CR) is one of the technologies that has been proposed to address the spectrum scarcity problem. It allows the users to access the temporally unoccupied spectrum. Therefore, it is aware of its frequency environment by sensing the atmosphere and provides service to the secondary or unlicensed users by utilizing the discovered holes

of vacant licensed spectrum channels. The opportunistic spectrum access is used every time the Primary or licensed Users (PU) are not operating in their frequency bands. Precise spectrum awareness is the core concern for the cognitive radio system which is used by the secondary user (SU). To achieve that, the communications of licensed operators i.e. PUs have to be sensed without any failure and the core focus for adaptive communication in an opportunistic manner is the sensing of vacant frequency bands i.e. spectrum sensing. Hence, spectrum sensing is an important part of cognitive radio systems and efficient spectrum utilization.

1.2 Cognitive radio architecture

As defined by the FCC [4]: *"Cognitive Radio is a radio that can change its transmitter parameters based on interaction with the environment in which it operates"*. The final objective of a CR is to utilize the un-used spectrum. In essence, this means that CR introduces intelligence to conventional radio such that it searches for a hole in the spectrum that is defined as "a licensed frequency band not being used by an incumbent at that time within a selected area". As most of the channels are already assigned to PUs with legacy rights, the key objective is to share the licensed spectrum bands without producing harmful interference to PUs. Hence one of the main functions of a CR is to track the spectrum channels that are not used by the PUs [5]. Spectrum usage opportunity is then exploited by CR as long as no spectrum activity is detected. If this band is re-acquired by the PU, the secondary user must vacate the band and adjust its transmission parameters to shift to another unoccupied spectrum channel. A graphic illustration of opportunistic spectrum deployment approach is presented in Figure 2.
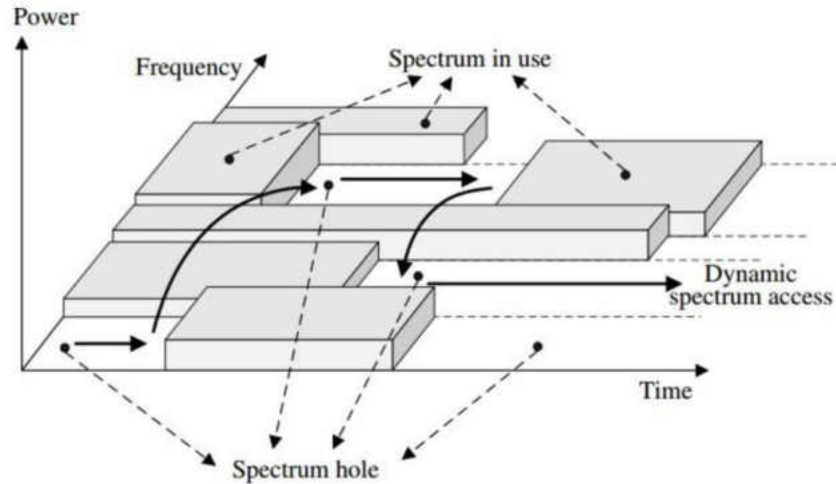
Figure 2. Illustration of spectrum holes and the concept of dynamic spectrum access [5]

From the definition, the two main characteristics of cognitive radio can be summarized as cognitive capability and re-configurability [3]. The first one enables the cognitive radio to interact with its environment in a real-time manner, and intelligently determine appropriate communication parameters based on quality of service (QoS) requirements. The CR system performs a set of processes, called a cognitive cycle shown in Figure 3. These processes are spectrum sensing, spectrum analysis, spectrum reasoning, and spectrum adaptation, which are described below:

- *Spectrum Sensing*: Either by cooperating or not, the cognitive radio nodes regularly monitor the RF environment. To improve the spectral usage efficiency, cognitive radio nodes should not only find spectrum holes by sensing some particular spectrum, but also monitor the whole spectral band.
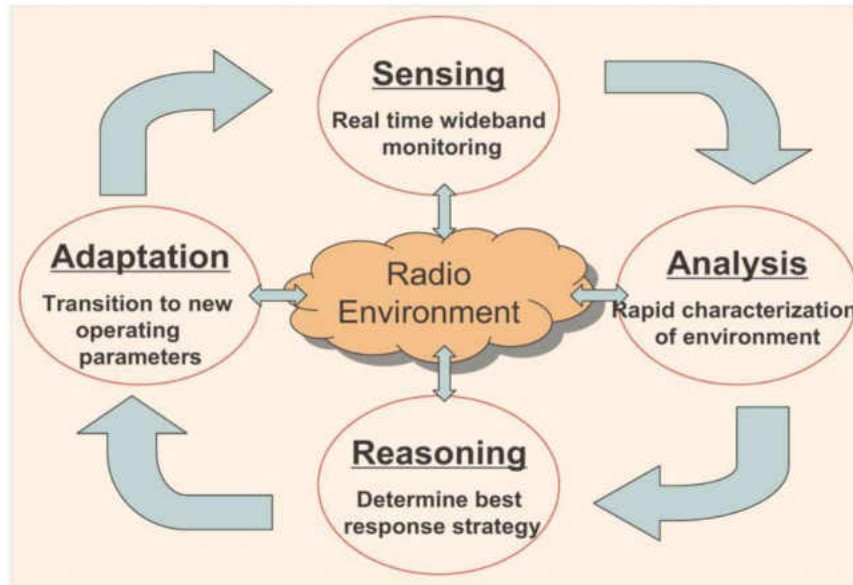
Figure 3. Cognitive Cycle [3]

- *Spectrum Analysis*: The characteristics of the spectral bands that are sensed through spectrum sensing are estimated. The estimation results, e.g., capacity, and reliability will be delivered to the spectrum decision process.

- *Spectrum Reasoning*: Based on the spectral analysis, CR takes decisions about what to do next according to the rules already set by design. The response strategy varies depending on the situation and the pool of resources available at that specific cognitive cycle.

- *Spectrum Adaptation*: According to the spectrum characteristics analysis and reasoning done above, an appropriate spectral band will be chosen for a particular cognitive radio node. Next the cognitive radio regulates new configuration parameters, e.g., data rate, transmission mode, and bandwidth of the transmission.

1.3   Spectrum sensing

One of the most important concepts in CR is the provision of opportunistic and dynamic spectrum access of the licensed frequency bands by the unlicensed secondary users. Hence, the main functionality of CR lies in efficient spectrum sensing so that whenever an opportunity of unused spectrum band is identified, CR may make use of it. Since cognitive radios are considered lower priority or Secondary Users (SU) of spectrum allocated to a primary user, an important condition is to avoid interference to potential PUs in their area. On the other hand, it is not required by the PU networks to change their structure for spectrum sharing with cognitive networks. Therefore, cognitive radios should be able to independently detect PU presence through spectrum sensing schemes.

Although spectrum sensing is traditionally considered as measuring the spectral occupancy by the primary user, in a more general term, it involves obtaining the spectrum usage characteristics across multiple dimensions such as time, space, frequency, and code [6]. The concepts and state-of-the-art in spectrum sensing will be discussed in chapter-II in detail.

1.4   Thesis organization

The objective of the thesis is to study and understand the three of the most basic spectrum sensing techniques i.e. energy detection, correlation based sensing, and matched filter sensing. Simulation platforms were developed for each of the three methods using GNU radio and Python interpreted language. The simulated performances of the three methods have been analyzed through several test matrices and also were compared to observe and understand the corresponding strengths and weaknesses. These simulation results provide the understanding and base for the hardware implementation of spectrum

sensing techniques and work towards a combined sensing approach with improved sensing performance with less complexity.

This thesis follows the following organization. Chapter-II presents the state-of-the-art in spectrum sensing for cognitive radio to understand the strengths and weaknesses of the available sensing techniques. Chapter-III explains the proposed simulation designs and methodologies of the intended spectrum sensing implementations. Chapter-IV describes the testing of the designs proposed in Chapter-III and discusses the results. Finally, in chapter-V a conclusion is drawn on the findings of the simulations and the future works are discussed.

CHAPTER II

STATE-OF-THE-ART IN SPECTRUM SENSING

Spectrum sensing is a technique used to characterize the occupancy state of the spectrum. It usually scans frequency bands in some predetermined order testing for occupancy [7]. It is used to recognize opportunistic spectrum by knowing which part of the spectrum is unoccupied and access that hole in the spectrum while avoiding interference with the primary users. Although, in one of the latest rulings, the FCC has eliminated the obligatory sensing necessity for unlicensed TV whitespaces but spectrum sensing is still mentioned as an important factor in allowing efficient secondary user access and would be considered for future unlicensed spectrum regulations [8]. The IEEE is also developing a standard, known as IEEE 802.22, for wireless regional area networks operating in unused television channels. Spectrum sensing is one of the cognitive features of this standard, which is used to identify vacant television channels. [9].

The fundamental goal of spectrum sensing is to decide between two hypotheses,

$$y[n] = w[n] \qquad\qquad H_0 \text{(white space)}$$

$$y[n] = h \times s[n] + w[n] \qquad\qquad H_1 \text{(occupied)} \qquad \text{(Equation 2.1)}$$

Where, $y[n]$ is the received signal by the cognitive radio, $s[n]$ is the primary user transmission, $w[n]$ is the noise of the Additive White Gaussian Noise (AWGN) channel, and h is the primary user's transmitter to the secondary user's receiver channel gain. H_0 is

a null hypothesis, meaning there is no primary user present in the band, while H_1 means the primary user's presence.

There can be two types of errors during spectrum sensing. One of these errors occurs when H_1 is detected by the system while H_0 is true. This phenomenon is known as a false alarm and in spectrum sensing the probability of false alarm of a detector is an important design parameter. False alarm leads to overlooking spectral opportunities and hinders the efficient operation of cognitive radio. The second error is made when H_0 is detected by the system while H_1 is true. This error is a result of a missed detection and hence makes the secondary user, interfere the primary transmissions and thus reducing the data rates for both the primary and the secondary system.

Generally, spectrum sensing in a cognitive radio system has to fulfil the constraints on both probability of false alarm and probability of missed detection. Since both probability of false alarm and probability of missed detection decreases as the number of samples increases, both constraints may be satisfied by selecting the number of samples to be a large number. For practical systems, working with a large number of samples is not always feasible because of the computational and hardware expense. For spectrum sensing algorithms, both threshold selection and performance analysis are preferred to be done analytically. However, in practice these are determined experimentally due to the large number of variables associated, such as the fading channel, synchronization errors, noise power uncertainty, etc.

Based on the literature spectrum sensing is divided into two types: cooperative sensing and non-cooperative sensing. The non-cooperative sensing is sub-divided into energy, feature, and matched filter based sensing. The cooperative sensing is sub-divided

into soft and hard combining. These sensing techniques and the sub-divisions will be discussed in the upcoming sections. This classification is shown in Figure 4:
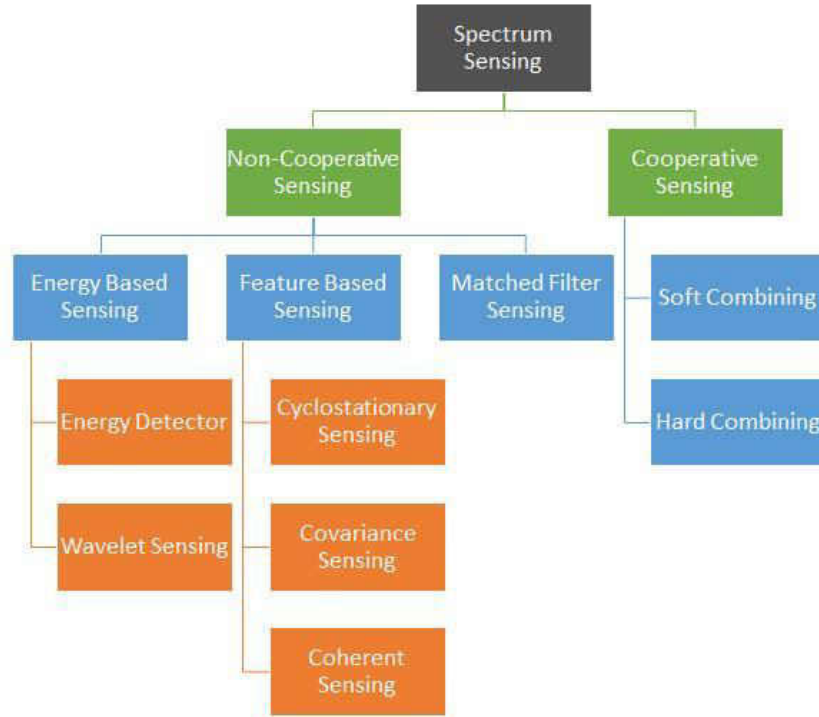
Figure 4. Classification of Spectrum Sensing Based on the approaches

2.1 Non-cooperative Sensing

In non-cooperative or standalone sensing, the data collection, signal processing and decision making is done locally in individual units. The ability of this technique to sense the spectrum and make decisions in standalone units has some inherent advantages. Non-cooperative sensing is often very simple and time efficient to implement. This sensing approach demands a smaller amount of computation thus the hardware implementation is less expensive. Moreover, the sensing time can be significantly smaller in this case because the sensing decision is made by the unit itself. Additionally, this method does not need any additional communication network between the nodes which would necessitate extra

wireless spectrum that would also require extra maintenance and cost. Non-cooperative sensing techniques are mainly divided into three categories: energy detection, feature detection, and matched filter sensing. Characteristics of these methods and their respective sub-categories have been discussed in the following categories.

2.1.1 Energy based sensing

2.1.1.1 Energy detection

Energy detection is the simplest of the methods in spectrum sensing since prior information about the signal and complex calculations are not required for detection. It computes the energy of the incoming signal for detection and thus does not depend on the modulation scheme of the primary user's signal. From the hypothesis in Equation 2.1, the detection statistics of the energy detector can be defined as the average (or total) energy of N observed samples:

$$T_{ED} = \frac{1}{N} \sum_{n=1}^{N} |y[n]|^2 \qquad \text{(Equation 2.2)}$$

Where, T_{ED} is the decision statistic, $y[n]$ is the sampled received signal, N is the total number of samples in a detection cycle. The assessment on whether the spectrum is being used by the primary user is drawn by comparing the detection statistic, T_{ED} with a pre-programmed threshold λ_{ED}. Although prior information about the received signal is not required, prior knowledge of noise power or a reliable estimate of it is necessary to obtain reliable performance [10]. The problem with fixed or static threshold is that it is

very prone to noise uncertainty. Sensing performance of the energy detection with static threshold degrades significantly if there is noise uncertainty present.

Consequently, if the signal power is below a certain level, the energy detector cannot distinguish the signal from a slightly larger noise power regardless of the detection time. This threshold is called the SNR wall [11]. For example, a real-valued signal of 1 dB noise uncertainty renders robust detection below SNR of -3.3 dB impossible [11]. The impact of the SNR wall phenomenon for energy detection is illustrated in Figure 5 [12]. The number of samples needed to meet the requirements for a 0.05 probability of false alarm and a 0.9 probability of detection for different levels of the noise uncertainty has been shown in the figure. Energy detection is also not efficient in discriminating between the secondary users, which are sharing the same channel as the primary user [13].
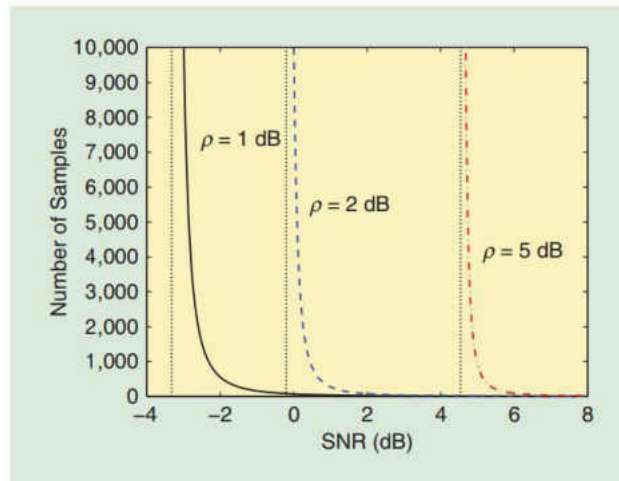
Figure 5. Number of samples essential to meet probability of false alarm of 0.05 and probability of detection of 0.9 using energy detection under noise uncertainty ρ [12].

Hence, to improve the efficiency of the energy detection technique, an improved version of energy detection method is proposed in [14]. This technique addresses the misdetection of primary transmission due to sudden drop in primary transmission power

12

by additionally keeping an updated list of latest sensing events. Sensing decision in every cycle is derived from an average decision statistic that calculated from that event list. This introduces a delay in showing actual detection during the transition time from H_1 to H_0 and vice versa.

The concept of double-threshold approach is suggested in [15] with the intention of finding and localizing narrowband signals. The utilization of double thresholds is capable of providing signal localization and separation. In low-SNR conditions, multiple antennas can be deployed for energy based spectrum detection to improve the detection performance [16]. Although the received signal gain is significant due to multiple antenna diversity in this approach, the antenna correlation degrades the detection performance. In [17], a technique is presented based on energy detection with wideband spectrum sensing. It mutually senses the signal strength levels within several frequency ranges. The aim is to improve the opportunistic throughput of the cognitive radios and decrease the interference to the primary users.

As the sensing performance is highly affected by the estimation error of noise power [18], a dynamic estimation style of noise power is suggested in [19]. In this method several signal classification algorithms are utilized for decoupling the signal and noise subspaces for noise floor estimation. Probability of false alarm P_{fa} can be defined as [20],

$$P_{fa} = Q(\frac{\lambda_{ED} - N\sigma_w^2}{\sqrt{2N\sigma_w^4}}) \qquad \text{(Equation 2.3)}$$

Where, Q ( ) is gaussian Q-function, λ_{ED} is the threshold, N is the sample number of samples and σ_w^2 is signal variance. The sensing threshold is adjusted in each iteration to achieve the constraints on probability of false alarm [21]. For threshold optimization, an

optimum and the adaptive threshold value is calculated in each cycle by employing the spectrum detection error function in [22]. Adaptive threshold control is also implemented in [23] with linear adoption on the energy threshold based on Signal to Interference plus Noise Ratio (SINR). The proposed system has been shown to attain a significant higher SU throughput than that with a fixed threshold approach while maintaining a decent constancy in false alarm and missed detection chances.

However, in a real world scenario, the system parameters that are assumed to be constant may vary over time which can induce deviation in the expected system output. This deviation can be reduced with adaptive threshold for energy detection in the presence of white Gaussian noise. The adaptive threshold is calculated with the noise power estimation for keeping the false alarm rate at a preferred point under different noise power levels [24]. Nevertheless, in this technique the concept of a dedicated noise estimation channel is introduced in which there will be no primary transmission present. This extra noise channel requirement might not be attainable given the fact that cognitive radios are deployed in the areas where spectrum is already scarce.

2.1.1.2 Wavelet sensing

This method works by a wavelet transformation of the power spectral density (PSD) of the received signal $y[n]$. The unused frequency bands can be discovered by finding the singularities of the PSD after the wavelet transformation. This method proposes benefits in terms of both application cost and flexible sensing for wideband channels [25]. Further development in the wavelet approach has been proposed in [26]. The PSD is first estimated for a wide bandwidth using compressive sampling and then the wavelet approach is applied

14

for edge detection to locate the different spectrum areas (black, gray, white spaces) in the estimated PSD.

Another wavelet approach for the wide-band spectrum estimation and spectrum hole detection has been proposed [27]. The idea in the proposed scheme is to directly sample the signal at the information rate of the signal. Conceptually this can be viewed as an analog-to-digital converter (ADC) operating at the Nyquist rate. After the PSD is reconstructed using a wavelet edge detector as shown in [26], the spectrum holes are detected using an energy detector in the frequency domain. A decision metric (DM) based approach has been proposed in [28] that is promised to substantially improve the sensing performance in terms of complexity and reliability, particularly at low SNR regions. A scheme for wideband spectrum sensing based on the analysis of singularities from their wavelet transform (WT) of multi scale information is also found in [29] that shows improvement over the current wavelet techniques at medium-to-low SNR regions.

2.1.2 Feature based sensing

Signals used in practical communication systems always contain some distinctive features such as transmit symbol rate, modulation, pulse shaping, etc. The specific features or properties that are inherent in modern modulation and coding techniques have aided in the design of efficient spectrum sensing algorithms. These features can be exploited for sensing and that enable us to achieve a detection performance that substantially surpasses the energy detector. Perhaps even more importantly, known signal features can be exploited to estimate unknown parameters such as the noise power. Therefore, making use

of known signal features effectively can bypass the problem of SNR walls discussed in the previous section.

Feature based detection refers to extracting features from the received signal and performing the detection based on the extracted features. For example, a typical feature used for detection is correlation based features. Additionally, cyclostationary-based detection has also received considerable attention. The advantage of the feature based detection over energy detection is that it typically shows the distinction between different signals or waveforms and is less susceptible to noise uncertainty. In the following sections, we give a short description of the different feature detection methods.

2.1.2.1 Cyclostationary based sensing

Cyclostationary feature sensing was first presented by [30]. In the majority of communication systems, the signals to be transmitted are modulated and combined with sine wave carriers, cyclic prefixes, hopping sequences, and pulse trains. But the additive noise is commonly wide-sense stationary (WSS) without any correlation. The periodicity of the majority signal transmission can be utilized to sense a random signal that retains a specific modulation category in the presence of noise. Such detection is called cyclostationary detection.

Unlike the energy detector that utilizes time-domain energy of the signal for test statistics T, the cyclostationary detection works by performing time-domain transformation to the frequency domain and then performing the hypothetical test. Cyclic Autocorrelation Function (CAF) is defined as,

$$R_y^\alpha(\tau) = E[\, y(t+\tau)y^*(t-\tau)e^{j2\pi\alpha t} \,] \hspace{2cm} \text{(Equation 2.4)}$$

Where, *y(t)* is the received signal, *E[.]* is the statistical expectation, α is the cyclic frequency and * denotes the complex conjugate. The spectral correlation function (SCF) is acquired by calculating the discrete Fourier transform of the cyclic auto-correlation function (CAF). Detection is then concluded by looking for the unique cyclic frequency by matching the peak in the SCF plane. Cyclostationary detection can be potentially employed to distinguish noise from the primary user's signal [31] and separates between different kinds of communications and primary systems [32].

Under noise uncertainty the energy detection is susceptible to large false alarm rates and is also unable to detect low power signals. In contrast, the cyclostationary sensing can distinguish noise from the primary user's transmission with superior sensing robustness in low SNR and noise uncertainty. An example of this type of technique is maximum cyclic autocorrelation selection based detection [33]. In this method the cyclic autocorrelation function is calculated for peak and non-peak values and then is compared in order to conclude if the primary user is present or not. This technique has the advantage of not requiring noise variance estimation and is robust in case of interference and noise uncertainty. An effective and dependable approach is proposed by merging signal classification with neural network and cyclic spectral analysis in [34]. One of the key attributes in this approach is to keep the computational requirement low. It is done by processing a large portion of the calculations offline using neural networks and thus the online calculation for signal detection is significantly reduced.

A promising sensing technique for multi-antenna cognitive radio that uses an adaptive cyclostationary beam forming is presented in [35]. The complexity level of the

resulting process is much smaller than that of the conventional cyclostationary detectors, but is higher than that of the energy detection. With all the advantages, the requirement for a multiple antenna system in this approach requires extra cost in terms of hardware. A non-parametric quickest detection scheme is suggested for sensing that utilizes energy detection in cooperation with cyclostationary features in [36]. Compared to traditional energy detection, this scheme reduces detection delays and thus achieves a greater percentage of channel usage. Although showing promising results, this detection scheme lacks the simplicity that is offered by some of the other combined sensing methods. For cyclostationary sensing a sub-optimal method is introduced in [37]. This method uses a lag set selection for the 2nd-order statistical testing that avoids the need for 4th-order cyclic cumulants which is hard to obtain. According to the authors, this technique offers superior detection performance in the low SNR region and the system is less complex than conventional cyclostationary sensing methods.

Cyclostationary detection is a commonly used feature detection technique. General feature detection also relates to other techniques that involve other features in the modulated primary transmission besides cyclostationarity. Such type of detection can utilize the different types of extracted features such as the level of strength of the primary transmission and its distribution over different frequency channels [38], [39], shape and bandwidth of a frequency channel [40], [41], power spectral density [42], center transmission frequency of the primary user [40], etc. A decent detection technique can also be achieved by fitting the features extracted at the receiving end with a priori data of primary transmission.

2.1.2.2 Covariance based sensing

Some of the features of a signal are also inherently present in the covariance matrix of the received transmission. Some transmitted signals demonstrate specific known features or structures to the covariance matrix. Covariance-based sensing utilizes these features to detect primary users. Since the statistical covariance matrices of the received signal and noise are generally not the same, the difference is utilized to distinguish the preferred signal element from background noise in [43] and [44]. For primary user detection, the eigenvalues found in the covariance matrix of the received signal can also be utilized [45]. According to random matrix theory, the ratio of the maximum to the minimum eigenvalue remains quantized and the sensing threshold can be extracted from those ratios [46].

Performance comparison between analytical results and simulations demonstrate the strength of Covariance Based Detection (CBD) [47]. As a modified version of the CBD, the Standard Condition Number (SCN) of the noise covariance matrix can also be utilized to evaluate the effect of noise correlation on eigenvalue-based sensing techniques [48]. Although these CBD techniques do not require a priori information of the primary signal or estimation of the noise power, they are presented to be stronger towards noise uncertainty.

2.1.2.3 Coherent sensing

Coherent sensing (also known as waveform-based sensing) can be utilized to make a decision about the presence of primary user transmission, if a particular feature can be

extracted from the received transmission [49]. In the procedure of coherent detection using a pilot pattern the decision statistic is expressed as follows [49]:

$$T = \frac{1}{N}\sum_{n=1}^{N} y[n]\hat{x}_p[n] \qquad \text{(Equation 2.5)}$$

Where, $x_p[n]$ is the known pilot-tone and $\hat{x}_p[n]$ is the normalized unit vector in the direction of the pilot-tone. Coherent sensing has the potential of being performed in the frequency domain also [42]. This sensing technique has been presented to show robustness in case of noise uncertainty and not restricted by the SNR wall effect as N is sufficiently large [49]. Furthermore, it outperforms energy detection in terms of sensing time [50], [51], as the required time of energy detection with a reliable result grows in a quadratic manner with the reduction in SNR, whereas that of coherent detection simply rises linearly [51]. On the other hand, particular information on the primary transmission waveform stands as a prerequisite for employing coherent sensing. A hybrid coherent/energy detection system for spectrum sensing using low–complexity and locally optimal decision metric has been proposed in [52]. The technique is a linear mixture of coherent and energy detection that merges the benefits of the individual metrics as it exploits both the pilot and the data symbols transmitted by the primary user.

2.1.3 Matched filter sensing

If secondary users possess certain information about the primary user's transmission, then the ideal detection method is matched filter detection [53]. A matched filter can compare the previously known features of a primary transmission with the

received signal to sense the existence of the primary user. Decision statistic, T_{MF} of matched filtering is given as [54]:

$$T_{MF} = \frac{1}{N}\sum_{n=1}^{N} y[n] * x_p[n] \underset{<}{\overset{>}{}} \lambda_{MF} \qquad \text{(Equation 2.6)}$$

Where, $y[n]$ is the received signal stream, $x_p[n]$ is the primary signal's known pilot signal, λ_{MF} is the matched filter threshold and N is the number of samples taken for calculation in a sensing cycle. The steps in a typical matched filter sensing cycle are illustrated in the block diagram shown in Figure 6. Since the matched filter requires a fewer number of received samples it has the advantage of a small sensing time and can demonstrate a definite sensing performance, such as a low chance of false alarm and missed detection [55]. However, the required number of signal samples also increases with the decrease in received signal SNR. The application of Automatic Modulation Classification (AMC) algorithm is introduced in [56]. This method improves the performance of sensing under low SNR situations. It works in three complex stages of feature key extraction, network training, and performance evaluation of the signal sensing.

Received Signal → Matched Filter → Threshold Comparison → Sensing Decision

Figure 6. Matched filter sensing steps.

Performance degradation of energy detection due to noise uncertainty and the SNR-Wall effect can be overcome by the statistical matched filtering method [57]. It is derived from the matched filter output by a ratio of maximum-to-mean absolute value. One more weakness of the energy detector is that it cannot distinguish the target signal from the

interfering signals, and due to that secondary users must be silenced in order to perform a successful sensing. Due to imperfect coordination during silent periods another secondary user may transmit and cause false alarms for the energy detector. If a matched filter is already present in a secondary user node, then matched filter assisted energy detection can significantly leverage the detection performance with less false alarm [58]. But this method only works if the node already has an unused matched filter, which is less likely to happen.

Although the matched filter is better than energy detection, its performance can significantly degrade in the presence of carrier frequency offset and phase noise. A trio of modified matched filter sensing methods termed as Block-Coherent Detector (BLCD), Second-Order Matched Filter-I (SOMF-I), and Second Order Matched Filter-II (SOMF-II) have been proposed as solutions to the problem [54]. However, from simulation data SOMF-II has been shown to be more robust between the three methods. This method is more complex and computationally expensive and appropriate if only carrier frequency offset, and phase noise are present.

A prerequisite for the matched filter sensing is precise information about the primary user's transmission, such as the working frequency, modulation scheme, bandwidth, etc. If incorrect data are provided for matched filtering, the sensing performance can show a significant amount of degradation [50], [59]. Furthermore, because it needs knowledge on all types of receiver signals and matching algorithms for wide-band spectrum sensing, it suffers from implementation complexity and high power consumption [60].

2.2 Cooperative sensing

In cooperative sensing the measurements of multiple sensing nodes are gathered in one master or central node and combine their measurements based on different approaches into one common decision. The concept of cooperative sensing is introduced as a solution to address the problems associated with the standalone or non-cooperative sensing like fading, shadowing, and noise uncertainty [61]. The block diagram in Figure 7 shows the typical steps involved in cooperative sensing. The primary user signal is received by the sensor nodes and their local decisions are then sent to the central node. Decision fusion is done in the central node and a central sensing decision is derived for efficient spectrum use. This method has been shown to increase the reliability of sensing, improve the detection likelihood, and decrease the false alarm rate to well defend a primary user.
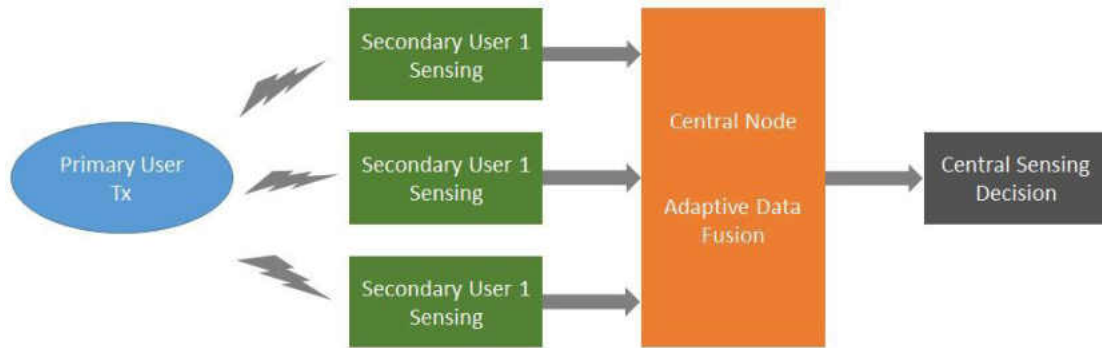
Figure 7. The cooperative spectrum sensing scheme.

A central controller, which is a secondary base-station, in a centralized cooperative spectrum sensing, assembles local sensing results from several secondary users. It makes a combined decision about the unfilled spectrum holes by means of some decision fusion instructions, and then notifies the secondary users which channels are right for entry. For distributed cooperative systems, secondary users interchange their local sensing outcomes

between themselves without needing to have a backbone infrastructure which decreases the expense.

Cooperative approach can be implemented based on any of the non-cooperative techniques that have been stated so far with the addition of central data analysis and decision making. Cooperative sensing utilizing energy detection [62], [63], wavelet sensing [64], cyclostationarity [65], [66], matched-filter [67] and covariance [68] has been suggested to address the problem of non-cooperative sensing of the corresponding types. The requirement for additional communication links between the central and the terminal nodes, structural complexity, delays in data analysis and decision making are some of the prominent challenges of cooperative sensing. A lot of research effort is concentrated in the field of cooperative sensing to address these issues. The notion of cooperative sensing is based on the fusion or combining of the sensing data or decisions of the sensors. Depending on the combining approaches, the cooperative sensing can be classified as soft combining and hard combining, which are discussed in the upcoming sections.

2.2.1 Soft combining

In this approach of cooperative sensing, all the user nodes transfer their individual soft decisions to a central fusion node that combines the soft values to one collective decision. This is comparable to the instance in which received data from all the individual nodes are available to the fusion center for access, and optimal sensing is performed based on all the data available. The energy detection has been used in [69] for the optimal cooperative sensing scheme to get data from the individual sensor nodes and combine the soft decisions by the weighted sum. However, if there exists a correlation between the

24

sensor nodes, the cooperation gain decreases severely and in case of one out of M no. of sensor nodes being untrustworthy, the sensitivity of every single sensor needs to be as good as that attained with M trusted user nodes [70].

2.2.2 Hard combining

In the soft combining approach, all the soft decisions by the standalone nodes are transmitted to the fusion center continually. However, in that method a large amount of data is required to be repeatedly transmitted to the fusion center, which is not always feasible. As a solution to this situation each of the sensor nodes makes its own sensing decision and transmits only the binary value or the hard decision to the central fusion node. In this kind of approach the central node combines the hard decisions from all the sensor nodes into a common decision by a voting rule [71]. Binary Phase Shift Keying (BPSK) signaling of the hard decisions to the fusion center has also been considered for some approaches. The corresponding optimal fusion rule is derived depending on the knowledge of the reporting channel SNRs and the local probability of the false alarm and detection [72].

2.3 Summary

In this chapter, we have described some of the state-of-the-art spectrum sensing methods and recent advances in cognitive radio. While doing so, we inevitably had to make choices and go through only some of the selected but popular parts of the current works on spectrum sensing. There are a few other matters that are worthy of mentioning which also have been attracting research interest recently. For example, working with the sensing methods that address more dynamic situations in terms of spectrum activity. For more

dynamic situations, the sensing time has to be really small even under very low SNR situations. Adaptive sensing and learning is also important while working with the dynamic spectrum environment. These challenges can be addressed with more research work on joint spectrum sensing and efficient resource utilization in cooperative sensing architecture.

CHAPTER III

SPECTRUM SENSING TECHNIQUES METHODOLOGIES

In spectrum sensing, the received signal *y[n]* is modeled as the sum of the transmitted PU signal *s[n]* multiplied by the channel gain *h* and the Additive White Gaussian Noise (AWGN) w*(n)*. The received sampled signal can be represented as:

$$y[n] = h \: x \: s[n] + w[n] \qquad \text{(Equation 3.1)}$$

Where, *y[n]* is the received signal by the cognitive radio, *s[n]* is the primary user transmission, w*[n]* is the noise of the Additive White Gaussian Noise (AWGN) channel, and *h* is the primary user's transmitter to the secondary user's receiver channel gain. These expressions are in terms of sample number n for digital domain calculation and *N* is the number of complex samples collected at the receiver for each sensing cycle. This model is illustrated in the Figure 8 below:
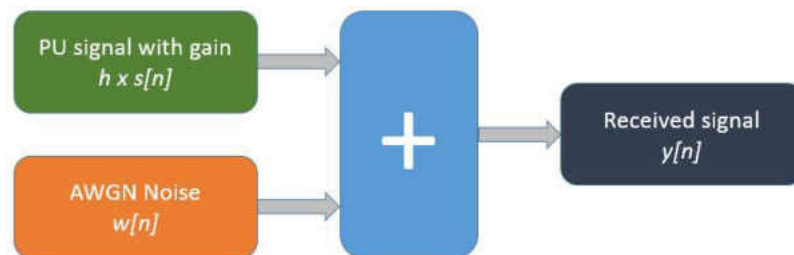
Figure 8. Signal Model Representation

From the above system model, spectrum sensing can be represented as the following binary hypothesis:

$$H_0: \; y[n] = w[n] \qquad\qquad \text{PU Signal absent}$$

$$H_1: \; y[n] = h \times s[n] + w[n] \qquad \text{PU Signal present} \qquad (\text{Equation 3.2})$$

Where, H_0 is a null hypothesis, which means there is no primary user present in the band, while H_1 means the primary user's presence. The simulation methodologies for energy detection, correlation based sensing, and matched filter sensing have been developed according to this model. An important aspect of developing the simulation setup is to choose the optimal tools and platform that have the strong base for research work and flexibility for future provision of hardware implementation.

3.1 Tools and platform of choice

In wireless radio communications research, a lot of components are implemented with hardware after the successful simulation of the projects. One of the important aspects of this thesis work is to keep the provision of hardware implementation in the future. Software Defined Radio (SDR), as a hardware platform, enables fast development of new wireless radio techniques, allowing the associated software to handle several protocols and frequencies, and executing real-time adaptive algorithms.

GNU Radio is a LINUX based open source project intended to ease the development of wireless radio communications projects with SDR. Due to its open source license, developers are able to share their processing cores, design custom modules and offer those for GNU Radio installation. Taking advantage of this feature, various signal

processing modules have been developed and tested already which enables complex waveforms to be created very fast. This allows easy reconfiguration of the application and tuning during real-time execution.

GNU Radio is designed based on the 'Python' language. It is easy and quick to learn and thus making it simple to construct connections between the signal processing modules. By linking processing modules with each other, a 'flow graph' can be created which enables the design of complex waveforms.
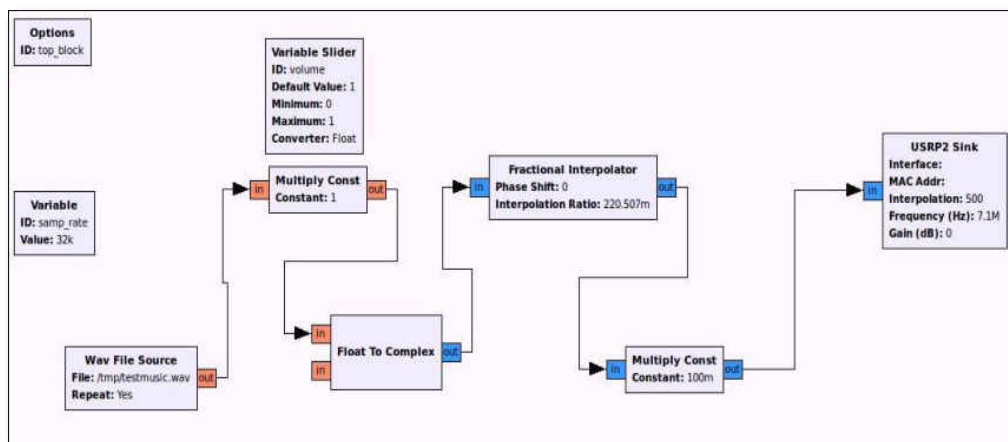
Figure 9. Simple AM transmitter with GNU Radio Companion (GRC).

GNU radio has an easy to use GUI interface known as GNU Radio Companion (GRC) which can be used to create very simple to complex signal flow graphs. In Figure 9, a simple AM transmitter that was created with few modules in the GNU radio environment which interfaces with an SDR box to transmit real AM signals. But if further modification is needed to perform a task that cannot be achieved by using the built in GRC blocks, custom blocks can be created that are written in python language and can later be

integrated in the GNU radio platform to be used as other built in blocks. This feature of GNU radio is very flexible and makes it an attractive choice for research projects.

Furthermore, GNU Radio environment has been designed to do real-time signal processing. Processing blocks have been written in C++, a compiled language, to achieve high signal throughput and performance needed for SDR applications. GNU Radio is also suitable for development of stable simulation projects. For all the simulation methodologies, first a standard flow graph was created in GRC and then that flow graph was further modified with python to achieve the required functions for the respective methods.
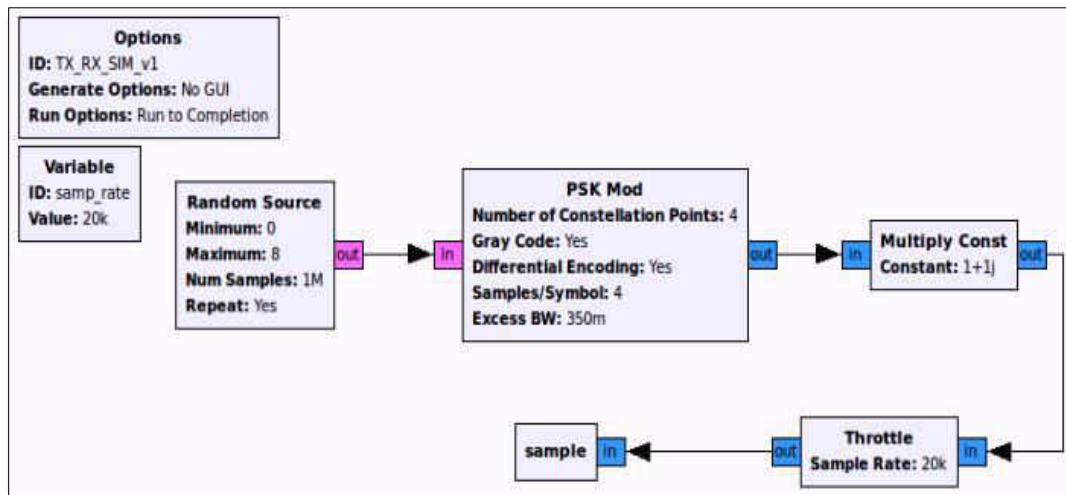
Figure 10. QPSK signal generation for simulation

Figure 10 shows the flow graph for QPSK signal generation using the standard design blocks present in GRC. It was created with the intention to be used in the simulations of the three sensing techniques. Figure 11 shows the constellation plot of the generated QPSK signal, *s[n]* from the GRC flow graph. There is no standard block available in GRC that contains the feature of spectrum sensing using one of the three techniques under

consideration. For that reason, a custom block named "sample" was created. It was prepared using the "out-of-the-tree" block creation cheat sheet from GNU radio.
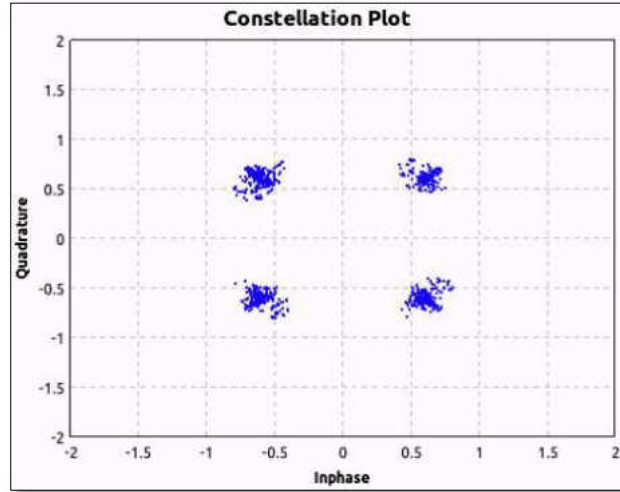
Figure 11. Constellation plot of the QPSK signal *s[n]*

The function of the custom block is to take the N number of QPSK modulated samples from previous block in the flow and that N number of QPSK modulated signal is considered as the PU signal *s[n]*. The value of the N which is also known as the sensing cycle length is user selectable before the execution of particular simulation.

From the flow graph shown in Figure 10 a script file was generated. Gaussian noise was simulated by editing that script file with the help of "NUMPY" and "SCIPY" modules available in python library. This noise emulates the channel noise associated with a Gaussian system which is considered as *w[n]*. The strength of the generated noise is user selectable to have the flexibility of achieving various SNR levels. By adding the noise, *w[n]* to the QPSK signal, *s[n]* the received signal, *y[n]* is generated and the sensing techniques can be applied to the received signal, *y[n]*. Before running this process the samples of length N in each cycle, the target SNR value and the total number of simulation

loop has to be provided or set by the user. The method of generating a QPSK signal, *s[n]*, noise *w[n]* and received signal, *y[n]* are shown in the flowchart in Figure 12.

The values of total number of sensing cycle or loop number, sample numbers in a sensing cycle N, and the value of SNR are specified at the beginning of the process. In the next step, a variable *count* initialized to '0' is used to count the loop number that is going to be executed. A decision is made in the following steps to check if the value of count is less than the value of loop or not. If count < loop is false which means all the sensing cycles have finished generating the received signal, *y[n]* then the process stops, but if it is true then the process goes to the next step. In the next two steps the QPSK signal, *s[n]* is generated and its power is adjusted according to the SNR value. After that, two more similar steps are completed to generate a noise signal, *w[n]* and adjust its power level to match the SNR value. Then the adjusted QPSK signal *s[n]* and noise signal *w[n]* are added to get the received signal *y[n]* in the next step. After that the value of count is incremented by one indicating that one cycle of the received signal, *y[n]* generation has been completed.
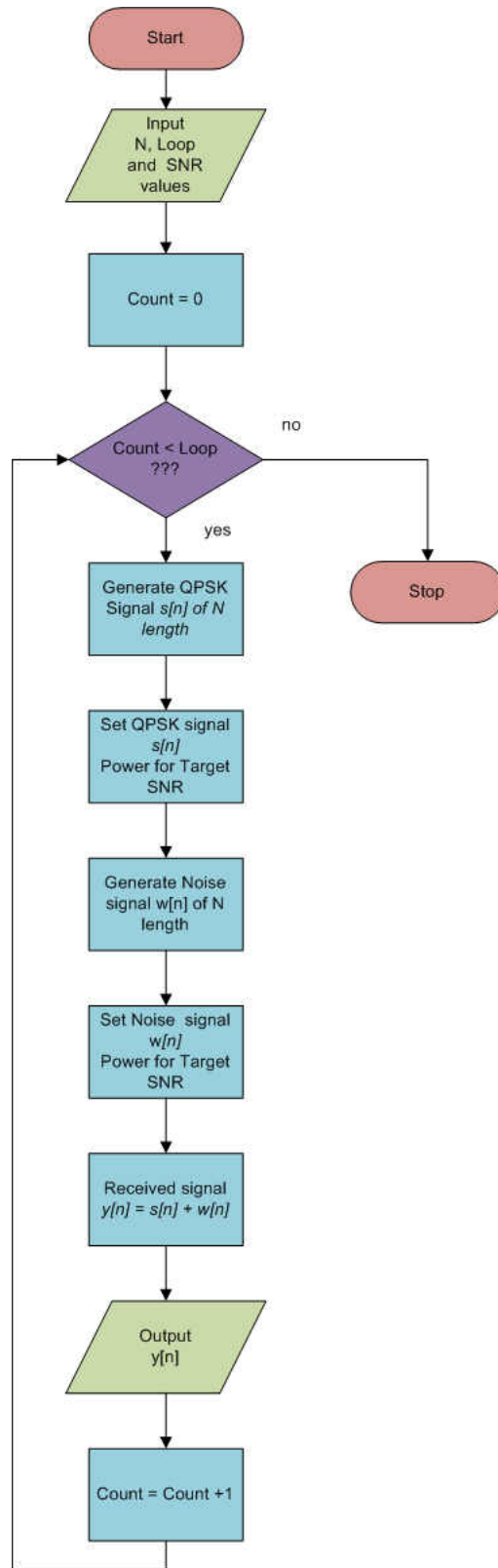
Figure 12. Flowchart of received signal *y[n]* generation

3.2 Energy detection

In its simplest form, the energy detection computes the energy of the received signal y[n] as a decision statistic T_{ED} and then compares T_{ED} with a predetermined fixed threshold λ_{ED}. The decision statistic can be expressed as:

$$T_{ED} = \frac{1}{N} \sum_{n=1}^{N} | y[n] |^2 \qquad \text{(Equation 3.3)}$$

Where, T_{ED} is the decision statistic, $y[n]$ is the sampled received signal, N is the total number of samples in a detection cycle. Decision statistic, T_{ED} can be calculated from the squared magnitude of the FFT averaged over N samples which is illustrated below in Figure 13.
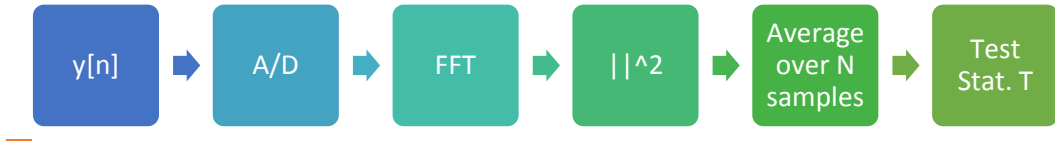
Figure 13. Block diagram for Decision Statistic T_{ED} calculation

Decision statistic, T_{ED} is computed in each sensing cycle of N samples and is compared to the threshold λ_{ED} to get the sensing result shown in equation 3.4:

$$T_{ED} < \lambda_{ED} \qquad \qquad \text{PU signal absent}$$

$$T_{ED} > \lambda_{ED} \qquad \qquad \text{PU signal present} \qquad \text{(Equation 3.4)}$$

For the calculation of the threshold λ_{ED} 'quite time approach' is often utilized. This refers to the time period when it is known that the primary user is not transmitting that is

only noise is present in the received signal *y[n]*. So on the approach, the decision statistic

calculated for quiet time is set as the threshold λ_{ED}.

But if N is large (N>250) central limit theorem can be used to approximate the test

statistic as Gaussian from which it can be derived as follows:

$T \sim$ Normal $(N\sigma_w^2 , 2N\sigma_w^4)$ Under *H₀*

$T \sim$ Normal $(N (\sigma_w^2 + \sigma_x^2) , 2N(\sigma_w^2 + \sigma_x^2)^2)$ Under *H₁*   (Equation 3.5)

Where, σ_w and σ_s is the standard deviation of noise and PU signal respectively. The

probability of detection P_d and false alarm P_{fa} can be evaluated as:

$$P_d = Q \left(\frac{\lambda - N (\sigma_w^2 + \sigma_s^2)}{\sqrt{2 N (\sigma_w^2 + \sigma_s^2)^2}} \right) \qquad \text{(Equation 3.6)}$$

$$P_{fa} = Q \left(\frac{\lambda_{ED} - N \sigma_w^2}{\sqrt{2 N \sigma_w^4}} \right) \qquad \text{(Equation 3.7)}$$

Where Q ( ) stands for the Gaussian Q-Function. An energy detector can meet any

desired P_d and P_{fa} simultaneously if the number of samples used in sensing is not limited.

The minimum number of samples required is a function of the signal to noise ratio (*SNR*)

and can be expressed as:

$$N = 2 \left[\left(Q^{-1}(P_{fa}) - Q^{-1}(P_d) \right) SNR^{-1} - Q^{-1}(P_d) \right]^2 \qquad \text{(Equation 3.8)}$$

So, for SNR << 1 regime a large number of samples are required to meet certain values of P_d and P_{fa} that corresponds to a reliable detection performance. Equation 3.7 can be further simplified as:

$$\lambda_{ED} = \sigma_w^2 \left(Q^{-1}(P_{fa})\sqrt{2N} + N \right) \qquad \text{(Equation 3.9)}$$

Thus for a Gaussian system, the threshold can be calculated from eq. 3.9 which requires that the values of noise standard deviation, σ_w and probability of false alarm, P_{fa} to be known. Probability of false alarm, P_{fa} is set as the target P_{fa} of the system while designing energy detector. The steps of the whole process are illustrated in the block diagram shown in Figure 14:
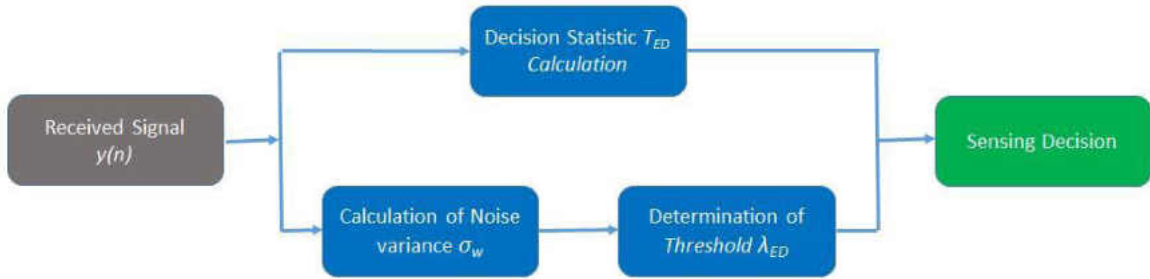
Figure 14. Block diagram of energy detection process

In the simulation, the noise is generated in the Python code which makes it possible to calculate the value of σ_w. For the purpose of performance evaluation, we can utilize the value of σ_w calculated from the generated noise to get threshold λ_{ED}. The detailed steps of simulating energy detection are shown in the flowchart in Figure 15. The output of the received signal generation block described in Figure 12 flowchart is further processed for energy detection steps. K-point FFT is done on the received signal, *y[n]* where K is user

selectable and has to be a number that can be expressed as 2^n. In the next step, FFT samples are squared and then averages over N samples to get the decision statistic, T_{ED}. The noise variance σ_w^2 is calculated from the noise stream $w[n]$ which is used to further compute threshold, λ_{ED}, from equation 3.9 in the next step. The decision statistic, T_{ED}, is compared with a threshold, λ_{ED}, to get the sensing decision. For $T_{ED} \geq \lambda_{ED}$ the output is H_1 (PU signal present) and for $T_{ED} < \lambda_{ED}$ the output is H_0 (PU signal absent). After getting the sensing decision the *count* value, a variable used for counting the total loop numbers, is incremented by one. The whole process is repeated until the total number of sensing cycles is completed.
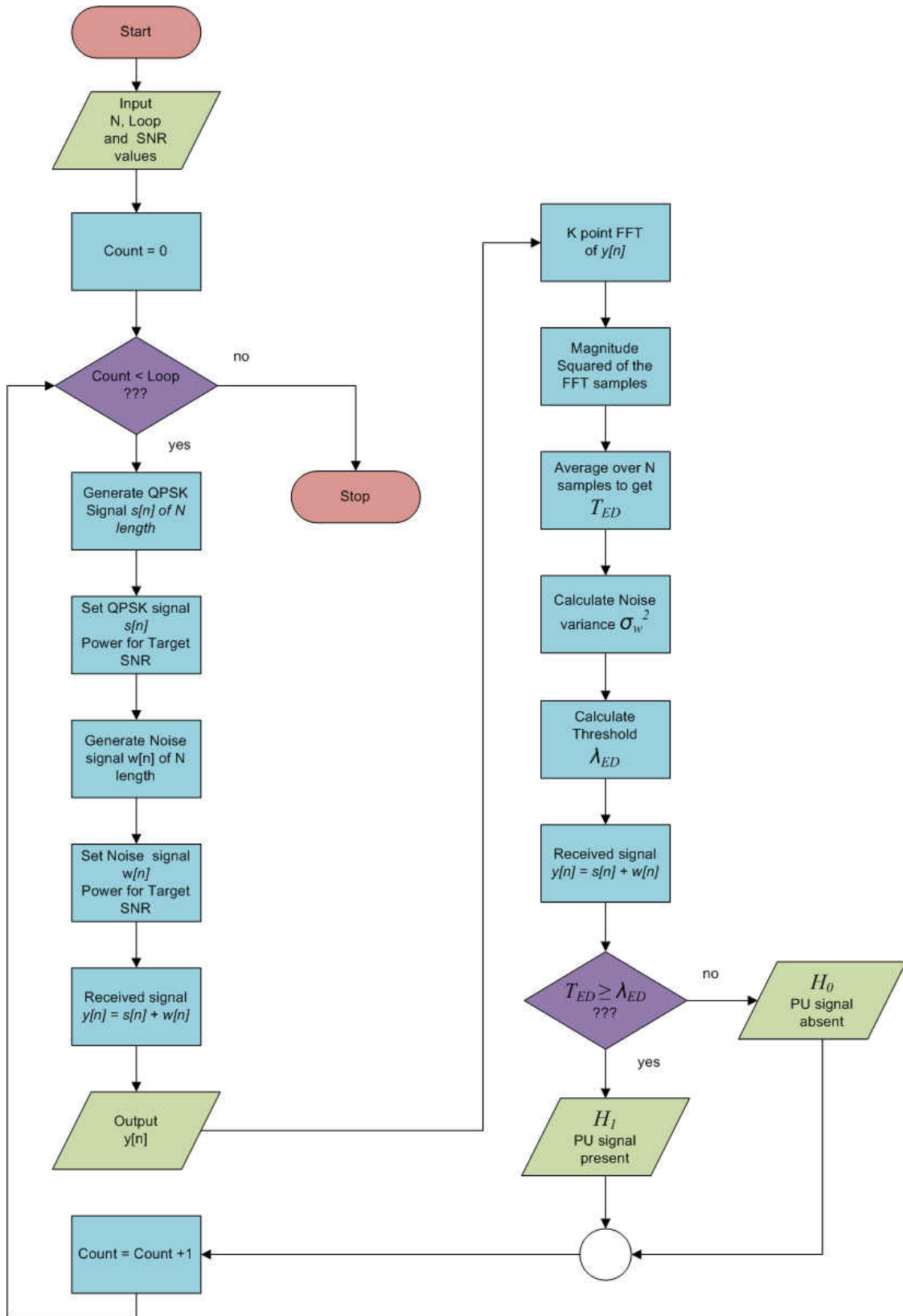
Figure 15. Flowchart of Energy Detection

3.3 Correlation based sensing

For the purpose of spectrum sensing, we can also exploit any features that exist in the deterministic transmitted signal that are not present in the noise. One such feature is the autocorrelation of the signal samples. In signal processing, given a signal *s(t)*, the continuous autocorrelation $R_f(\tau)$ at lag(τ) is defined as:

$$R_f(\tau) = \int_{-\infty}^{\infty} s(t)\, s^*(t-\tau)dt \qquad \text{(Equation 3.10)}$$

Where *s** represents the complex conjugate of *s* and τ is the time lag. An 'Autocorrelation Function' is one which is obtained by plotting the autocorrelation values for various time lags. If two successive values of an autocorrelation function of a signal are close to each other, then that means the signal is more correlated and if the values are significantly differ from each other then it is said that the signal is least correlated or uncorrelated.

In spectrum sensing, noise is a factor which greatly affects the quality of sensing. Signals affected by white Gaussian noise are, in general, difficult to interpret. By definition, Gaussian noise is uncorrelated and the autocorrelation function of a Gaussian noise stream results in a sharp spike at zero lag while the values of the rest of lags are close to zero as shown in Figure 16 (a). However, for a deterministic signal the autocorrelation function can present high values that depend on the transmit symbol rate, modulation, and pulse shaping. Due to the inherent nature of the signal, correlation is present in this transmitted signal and thus the values of zero lag and first lag of the autocorrelation function is very

close as shown in Figure 16 (b). This contrast in behavior of the noise and signal in the autocorrelation domain can be utilized for the purpose of spectrum sensing.
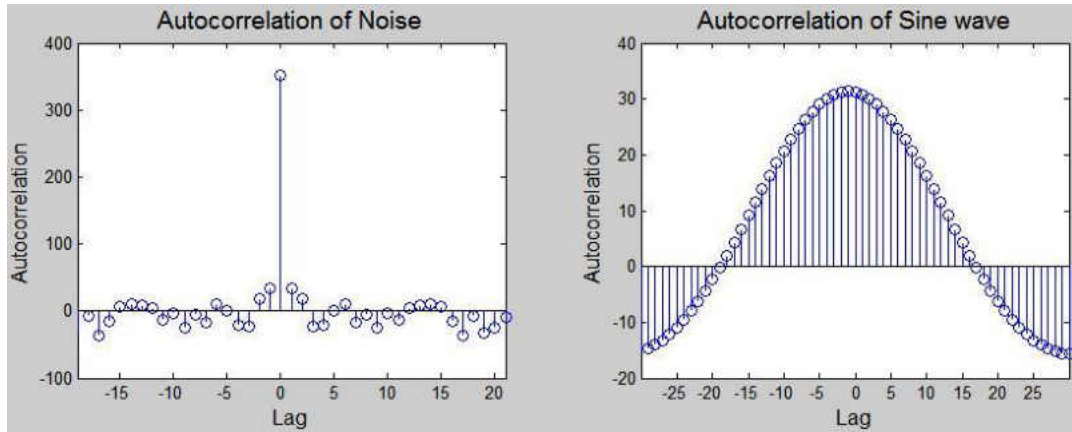
Figure 16. (a) Autocorrelation of noise. (b) Autocorrelation of Sine wave.

Figure 17 shows the main steps involved in the spectrum sensing based autocorrelation.
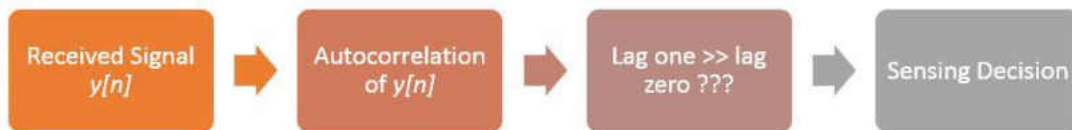
Figure 17. Correlation based Sensing Steps

In the first step, the autocorrelation is performed on the received signal *y[n];* lag zero and lag one are compared and the sensing decision is made using the following hypothesis:

$$Lag\ zero \gg lag\ one \qquad \text{PU Signal absent}$$

$$Lag\ zero \approx lag\ one \qquad \text{PU Signal present} \quad \text{(Equation 3.11)}$$

40

If the value of 'lag one' is much smaller than the value of 'lag zero', then the primary user transmission is absent; however, if the 'lag zero' and 'lag one' values are close, the primary user transmission is considered to be present. Figure 18 shows the flowchart of the autocorrelation based sensing technique.
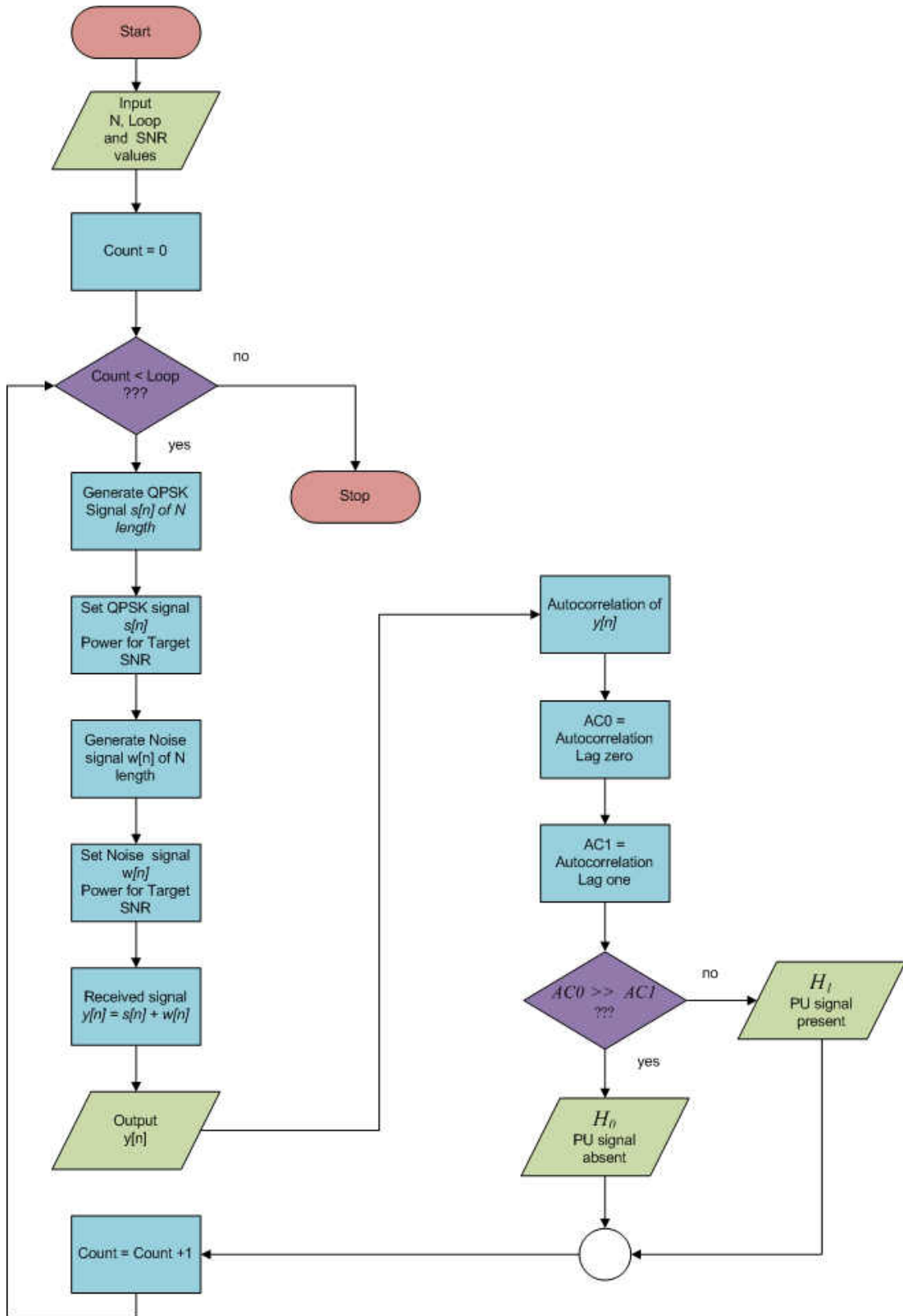
Figure 18. Flowchart of correlation based sensing

The output of the received signal generation block, described in Figure 12 flowchart, is also used for the autocorrelation based sensing. Autocorrelation is performed on the received signal, $y[n]$ and then the lag(0) and lag(1) values of the autocorrelation output are set as $AC0$ and $AC1$, respectively. If $AC0 >> AC1$ is 'true' then the output is H_0 (PU signal absent) and when $AC0 >> AC1$ is 'false' then the output is H_1 (PU signal present). After getting the sensing decision the *count* value, a variable used for counting the total number loops, is incremented by one. The whole process is repeated until the total number of sensing cycles is completed.

3.4 Matched filter sensing

Matched filter is considered as one of the optimum techniques for spectrum sensing if the knowledge of the primary user waveform is available. In this technique, filtering is done by matching the received signal with some pre-collected and saved pilot of the same PU signal stream. The main steps of this technique are shown in Figure 19:

Figure 19. Matched filter sensing steps.

The matched filter block in Figure 19 is further expanded in Figure 20 in which the received signal $y[n]$ is convolved with pre-collected pilot $x_p[n]$ and then averaged over N samples to get the matched filter decision statistic, T_{MF}, which is later compared to the matched filter threshold, λ_{MF}, to get the sensing decision. T_{MF} is given by

$$T_{MF} = \frac{1}{N}\sum_{n=1}^{N}\left(y[n] * x_p[n]\right) \qquad \text{(Equation 3.12)}$$

Where, $y[n]$ is the received signal stream, $x_p[n]$ is the primary signal's known pilot signal, and N is the number of samples taken for calculation in a sensing cycle. When the spectrum condition is in H_0, the decision statistic, T_{MF}, results from the convolution between Gaussian noise and the pre-collected pilot signal averaged over N samples. In H_1 situation, T_{MF} results from the convolution of PU signal mixed with the Gaussian noise and the pre-collected pilot signal averaged over N samples.
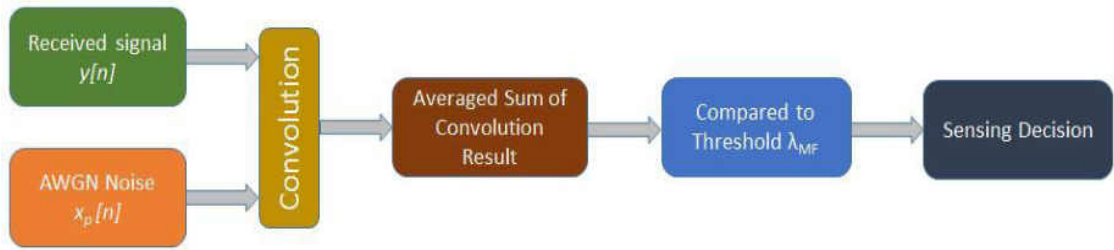
Figure 20. Matched filter sensing algorithm.

Matched filter threshold, λ_{MF}, is derived from the 'quiet time approach' which refers to the time period. This time period is the time when the primary user is not transmitting. Therefore, only the noise is present in the received signal $y[n]$. Thus for the quiet time period, the matched filter threshold, λ_{MF}, is equal to the matched filter decision statistic, T_{MF}. The block diagram of matched filter sensing is illustrated in Figure 20. If λ_{MF} is determined, the binary hypothesis is given as:

$$T_{MF} < \lambda_{MF} \qquad \text{Primary User absent}$$

$$T_{MF} > \lambda_{MF} \qquad \text{Primary User present} \qquad \text{(Equation 3.13)}$$

The detailed steps for matched filter sensing is illustrated in the flowchart shown in Figure 21. As for the other methods, the output of the received signal generation block, previously described in Figure 12, is also used for matched filter based sensing. After generating the received signal, $y[n]$, the pilot signal, $x_p[n]$, is read from database. Then, the convolution is done between $y[n]$ and $x_p[n]$. In the next step, the averaged sum of the convolution samples is calculated. To get the sensing decision, T_{MF} and λ_{MF} are compared. For $T_{MF} \geq \lambda_{MF}$ the output is H_1 (PU signal present) and for $T_{MF} < \lambda_{MF}$ the output is H_0 (PU signal absent). As for the previous techniques, after getting the sensing decision the *count* value, a variable used for counting the total number loops, is incremented by one. The whole process is also repeated until the total number of sensing cycles is completed.
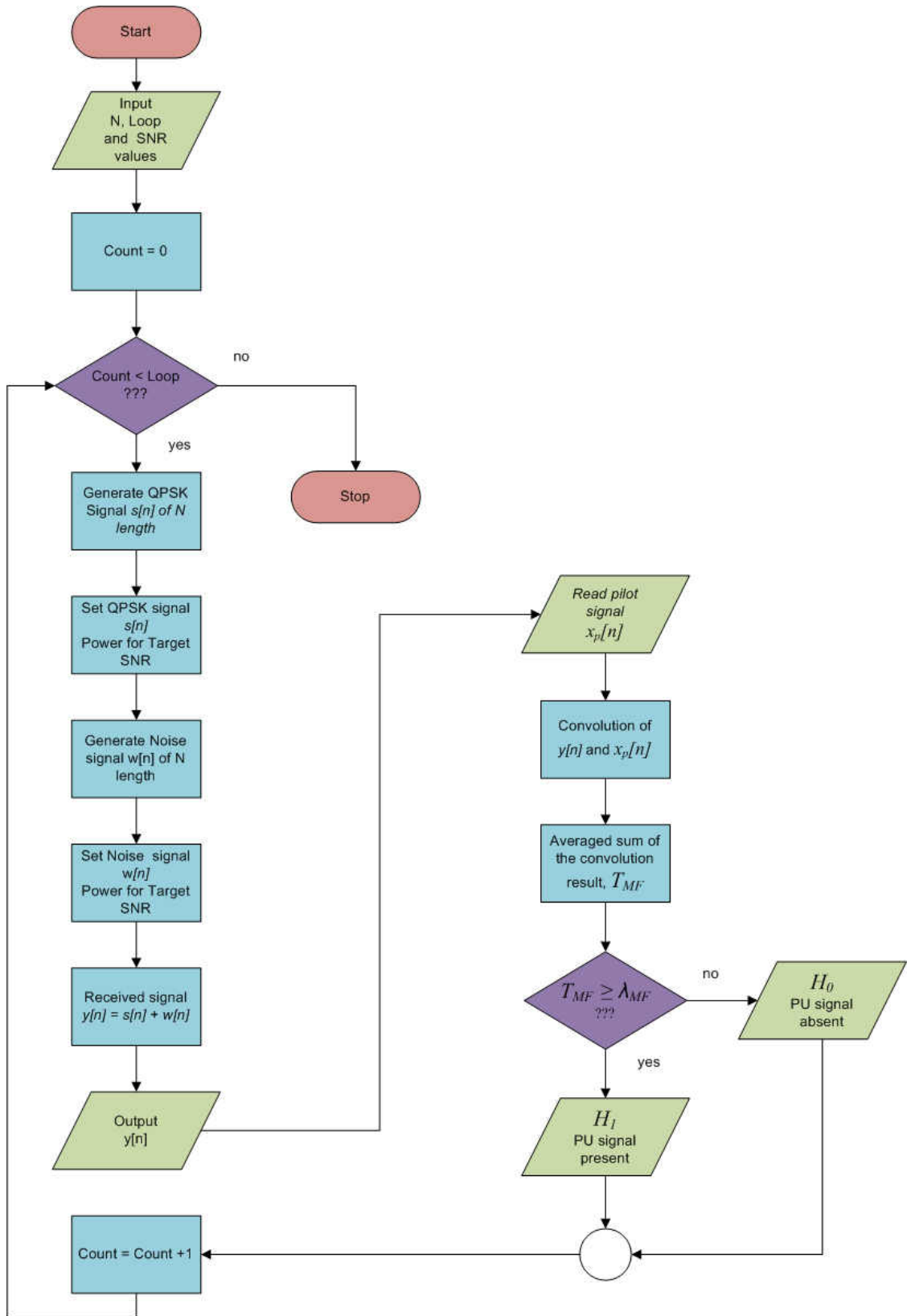
Figure 21. Flowchart of the matched filter sensing technique

CHAPTER IV

RESULTS AND DISCUSSION

4.1 Simulation parameters

In this work, the primary user signal is generated using the GRC flow graph shown in Figure 10 given in chapter III. The primary signal considered is a QPSK modulated signal and its power is set at -70 dBm. The AWGN noise is simulated as Gaussian noise and is generated with python code according to the target value of *SNR*. The *SNR* values range from -20 dB to +20 dB. Spectrum sensing is carried out by taking *N* samples of the primary signal as the received signal and then performing one of the three sensing techniques under investigation. The signal generation and main sensing steps are shown in the block diagram of Figure 22 below.
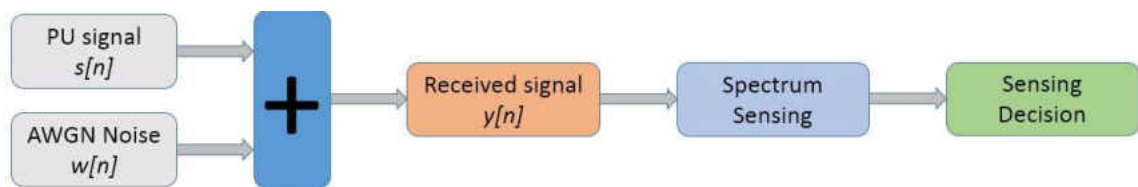
Figure 22. Main steps of the spectrum sensing techniques.

Two important parameters for evaluating the performance of spectrum sensing techniques are used: the probability of detection, P_D, and the probability of false alarm, P_{FA}. P_D and P_{FA} are calculated using the following equations:

$$P_D = \frac{N_D}{N_T} \qquad \text{(Equation 4.1)}$$

$$P_{FA} = \frac{N_{FA}}{N_T} \qquad \text{(Equation 4.2)}$$

Where, N_D is the number of total detections, N_{FA}, is the number of total detections, and N_T is the number of total experiments. To assess the performance of the sensing methods under investigation, we the evaluation parameters P_D, P_{FA}, λ, N, and SNR. Table 1 lists the evaluation parameters for each method under investigation. The threshold for energy detection, λ_{ED}, can be calculated from Equation 3.9 mentioned in chapter III. Thus, the energy detection technique has two more evaluations parameters compared to the two other methods.

Energy Detection	Correlation Sensing	Matched Filter Sensing	Comparison of the 3 Methods
P_D vs SNR (Variable λ_{ED})	P_D vs SNR	P_D vs SNR (Variable λ_{MF})	P_D vs SNR
P_D vs N (variable SNR)	P_D vs N (Variable SNR)	P_D vs N (variable SNR)	P_D vs N (Variable SNR)
P_{FA} vs SNR (Variable λ_{ED})	P_{FA} vs SNR	P_{FA} vs SNR (Variable λ_{MF})	P_{FA} vs SNR
P_{FA} vs λ_{ED} (Variable SNR)			
P_D vs P_{FA} (Variable SNR)			

Table 1. Performance evaluation matrices for the 3 methods

4.2 Energy detection

As discussed in chapter III, the expressions of P_D and P_{FA} for energy detection are given by:

$$P_D = Q \left(\frac{\lambda_{ED} - N \, (\sigma_w^2 + \sigma_s^2)}{\sqrt{2 \, N \, (\sigma_w^2 + \sigma_s^2)^2}} \right) \qquad \text{(Equation 4.3)}$$

$$P_{FA} = Q \left(\frac{\lambda_{ED} - N \, \sigma_w^2}{\sqrt{2 \, N \, \sigma_w^4}} \right) \qquad \text{(Equation 4.4)}$$

Where, σ_w and σ_s are the standard deviations of noise and PU signal respectively, N is the number of samples, λ_{ED} is the threshold, and Q ( ) stands for the Gaussian Q-Function. By changing the values of these variables we can test the performance of energy detection in terms of P_D and P_{FA}. For the performance evaluation of energy detection, the following metrics have been considered in the simulations:

1) Probability of Detection, P_D, vs SNR (Variable Threshold)

2) Probability of Detection, P_D, vs Sample Number, N (Variable SNR)

3) Probability of False Alarm, P_{FA}, vs SNR (Variable Threshold)

4) Probability of False Alarm, P_{FA}, vs Threshold (Variable SNR)

5) Probability of Detection, P_D, vs Probability of False Alarm, P_{FA} (Variable SNR)

The probability of detection, P_D, has been simulated by varying the *SNR* for different levels of threshold, λ_{ED}. The expression for λ_{ED} is given as:

$$\lambda_{ED} = \sigma_w^2 \left(Q^{-1}(P_{FA})\sqrt{2N} + N \right) \qquad \text{(Equation 4.5)}$$

In this equation, σ_w and N are known. The PU signal strength is set at -70 dBm. The SNR range is varied from -20 dB to +20 dB by varying the noise power. P_{FA} is set to be

0.2 to get the value of λ_{ED} using equation 4.5. After determining the value of λ_{ED} the threshold was varied by multiplying factors to observe the effect of threshold variation on P_D. For example, if the value of λ_{ED}, for $P_{FA} = 0.2$ and $N = 1000$, is 142, then for a threshold factor 1.3, the final threshold will be 142*1.3 = 426. The size FFT for determining the decision statistic, T_{ED}, was set at 128. For each set of parameters the simulation is run for 1000 cycles to get an average result of P_D. The values of the parameters for this simulation are shown in Table 1.

Parameters	Values
Sample Number, N	1000
PU signal strength	-70 dBm
SNR range	-20 dB to +20 dB
FFT size	128
P_{FA}	0.2
Threshold factors	1, 2, 3, 4
Sensing cycle no.	1000

Table 2. Values of simulation parameters for P_D vs SNR (Variable λ_{ED}).

The simulation results are shown in Figure 23. From these results we can see that P_D for energy detection increases with the increase in SNR. It also shows the effect of threshold on P_D which decreases with the increase of the threshold. This figure also shows that the method achieves a detection probability of 100% for SNR values of -10 dB for the threshold 1, +2 dB for the threshold 2, 5 dB for the threshold 2, 7 dB for the threshold 4. This means that to achieve a detection probability of 100%, the threshold has to be under 1 and the SNR higher than -10 dB.
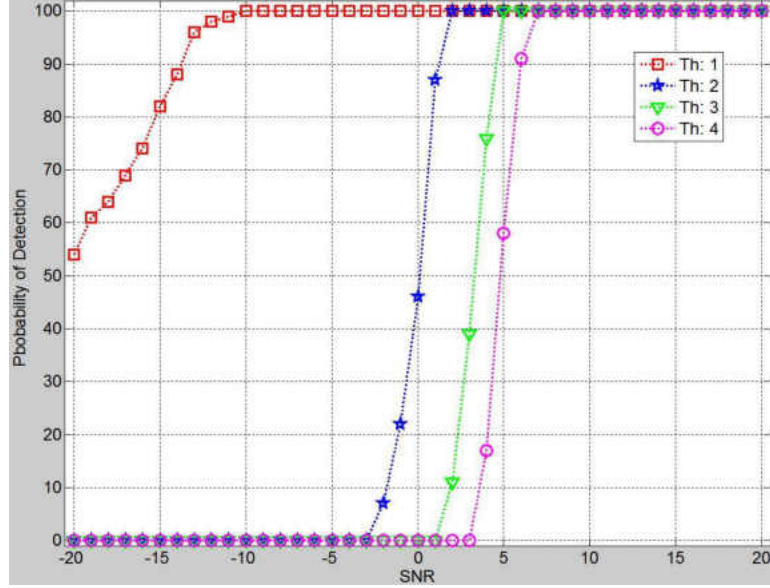
Figure 23. Energy detection simulation results for P_D vs SNR (Variable λ_{ED}).

We also investigated the impact of the number of samples on the probability of detection. This investigation was performed by varying the number of samples, *N,* for different levels of *SNR*. N is varied from 100 to 1000 in steps of 100. The PU signal strength is set at -70 dBm. The SNR values range from -20 dB to -4 dB by varying the noise power. P_{FA} is set at 0.2 to get the value of λ_{ED} using equation 4.5. The size FFT for determining the decision statistic is set at 128. For each set of parameters the simulation is run for 1000 cycles to get an average result of for P_D. The values of the parameters for this simulation are shown in the Table 3.

Parameters	Values
Sample Number, *N range*	100 to 1000
PU signal strength	-70 dBm
SNR range	-20 dB, -16 dB, -12 dB, -8 dB, -4 dB
FFT size	128
P_{FA}	0.2
Sensing cycle no.	1000

Table 3. Values of simulation parameters for P_D vs N (variable *SNR*).

The results of this investigation are shown in Figure 24. As expected, P_D for the energy detection method increases with the increase of the number of samples N. This figure also shows that P_D increases with the increase of SNR. Therefore, to achieve a high probability of detection at a specific SNR the number of samples has to be more than 800.
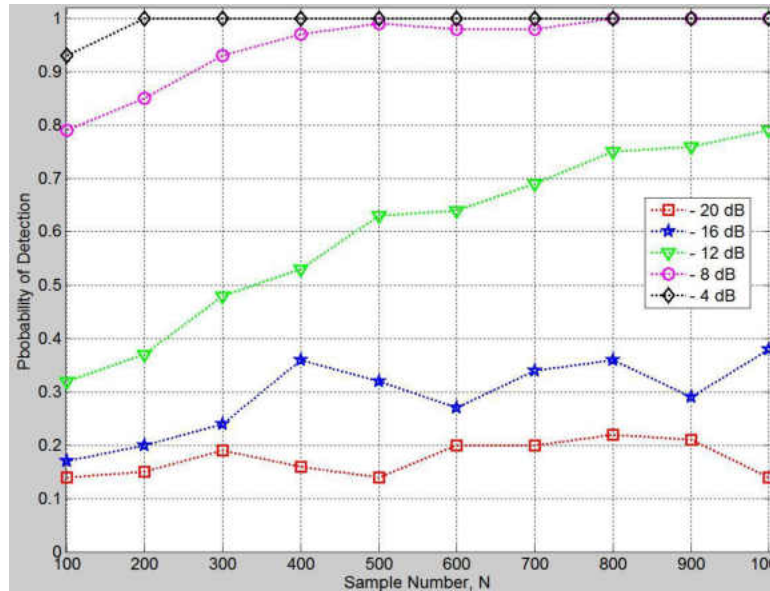
Figure 24. Energy detection simulation results for P_D vs N (Variable *SNR*).

The results of Figures 23 and 24 show very high detection probability at very low SNR values. However, at low SNR the noise is dominant. Because the energy detection based sensing measures the energy of the incoming signal mixed with the noise, it is likely that the probability of detection does not reflect the presence of the signal. To investigate this matter, we analyzed the probability of false alarm, P_{FA}. For this later probability we used the same simulation parameters as for the probability of detection. These parameters

are given on page 50. The values of the parameters for this simulation are shown in the following Table 4.

Parameters	Values
Sample Number, N	1000
PU signal strength	-70 dBm
SNR range	-20 dB to +20 dB
FFT size	128
P_{FA}	0.2
Threshold factors	1, 2, 3, 4
Sensing cycle no.	1000

Table 4. Values of simulation parameters for P_{FA} vs SNR (Variable λ_{ED}).

The simulation results of P_{FA} vs SNR for different threshold levels are shown in Figure 25. As expected, P_{FA} corresponding to energy detection decreases when SNR increases. It also shows the effect of the threshold on P_{FA}. The probability of false alarm decreases with the increase of the threshold λ_{ED}. To achieve a P_{FA} of approximately 0%, the threshold should be at least equal to 4 for SNR values higher than -5dB.
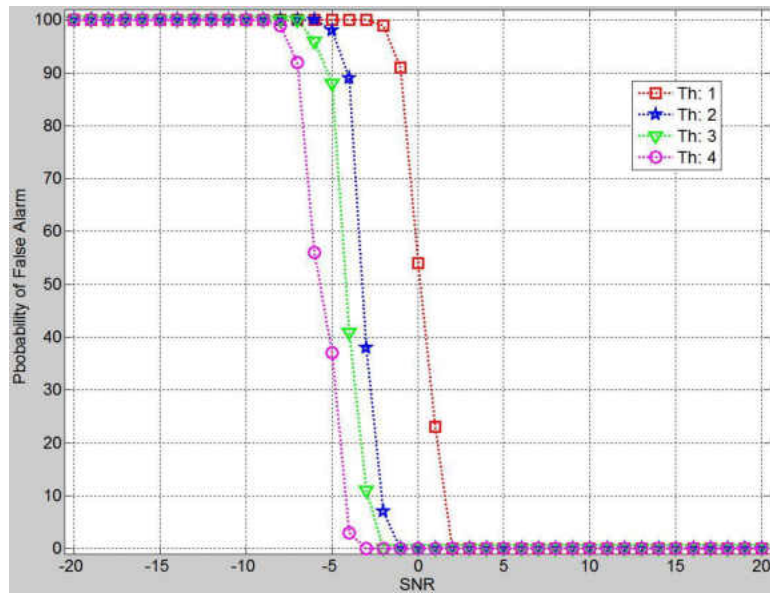
Figure 25. Energy detection simulation results for P_{FA} vs SNR (Variable λ_{ED}).

53

Figure 26 shows the results of P_{FA} vs λ_{ED} for different SNR values. As can be seen in this figure the probability of false alarm decreases with the increase in the threshold, λ_{ED}. One can also see the effect of *SNR* on P_{FA}; the number of false alarms decrease with the increase in the SNR value. Increasing the value of *SNR* shows to be also affecting the switching threshold point at which the P_{FA} curve starts to fall from 100% to 0%. For a higher value of threshold, the SNR value has to be high to achieve a low level of false detections.
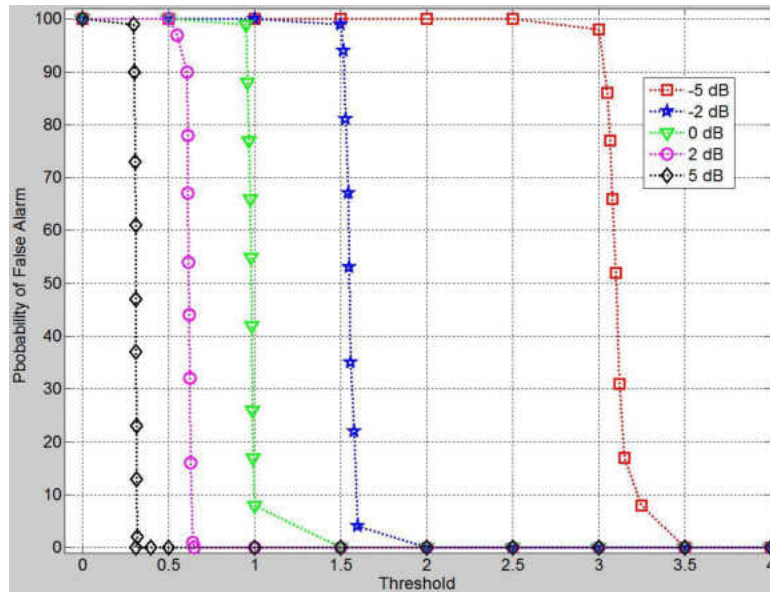
Figure 26. Energy detection simulation results for P_{FA} vs λ_{ED} (Variable *SNR*).

In summary, the results of the probability of detection and those of the probability of false alarm corresponding to the energy detection cannot be analyzed separately. Knowing the relationship between the two probabilities gives one a better understanding of the method's performance. We have investigated this relationship using the parameters given in Table 5.

Parameters	Values
Sample Number, N	1000
PU signal strength	-70 dBm
SNR levels	-20 dB, -16 dB, -12 dB, -8 dB, -4 dB
FFT size	128
P_{FA}	0 to 1
Sensing cycle no.	1000

Table 5. Values of simulation parameters for P_D vs P_{FA} (Variable SNR).

The results are shown in Figure 27. This figure shows that P_D for energy detection increases with the increase of the probability of false alarm, P_{FA}. It also shows how the SNR impacts on P_D. In other words, this method is only effective if the signal is stronger than the noise (high value of SNR). For a given situation where the SNR is around -15dB, the two probabilities are almost equal. However, more the SNR value increases and more PD increases while PF decreases.
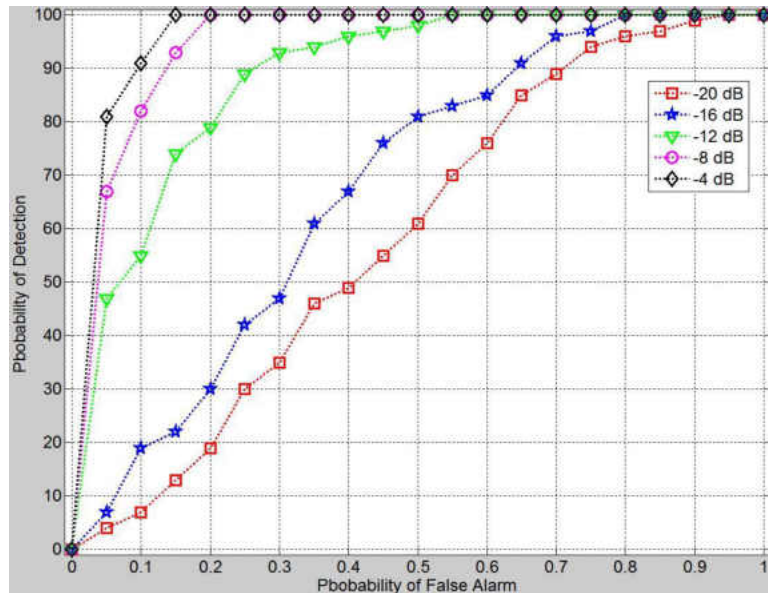
Figure 27. Energy detection simulation results for P_D vs P_{FA} (Variable SNR).

4.3 Correlation based sensing

The correlation based sensing method has been described on page 39 of chapter III. First, the autocorrelation is done on the received signal $y[n]$ and then the lag zero and lag one of the autocorrelation function are compared to get the sensing decision. Since there is no threshold calculation involved in this technique, the performance evaluation has been carried out based on the following parameters:

1) Probability of Detection, P_D vs SNR (Variable Threshold)

2) Probability of Detection, P_D vs Sample Number, N (Variable SNR)

3) Probability of False Alarm, P_{FA} vs SNR (Variable Threshold)

The probability of detection, P_D, is calculated for different values of *SNR*. The number of samples, N, is set at 1000. The PU signal strength is set at -70 dBm. The *SNR* ranges from -20 dB to +20 dB by varying the noise power. Received signal is generated by adding PU signal and noise signal. Autocorrelation is performed on the received signal, $y[n]$, to get the autocorrelation function. From the autocorrelation function, the values of lag zero (AC_0) and lag one (AC_1) are compared. If $AC_1 \leq (2\%$ of $AC_0)$ then PU signal is not present and vice versa. Here, the correlation sensing threshold, λ_{CS} is 1% of AC_0. The threshold factors are selected as 1, 2, 3, and 4 which represents 2%, 4%, 6%, and 8% of AC_0. For each set of parameters the simulation is run for 1000 cycles to get an average result of P_D. The values of the parameters for this simulation are shown in Table 6 below.

Parameters	Values
Sample Number, N	1000
PU signal strength	-70 dBm
SNR range	-20 dB to +20 dB
Sensing cycle no.	1000

Table 6. Values of simulation parameters for P_D vs SNR.

The simulation results are shown in Figure 28. From these results we can see that P_D for the autocorrelation base sensing technique increases in the increase of SNR. For a threshold factor 4, the transition of P_D from a low level to a high level happens between the SNR range of -7 dB and -4 dB. P_D reaches 100% at approximately -4 dB and stays at that level for higher SNR values. These results obtained from the correlation method is within the acceptable range of spectrum sensing found in the literature [14, 28, 48].
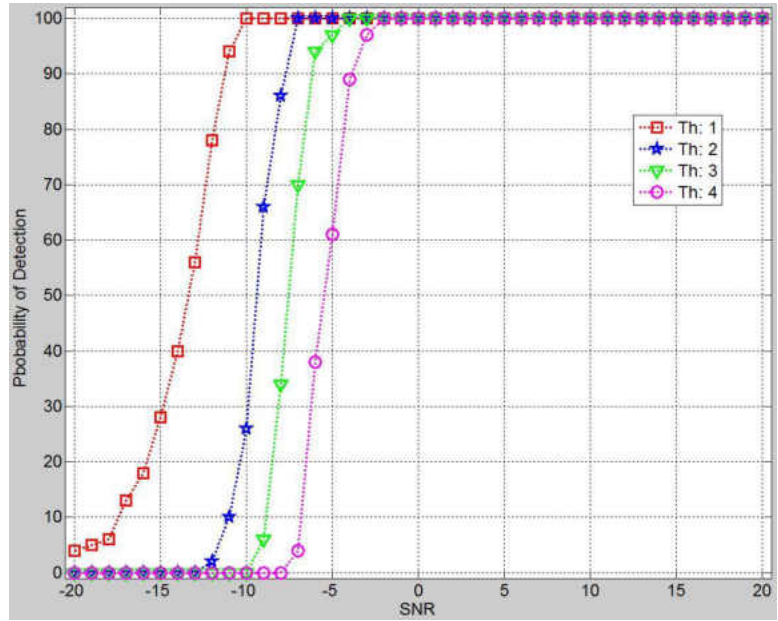
Figure 28. Correlation sensing simulation results for P_D vs SNR (Variable Threshold).

As for the correlation based sensing method, we also investigated the impact of the number of samples on the probability of detection, P_D, when using the autocorrelation function. In this simulation, the value of N varies from 100 to 1000. The PU signal strength is set at -70 dBm. The SNR levels have been set at -20 dB, -16 dB, -12 dB, -8 dB, and -4 dB by varying the noise power. Received signal is generated by adding PU signal and noise signal. Autocorrelation is performed on the received signal to the autocorrelation function. From the autocorrelation function, the values of lag zero (AC_0) and lag one (AC_1) are compared. For a threshold factor 4, if $AC_1 \leq (8\%$ of $AC_0)$ then PU signal is not present and vice versa. For each set of parameters the simulation was for 1000 cycles to get an average result of P_D. The values of the parameters for this simulation are shown in Table 7.

Parameters	Values
Sample Number, *N range*	100 to 1000
PU signal strength	-70 dBm
SNR range	-20 dB, -16 dB, -12 dB, -8 dB, -4 dB
Sensing cycle no.	1000

Table 7. Values of simulation parameters for P_D vs N (*Variable SNR*).

The results of this simulation are shown in Figure 29. Unlike for the energy detection method, N has almost no effect on P_D when using the correlation based sensing method. Thus, it can be concluded that the performance of the correlation based sensing does not improve with the increase of the number of samples, *N*. Thus, using a small number of sample can decrease the processing time. This result of P_D independent of N, however, contradicts the results obtained by other researchers in the same laboratory.
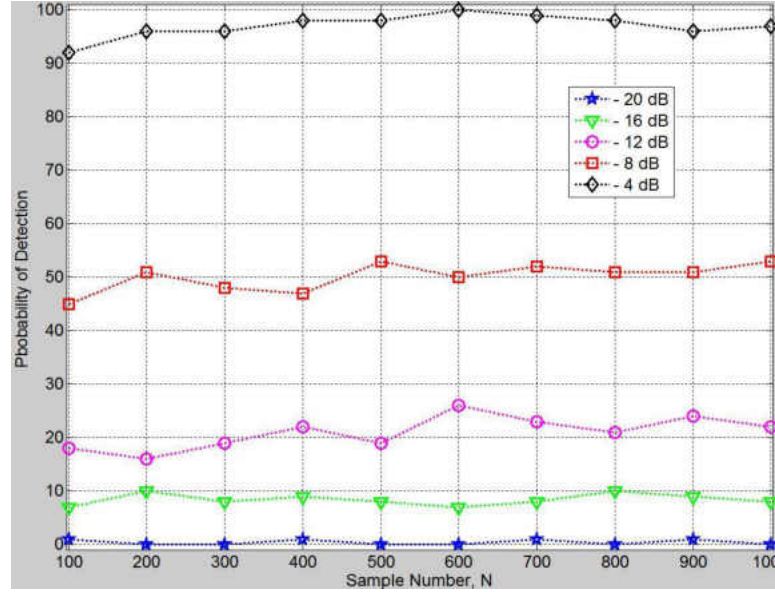
Figure 29. Correlation based sensing simulation results for P_D vs N (Variable *SNR*).

The probability of false alarm, P_{FA} has been simulated by varying the *SNR*. In this simulation, the value of N is also set at 1000. The PU signal strength is set at -70 dBm. The SNR range was varied from -20 dB to +20 dB by varying the noise power. The received signal, *y[n]* is generated by adding PU signal and noise signal. Autocorrelation is performed on the received signal to the autocorrelation function. From the autocorrelation function, the values of lag zero (AC_0) and lag one (AC_1) are compared. . If $AC_1 \leq$ (4% of AC_0) then PU signal is not present and vice versa. Here, the correlation sensing threshold, λ_{CS} is 4% of AC_0. The threshold factors are selected as 1, 2, 3, and 4 which represents 4%, 5%, 6%, and 7% of AC_0. For each set of parameters the simulation has been run for 1000 cycles to get an average result of P_D. The values of the parameters for this simulation are shown in the following Table 8.

Parameters	Values
Sample Number, N	1000
PU signal strength	-70 dBm
SNR range	-20 dB to +20 dB
Sensing cycle no.	1000

Table 8. Values of simulation parameters for P_{FA} vs SNR.

The simulation results of the probability of false alarm versus SNR are shown in Figure 30. From the results we can see that P_{FA} is very small for the entire SNR range for threshold factor 1 and gets to better levels for threshold factor 2, 3, and 4. This means that correlation based sensing method is very effective in distinguishing between the noise and signals. As was explained in Figure 16, page 40, the Gaussian noise does not have any correlation with itself. Thus, if there is no primary transmission present in the received signal, the output of the autocorrelation function will be uncorrelated and vice versa. This characteristic of the correlation sensing is the cause behind the results found in Figure 30.
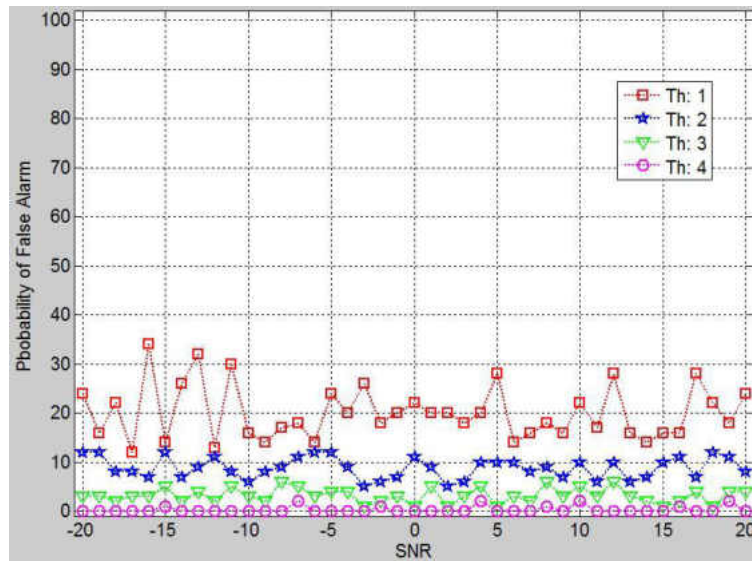
Figure 30. Correlation sensing simulation results for P_{FA} vs SNR (Variable Threshold).

4.4 Matched filter sensing

In this technique, filtering is done by matching the received signal with some pre-collected and saved pilot of the same PU signal stream. The received signal is convolved with pre-collected pilot and then averaged over N samples to get the decision statistic, T_{MF}, which is later compared to λ_{MF} for getting the sensing decision. To evaluate the performance of matched filter, we used the following metrics:

1) Probability of Detection, P_D vs SNR (Variable Threshold)

2) Probability of Detection, P_D vs Sample Number, N (Variable SNR)

3) Probability of False Alarm, P_{FA} vs SNR (Variable Threshold)

The probability of detection, P_D, for matched filter sensing has been simulated by varying the *SNR* for different levels of threshold, λ_{MF}. In this simulation, the values of σ_w and N are known. The PU signal strength is set at -70 dBm. The *SNR* ranges from -20 dB to +20 dB by varying the noise power. Matched filter threshold, λ_{MF}, is collected from the 'quite time approach' as described in chapter III. After determining the value of λ_{MF}, the threshold is calculated by multiplying factors to observe the effect of threshold variation on P_D. The received signal and the pre-saved pilot signal are convolved and the convolution result is averaged over N samples to get matched filter decision statistic, T_{MF}. For each set of parameters the simulation is run for 1000 cycles to get an average result of P_D. The values of the parameters for this simulation are shown in the following Table 9.

Parameters	Values
Sample Number, N	1000
PU signal strength	-70 dBm
SNR range	-20 dB to +20 dB
Threshold factors	1, 2, 3, 4
Sensing cycle no.	1000

Table 9. Values of simulation parameters for P_D vs SNR (Variable λ_{MF}).

The results are shown in Figure 31. These results show that P_D for matched filter sensing increases with the increase in SNR. The figure also shows the P_D for different levels of threshold values. The transition from 0% to 100% of P_D happens at lower SNR range for matched filter sensing than for the other 2 methods.
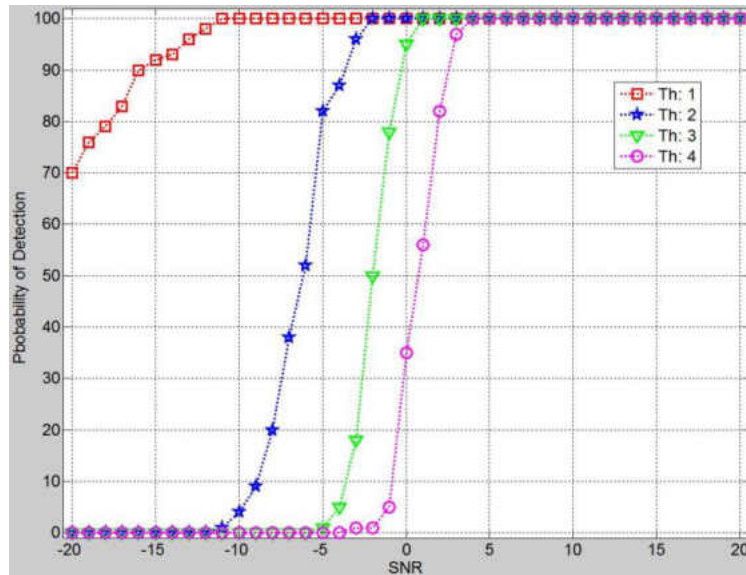
Figure 31. Matched filter sensing simulation results for P_D vs SNR (Variable λ_{MF}).

The effect of increasing the number of samples, N, on P_D for matched filter sensing has also been investigated in this simulation. P_D is evaluated by varying the number of samples, N for different levels of SNR. The value of N varies from 100 to 1000 in steps of

100. The PU signal strength is set at -70 dBm. The SNR values are selected are -20 dB to -4 dB by varying the noise power. Matched filter threshold, λ_{MF}, and decision statistic, T_{MF}, are calculated using the steps described in the previous section. For each set of parameters the simulation has been run for 1000 cycles to get an average result of P_D. The values of the parameters for this simulation are shown in Table 10.

Parameters	Values
Sample Number, N range	100 to 1000
PU signal strength	-70 dBm
SNR range	-20 dB, -16 dB, -12 dB, -8 dB, -4 dB
Sensing cycle no.	1000

Table 10. Values of simulation parameters for P_D vs N (variable SNR).

Figure 32 illustrates the matched filter sensing results for P_D vs N (variable SNR). As expected, P_D increases with the increase in N and for a high level of SNR, the detection also improves. Furthermore, in cases of higher levels of SNR, smaller numbers of samples are required to achieve 100% of P_D. The results also show an important fact about matched filter sensing. It reaches 100% detection rate faster than the other two methods discussed previously.
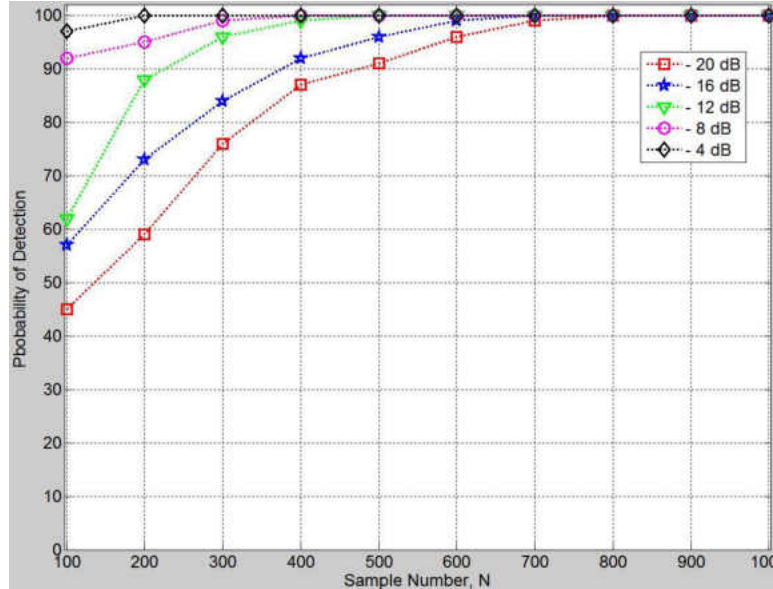
Figure 32. Matched filter sensing simulation results for P_D vs N (Variable *SNR*).

Figure 32 shows that a high level of P_D can be achieved even for a very low *SNR* value. However, there can be still a high level of without the presence of the primary transmission in the received signal. This phenomenon is known as a false alarm and the way to investigate this is the probability of false alarm, P_{FA}. In this section, the performance of P_{FA} has been simulated by varying the *SNR* for different levels of threshold, λ_{MF}. The performance evaluation of P_{FA} is done by varying *SNR* and the threshold, λ_{MF}. The sample number, N is set to be 1000 and the PU signal strength is set at -70 dBm. The *SNR* range was varied from -20 dB to +20 dB by varying the noise power. T_{ED} and λ_{MF} are calculated as previously described and λ_{MF} was varied by multiplying factors to observe the effect of threshold variation on P_D. For each set of parameters the simulation has been run for 1000 cycles to get an average result of P_{FA}. The values of the parameters for this simulation are shown in the following Table 11.

Parameters	Values
Sample Number, N	1000
PU signal strength	-70 dBm
SNR range	-20 dB to +20 dB
Threshold factors	1, 2, 3, 4
Sensing cycle no.	1000

Table 11. Values of simulation parameters for P_{FA} vs SNR (Variable λ_{MF}).

The simulation results for P_{FA} vs SNR for different threshold values are shown in Figure 33. As expected, P_{FA} for the matched filter sensing decreases with the increase in SNR, but the rate of decrease is smaller than the rates corresponding to energy detection (Figure 26). Increasing the value of λ_{MF} shows to be affecting the switching point in terms of SNR at which the P_{FA} curve starts to fall from 100% value to 0%.
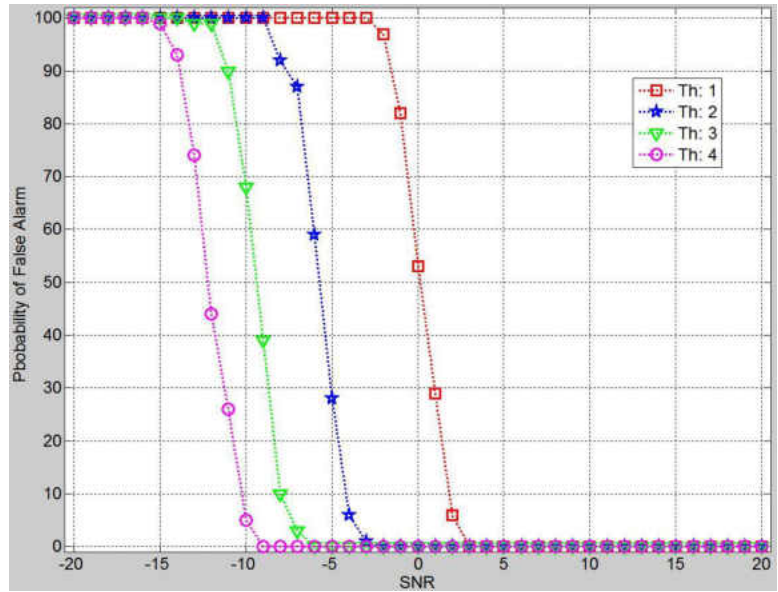
Figure 33. Matched filter sensing simulation results for P_{FA} vs SNR (Variable λ_{MF}).

4.5 Comparison of the three methods

In this section the performance of energy detection, correlation based sensing, and matched filter sensing have been compared to understand the strengths and weaknesses of the three methods. For this comparison, the following metrics are used:

1) Probability of Detection, P_D vs SNR

2) Probability of Detection, P_D vs Sample Number, N (Variable SNR)

3) Probability of False Alarm, P_{FA} vs SNR

First, we simulated the probability of detection, P_D, for all the three methods by varying the *SNR and* the threshold. The sample number, N, is set at 1000 for all the methods and the PU signal strength is set at -70 dBm. The SNR is varied in the range of -20 dB to +20 dB by varying the noise power. For each set of parameters the simulation was run 1000 cycles to get an average result of P_D. The values of the parameters for this simulation are shown in Table 12.

Parameters	Values
Sample Number, N	1000
PU signal strength	-70 dBm
SNR range	-20 dB to +20 dB
FFT size	128
P_{FA}	0.2
Threshold factor	1
Sensing cycle no.	1000

Table 12. Values of simulation parameters for P_D vs SNR.

The comparison results are shown in Figure 34. This figure shows that P_D increases with the increase in SNR for all the 3 methods. P_D of energy detection and the one of the matched filter sensing are very close which suggest that they have almost the same

performance. However, the correlation based method shows to require higher SNR to be performing at 100% level of P_D.
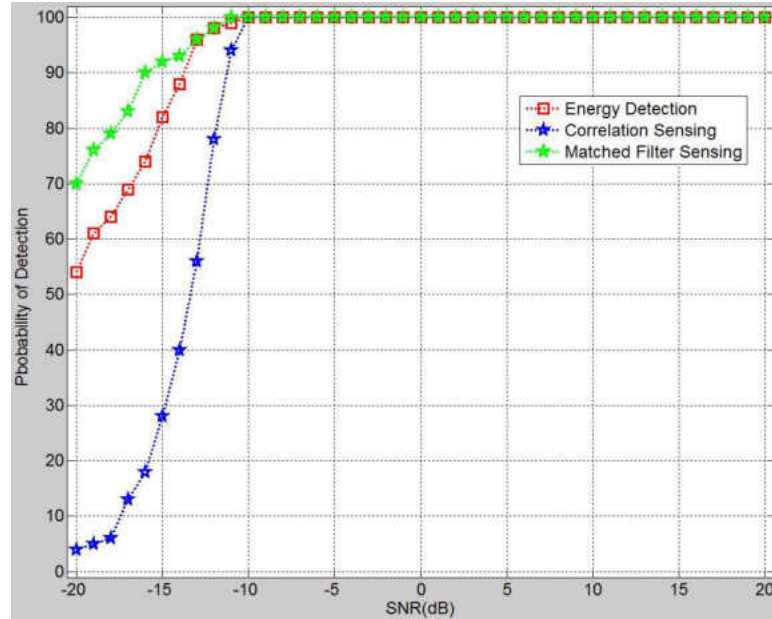
Figure 34. Comparative simulation results for P_D vs SNR.

Figure 35 shows the results of P_D vs number of samples. As can be seen P_D for energy detection and matched filter increases with the increase in N. P_D of the matched filter, however, reaches 100% detection rate faster than the one of energy detection. On the other hand, for correlation based sensing technique, P_D does not seem to change with the increase in N.
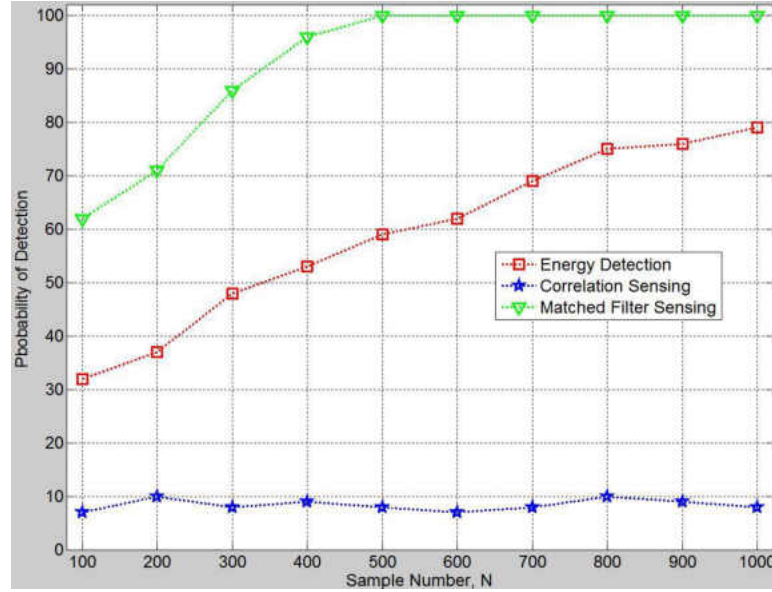
Figure 35. Comparative simulation results for P_D vs N

The probability of false alarm, P_{FA} has been simulated by varying the *SNR* for all three methods. In this test matrix three sensing methods are compared with the performance of P_{FA} is evaluated by varying SNR. The sample number, N is set to be 1000 and the PU signal strength is set at -70 dBm. The SNR range was varied from -20 dB to +20 dB by varying the noise power. For each set of parameters the simulation has been run for 1000 cycles to get an average result of P_{FA}. The values of the parameters for this simulation are shown in the following Table 13.

Parameters	Values
Sample Number, N	1000
PU signal strength	-70 dBm
SNR range	-20 dB to +20 dB
FFT size	128
P_{FA}	0.2
Threshold factor	1
Sensing cycle no.	1000

Table 13. Values of simulation parameters for P_{FA} vs *SNR*.

As discussed previously, P_D does not always give the actual sensing performance. That happens because in cases of very low SNR, the noise power is very high and the cognitive radio might detect a primary user even there is none present. This is known as a false alarm and to evaluate this, simulation for the probability of false alarm, P_{FA} is done. Figure 36 illustrates the results of P_{FA} vs *SNR* for 3 methods under investigation. From these results we can see that P_{FA} for the matched filter sensing decreases with the increase in SNR except for correlation based sensing. It also shows the effect of threshold increment on P_{FA}. With correlation method, for very low level of *SNR*, P_{FA} remains zero or very close to zero as shown in the figure. It is due to the fact that correlation based sensing method is very good at separating noise from PU signal at all SNR levels. However, energy detection and matched filter sensing are on par with each other in terms of performance.
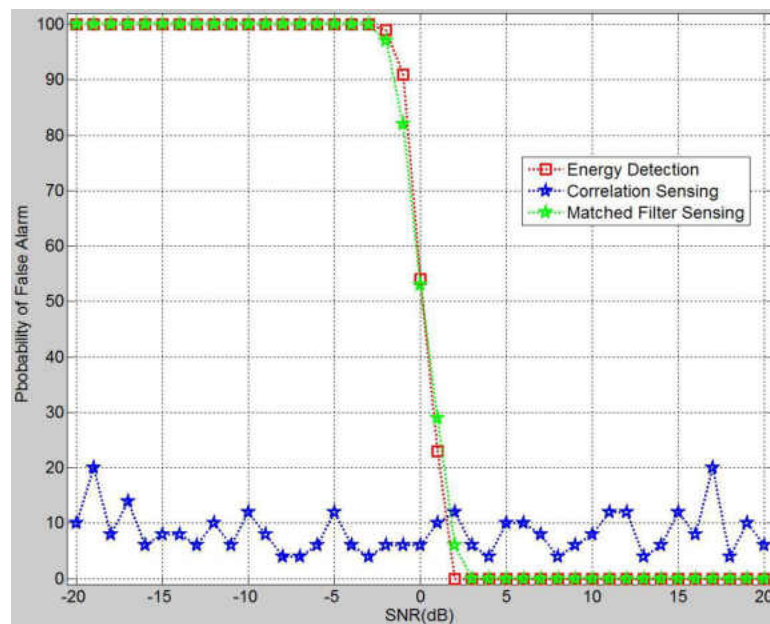
Figure 36. Comparative simulation results for P_{FA} vs *SNR*.

4.6 Discussion

In this chapter, three basic spectrum sensing methods for identifying underutilized radio spectrum have been simulated and their corresponding results have been discussed. Between the three methods tested, energy detection is the simplest one. However, results show that the probability of false alarm of this method is high in situations with low SNR. To get better detection performance with energy detection, the number of samples, N, as well as the threshold value has to be high which work only for high power signals; thus this technique is suitable for simple, fast, and less costly spectrum sensing with some compromise in performance.

Although matched filter sensing has shown a similar probability of false alarm as the energy detection technique, matched filter method achieves a higher level of detection for lower number of samples, N compared to energy detection. In fact, matched filter method reaches close to 100% probability of detection at lower SNR such as -5 dB shown in Figure 34 compared to the other two methods discussed previously. On the other hand, autocorrelation based sensing shows a similar level of detection performance to the energy detection, but offers a very low probability of false alarm compared to the other two methods tested. Thus, both correlation based sensing and matched filter sensing techniques have their respective advantages over energy detection.

Energy detection does not require any assumptions on the primary signal. Unfortunately, this also means that energy detection cannot distinguish between signals and interference. Moreover, energy detection is more susceptible to noise uncertainty that renders detection below certain SNR values, regardless of the number of samples (i.e., the

SNR wall behavior). Hence, energy detection requires an accurate noise level estimate, which is not feasible. The benefit of energy detection is that it is computationally very inexpensive. On the other hand, correlation based sensing has the advantage of having a very low probability of false alarm compared to energy detection and matched filter methods. Finally, matched filter sensing has the benefit of having the fastest detection in terms of number of samples, N, but suffers from high probability of false alarm. Furthermore, prior knowledge of the primary transmission is a prerequisite for matched filter method which might not be available to the cognitive radio system. Thus, selecting one of these three sensing techniques will depend on the operating SNR range, noise uncertainty of the transmission channel, processing power on the sensing unit, and available information about the primary user's transmission. Table 14 lists some of the advantages and disadvantages of the three sensing methods.

Sensing Type	Advantages	Disadvantages
Energy Detection	> Easy to implement > Prior knowledge of primary signal not required	> High false alarm rate > Unreliable in low SNR values
Correlation Based Sensing	> Robust against noise uncertainty > Can distinguish between primary signal and noise	> Higher data processing
Matched Filter Sensing	> Better detection at low SNR region > Needs less signal samples for good sensing performance	> Prior knowledge of primary signal is required > One sensing system works for only one type of primary user

Table 14. Advantages and disadvantages of the three sensing methods

CHAPTER V

CONCLUSIONS AND FUTURE WORK

The use of radio frequencies has increased dramatically during the past few decades. As a result, the radio frequency spectrum is becoming more and more scarce. Efficient and reliable operation in this scares environment calls for flexible and intelligent automatic systems capable of adapting to the existing radio environment. In order to facilitate learning and adaptation, these systems must observe the radio environment and sense the spectrum and become aware its state. Cognitive radios are future of communication devices that are capable of learning from the environment. Accurate and efficient spectrum sensing operations are the tasks that the future communication systems will need to accomplish for optimum performance.

In this thesis, a comparative study of the three basic spectrum sensing methods i.e. energy detection, correlation base sensing, and matched filter sensing have been presented. This study includes methodologies and simulation results of the three techniques of sensing developed on the GNU Radio platform. Results show that each of the sensing methods has strengths and weaknesses. The energy detection method, for example, has the advantage of being very simple to implement. However, the detection performance of energy detection degrades significantly when the SNR is very low. On the other hand, autocorrelation based sensing shows very low probability of false alarm even under very

low SNR. Nonetheless, it requires a larger number of samples for getting a good detection performance compared to the other two methods of sensing. The matched filter method required the least number of samples to achieve 100% detection rate among the three sensing techniques, but shows high probability of false alarm comparable to the one of energy detection. Thus, the results obtained from the simulations provide a decent understanding of the three basic sensing methods under investigation.

In the future, the main challenge will be the hardware implement of theses sensing techniques using the GNU radio platform. GNU radio is an open source platform and supports seamless compatibility with software defined radio (SDR) hardware. All the python codes and custom GNU radio blocks that have been developed for the simulations in this thesis are ready for SDR interfacing and will provide a strong base for hardware implementation. The effect of carrier to noise interference ratio (CINR) is an important parameter which also have to be considered for hardware implementation. Another future prospect of this work is to combine these three simple sensing techniques in a way so that the weaknesses can be rectified while keeping the combine sensing method less complex.

APPENDIX

PYTHON CODE FOR SPECTRUM SENSING BLOCK

```
#!/usr/bin/env python
######################################################
# Gnuradio Python Flow Graph
# Title: Tx Rx Sim V1
# Generated: Thu Jan 23 16:49:56 2014
######################################################

from gnuradio import blocks
from gnuradio import digital
from gnuradio import eng_notation
from gnuradio import gr
from gnuradio.eng_option import eng_option
from gnuradio.gr import firdes
from optparse import OptionParser
import howto
import numpy
from scipy.fftpack import dct
from statsmodels.robust.scale import mad
import random
import numpy as np
import time
import sys
from numpy import linalg as ln
import math

class TX_RX_SIM_v1(gr.top_block):

    def __init__(self):
        gr.top_block.__init__(self, "Tx Rx Sim V1")

        ######################################################
        # Variables
        ######################################################
        self.samp_rate = samp_rate = 20000
```

```python
            ##################################################
            # Blocks
            ##################################################
            self.random_source_x_0 = gr.vector_source_b(map(int,
numpy.random.randint(0, 8, 1000000)), True)
            self.howto_sample_0 = howto.sample()
            self.digital_psk_mod_0 = digital.psk.psk_mod(
              constellation_points=4,
              mod_code="gray",
              differential=True,
              samples_per_symbol=4,
              excess_bw=0.35,
              verbose=False,
              log=False,
              )
            self.blocks_multiply_const_vxx_0 =
blocks.multiply_const_vcc((1+1j, ))


            ##################################################
            # Connections
            ##################################################
            self.connect((self.random_source_x_0, 0),
(self.digital_psk_mod_0, 0))
            self.connect((self.digital_psk_mod_0, 0),
(self.blocks_multiply_const_vxx_0, 0))
            self.connect((self.blocks_multiply_const_vxx_0, 0),
(self.howto_sample_0, 0))


    def get_samp_rate(self):
          return self.samp_rate

    def set_samp_rate(self, samp_rate):
          self.samp_rate = samp_rate

    def get_spectrum(self):
          return self.howto_sample_0.get_spectrum()


if __name__ == '__main__':
    parser = OptionParser(option_class=eng_option, usage="%prog:
[options]")
    (options, args) = parser.parse_args()
    tb = TX_RX_SIM_v1()
    tb.start()
```

```
############ Variables Declaration ##############

      # Number of samples taken for each sensing cycle
      length = 1000

      # Total number of sensing cycles
      loop   = 1000

      # SNR range for the simulation
      snrlist = [-20,-19,-18,-17,-16,-15,-14,-13,-12,-11,-10,-9,-
8,-7,-6,-5,-4,-3,-2,-
1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]

      # Threshold factor
      thlist    = [1, 2, 3, 4]

      # Target Probability of False alarm for Energy detection
      fa = 0.8416 # Pfa = 0.2

      # mean and standard deviation for Gaussian noise
      mu, sigma = 0, 1e-3


############ Variables Declaration ##############

      # For loop for varying threshold
      for th in thlist:

            # For loop for varying SNR
            for aa in snrlist:
                  snr = float(aa)
                  count = 0
                  sumFT = 0.0
                  sumDT = 0.0
                  sumAC = 0.0
                  sumMF = 0.0
                  suma  = 0.0
                  sumvar= 0.0
                  thfa  = th
                  FT    = 0.0
                  MFT     = 0.0
                  std_n = 0.0
                  std_s = 0.0

                  while count < loop:
                        # Noise generation
```

```python
                noise_1        = np.random.normal(mu, sigma,
length)             #Simulated Noise
                noise_2        = np.random.normal(mu, sigma,
length)             #Simulated Noise

                noise_re        =
np.array(noise_1,dtype=complex)
                noise_im        =
np.array(noise_2,dtype=complex)
                J                        = complex(0,1)
                noise_im        = noise_im * pow(J,5)

                noise_array     = noise_re + noise_im
                noise_array1 = (1 /
np.sqrt(np.cov(noise_array)) ) * noise_array

                # QPSK Signal from USRP-Signal Generator
block
                signal        = tb.get_spectrum()

                signal_array = np.array(signal)
                signal_array1 = (1 /
np.sqrt(np.cov(signal_array)) ) * signal_array

                # Reading Pilot signal for Matched Filter
Sensing
                read_data = np.loadtxt('pilot.txt')

                # Set Signal and Noise Power at -70 dBm

                dbm = -70.0

                signal_array1 = np.sqrt(10**((dbm/10)-3)) *
signal_array1;

                noise_array1  = np.sqrt(10**((dbm/10)-3)) *
noise_array1;

                # SNR adjustment for Signal and Noise

                noise_array =
np.sqrt(np.cov(signal_array1)*10**-
(snr/10))*(1/np.sqrt(np.cov(noise_array1))) * noise_array1;
                #### noise adjustment
                signal_array= signal_array1;
```

77

```python
                        # Noise variance for threshold calculation
                        noisez_var = np.cov(noise_array)


                        # For Probability of Detection
                        spectrum_on = noise_array + signal_array

                        # For Probability of False alarm
                        spectrum_off = noise_array

                        spectrum = spectrum_on


        ##################################################
        # -------- Energy Detection --------------------
        ##################################################

                        # Decision Statistic T calculation
                        float_samples = abs(spectrum )

                        sample_fft=np.fft.fft(float_samples,
        n=length)

                        des_stat = (np.sum( (
        np.square(abs(sample_fft)) ) ))/length
                        des_stat_FT = des_stat/2

                        # Threshold Calculation
                        threshold =  ( fa * np.sqrt(2 * length) +
        length ) * noisez_var

                        # Sensing Decision
                        if des_stat_FT <= threshold:
                            op = 0
                        else:
                            op = 1


                        if op == 1:
                            string_op = 'H1'
                        else:
                            string_op = 'H0'
```

```
###################################################
# ------------- Correlation  Sensing -----------
###################################################
                    spectrum_abs = abs(spectrum)
                    # Autocorrelation calculation
                    ac = np.correlate(spectrum, spectrum,
"full")

                    lag0 = abs(ac[length-1])
                    lag1 = abs(ac[length])

                    # Sensing Decision
                    if lag1 <= lag0 * 0.05 * thfa:
                         ac_i = 0
                    else:
                         ac_i = 1

                    if ac_i == 1:
                         string_op = 'H1'
                    else:
                         string_op = 'H0'




###################################################
# ------------- Matched Filter -----------------
###################################################
                    # Decision Statistic T calculation
                    spectrum_abs = abs(spectrum)
                    ac = np.convolve(read_data, spectrum,
"full")

                    T_MF = np.mean(abs(ac))

                    # Matched Filter Threshold from " Quiet
Time Approach"
                    MF_TH = 0.000181802375265*thfa          # N =
1000


                    # Sensing Decision
                    if T_MF <= MF_TH:
                         mf_i = 0
                    else:
                         mf_i = 1
```

```python
                    if mf_i == 1:
                            string_op = 'H1'
                    else:
                            string_op = 'H0'


                    string_snr = "%g" % snr

                    count = count+1
                    # Snapshots loop ends here

            # Converting data types for writing on txt file
            d_rate = 100*sumFT/(loop)
            string_d_rate = "%g" % int(d_rate)

            d_rate_AC = 100*sumAC/(loop)
            string_d_rate_AC = "%g" % int(d_rate_AC)

            d_rate_MF = 100*sumMF/(loop)
            string_d_rate_MF = "%g" % int(d_rate_MF)


            FT = FT/loop
            string_FT = "%g" % int(FT)

            std_n = std_n/loop
            string_std_n = "%.8f" % (std_n)

            std_s = std_s/loop
            string_std_s = "%.8f" % (std_s)


            # Write on txt file
            text_file = open("write_rx.txt","a")

            text_file.write("\n" + "SNR: " + string_snr + "
dB" +"\t\t DRate: " + string_d_rate+"\t\t DRate_AC: " +
string_d_rate_AC + "\t\tDRate_MF: " + string_d_rate_MF )

            text_file.close()


        text_file = open("write_rx.txt","a")

        text_file.write("\n\n\n" )
```

```
        text_file.close()


#     raw_input('Press Enter to quit: ')
      tb.stop()
```

REFERENCES

[1]   K. Patil, R. Prasad, and K. Skouby, "A survey of worldwide spectrum occupancy measurement campaigns for cognitive radio," *International Conference on Devices and Communications (ICDeCom),* pp. 1-5, 2011.

[2]   M. A. McHenry, "NSF spectrum occupancy measurements project summary," *Shared Spectrum Company,* 2005.

[3]   I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *The International Journal of Computer and Telecommunications Networking,* vol. 50, pp. 2127-2159, 2006.

[4]   FCC, "ET Docket No. 02- 135 " *Notice of proposed rule making and order,* 2002.

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

[6]   H. Arslan, *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems*: Springer, 2007.

[7]   K. Arshad, S. Chantaraskul, X. Gelabert, C. Germond, J. Kronander, M. I. Rahman*, et al.*, "Spectrum Sensing," *End to End Efficiency (E3) White Paper,* 2009.

[8]   U. S. FCC, "Second Memorandum Opinion and Order," *In the Matter of Unlicensed Operation in TV Broadcast Bands,* vol. 10-174, 2010.

[9]   S. J. Shellhammer, "Spectrum Sensing in IEEE 802.22," *2nd International Workshop on Cognitive Information Processing (IAPR),* 2008.

[10]   A. Sahai, N. Hoven, and R. Tandra, "Some Fundamental Limits on Cognitive Radio," *Forty-second Allerton Conference on Communication, Control, and Computing,* pp. 1-11, 2004.

[11]   R. Tandra and A. Sahai, "SNR Walls for Signal Detection," *IEEE Journal of Selected Topics in Signal Processing,* vol. 2, pp. 4-17, 2008.

[12]   E. Axell, G. Leus, E. Larsson, and H. Poor, "Spectrum Sensing for Cognitive Radio : State-of-the-Art and Recent Advances," *IEEE Signal Processing Magazine,* vol. 29, pp. 101-116, 2012.

[13]   N. Sai Shankar, C. Cordeiro, and K. Challapali, "Spectrum agile radios: Utilization and sensing architectures," *1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* pp. 160-169, 2005.

[14]   Lo, x, B. pez, x, M. tez, and F. Casadevall, "Improved energy detection spectrum sensing for cognitive radio," *IET Communications,* vol. 6, pp. 785-796, 2012.

[15]   J. Vartiainen, H. Sarvanko, J. Lehtomaki, M. Juntti, and M. Latva-aho, "Spectrum Sensing with LAD-Based Methods," *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC),* pp. 1-5, 2007.

[16]   V. R. S. Banjade, N. Rajatheva, and C. Tellambura, "Performance Analysis of Energy Detection with Multiple Correlated Antenna Cognitive Radio in Nakagami-m Fading," *IEEE Communications Letters,* vol. 16, pp. 502-505, 2012.

[17]   Q. Zhi, C. Shuguang, A. H. Sayed, and H. V. Poor, "Wideband Spectrum Sensing in Cognitive Radio Networks," *IEEE International Conference on Communications (ICC),* pp. 901-906, 2008.

[18]   R. Tandra and A. Sahai, "Fundamental limits on detection in low SNR under noise uncertainty," *International Conference on Wireless Networks, Communications and Mobile Computing,* vol. 1, pp. 464-469, 2005.

[19]   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," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* pp. 170-179, 2005.

[20]   D. Cabric, A. Tkachenko, and R. W. Brodersen, "Experimental study of spectrum sensing based on energy detection and network cooperation," *1st International Workshop on Technology and Policy for Accessing Spectrum (TAPAS),* vol. 222, pp. 1-8, 2006.

[21]   F. Weidling, D. Datla, V. Petty, P. Krishnan, and G. J. Minden, "A framework for R.F. spectrum measurements and analysis," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* pp. 573-576, 2005.

[22]   O. Dong-Chan and L. Yong-Hwan, "Energy detection based spectrum sensing for sensing error minimization in cognitive radio networks," *International Journal of Communication Networks and Information Security (IJCNIS),* vol. 1, pp. 1-5, 2009.

[23]   L. Xiang, W. Bin, W. Hong, H. Pin-Han, B. Zhiqiang, and P. Lili, "Adaptive Threshold Control for Energy Detection Based Spectrum Sensing in Cognitive Radios," *IEEE Wireless Communications Letters,* vol. 1, pp. 448-451, 2012.

[24]   C. H. Lim, "Adaptive energy detection for spectrum sensing in unknown white Gaussian noise," *IET Communications,* vol. 6, pp. 1884-1889, 2012.

[25]   Z. Tian and G. B. Giannakis, "A wavelet approach to wideband spectrum sensing for cognitive radios," *1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom),* pp. 1-5, 2006.

[26]   Z. Tian and G. B. Giannakis, "Compressed Sensing for Wideband Cognitive Radios," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* vol. 4, pp. 1357-1360, 2007.

[27]   Y. L. Polo, Y. Wang, A. Pandharipande, and G. Leus, "Compressive wide-band spectrum sensing," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 2337-2340, 2009.

[28]   C. K. Tan and W. K. Lim, "Reliable and low-complexity wavelet-based spectrum sensing for cognitive radio systems at low SNR regimes," *Electronics Letters,* vol. 48, pp. 1565-1567, 2012.

[29]   S. E. El-Khamy, M. S. El-Mahallawy, and E. S. Youssef, "Improved wideband spectrum sensing techniques using wavelet-based edge detection for cognitive radio," *International Conference on Computing, Networking and Communications (ICNC),* pp. 418-423, 2013.

[30]   W. A. Gardner, "Signal interception: A unifying theoretical framework for feature detection," *IEEE Transactions on Communications,* vol. 36, pp. 897-906, 1988.

[31]   D. Cabric and R. W. Brodersen, "Physical layer design issues unique to cognitive radio systems," *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC),* vol. 2, pp. 759-763, 2005.

[32]   J. Lunden, V. Koivunen, A. Huttunen, and H. V. Poor, "Spectrum sensing in cognitive radios based on multiple cyclic frequencies," *2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom),* pp. 37-43, 2007.

[33]   K. Muraoka, M. Ariyoshi, and T. Fujii, "A Novel Spectrum-Sensing Method Based on Maximum Cyclic Autocorrelation Selection for Cognitive Radio System," *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* pp. 1-7, 2008.

[34]   A. Fehske, J. Gaeddert, and J. Reed, "A new approach to signal classification using spectral correlation and neural networks," *First IEEE International Symposium on*

New Frontiers in Dynamic Spectrum Access Networks (DySPAN), pp. 144-150, 2005.

[35] K. L. Du and M. Wai-Ho, "Affordable Cyclostationarity-Based Spectrum Sensing for Cognitive Radio With Smart Antennas," *IEEE Transactions on Vehicular Technology,* vol. 59, pp. 1877-1886, 2010.

[36] L. Yang and S. K. Jayaweera, "Dynamic Spectrum Tracking Using Energy and Cyclostationarity-Based Multi-Variate Non-Parametric Quickest Detection for Cognitive Radios," *IEEE Transactions on Wireless Communications,* vol. 12, pp. 3522-3532, 2013.

[37] S. Juei-Chin and E. Alsusa, "An Efficient Multiple Lags Selection Method for Cyclostationary Feature Based Spectrum-Sensing," *IEEE Signal Processing Letters,* vol. 20, pp. 133-136, 2013.

[38] G. Vardoulias, J. Faroughi-Esfahani, G. Clemo, and R. Haines, "Blind radio access technology discovery and monitoring for software defined radio communication systems: problems and techniques," *Second International Conference on 3G Mobile Communication Technologies,* pp. 306-310, 2001.

[39] M. Mehta, N. Drew, G. Vardoulias, N. Greco, and C. Niedermeier, "Reconfigurable terminals: an overview of architectural solutions," *IEEE Communications Magazine,* vol. 39, pp. 82-89, 2001.

[40] T. Yucek and H. Arslan, "Spectrum Characterization for Opportunistic Cognitive Radio Systems," *IEEE Military Communications Conference (MILCOM),* pp. 1-6, 2006.

[41] J. Palicot and C. Roland, "A new concept for wireless reconfigurable receivers," *IEEE Communications Magazine,* vol. 41, pp. 124-132, 2003.

[42] Q. Zhi, S. J. Shellhammer, Z. Wenyi, and A. H. Sayed, "Spectrum Sensing by Cognitive Radios at Very Low SNR," *IEEE Global Telecommunications Conference (GLOBECOM),* pp. 1-6, 2009.

[43] Z. Yonghong and L. Ying-Chang, "Spectrum-Sensing Algorithms for Cognitive Radio Based on Statistical Covariances," *IEEE Transactions on Vehicular Technology,* vol. 58, pp. 1804-1815, 2009.

[44] Z. Yonghong and L. Ying-Chang, "Covariance Based Signal Detections for Cognitive Radio," *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* pp. 202-207, 2007.

[45] Z. Yonghong and L. Ying-Chang, "Maximum-Minimum Eigenvalue Detection for Cognitive Radio," *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC),* pp. 1-5, 2007.

[46] A. M. Tulino and S. Verdú, *Random Matrix Theory and Wireless Communications* vol. 1, 2004.

[47] J. Ming, L. Youming, and R. Heung-Gyoon, "On the Performance of Covariance Based Spectrum Sensing for Cognitive Radio," *IEEE Transactions on Signal Processing,* vol. 60, pp. 3670-3682, 2012.

[48] S. K. Sharma, S. Chatzinotas, and B. Ottersten, "Eigenvalue-Based Sensing and SNR Estimation for Cognitive Radio in Presence of Noise Correlation," *IEEE Transactions on Vehicular Technology,* vol. 62, pp. 3671-3684, 2013.

[49] A. Sahai, R. Tandra, S. M. Mishra, and N. Hoven, "Fundamental design tradeoffs in cognitive radio systems," *Proceedings of the first international workshop on Technology and policy for accessing spectrum,* pp. 1-9, 2006.

[50] D. Cabric, A. Tkachenko, and R. W. Brodersen, "Spectrum Sensing Measurements of Pilot, Energy, and Collaborative Detection," *IEEE Military Communications Conference (MILCOM),* pp. 1-7, 2006.

[51] H. Tang, "Some physical layer issues of wide-band cognitive radio systems," *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* pp. 151-159, 2005.

[52] F. Moghimi, R. Schober, and R. K. Mallik, "Hybrid Coherent/Energy Detection for Cognitive Radio Networks," *IEEE Transactions on Wireless Communications,* vol. 10, pp. 1594-1605, 2011.

[53] J. Proakis, *Digital Communications*: McGraw Hill, New York, 2001.

[54] Y. F. Sharkasi, D. McLernon, and M. Ghogho, "Robust spectrum sensing in the presence of carrier frequency offset and phase noise for cognitive radio," *Wireless Telecommunications Symposium (WTS),* pp. 1-5, 2012.

[55] A. Sahai and D. Cabric, "Spectrum sensing: fundamental limits and practical challenges," *IEEE International Symposium on New Frontiers Dynamic Spectrum Access Networks (DySPAN),* pp. 1-21, 2005.

[56] J. J. Popoola and R. van Olst, "A Novel Modulation-Sensing Method," *IEEE Vehicular Technology Magazine,* vol. 6, pp. 60-69, 2011.

[57] C. Jiang, Y. Li, W. Bai, Y. Yang, and J. Hu, "Statistical matched filter based robust spectrum sensing in noise uncertainty environment," *IEEE 14th International Conference on Communication Technology (ICCT),* pp. 1209-1213, 2012.

[58] L. Ma, Y. Li, and A. Demir, "Matched filtering assisted energy detection for sensing weak primary user signals," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 3149-3152, 2012.

[59] C. Hou-Shin, G. Wen, and D. G. Daut, "Signature Based Spectrum Sensing Algorithms for IEEE 802.22 WRAN," *IEEE International Conference on Communications (ICC),* pp. 6487-6492, 2007.

[60] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," *Thirty-Eighth Asilomar Conference on Signals, Systems and Computers,* vol. 1, pp. 772-776, 2004.

[61] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," *1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN),* pp. 131-136, 2005.

[62] A. Singh, M. R. Bhatnagar, and R. K. Mallik, "Cooperative Spectrum Sensing in Multiple Antenna Based Cognitive Radio Network Using an Improved Energy Detector," *IEEE Communications Letters,* vol. 16, pp. 64-67, 2012.

[63] S. Atapattu, C. Tellambura, and J. Hai, "Energy Detection Based Cooperative Spectrum Sensing in Cognitive Radio Networks," *IEEE Transactions on Wireless Communications,* vol. 10, pp. 1232-1241, 2011.

[64] Y. Wang, A. Pandharipande, Y. L. Polo, and G. Leus, "Distributed compressive wide-band spectrum sensing," *Information Theory and Applications Workshop,* pp. 178-183, 2009.

[65] H. Sadeghi, P. Azmi, and H. Arezumand, "Cyclostationarity-based soft cooperative spectrum sensing for cognitive radio networks," *IET Communications,* vol. 6, pp. 29-38, 2012.

[66] S. Srinu and S. L. Sabat, "Cooperative wideband sensing based on entropy and cyclic features under noise uncertainty," *IET Signal Processing,* vol. 7, pp. 655-663, 2013.

[67] T. Kieu-Xuan and I. Koo, "Cooperative spectrum sensing using Kalman filter based adaptive fuzzy system for cognitive radio networks," *KSII Transactions on Internet and Information Systems,* vol. 6, pp. 307-324, 2012.

[68] K. Jaeweon, C. Chan-Byoung, and J. G. Andrews, "Cooperative Spectral Covariance Sensing under Correlated Shadowing," *IEEE Transactions on Wireless Communications,* vol. 10, pp. 3589-3593, 2011.

[69] J. Ma, G. Zhao, and Y. Li, "Soft Combination and Detection for Cooperative Spectrum Sensing in Cognitive Radio Networks," *IEEE Transactions on Wireless Communications,* vol. 7, pp. 4502-4507, 2008.

[70] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative Sensing among Cognitive Radios," *IEEE International Conference on Communications (ICC),* vol. 4, pp. 1658-1663, 2006.

[71] Z. Quan, S. Cui, H. V. Poor, and A. H. Sayed, "Collaborative wideband sensing for cognitive radios," *IEEE Signal Processing Magazine,* vol. 25, pp. 60-73, 2008.

[72] C. Biao, J. Ruixiang, T. Kasetkasem, and P. K. Varshney, "Channel aware decision fusion in wireless sensor networks," *IEEE Transactions on Signal Processing,* vol. 52, pp. 3454-3458, 2004.