

Ph.D. Thesis

**Studies on Spectrum Sensing algorithms and real-time
implementation for cognitive radio**

A thesis submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Electronics Science

By

D. K. Sunil

(09PHPE02)



Supervisor:

Prof. Samrat L. Sabat

Centre for Advanced Studies in Electronics Science and Technology

(CASEST)

School of Physics

University of Hyderabad

Hyderabad – 500046, India

December 2018



Centre for Advanced Studies in Electronics Science and Technology
(CASEST)
School of Physics
University of Hyderabad
Hyderabad, India

DECLARATION

I, Mr.**D.K.Sunil** hereby declare that this thesis entitled “**Studies on Spectrum Sensing algorithms and real-time implementation for cognitive radio**” submitted by me under the guidance and supervision of Dr. Samrat L. Sabat is a bonafide research work which is also free from plagiarism. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma. I hereby agree that my thesis can be deposited in Shodganga/INFLIBNET. A report on plagiarism statistics from the University Librarian is enclosed.

D.K.Sunil

Ph.D (Electronics Science)

Reg. No: 09PHPE02

Dr. Samrat L. Sabat

Professor

CASEST

School of Physics

University of Hyderabad



Centre for Advanced Studies in Electronics Science and Technology
(CASEST)
School of Physics
University of Hyderabad
Hyderabad, India

CERTIFICATE

This is to certify that the thesis entitled **Studies on Spectrum Sensing algorithms and real time implementation for cognitive radio** submitted by **D.K.Sunil** bearing registration number 09PHPE02 in partial fulfilment of the requirements for award of Doctor of Philosophy (Electronics Science) in the Centre for Advanced Studies in Electronics Science and Technology (CASEST), School of Physics is a bonafide work carried out by him under my supervision and guidance.

This thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for award of any degree or diploma.

Further, the student has the following publications before submission of the thesis for adjudication and has produced evidence for the same in the form of acceptance letter or the reprint in the relevant area of his research:

1. Krishnamurthy Sunil Devanahalli, and Samrat L. Sabat. Blind SNR Estimation for M-ARY Frequency Shift Keying Signal Using Covariance Technique , AEU - International Journal of Electronics and Communications (Elsevier), vol 70, issue 10 ,2016, pp. 1388 - 1394, ISSN:1434-8411.

Chapter of dissertation where this publication appears : Chapter 4.

2. D. K. Sunil and S. L. Sabat, Spectrum sensing using envelope tracking and signal moment, in Proceeding of IEEE International Conference on Signal Processing and Communication (ICSC), Noida, Dec,2016, pp. 502-507, Electronic ISBN: 978-1-5090-2684-5.

Chapter of dissertation where this publication appears : Chapter 6.

And has made presentations in the following conferences:

1. D. K. Sunil and S. L. Sabat, Real Time Evaluation of Energy Detection and Covariance based Spectrum Sensing Algorithms, 5th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace and Electronic Systems (WVITAE), Hyderabad, Dec 13-16,2015.

Chapter of dissertation where this publication appears : Chapter 5.

2. D. K. Sunil and S. L. Sabat, Spectrum sensing using envelope tracking and signal moment, International Conference on Signal Processing and Communication (ICSC), Noida,Dec 26-28, 2016.

Further the student has passed the following courses towards fulfillment of coursework requirement for Ph.D.

	Course Code	Name	Credits	Pass/Fail
1	PE 900	Semiconductor Device Physics	4	Pass
2	PE 901	IC Fabrication Technology	4	Pass
3	PE 902	RF/Microwave IC's Theory	4	Pass
4	PE 903	VLSI Design Technology Theory	4	Pass

Supervisor

Dr.Samrat L.Sabat

Head

CASEST

Dean

School of Physics

To

My father

Acknowledgements

Foremost I would like to express my sincere gratitude to my research supervisor Prof. Samrat.L.Sabat for all the support and guidance throughout the progress of this research work. His encouragement, guidance and patience have made this thesis possible.

I take this opportunity to thank Prof. K.P.N.Murthy, Prof.G.Rajaram, Prof. M. Ghanshyam Krishna, Prof.K.C.James Raju for their valuable suggestions during the course of this research work.

I would like to thank the management of M/s.Hindustan Aeronautics Limited for permitting me to pursue this research work and for the use of equipment and facilities. My heartfelt thanks to Jyoti Gupta, Shawn James and Arun Lal for their assistance.

I am thankful to Dr.Shesham Srinu, Dr.Mundla Narasimhappa and other research scholars for their co-operation and suggestions during the research. I am thankful to Mr.Abraham and all administrative staff of CASEST.

I wish to express my gratitude to my wife Dr.L.N.Shubha who has motivated me time and again to cross the finishing line. I would like to thank my parents and my children for their patience and understanding over so many years.

Devanahalli Krishnamurthy Sunil

Abstract

In the recent past Cognitive Radio (CR) technology has received increased attention to solve the spectrum scarcity problem using the opportunistic spectrum re-usage technique. It allows secondary users to use the unlicensed channels opportunistically without causing interference to the licensed users. It involves mainly two functionalities namely spectrum sensing and spectrum management. Spectrum sensing identifies vacant spectrum bands, i.e., spectrum holes for the opportunistic use of secondary users in the network. Robust spectrum sensing is essential to avoid interference to existing 'licensed' users and maximize the spectrum utilization. There exist mainly two class of sensing algorithms: data aided and blind. The data aided algorithm requires the characteristics of the signal a priori for successful detection. However, in the real-time radio environment, it is not always possible to know the signal characteristics a priori. Hence blind algorithms are essential for reliable spectrum sensing in the real-time environment.

In the sensing domain, although energy detection is popularly being used due to its low computational complexity, its performance suffers due to noise uncertainty. Although it is a blind technique, but it requires information about the noise variance. The estimation accuracy of received signal noise variance affects the detection performance. This thesis proposes an improved energy detection technique where the threshold is adjusted with respect to the noise variance. The noise variance is estimated using the Linear predictor method, i.e., the Burg method. This thesis performs a detailed performance analysis of the energy detection algorithm with noise variance estimation in the case of a single node and multi-node sensing under white and colored noise characteristics of the channel. Since the accuracy of sensing depends on the accuracy of the noise variance estimator, a detailed study of different types of noise estimator and its impact on sensing accuracy is also carried out.

The sensing performance can also be improved by estimating the signal to noise ratio (SNR) of the received signal. This work also presents a blind SNR estimation technique to match the real-time requirement. The proposed SNR estimation technique simultaneously performs spectrum sensing without any extra computational overhead. Its performance is analyzed for a wide range of SNR

ranging from -20 dB to +20 dB.

In the real-time environment many times, the received signal does not correlate samples. Thus the popular methods that exploit correlation of the covariance matrix fail during sensing; this work proposes a time domain sensing technique utilizing the envelope tracking of the fourth moment of the received signal.

In general, the performance achieved in the algorithmic simulation is not achieved in real time environment due to inadequate modeling. Thus, in the CR domain, at one end there is a continuous demand to develop robust sensing algorithms, and at another end, there is equally demand to perform real-time validation of the algorithms. This thesis presents the real-time implementation of the sensing algorithms such as Energy detection, Covariance detection and moment based detection algorithm in an experimental testbed. It further analyses the locking time for each of the algorithm.

The main contributions of this thesis are as follows:

1. The analysis of linear predictor based noise variance estimator in white and its extension to multi-node sensing. Comparative analysis of different weight estimation techniques, in the case of multi-node sensing.
2. Propose a blind Signal to Noise Ratio (SNR) estimation algorithm that can perform both estimation and sensing, without any extra computational overhead.
3. Propose a time domain envelop tracking method for spectrum sensing.
4. Performance validation of a set of spectrum sensing algorithms such as energy detection, covariance method and envelop tracking method in an experimental testbed.

Thus, the results in the thesis provide the performance of a set of spectrum sensing algorithms in both simulation and real-time experimental testbed under different radio environment.

Contents

Acknowledgements	v
Abstract	vi
Abbreviations	xiv
Notations	xvi
Publications	xvii
1 Introduction	1
1.1 Introduction	1
1.2 Challenges	4
1.3 Research objective	5
1.4 Thesis contribution	5
1.5 Thesis organization	6
2 Spectrum sensing and Hardware test setup	9
2.1 Objective	9
2.2 Spectrum sensing techniques	9
2.2.1 Performance indicators	10
2.2.2 Energy detection	11
2.2.3 Covariance based sensing	12
2.2.4 Eigenvalue based sensing	14
2.2.5 Cyclostationarity based sensing	15
2.2.6 Matched Filter based sensing	16
2.2.7 Cooperative sensing	17
2.3 Hardware implementation	21
2.3.1 Hardware details	21
2.3.2 Instrumentation and Hardware test setup	22
3 Energy detection with noise estimation	25
3.1 Objective	25

3.2	Introduction	25
3.3	System model	27
3.4	Single node spectrum sensing using improved Energy detection technique	28
3.5	Noise variance estimator	29
3.5.1	Auto regressive variance estimator	29
3.5.2	Threshold adaptation	30
3.6	Unbiased estimator	30
3.7	Colored noise	30
3.8	Cooperative sensing using improved energy detection	31
3.8.1	Soft Decision fusion	32
3.8.1.1	Weight estimation using Log Likelihood Ratio test	33
3.8.1.2	Equal gain Soft decision Fusion	33
3.8.1.3	Weight estimation using Differential Evolution algorithm	34
3.9	Simulation details	35
3.10	Results and discussions	38
3.11	Conclusions	46
4	SNR estimation for spectrum sensing	49
4.1	Objective	49
4.2	Introduction	49
4.3	SNR estimation algorithm	51
4.3.1	Signal detection	51
4.3.2	Approximation of test statistic	52
4.3.3	System model	54
4.4	Partially Data Aided ML estimator for M-ARY FSK	54
4.5	Simulation results and discussions	56
4.6	Conclusions	64
5	Real time implementation	65
5.1	Objective	65
5.2	Introduction	65
5.3	Algorithm evaluation methodology	66
5.3.1	Step 1: Matlab Simulation	67

5.3.2	Step 2: Simulink simulation with Sysgen blocks (Fixed point simulation)	68
5.3.3	Step 3: Hardware Co-Simulation	69
5.3.4	Step 4: Real time test with instrumentation	72
5.4	SNR calibration	76
5.5	Simulation and real-time test results	76
5.6	Results and discussion	77
5.7	Conclusions	81
6	Spectrum sensing with signal moment	83
6.1	Objective	83
6.2	Introduction	83
6.2.1	M4-Edge	84
6.2.1.1	Envelope tracking	84
6.2.1.2	Moments	86
6.2.1.3	Decision	86
6.3	Hardware details, calibration and signal processing	90
6.3.1	Hardware details	90
6.3.2	Simulation and Real-time test results	91
6.4	Results and discussions	91
6.5	Conclusions	96
7	Conclusions and Future work	99
7.1	Conclusions	99
7.2	Future work	102

List of Figures

1.1	The cognitive cycle [24]	4
2.1	Signal model	9
2.2	Cooperative sensing scenario	19
2.3	Test setup for algorithm evaluation	22
3.1	Signal model	27
3.2	Differential Evolution algorithm flow	34
3.3	Plots of computed energy, input and output pulse templates	36
3.4	The one-sided power spectral density vs normalised frequency	37
3.5	The probability of detection vs SNR for additive white noise.	39
3.6	The probability of detection for LP and Unbiased and Without Estimation for blue noise	39
3.7	The probability of detection for LP and Unbiased and Without Estimation for red noise	40
3.8	Detection for the cases of no fade and flat fading Rayleigh channel	41
3.9	Detection performance for additive white noise with non-unity variance	43
3.10	Effect of noise variance estimation on cooperative sensing	44
3.11	Single node and multinode cooperative sensing	44
3.12	Effect of noise variance estimation on ROC for cooperative sensing	45
3.13	Effect of noise variance estimation on Complementary ROC for cooperative sensing	46
3.14	Effect of noise variance estimation on ROC for cooperative sensing	47
4.1	Plot of γ , $\gamma(\text{peak})$ and δ_{MAV} for a BFSK burst	53
4.2	Plot of γ for a BFSK burst	54
4.3	Exponential curve fit of γ	55
4.4	P_d vs SNR for M-FSK signal in Rayleigh flat fading channel	58
4.5	Normalised Mean Square Error for M-ARY FSK for M=2,4,8	60
4.9	Normalised Mean Square Error for 8FSK for Rayleigh flat fading	60
4.6	Normalised Mean Square Error for M-ARY FSK for M=2,4,8	61

4.10	Normalised Mean Square Error for M=2,4 and 8 for Rayleigh flat fading	61
4.7	Plot of γ for BFSK	62
4.11	Normalised Mean Square Error for BFSK for Rayleigh fading with Doppler	62
4.8	Partially Data Aided ML estimation of SNR for M-ARY FSK for M =2,4,8	63
5.1	Block diagram of spectrum sensing architecture	66
5.2	System Generator block for SNR creation	70
5.3	High level schematic of ED implementation using SYSGEN blocks.	70
5.4	Data flows between Simulink and ML605 FPGA board for hardware co-simulation.	71
5.5	Functional setup for Real time performance evaluation.	72
5.6	Realtime Spectrum Sensing algorithm evaluation Setup	73
5.7	Flow chart of CAV implementation on Virtex 6 FPGA	74
5.8	Intermediate frequency to Baseband signal processing	75
5.9	64 K FFT of the noisy signal at an SNR of -11.5 dB	75
5.10	Real time spectrum sensing capture on digital oscilloscope	78
5.11	P_d vs SNR Comparison of two schemes in real-time on FPGA with pulsed input.	79
5.12	Sensing time vs SNR of the pulsed signal in real-time on FPGA	79
5.13	P_d vs SNR Comparison of ED and CAV schemes in all four stages	80
5.14	P_d vs SNR for ED and CAV for BFSK and DVBT.	80
6.1	Envelope tracking at an SNR of -10 and -16 dB	88
6.2	Plots of the <i>actual burst length</i> and <i>detected burst length</i>	89
6.3	High level schematic of M4-Edge implementation using SYSGEN blocks.	89
6.4	Real-time Spectrum Sensing evaluation Setup	90
6.5	Plots of the Fourth Central Moment for mean, upper and lower envelope	92
6.6	P_d vs SNR Comparison of three schemes in real-time on FPGA	93
6.7	Sensing time vs SNR of the pulsed signal in real-time on FPGA	94
6.8	P_d vs SNR Comparison of three schemes in all four stages	95
6.9	P_d vs SNR for ED, M4-Edge and CAV for BFSK and DVBT	96

List of Tables

3.1	Detection performance at different sample sizes	42
4.1	Exponential Curve fitting statistics	57
4.2	Curve fitting values for M=2 and M= 8	59
5.1	FPGA resource estimation for energy detection	71
5.2	Signal capture conditions	75
5.3	SNR calibration : Actual vs Computed on FPGA	76
5.4	Detection sensitivity vs number of sample per frame	77
5.5	Resource utilization on FPGA	81
6.1	Resource utilization of different algorithms on FPGA	96
7.1	Detection performance at sample size of 512	100

Abbreviations

ADC : Analog to Digital Converter
AR : Auto Regressive
AWGN : Additive White Gaussian Noise
BER : Bit Error Rate
BFSK : Binary Frequency Shift Keying
BS : Base Station
BSC : Base Station Centre
CAV : Covariance Absolute Value
CAV : Covariance Absolute Value
CBS : Cognitive Base Station
CDF : Cumulative Distribution Function
CNR : Carrier to Noise Ratio
CPE : Consumer Premises Equipment
CR : Cognitive Radio
CRN : Cognitive Radio Network
CROC : Complementary Receiver Operating Characteristic
CRS : Cognitive Radio System
CSD : Cyclostationary Spectral Density
CSS : Cooperative Spectrum Sensing
DE : Differential Evolution
DFB : Digital Filter Bank
DYSPAN : Dynamic Spectrum Access Network
ED : Energy Detection
EME : Energy with Minimum Eigenvalue
ETSI : European Telecommunications Standards Institute
FC : Fusion Centre
FCC : Federal Communications Commission
FPGA : Field Programmable Gate Array
GSM : Global System for Mobile
GSMA : GSM Association
IoT : Internet of Things

ISM : Industrial Scientific and Medical
LAN : Local Area Network
LLR : Log Likelihood Ratio
LPC : Linear Prediction Coding
M4-Edge : Detection using signal envelope and moment
MAN : Metropolitan Area Network
MCAS : Maximum Cyclic Autocorrelation Selection
OFDM : Orthogonal Frequency Division Multiplexing
PDA : Personal Digital Assistant
PDA MLE : Partially Data Aided Maximum Likelihood Estimator
PSD : Power Spectral Density
PU : Primary User
ROC : Receiver Operating Characteristic
RRS : Reconfigurable Radio System
SCM : Sample Covariance Matrix
SDR : Software Defined Radio
SNR : Signal to Noise Ratio
SU : Secondary User
USRP : Universal Software Radio Processor
UWB : Ultra Wide Band
VANET : Vehicular Network
WAN : Wide Area Network
WARP : Wireless Open Access Research Platform
WGC : Weighted Gain Combining
WPAN : Wireless Personal Area Networks
WRAN : Wireless Radio Access Network

Notations

dBm : dB below 1 milliWatt

dBW : db below 1 Watt

γ : threshold

$s(t)$: signal voltage

$n(t)$: noise voltage

$N_0/2$: Two sided noise power density

$\mathcal{N}(0, \sigma^2)$: Normal distribution with mean zero; variance of σ^2

H_0 : Null hypothesis

H_1 : True hypothesis

P_d : Probability of detection

P_f : Probability of false alarm

P_m : Probability of mis-detection

$U(k)$: Energy of the k^{th} frame

N_s : Number of samples per frame.

$Q(\cdot)$: Gaussian tail probability function.

$R_y(k)$: Sample covariance matrix

$\lambda(l)$: Sample autocorrelation

F_1^{-1} : CDF of the Tracy-Widom distribution of order one

w_i : Weights assigned to the i^{th} CR

Publications

I. Research Papers in Peer Reviewed International Journals

1. **Devanahalli Krishnamurthy Sunil** and S. L. Sabat, “Blind SNR estimation for M-ARY Frequency Shift Keying signal using covariance technique ”, AEU-International Journal of Electronics and Communications (Elsevier), vol. 70, no. 10, pp. 1388-1394, 2016.

II. Research Papers in Peer Reviewed International Conferences

1. **D.K.Sunil** and S. L. Sabat, “Real Time Evaluation of Energy Detection and Covariance based Spectrum Sensing Algorithms ”, The Fifth International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace and Electronic Systems (Wireless VITAE), 2015.
2. **D.K.Sunil** and S. L. Sabat, “Spectrum sensing using envelope tracking and signal moment”, in Proceedings of International Conference on Signal Processing and Communication (ICSC), pp. 502-507, IEEE, 2016.
3. H. Yerranna, S. L. Sabat, **D. K. Sunil**, and S. K. Udgata, “Real time performance evaluation of energy detection based spectrum sensing algorithm using WARP board ”, in Proceedings of International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2719-2723, Sept 2016.

III. Draft Research Papers

1. **D.K.Sunil** and S. L. Sabat, “Improved Single node and Cooperative Multi-node Spectrum Sensing using noise variance estimation ”.

IV. Other Papers in Peer Reviewed International Conferences

1. **D.K.Sunil**, “Networks for NCW: Concepts and issues ”, in Proceedings of International Conference on Computer and Netork Technology (ICCNT), pp. 288-292, World Scientific, 2010.
2. T.Ramesh, B Malarkodi, **D.K.Sunil**, “Aeronautical channel parameter study and analysis for combat aircraft data communication”, International Conference on Electrical,Electronics, Computer Science and Information Technology(ICEECSIT), 2018.

Chapter 1

Introduction

1.1 Introduction

The rapid proliferation of wireless technology combined with the advantages of pervasive wireless computing and communication led to many societal benefits. According to a GSMA report [1], the number of mobile Internet subscribers has increased dramatically from 2.3 billion in 2008 to 5.0 billion in 2017 and is predicted to rise to nearly 6.0 billion by 2020. The impact of wireless technology on the cell phone to sensor network to the Internet of Things is tremendous on building a smart X, i.e., home/city/agriculture/healthcare etc. This explosion of wireless applications on different segment demands more radio spectrum.

In a paper released by the US Federal Communications Commission (FCC) Spectrum Policy Task Force, it is reported that the use of existing spectrum is non optimal [2]. Further, it mentions that (a) Spectrum is allocated by regulatory agencies exclusively to an operator and (b) The spectrum is underutilized by the operator.

There exists a vast temporal and geographic variations in the use of allocated spectrum with utilization ranging from 15% to 85% in major US metropolitan areas [2]. In another study, Berkeley University reported that the spectrum utilization in the frequency range 0-6 GHz is 0.25% to 54.4% [3]. A study in Singapore [4] reported that the spectrum usage in the frequency band 80 to 5850 MHz is 4.54%, and most of the time 66% of the spectrum between 174-230 MHz and 614-790 MHz band is unoccupied. A European agency studied spectrum occupancy across three cities in Europe and concluded that the average spectrum usage is between 6.5% and 10.7% in the frequency range of 400 - 3000 MHz [5]. Similarly, in India, the bands ranging from 170 MHz to 800 MHz, 800 MHz to 1000 MHz, and ISM band ranging from 2.4 GHz to 2.5 GHz were studied for spectrum occupancy. The study concluded that the first and third bands are underutilized whereas the cellular band is utilized effectively [6]. Thus, from different studies, it is clear

that spectrum scarcity is not due to the unavailability of the spectrum but due to the fixed spectrum licensing policy in use. Real-time spectrum occupancy prediction is investigated for dynamic spectrum access for Cognitive radio by the use of time-series models and ML techniques [7]. If the user who does not have a license to use the spectrum, known as a secondary user (SU), can use the underutilized spectrum without causing interference to the licensed user, known as a Primary User (PU), then the spectrum scarcity problem can be resolved. This technique of spectrum reuse is also termed as dynamic spectrum access. Essentially, Cognitive Radio (CR) enables this dynamic spectrum access technique for utilization of the spectrum efficiently.

Cognitive Radio is a radio that can sense the operating radio environment and adapt its transmission parameters accordingly to achieve the best application performance. It can detect the parameters like transmission frequency, bandwidth, power, modulation, waveform etc. that helps the secondary users to identify the best available spectrum for its use. The secondary users use the best available channel/spectrum and coordinate with other users during communication. Some of the applications of CR include, but not limited to, Wide Area Mobile Networks, Energy Efficient Wireless Networks, Public Safety Communications, Wireless Network for Internet of Things (IoT), Vehicular Networks (VANET), Battlefield Ad-Hoc networks etc.

Simon Haykin has defined CR as “.. an intelligent wireless communication system that is aware of its surrounding environment (i.e., outside world), and uses the methodology of understanding by building to learn from the environment and adapt its internal states to statistical variations in the incoming RF stimuli by making corresponding changes in certain operating parameters (e.g., transmit-power, carrier frequency and modulation strategy) in real-time, with two primary objectives in mind: a) Highly reliable communications whenever and wherever needed and b) Efficient utilization of the radio spectrum” [8]. A typical CR cycle is shown in Fig 1.1. It has four main tasks, i.e., (a) detecting vacant spectrum bands known as spectrum sensing (b) analyzing radio environment and selecting the best frequency band known as spectrum decision and (c) coordinating spectrum access with other users known as spectrum sharing and (d) vacating the band when PU reclaims it known as spectrum mobility [8],[9]. It is achieved due to its inherent functionalities like a) spectrum sensing, b) spectrum management and c) spectrum allocation and sharing. The SU after sensing will know the available

bands. Then it enables spectrum management and allocation functionality.

In the Cognitive Radio Network (CRN), if the primary user coexists with the secondary user, then the primary user gets priority to use the spectrum [10]. In this scenario, when a Primary User (PU) appears in the band, the Secondary User (SU) vacates the channel and uses other frequencies according to the channel characteristics. So the SU needs to continuously sense the wide-band spectrum for hopping to the available vacant channel, known as spectrum hole [11]. On the other hand, when secondary users are allowed to use the spectrum in the absence of primary users, the secondary users need to be vigilant about the reclaiming by PU. Thus in the CRN, the secondary users need to sense the radio spectrum for spectrum holes. In the CR domain, this mechanism is known as spectrum sensing. After sensing the SU will be aware of available bands. Then SU enables spectrum management and allocation functionality. The current thesis studies spectrum sensing, whereas the other two tasks are out of the scope of this thesis. Spectrum sensing enables the CR (a) to get information about PU activities in the band (b) presence of spectrum hole. With this information, CR can adapt its transmission characteristics and communicate in vacant bands. Thus spectrum sensing is the primary and critical function of the cognitive radio.

Spectrum sensing is a signal processing algorithm that senses the radio environment (which may have heterogeneous spectrum signatures from multiple devices) under stringent noise and channel impediments. Although, there exist different sensing algorithms, the most popular among them are energy detection [12], matched filter [13], cyclostationary [14], entropy based [15] and covariance techniques [16]. Other algorithms have looked at sub-space methods, wavelet and random matrix methods [17],[18],[19],[20]. The most challenging task is to detect the presence of a signal in a negative Signal-to-Noise Ratio environment with a faded channel. The sensing algorithms are extended to cooperative sensing to tackle the problem of hidden node and fading [21],[22],[23]. All these algorithms compete with each other in terms of detection performance and complexity. To qualify a sensing algorithm to be useful in real-time, a careful study of its detection performance under noise uncertainties, algorithmic complexity, implement-ability, real type prototyping is essential. It is equally important to evaluate the performance of the sensing algorithm in real-time. It motivates to develop different sensing algorithms and evaluate its real-time performance on a hardware platform.

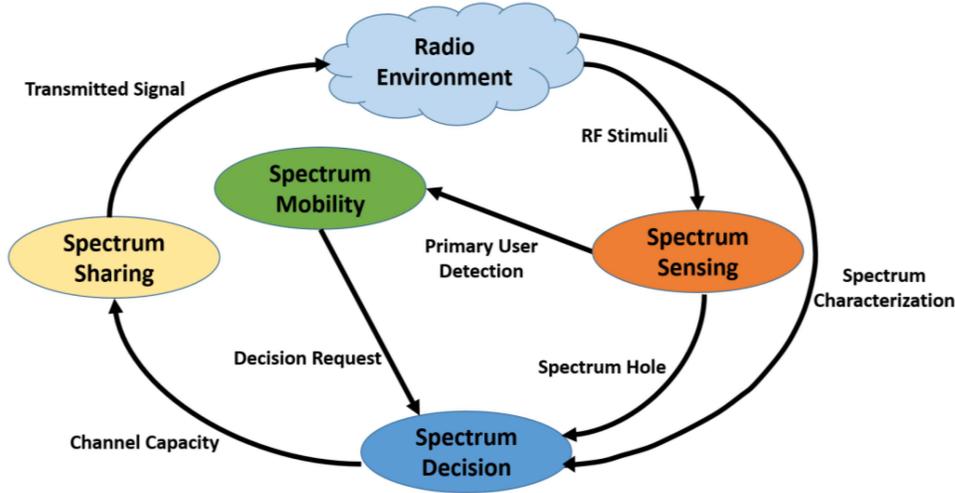


Figure 1.1: The cognitive cycle [24]

1.2 Challenges

To make the CR technology deployable, the CR should decide which frequency band to use? It further leads to questions like : does the radio need to find the available frequency band and should it be able to decide which band is the best frequency band from the available bands?

Thus in spectrum sensing the first challenge is to find the available frequency band then to analyse it. Many of the spectrum sensing algorithms assume ideal channel conditions. In reality, signals fade due to movement or obstructions in the channel. This fade has to be equalized in the algorithm for robust detection. Alternately, the waveform must have fade resistance. Simple methods like energy detection fail at negative SNR conditions and algorithms that exploit the other dimensions of the signal like eigen values require complex matrix operations thus increasing algorithmic complexity. Issues like synchronization, noise figure of the RF section, signal conditioning, sampling related issues, finite word effects also add to the algorithmic complexity. Many of the reported algorithms require the information about the signal apriori, which cannot always be satisfied in real-time. So the development of blind sensing algorithm to work at negative SNR is a challenging task.

The other challenge is the sensing algorithm should perform satisfactorily as per the standards in a real-time environment. It demands the validation of sensing algorithm on a experimental testbed in real-time.

1.3 Research objective

The objective of this research work is to study some of the problems associated with the spectrum sensing namely :

1. Channel SNR estimation at a low Signal-to-Noise Ratio.
2. Noise power estimation to improve the detection probability of energy detection at low Signal-to-Noise Ratio.
3. Investigate blind sensing methods with low computational complexity.
4. Investigate real-time implementation of sensing algorithms to validate the simulation results.

1.4 Thesis contribution

We investigated the noise power estimation and used this for adapting the threshold for sensing. The noise variance is estimated using the unbiased estimate of the variance and an estimator based on linear prediction method. The results demonstrate that it improves the efficiency of Energy detection by giving information for varying threshold based on noise power. Further, this analysis is extended to colored noise environment and cooperative sensing.

The second contribution of this work is the development of a novel SNR estimation method based on the covariance of the received signal. The SNR is estimated by comparing the test statistic, derived from the Sample Covariance Matrix (SCM) of the received signal, with a calibrated signal. The estimated SNR is the value of the SNR that minimizes the difference between the computed and calibrated test statistics. The algorithm demonstrates good performance for positive and negative SNRs. Binary Frequency Shift Keying (BFSK) is used as the waveform for the study.

A time domain fourth order moment based sensing algorithm (M4-Edge) is developed to detect uncorrelated received signals. The simulated results were compared with energy detection and Covariance Absolute Value (CAV). The analysis demonstrates improved performance as compared with CAV at low sample sizes. The proposed algorithm performs well even when the signal is not correlated.

The performance of the algorithm is compared with respect to the probability of detection and SNR wall.

We performed a real-time spectrum sensing using Virtex-6 FPGA. Three algorithms namely energy detection, CAV and M4-Edge are tested in real-time conditions. The experimental results are analysed and compared in terms of processing time, detection time and the probability of detection. The result and analysis presented in this thesis thus provides a set of spectrum sensing algorithms and real-time performance analysis for different environments.

1.5 Thesis organization

Chapter 1: Introduction

This chapter describes the objective of the research work, research contribution and the details about the thesis organization. The need for cognitive radio is explained.

Chapter 2: Spectrum Sensing Techniques and Hardware test setup

Chapter 2 presents the various spectrum sensing methods and details the test setup and the hardware details of the Virtex-6 FPGA board.

Chapter 3: Improved Energy detection using Noise variance estimation

This chapter presents a method of estimating noise power using a linear predictive method. The autocorrelation matrix of the signal is evaluated and the noise power is estimated using the Yule-Walker equations. The Burg method is used for estimating the linear predictive coefficients. An unbiased estimator for the noise power is also computed for comparison. It is shown that the methods improve the detection capability compared to the energy detection with no noise power estimation. The estimator is also used to evaluate improvement in energy detection for cooperating nodes using combining schemes like LLR, EGC and DE.

Chapter 4: SNR estimation for spectrum sensing

This chapter presents a blind (Non-Data-Aided) SNR (Signal-to-Noise Ratio) estimation algorithm for an M-ARY Frequency Shift Keying (FSK) signal in Rayleigh and Rician fading channels with Additive White Gaussian Noise (AWGN). The

SNR is estimated by comparing the test statistic of the received signal with a calibrated signal. The estimated SNR is the value of the SNR that minimizes the difference between the computed and calibrated test statistics. The test statistic of both the received and calibrated signal is calculated using the Sample Covariance Matrix (SCM). The performance of the proposed algorithm is compared with the Partially Data Aided Maximum Likelihood Estimator (PDA MLE).

Chapter 5: Real-time implementation of spectrum sensing algorithms on Virtex-6 FPGA.

This chapter presents the hardware details of the test setup used for this work. The methodology of simulation and signal processing are detailed. A detailed methodology for real-time implementation and evaluation of spectrum sensing algorithms is presented. The real-time performance of two algorithms namely Energy Detection (ED) and CAV (Covariance Absolute Value) is evaluated on a Virtex-6 FPGA platform. BFSK signal corrupted with Additive White Gaussian Noise (AWGN) is used for real-time performance evaluation. The probability of detection, sensing time and resource utilisation are used as the metrics for measuring the efficiency of an algorithm.

Chapter 6: Spectrum sensing with envelope tracking and signal moment

This chapter presents a moment based algorithm namely M4-Edge algorithm spectrum sensing. This algorithm overcomes the limitations of CAV and ED algorithms. The proposed algorithm tracks the envelope of the signal burst, of the primary user, in the time domain. The fourth central moment of the envelope is evaluated and compared with a threshold to detect the rising and falling edges of the burst and hence detects the presence of a signal. Further, the algorithm is implemented on a Xilinx Virtex-6 Field Programmable Gate Array development board for evaluating its real-time performance. In the real-time, the performance of the proposed algorithm is compared with ED and CAV algorithm by considering both BFSK and DVBT signal corrupted by Additive White Gaussian Noise (AWGN) and flat fading. The implementation results are compared on parameters like sensing time and logic blocks utilization.

Chapter 7: Conclusions and Future work

This chapter presents the research contributions of the proposed algorithms which are summarized in the context of the efficiency of spectrum sensing and implementation aspects. It also presents the future scope of the research.

Chapter 2

Spectrum sensing techniques and Hardware test setup

2.1 Objective

The objective of this chapter is to introduce Spectrum Sensing (SS) techniques for cognitive radio. The test setup and the hardware details using the Virtex 6 Field Programmable Gate Array (FPGA) board are detailed..

2.2 Spectrum sensing techniques

In the wireless communication, the receiver model is shown in the Fig 2.1

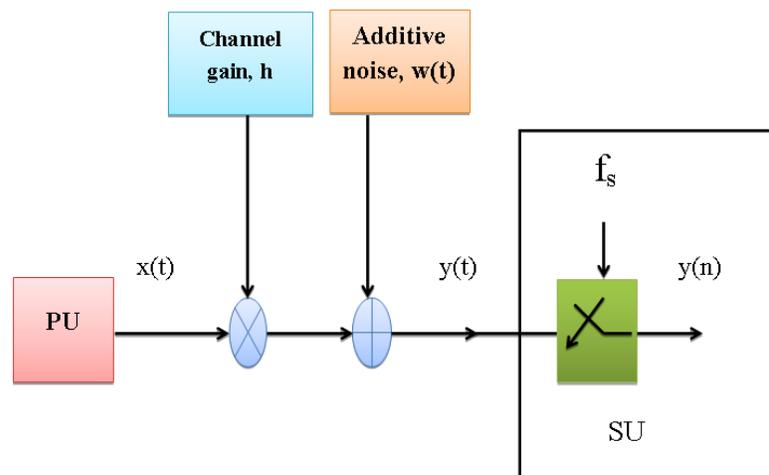


Figure 2.1: The signal model shows the various components of the chain from transmitter to the sampled signal. The channel effect is multiplicative whereas the noise is additive.

In the figure, the received signal $y(t)$ is :

$$y(t) = hx(t) + w(t) \quad (2.1)$$

where, $x(t)$ is the transmitted signal and $w(t)$ is a zero mean white noise with variance σ^2 , h is the channel gain (a random variable with a known or unknown probability density function). The sampled signal is denoted by $y(n)$. Spectrum sensing is the ability to detect the presence of an active PU signal in the sensing band. Mathematically it is nothing but testing a binary hypothesis to identify the presence of PU signal.

In a given time the scanned band is said to be vacant if only noise is detected whereas it is considered to be occupied by PU if a PU signal and noise are detected. Thus mathematically the hypothesis is

$$\begin{aligned} H_0 : y(n) &= w(n) \\ H_1 : y(n) &= x(n) + w(n), \quad n = 1, 2, \dots, N \end{aligned} \quad (2.2)$$

Where the hypotheses H_0 and H_1 indicate the *null hypothesis* (absence of signal) and *true hypothesis* (presence of signal) respectively and $y(n)$ is the received signal. $x(n)$ and $w(n)$ correspond to the primary user signal and noise respectively. The detector output, the test statistic $U(n)$, is then computed and compared with a threshold $\gamma(n)$ to decide the presence or absence of a signal.

$$U(n) \underset{H_0}{\overset{H_1}{\geq}} \gamma(n) \quad (2.3)$$

In the case of sequential sensing the decision rule can be expressed as

$$U(k) \underset{H_0}{\overset{H_1}{\geq}} \gamma \quad (2.4)$$

Where $U(k)$ is the test statistic of the k^{th} frame, and γ is the sensing threshold. The test statistic depends on the detection algorithm, i.e., for energy detection, it is the energy of the received signal [12], for entropy detection it is the entropy of the signal [25].

2.2.1 Performance indicators

The performance of sensing algorithm is measured by three probabilities such as Probability of false alarm (P_f), Probability of detection (P_d) and Probability of mis-detection P_m . P_f measures the probability of declaring the presence of signal

when actually there is no signal. Mathematically,

$$P_f = Pr(U > \lambda | H_0) \quad (2.5)$$

P_d : Probability of detection measures the probability of declaring the presence of signal when actually signal is present.

$$P_d = Pr(U > \lambda | H_1) \quad (2.6)$$

It is always aimed to develop sensing algorithms that have high P_d and low P_f . There exists a number of sensing algorithms to detect the presence of a signal in the channel. These techniques provide opportunities to the secondary users for utilizing the vacant spectrum bands opportunistically. The popular techniques are namely: energy detection, covariance-based detection, Eigen value based detection and matched filter based detection among many others.

2.2.2 Energy detection

Energy detection is the simplest spectrum sensing technique which can blindly detect any signal. The earliest work on Energy detection (ED) treats it as a binary hypothesis problem and derives ROC (Receiver Operating Curve) for various time-bandwidth products [12]. Closed form equations were given for energy detection over fading channels [26],[27]. Energy detection technique has the lowest complexity, but it is sensitive to noise variance and signal power [28]. In addition, ED suffers from a limitation called *SNR Wall*. It is defined as the *SNR* up to which the algorithm can detect the presence of the primary user with the desired P_d , which cannot be improved by increasing the number of samples [29].

The decision statistic for the energy detector is expressed as the frame energy as:

$$U(k) = \frac{1}{N_s} \sum_{n=1}^{N_s} |y(n)|^2 \quad (2.7)$$

Where, $U(k)$ is the energy of the k^{th} frame, k and n are the frame and sample indices respectively and N_s is the number of samples per frame. It provides information about the signal at k^{th} frame for a particular band and is used for detecting PU signal by testing the binary hypothesis. For low *SNR* conditions, $N_s = \mathcal{O}(SNR^{-2})$. Thus if the noise statistics are known, then by using a larger

sensing time signals can be detected at low SNR. Intuitively a longer sensing time is indicated in low SNR conditions. It implies that if N_s is large, then as per the Central Limit Theorem, the test statistic can be approximated as a Gaussian [30]:

$$T(\mathbf{U}) | H_0 : \mathcal{N}(\sigma^2, 2\sigma^4/N_s) \quad (2.8)$$

$$T(\mathbf{U}) | H_1 : \mathcal{N}(P + \sigma^2, 2(P + \sigma^2)^2/N_s) \quad (2.9)$$

Where, P , σ^2 , H_0 and H_1 represents the signal power, noise variance and null and true hypothesis respectively. \mathcal{N} denotes the Normal distribution. In the case of Energy detection, the threshold γ can be set as [30]:

$$P_f = Q\left(\frac{\gamma - \sigma^2}{\sqrt{(2/N_s)\sigma^2}}\right) \quad (2.10)$$

$$P_d = Q\left(\frac{\gamma - (P + \sigma^2)}{\sqrt{(2/N_s)(P + \sigma^2)}}\right) \quad (2.11)$$

$$\gamma = \sigma^2(Q^{-1}(P_f)\sqrt{(2/N_s)} - 1) \quad (2.12)$$

Where, P_d is the probability of detection, P_f is the desired probability of false alarm, $Q(\cdot)$ is the Gaussian tail probability function.

Energy detection is computationally simple and assumes no information of the signal characteristics. However, it suffers from the ‘SNR Wall’ limitation, which says that the detection performance cannot be increased beyond a certain limit even if the number of samples is increased. It is because the detection threshold fixing becomes difficult when the noise variance is comparable to the signal energy. Other problems with energy detection are that it cannot differentiate between interference and PU signal, and it performs poorly at low SNR [31] and for spread spectrum signals [32].

2.2.3 Covariance based sensing

The Covariance Absolute Value (CAV) algorithm is based on the concept that the covariance matrix of an un-correlated signal is a diagonal matrix [16],[33]. Under the condition of a large number of received samples, the Sample Covariance Matrix (SCM) of the received signal $y(n)$ is an approximation of the covariance matrix

$R_y(k)$:

$$R_y(k) = \begin{pmatrix} \lambda(0) & \lambda(1) & \cdots & \lambda(L-1) \\ \vdots & \ddots & \cdots & \vdots \\ \lambda(L-1) & \lambda(L-2) & \cdots & \lambda(0) \end{pmatrix} \quad (2.13)$$

where,

$$\lambda(l) = \frac{1}{N_s} \sum_{n=0}^{N_s-1} y(n)y(n-l); l = 0, 1..L-1 \quad (2.14)$$

$$T_1(k) = \frac{1}{L} \sum_{a=1}^L \sum_{b=1}^L |r_{ab}| \quad (2.15)$$

$$T_2(k) = \frac{1}{L} \sum_{a=1}^L |r_{aa}| \quad (2.16)$$

$$\gamma(k) = T_1(k)/T_2(k) \quad (2.17)$$

Where, $R_y(k)$, N_s and $\lambda(l)$ corresponds to the sample covariance matrix, the number of samples per frame and the sample autocorrelation respectively. The SCM is symmetric and Toeplitz. r_{aa} and r_{ab} are the elements of $R_y(k)$, k is the frame index and L is a smoothing constant. In the scanned band, $\gamma(k) > 1$ indicates the presence of a signal; else there is no signal. For large N_s , $\lambda(l)$, $T_1(k)$ and $T_2(k)$ can be approximated to a Gaussian distribution for which P_f , P_d , and γ_1 can be derived as [16]:

$$P_f = 1 - Q\left(\frac{\left(\frac{1}{\gamma_1}\right)(1 + (L-1)\sqrt{\frac{2}{N_s\pi}}) - 1}{\sqrt{\frac{2}{N_s}}}\right) \quad (2.18)$$

$$P_d = 1 - Q\left(\frac{\frac{1}{\gamma_1} + \frac{\gamma_L SNR}{\gamma_1(SNR+1)} - 1}{\sqrt{\frac{2}{N_s}}}\right) \quad (2.19)$$

$$\text{where, } \gamma_1 = \frac{1 + (L-1)\sqrt{\frac{2}{N_s\pi}}}{1 - Q^{-1}(P_f)\sqrt{\frac{2}{N_s}}} \quad (2.20)$$

The CAV algorithm is a blind method as it does not require any information of the signal constellation. It removes the need for synchronisation to the incoming signal which is always difficult for a time-varying channel. One disadvantage of

this method is that it requires a large number of samples at lower SNR. The number of samples required is given as [16]

$$N_c = 2 * \left(\frac{Q^{-1}(P_f) - Q^{-1}(P_d) + (L - 1)(\sqrt{\pi})}{\gamma_L SNR} \right)^2 \quad (2.21)$$

For example, with $P_f=0.1$, $L=32$, $\gamma_L=1$, and $SNR=-20$ dB the number of samples required is approximately $3.3 * 10^7$ samples. It can be difficult to obtain in a real time environment. If the number of samples are limited then the threshold as given by (2.20) increases from the ideal value and the detection performance reduces significantly.

2.2.4 Eigenvalue based sensing

This technique is based on the concept that the largest eigenvalue of the Sample Covariance Matrix corresponds to the signal and the smallest eigenvalue corresponds to noise [34],[35]. in the case of Eigen based sensing, two test statistics, T_1 and T_2 are evaluated as

$$T_1 = \frac{\lambda_{max}}{\lambda_{min}} \geq \gamma_1 \quad (2.22)$$

$$T_2 = \frac{\varepsilon}{\lambda_{min}} \geq \gamma_2 \quad (2.23)$$

$$\text{where } \gamma_1 = \frac{(\sqrt{N_s} + \sqrt{ML})^2}{(\sqrt{N_s} - \sqrt{ML})^2} \left(1 + \frac{(\sqrt{N_s} + \sqrt{ML})^{-2/3}}{(N_s ML)^{1/6}} F_1^{-1}(1 - P_f) \right) \quad (2.24)$$

$$\text{and } \gamma_2 = \left(\sqrt{\frac{2}{MN_s}} Q^{-1}(P_f) + 1 \right) \frac{N_s}{(\sqrt{N_s} - \sqrt{ML})^2} \quad (2.25)$$

$$(2.26)$$

Where, ε is the signal energy as (3.2), λ_{max} and λ_{min} are the largest and smallest eigenvalues of the Sample Covariance Matrix (SCM), M is the oversampling factor, F_1^{-1} is the Cumulative Distribution Function of the Tracy-Widom Distribution of Order 1, P_f is the probability of false alarm.

Signal detection is declared if either of (2.22) or (2.23) is satisfied. If (2.22) is used for detection it is called the Max-Min Eigenvalue (MME) technique and if (2.23) is used then it is called the Energy with Minimum Eigenvalue (EME) technique. This is a blind technique as it is based on a test statistic derived from the signal and no apriori information about the signal characteristics is required. Eigen

based methods perform well with noise uncertainty. If the signal is correlated, then the performance is better than energy detection. This method does not require synchronisation making it a blind method. When the sample size is small it is difficult to detect closely spaced signals below a certain threshold, which is a function of noise variance [36]. It is a computationally complex method as the covariance matrix is decomposed and then the eigen values are computed. Iterative algorithms to reduce the complexity are proposed in [37].

2.2.5 Cyclostationarity based sensing

The transmitted waveforms have cyclical spectral components which can be used to detect signals. Cyclostationary features refer to the Fourier Transform of the Autocorrelation of the signal. Each signal has a unique set of spectral features which are used for checking the spectrum occupancy. If the autocorrelation of the signal $R_y(t, \tau)$ is periodic as

$$R_y(t, \tau) = R_y(t + T, \tau) \quad (2.27)$$

where t is the time variable, τ is the autocorrelation lag, then the periodic autocorrelation function in terms of Fourier series is given as [17],[14],[38],[39]:

$$R_y(t, \tau) = \sum_{\alpha=-\infty}^{\infty} R_y^\alpha(\tau) \exp(j2\pi\alpha t) \quad (2.28)$$

$$\text{where } R_y^\alpha(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_T y(t + \tau/2) y^*(t - \tau/2) \exp(-j2\pi\alpha t) dt \quad (2.29)$$

The Fourier transform of $R_x^\alpha(\tau)$ can be expressed as

$$S_y^\alpha(f) = \int_{-\infty}^{\infty} R_y^\alpha(\tau) \exp(-j2\pi f\tau) d\tau \quad (2.30)$$

$$S_y^\alpha(f) = \lim_{\Delta t \rightarrow \infty} \lim_{T \rightarrow \infty} \int_{-\Delta t/2}^{\Delta t/2} Y_T(t, f + \alpha/2) Y_T^*(t, f - \alpha/2) dt \quad (2.31)$$

$S_y^\alpha(f)$ is called the Cyclostationary Spectral Density (CSD) function. If α is zero then (2.31) reduces to the Power Spectral Density (PSD). Noise contributes only to the PSD and does not create any distinct peak in the CSD, whereas the CSD peaks for distinct periodic α of the signal [40]. Signal detection at low SNR is

possible by comparing the CSD value for the signal to a CSD value computed for pure noise. The strength of CSD can be affected by fading channel and frequency offset, due to oscillators or Doppler, and hence it requires synchronisation. Thus in summary, Cyclostationary detection is not a blind technique, and it requires high sample rate and thus is computationally complex. Complexity reduction is proposed using a maximum cyclic autocorrelation selection (MCAS) in [41]. When the CR is mobile the Doppler frequency shift introduces a Cyclic frequency offset, which is estimated using a complex exponential basis model [42].

2.2.6 Matched Filter based sensing

Matched filtering maximizes the SNR at the detector [13]. If $y(t)$ is the transmitted signal, defined over $0 \leq t \leq T$, then the matched filter that maximizes the signal to noise ratio at the output of the filter is :

$$h(t) = \begin{cases} y(T-t); & 0 \leq t \leq T \\ 0 & ; \text{elsewhere} \end{cases} \quad (2.32)$$

Thus a matched filter has an impulse response that is a reversed and time-shifted version of the input signal. The optimum method for signal detection, but the details of the transmitted signal must be fully known. It achieves detection in a short time to achieve a given Probability of False alarm or miss detection at low SNRs [13]. The matched filter is a coherent detector that requires the prior knowledge at the PU signals. The sensing method is the optimal sensing method and it maximizes the SNR at the output of the detector. However, the disadvantage with this method is that it requires prior information about the PU signal at the SU receiver. It is the advantage with this method is that it achieves detection in a short time at a low SNR. The test statistics for matched filter detector is

$$U(t) = \sum_0^T y(t)y(T-t) \quad (2.33)$$

Where, $U(t)$ is the output of the matched filter, $y(t)$ is the input PU signal. The test statistic is then compared with a threshold to decide the presence of a signal. In this method, the Probability of detection and misdetection can be calculated

using Neyman Pearson criteria as

$$P_d = Q\left(\frac{\lambda - (U + \sigma_w^2)}{\sqrt{\frac{2}{N}(U + \sigma_w^2)}}\right) \quad (2.34)$$

$$P_f = Q\left(\frac{\lambda}{\sqrt{\frac{2}{N}\sigma_w^2}}\right) \quad (2.35)$$

Where λ , U and σ_w^2 are threshold, average PU signal energy and noise variance respectively. The sensing threshold λ can be expressed as

$$\lambda = Q^{-1}(P_f)\sqrt{\frac{2}{N}\sigma_w^2} \quad (2.36)$$

Apart from these algorithms, Moment based sensing algorithms were reported in the literature for spectrum sensing [43],[44]. In [28] noise variance was estimated using the optimal moment pair for improving the performance of energy detection technique. The estimators were derived for BPSK and QAM constellations. In [45] a fourth order detector was derived for detection of linearly modulated signals.

2.2.7 Cooperative sensing

In the radio environment, when CR nodes are located far away from the PU or due to deep fade or shadowing the PU signal strength received at the CR may be very low for robust detection. In such conditions, it is possible that the CR may assume that the PU is not present and start transmission. It is called as the *HiddenNode* problem. Multipath fading can create deep nulls at a particular location due to destructive interference of the EM waves. However, a point even at a small distance away may not experience such a severe fade as the path lengths are different. For example, the wavelength of a signal at 800 MHz is 37.5 cm. The signal phase will be 180 degrees out of phase at a distance of 18.75 cm and the fade would be significantly different. This problem can be avoided by the combined cooperative decision derived from spatially collected observations from spatially located CR users. With this the overall detection performance increases. Thus, cooperative sensing is the alternative sensing approach to nullify the effect of shadowing, fading and multipath issues [23].

In cooperative sensing, each spatially located CR share the sensing informa-

tion to a fusion center (FC), wherein the global decision about the presence of PU is taken. Depending on the way each CR share information, cooperative sensing is classified into three categories: Centralized [46], distributed [47],[48] and relay assisted [49]. In a centralized cooperative sensing scheme, each CR users in the network share their local sensing information to an identified fusion center through a control channel. The fusion center takes the final decision about the presence of PU by combining all the local sensing information. The FC transmits this to all SU in the network. In a distributed cooperative sensing scheme, each CR users communicate among themselves and converge to a final decision about the presence of PU in the scanned band unlike a FC in case of the centralized scheme [50]. The relay assisted sensing scheme uses the sensing channel and control channel dynamically depending on its characteristics and perform sensing like multihop cooperative sensing. In the current thesis, centralized cooperative sensing is used.

Cooperative spectrum sensing [51],[52],[21] results in improved sensing time as well as improved accuracy. In cooperative sensing as more nodes are involved the detection is more robust. Therefore the periodicity at which individual nodes sense the spectrum can be increased. Secondly, by cooperating with other nodes, the PU's location can be determined. However, there are a few problems with cooperative sensing like increased computational overheads, an additional control channel for inter-CR communication and malicious attacks or spoofing to bring down the network. In the cooperative sensing, multiple radios, spatially located, are utilized for deciding the presence of a signal. Each radio individually uses signal processing algorithms mentioned as above subsections and report the local sensing information to a fusion center/radio. The fusion center uses a fusion algorithm to combine the data and take the final cooperative decision. The detail about cooperative sensing can be found in [21]. This is illustrated in Fig 2.2. Various Fusion strategies were proposed for robust cooperative sensing

1. Spatio-temporal Fusion

In this case, the CBS(Cognitive Base Station) makes a decision every T seconds based on reports from the CR nodes either in Synchronised or Non-Synchronised form. In the case of synchronized reporting, the Cognitive Base Station (CBS) makes a decision every T seconds. The reports from the CR nodes arrive at different times based on their distance from the CBS and their processing times. The CR nodes need to ensure that the decisions

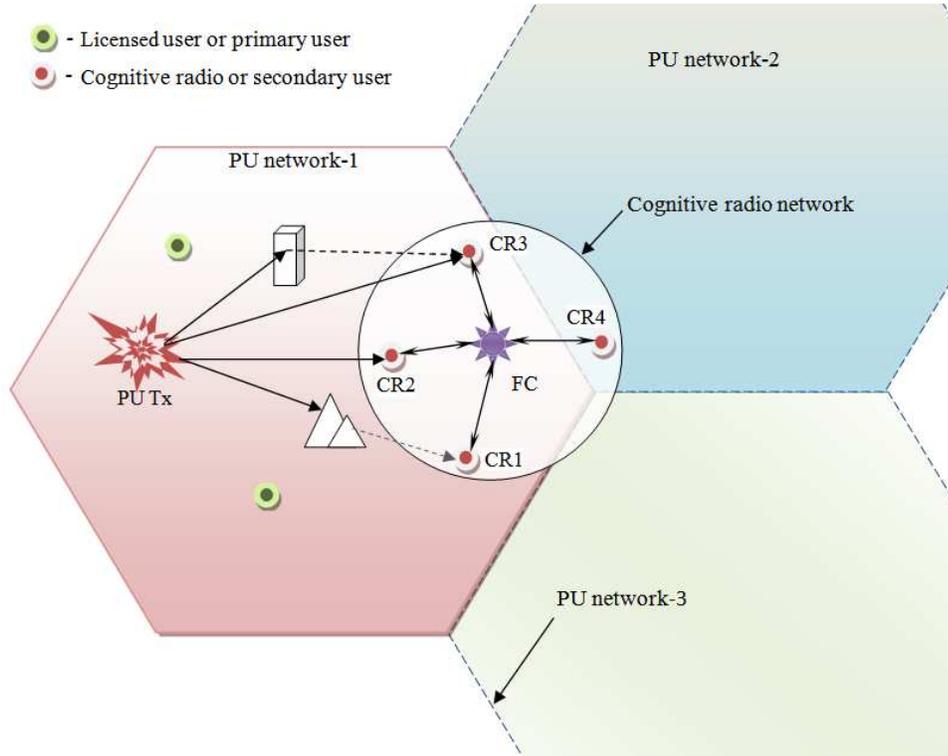


Figure 2.2: Cooperative sensing scenario [53]

are conveyed within the time slot and therefore have to ensure that the time for sensing, processing, and transmission are well within limits. It is an unnecessary overhead.

In the case of non-synchronised reporting, the CR nodes send in their decision as soon as it is available. This results in varying number of decisions reaching the CBS in T seconds.

2. Hard decision fusion

In this case, the individual nodes decide on the techniques and strategies for deciding if the PU is present. The node makes a binary hypothesis decision and sends a value of 1 or 0 to the fusion center. At the fusion center the fusion can be carried out in two ways: Chair-Varshney log-likelihood ratio test or M-out-of-N voting.

a) Chair-Varshney Fusion

This is based on the log-likelihood ratio test. The CR node sends in addition to the binary decision, Probability of False alarm P_f and Probability of Mis-detection P_m . In addition the apriori probabilities of the events H_0 and H_1

given by $P(H_0)$ and $P(H_1)$ respectively. The Chair-Varshney criterion for fusion decision, d_k is [54]:

$$d_k = \begin{cases} 1 & ; \mu \geq 0 \\ -1 & ; \mu < 0 \end{cases} \quad (2.37)$$

$$\text{where } \mu = a_0 + \sum_{i=1}^N a_k(i)u_k(i) \quad (2.38)$$

$$a_k(i) = \log \left(\frac{1 - P_m(i)}{P_f(i)} \right); \text{ if } u_k = 1; \quad (2.39)$$

$$a_k(i) = \log \left(\frac{1 - P_f(i)}{P_m(i)} \right); \text{ if } u_k = -1; \quad (2.40)$$

Alternately $a_0, a_k(i)$ can be computed at the CR node and then transmitted to the fusion center. However it is a challenging task to compute P_f and P_m at the CR node, as it requires reference data in the transmission.

b) K-out-of-M Fusion

Another fusion rule is based on voting. If K out of M number of CR nodes report H_1 then decide on H_1 . If $u_k(i)$ are the decisions reported by the CRs then the fusion decision d_k can be evaluated as:

$$d_k = \begin{cases} 1 & ; \mu \geq K \\ -1 & ; \mu < 0 \end{cases} \quad (2.41)$$

$$\text{where } \mu = \sum_{i=1}^M 0.5(u_k(i) + 1) \quad (2.42)$$

$$P_m = \sum_{j=0}^{K-1} \frac{M!}{(M-j)!j!} P_d^j (1 - P_d)^{M-j} \quad (2.43)$$

$$P_f = 1 - \sum_{j=0}^{K-1} \frac{M!}{(M-j)!j!} P_f^j (1 - P_f)^{M-j} \quad (2.44)$$

where M is the number of CR nodes and P_d, P_m, P_f are the probabilities of detection, mis-detection and false alarm. If we let $K=1$ in (2.41), then the decision is equivalent to the *OR* rule and the *AND* rule can be implemented by letting $K = M$. Majority voting can be implemented by letting $K = M/2$. The *OR* rule minimises P_m but P_f could be higher whereas the reverse is true for the *AND* rule. The majority voting rule is a compromise between

the two.

3. Soft Decision Fusion

In this fusion method [55], the test statistics are transmitted to the fusion center and the fusion center takes the cooperative decision. Assume that the CR nodes send their energy test statistic to the fusion center. The test statistic at the fusion center is

$$\hat{\xi} = \sum_{i=1}^M w_i \xi_i = \mathbf{W}\boldsymbol{\Upsilon}^T \quad (2.45)$$

where w_i are the weights assigned to the i^{th} CR and ξ is the fused test statistic and ξ_i are the individual test statistic received from the CR nodes, $W = [w_1, w_2 \dots w_M]$ and $\boldsymbol{\Upsilon} = [\xi_1, \xi_2 \dots \xi_M]$. Here the fusion center is free to use any decision rule: either Likelihood ratio test(LLR) or Equal Gain Combining(EGC). Performance comparison of hard and soft decision fusion schemes is studied for a Rayleigh channel for different network parameters [56].

2.3 Hardware implementation

In literature, although many sensing algorithms were reported, its hardware implementation and real-time study are limited. A working prototype for experimenting with different spectrum sensing techniques is demonstrated [21]. A Berkeley Emulation Engine BEE2 testbed with 18 radio cores is developed for configuring multiple CR users in 2.4 GHz ISM band [21]. An efficient gradient-based wideband sensing is demonstrated using USRP(Universal Software Radio Processor) boards in real time. Hardware implementation of Energy detector on a wireless testbed is reported in [57] wherein sensing time was varied to achieve a predetermined P_d at low SNR regime. Few studies have focussed on sensing time for a given modulated signal of finite duration.

2.3.1 Hardware details

FPGA (Field Programmable Gate Array) have high hardware density and operating speeds making them ideal for implementation of spectrum sensing algorithms.

The ML 605 FPGA board from Xilinx is a versatile board featuring the Virtex-6, XC6VLX240T-1FFG1156, FPGA. The detailed features of the ML605 FPGA Board are listed in [58]. The board has a USB and JTAG interface for downloading code and debugging. The code is downloaded into the 32 MB flash. The board also has a tri-speed 10/100/1000 Mbps ethernet. It ensures a high speed link for data transfer with the host computer. The board has 512 MB DDR3 RAM and 8Kb I2C EEPROM for storing program data. The board has a 200 MHz oscillator.

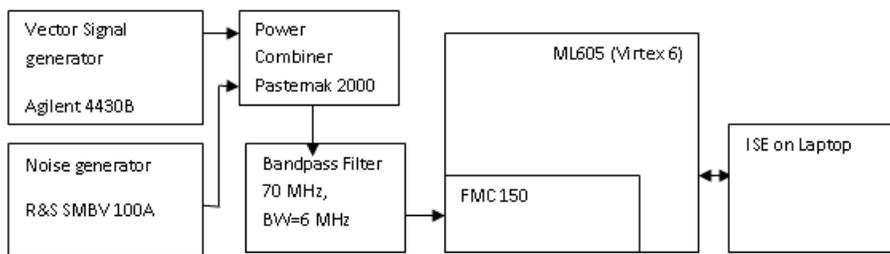


Figure 2.3: The test setup for algorithm evaluation including instruments and ML605 Xilinx FPGA board, ADC/DAC board FMC 150 and Xilinx ISE running on laptop.

2.3.2 Instrumentation and Hardware test setup

The test setup for the real time evaluation is shown in Fig 6.4. The Vector signal generator, Agilent 4430B, is capable of generating frequencies from 250 KHz to 1 GHz with a resolution of 0.01 Hz. It is capable of generating both analog and digital modulations either from internal or external baseband inputs. The generator is capable of generating AM, FM and Phase modulations from its internal source with an amplitude accuracy of ± 0.2 dB. Modulated signal bandwidths more than 10 MHz can be generated. The signal generator is programmed to generate BFSK signal from internal source with a data rate of 1 MHz. The signal generator has a 'Symbol Sync' clock output on the rear panel which is used for symbol synchronisation. This clock is used as the *gating clock* for the BFSK output and is fed in parallel to the Tektronix oscilloscope. The output from the FPGA board is the 'detected symbol' clock which is fed to the other port of the oscilloscope. The noise generator, Rhode and Schwarz SMBV 100A, can generate additive white Gaussian noise separately for I and Q channels. The Carrier-to-Noise Ratio (CNR) can be

adjusted from -30 to + 30 dB in steps of 0.1 dB. The generator is capable of noise bandwidth from 1 KHz to 120 MHz. The signal and noise are combined using a 3 dB power combiner which can operate from 2 to 500 MHz with an insertion loss of 1 dB and an amplitude balance of 0.3 dB [59]. It provides inter-port isolation of 25 dB thus avoiding any interaction between the signal and noise generators. In our test setup, the SMBV generator is used as a white noise source whose power is varied to create different SNR at the output of the power combiner. The real-time test is conducted at a center frequency of 70 MHz. Accordingly, a bandpass filter is introduced after the power combiner to allow only the frequency of interest to pass to the signal acquisition board. The filter is a bandpass filter with center frequency of 70 MHz and a \pm -3 dB power bandwidth of \pm 3 MHz. The insertion loss is -5.17 dB. The signal is then fed to the ADC/DAC FM150 [60] which is a FPGA Mezzanine card with two 14-bit Analog-to-Digital convertor (ADC) and two 16-bit Digital-to-Analog convertor channels capable of 250 and 800 Mega samples per second (Msps). The signal is coupled using a coaxial cable to the MMCX coaxial connector on the front panel of the FM150. The Virtex-6 FPGA on the ML 605 reads the digitized signal . The binary file of the spectrum sensing algorithm is downloaded from the laptop which runs the Xilinx ISE IDE. The algorithm generates a decision about the presence of the signal. This is indicated by changing the state of a discrete output to 1 from 0 which is available on a SMB connector port connected using a coaxial cable to the second channel of the oscilloscope. Thus on the oscilloscope, the *input signal burst* and the *detected signal burst* can be observed.

**This page is intentionally left
blank**

Chapter 3

Improved energy detection using noise variance estimation

3.1 Objective

Energy detection is a computationally simple spectrum sensing algorithm for Cognitive Radio. However, it suffers from a phenomenon called *SNR Wall*, wherein the energy detector cannot detect a signal if its power is less than the uncertainty of the noise power, irrespective of sample size. However, if the noise power can be estimated accurately, the detection probability can be improved by using adaptive threshold technique. The objective of this chapter is to use two noise variance estimators namely: unbiased and Autoregressive Linear Predictor(LP) for adapting the threshold to improve the performance of energy detection. The estimators are evaluated for three cases of additive noise : white, blue and red.

3.2 Introduction

The earliest work on energy detection treats it as a binary hypothesis problem and derives Receiver Operating Characteristic (ROC) for various time-bandwidth products [12]. Closed form equations were given for energy detection over fading channels in [26],[27]. The Energy Detection technique has the lowest complexity but is sensitive to noise variance and signal power [28]. The performance of Energy detection technique deteriorates, if the signal power is less than the noise power uncertainty. It is defined as the *SNR wall*, i.e. the *SNR* up to which the algorithm can detect the presence of the primary user with the desired P_d , and the P_d cannot be improved upon by increasing the number of samples. However, the performance of energy detection can be improved if the noise variance is known a priori [29]. Noise variance estimation has been investigated in [61],[62],[63]. In [61] the noise variance is estimated on the basis of the shortest half sample method.

The scatter of the signal is used as a measure of the noise variance. The signal is sectioned into several groups of sub-samples of varying sizes to estimate the mean of the shortest half sample and the least median of its amplitude squared. This method works for a signal with outliers up to 50% of the samples. However, it does not work well for small sample size and if the outliers are densely distributed as compared to noise-only samples. In [62] the spectrum sensing error is sought to be minimized by estimating the noise variance. In this method, the energy in the k^{th} sub-band of the polyphase Digital Filter Bank (DFB) is computed. The optimum energy detection threshold is formulated as a constrained optimisation problem using the Lagrangian multiplier method. The threshold, estimated using an Autoregressive (AR) model, is updated using the gradient based technique. The spectrum sensing error for two threshold values is detailed for different sensing error weights. However, there are no details provided for the probability of detection at various SNRs or the Receiver Operating Characteristic (ROC) which are critical performance indicators for cognitive radio applications.

Thus there is a need to develop algorithms for improving the performance of energy detection. Also, as the real-time channel may not encounter white noise only, there is a need for studying the performance of energy detection in colored noise environment.

In this chapter, we propose a threshold adaptation scheme based on the estimated noise to improve the detection performance. The noise variance is estimated using a 2-pole autoregressive filter. The filter parameters are related to the autocorrelation sequence by the Yule-Walker equations. This matrix is Toeplitz, a matrix in which each element of the main diagonal is constant, which is efficiently inverted using the Levinson-Durbin recursion. The noise variance is then computed using these filter parameters. The performance improvement is investigated for three different noise types i.e., white, blue and red. Finally, the detection performance with and without noise estimation are compared. The detection performance with different weight estimation techniques are compared with and without noise estimation. Signal node sensing algorithm encounters hidden node issue and cooperative sensing is preferable in such an environment, at the cost of increased algorithm and hardware complexity.

In this thesis the single node energy detection using adaptive threshold technique is extended to cooperative sensing for enhancing the probability of detection. In the Centralized Spectrum Sensing (CSS), each node measures the energy locally

and transmits it to a fusion centre. Fusion center aggregates the weighted energy and compares it with a threshold to find the status of the channel. The weights associated with each node plays a critical role affect the detection performance. There exist different methods to find weight values for each node. In this thesis, we use a heuristic method namely differential evolution algorithm to evaluate the weight value associated with each node to maximize the probability of detection.

3.3 System model

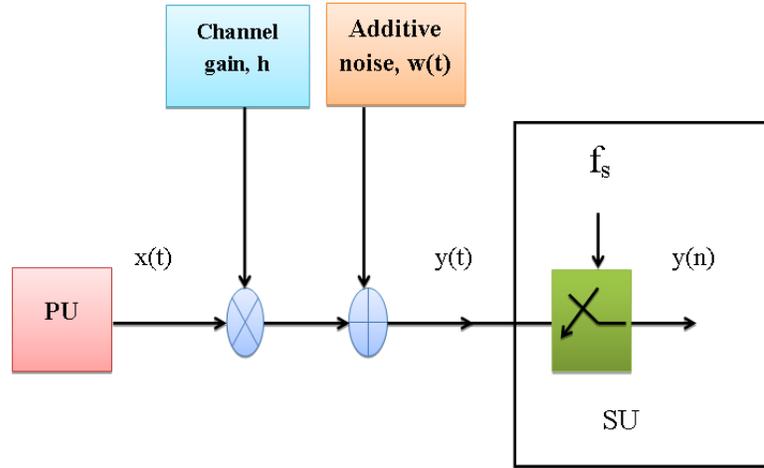


Figure 3.1: The signal model shows the various components of the chain from PU transmitter to the SU receiver. The channel effect is multiplicative whereas the noise is additive.

In wireless communication, the receiver model is shown in the Fig 3.1. Consider a Binary Frequency Shifted Keyed (BFSK) signal as the received signal $y(t)$:

$$y(t) = hx(t) + w(t) \quad (3.1)$$

where $x(t)$ is the transmitted signal and $w(t)$ is a zero mean white noise with variance σ^2 , h is the channel gain (a random variable with a known or unknown probability density function). The sampled signal is denoted by $y(n)$. The channel is assumed to be flat fading. It is a channel in which the signal has a single path from source to the receiver and the envelope of the signal fade follows the Rayleigh distribution. It is called a flat fading channel.

3.4 Single node spectrum sensing using improved Energy detection technique

In the case of single node spectrum sensing, a CR user is sensing the spectrum of interest. In the Energy detection technique, the frame energy is computed as :

$$U(k) = \frac{1}{N_s} \sum_{n=1}^{N_s} |y(n)|^2 \quad (3.2)$$

where $U(k)$ is the energy for the k^{th} frame, k and n are the frame and sample indices respectively and N_s is the number of samples per frame. For low SNR conditions, $N_s = \mathcal{O}(1/SNR^2)$. In low SNR conditions intuitively a longer sensing time is indicated. This implies N_s is large, the test statistic can be approximated as a Gaussian, as per the Central Limit Theorem [29]. The presence of active PU is decided using the test :

$$U(k) \geq \lambda \quad (3.3)$$

The threshold γ can be set as [29]:

$$\gamma = \sigma^2(Q^{-1}(P_f)\sqrt{(2/N_s)} - 1) \quad (3.4)$$

where P_f is the desired probability of false alarm, $Q(\cdot)$ is the Gaussian tail probability function, σ^2 is the noise power. The main advantage of the energy detector is that it works irrespective of the signal. However, it depends on an assumption that the noise power is fully known. In the real radio environment, this is not the case. Detection at low SNR can be improved by increasing the observation time i.e., the increase the number of samples. However, it was demonstrated that there is a phenomenon called *SNR Wall* which prevents detection of the signal at arbitrarily low SNR even if the sample size is increased [29]. If σ^2 is the noise power and the range of noise powers lie in the interval $(\frac{\sigma^2}{\rho}, \rho\sigma^2)$, then

$$SNR_{Wall} = \rho^2 - 1/\rho \quad (3.5)$$

where $0 < \rho < 1$. If the noise power can be estimated from the received signal then detection at lower SNR is possible. Methods of estimating noise power are detailed in the following sections.

3.5 Noise variance estimator

3.5.1 Auto regressive variance estimator

Non-Parametric methods of power spectrum estimation are computed using the FFT (Fast Fourier Transform) algorithm. However, long data series are required for good frequency resolution. Parametric methods are based on constructing a model for the generation of the signal with model parameters that can be estimated from the input data series $y(n)$ [64]. The signal model is assumed as per (3.1) with $h=1$ for a channel without fade. The linear filter $H(z)$ is an *all-pole* filter, which is characterised by its p coefficients a_1, a_2, \dots, a_p . The parametric model represented by (3.6) is called Auto Regressive (AR) model of order p . This model is also called as the All-Pole model and is well suited for modeling spectra with peaks. In our case, the BFSK spectrum has two sharply defined peaks, and therefore the AR model is well suited to model this waveform.

$$H(z) = 1/A(z) = 1 / \left\{ 1 + \sum_{i=1}^p a_i z^{-i} \right\} \quad (3.6)$$

Where, a_i satisfy the following Yule-Walker equations [65]:

$$\sum_{k=1}^p a_k R_x(|i-k|) = R_x(i); \quad i > 0 \quad (3.7)$$

where R_x, R_y are the autocorrelation coefficients of $x(n)$ and $y(n)$ respectively and the relationship between them is as :

$$R_x(0) = R_y(0) - \sigma_w^2 \quad (3.8)$$

$$R_x(i) = R_y(i); \quad |i| > 0; \quad (3.9)$$

The noise variance $\hat{\sigma}_w^2$ is estimated as:

$$\hat{\sigma}_w^2 = \sum_{i=1}^p a_i \left\{ \hat{R}_y(i) + \sum_{k=1}^p a_k \hat{R}_y(|i-k|) \right\} / \sum_{i=1}^p a_i^2 \quad (3.10)$$

The AR coefficients, a_i are obtained by inverting the correlation matrix using the Levinson-Durbin algorithm. Alternately they can be obtained using the Burg

method [66] which is based on the minimisation of backward and forward errors in linear predictors, such that the Levinson-Durbin recursion is satisfied. The estimator given by (3.10) is labeled as LP estimator.

3.5.2 Threshold adaptation

The detection threshold is adapted as :

$$\hat{\gamma}(k) = \gamma(k)\hat{\sigma}_w^2(k) \quad (3.11)$$

where, $\hat{\sigma}_w^2$ is the estimated noise power, $\hat{\gamma}(k)$ and $\gamma(k)$ are the adapted and original threshold respectively [67]

3.6 Unbiased estimator

The sample variance $\hat{\sigma}^2$ is the second sample central moment. It is an unbiased estimator for the population variance and is given as:

$$\hat{\sigma}^2 = \frac{1}{N_s - 1} \sum_{i=1}^{N_s} (y_i - \mu)^2 \quad (3.12)$$

where N_s is the number of samples per frame and μ is the sample mean.

3.7 Colored noise

In general, white noise is assumed for modeling additive noise in the signal. It has a flat spectrum across all frequencies. Pure white noise is not encountered in real channels, as it would need to have infinite power to span infinite bandwidth. Practically, white noise is bandlimited over B Hz and is given by

$$S_w(f) = \sigma^2; f \leq |B| = 0; otherwise \quad (3.13)$$

However, in many cases, the additive noise is not white only. The additive received noise could have been added due to either thermal or interference noise. In general, these are called 'colored noise'. Thermal noise, also known as Johnson noise, is generated by the random motion of electrons. The spectral density of

thermal noise is given as

$$S_w(f) = \frac{kT}{2} \quad W/Hz \quad (3.14)$$

where k is the Boltzmann constant, $k = 1.38 \times 10^{-23}$ Joules per degree Kelvin, T is the absolute temperature in degree Kelvin. As this is only a function of temperature, its spectrum is flat and can be called as 'white noise'. Three types of colored noise are frequently encountered in literature: blue, pink, brown or red noise. Blue noise has a power spectral density which is proportional to frequency. It increases at the rate of 3 dB/octave, i.e. 10 dB/decade. This kind of noise is encountered when the thermal noise is induced into the MOSFET gate. The induced noise increases with frequency. *Pink* noise is also called flicker noise or $1 \setminus f$ noise. The frequency spectrum of flicker noise has the same power in frequency bands that are proportionally wide (logarithmic spacing). It also has a Gaussian PDF but its PSD is proportional to $1 \setminus f$. It is an important form of noise, in RF oscillators and it influences the transmitted waveform. *Brown* or *Red* noise has a spectral density that has a power density that decreases at 6 dB per octave with increasing frequency (density proportional $1 \setminus f^2$) over the frequency range. It can be generated by simulating Brownian motion and has an underlying Gaussian PDF but the PSD falls off at $1 \setminus f^2$ rate. The power spectral density can be expressed as :

$$S_w(f) = \frac{k}{f^\alpha}; \quad (3.15)$$

$$\alpha = 1 : \textit{Pink noise}; \quad (3.16)$$

$$= 2 : \textit{Red noise}; \quad (3.17)$$

where k is a constant and $\alpha = 1, 2$.

3.8 Cooperative sensing using improved energy detection

In the cooperative sensing [51],[52],[23],[21], multiple radios are utilized for taking the decision about the presence of signal. Each radio individually uses signal processing algorithms detailed in sections 3.4-3.6 and report to a fusion centre/radio. The fusion centre uses a fusion algorithm to combine the data and take a decision.

The hypothesis tests for co-operative sensing, with M nodes in co-operation is defined as :

$$\begin{aligned} H_0 : y_m(n) &= w_m(n) \quad , \quad m = 0, 1, \dots, M - 1 \\ H_1 : y_m(n) &= h_m x_m(n) + w_m(n), \quad n = 0, 1, \dots, N - 1 \end{aligned} \quad (3.18)$$

Where, h_m is the channel gain of m^{th} node.

A pictorial representation of centralized co-operative sensing is shown in Fig 2.2. It explains that, each secondary user or CR receives PU signal under multiple channel conditions. Each CR communicates either the local decision (in case of hard fusion) or test statistics (in case of soft fusion) to the Fusion Centre (FC). The FC aggregates the received information from all the CR in co-operation and takes the final decision.

In this work, the improved energy detection method is used as test statistics. There are two types of fusion techniques: soft and hard. Although the hard fusion decision algorithm has computational advantages, it is not reliable. Thus the current work concentrates on soft fusion logic at the fusion centre.

In the case of soft fusion logic, the weight associated with each CR plays an important role to achieve a reliable P_d . There are different strategies to evaluate the weight values such as Log Likelihood Ratio (LLR), Equal Gain Combining) EGC and Weighted Gain Combining (WGC). Out of all these, LLR method is the optimal method. The next subsection details about soft decision fusion technique.

3.8.1 Soft Decision fusion

In this fusion method [55], the test statistics are transmitted to the fusion Base Station Centre (BSC) for taking the decision. Here the fusion centre is free to use any decision rule: either Likelihood ratio test or Equal gain combining. Assume that the CR nodes send their energy test statistic to the fusion BS and it is fused as

$$\hat{\xi} = \sum_{i=1}^M w_i \xi_i = \mathbf{W} \mathbf{\Upsilon}^T \quad (3.19)$$

where w_i are the weights assigned to the i^{th} SU and $\hat{\xi}$ is the fused test statistic and ξ_i is the test statistic received from the i^{th} CR nodes and $W = [w_1, w_2 \dots w_M]$ and $\mathbf{\Upsilon} = [\xi_1, \xi_2 \dots \xi_M]$. The performance of test statistic depends on the weight values. The weight values can be estimated using different approaches like Log

Likelihood Ratio (LLR) or Equal Gain Combining (EGC) or Weighted Gain Combining (WGC).

3.8.1.1 Weight estimation using Log Likelihood Ratio test

Assuming that all ξ_i are independent and $Pr(\xi_i|H_1), Pr(\xi_i|H_0)$ are Gaussian and N is large, then the soft fusion is derived from the Likelihood ratio Test (LRT) given as:

$$LRT = \prod_{i=1}^M \frac{Pr(\xi_i|H_1)}{Pr(\xi_i|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda \quad (3.20)$$

$$(3.21)$$

The weight to each CR can be evaluated using log likelihood ratio test [68]

$$\log_{10} \left(\frac{Pr(\xi_i|H_1)}{Pr(\xi_i|H_0)} \right) \underset{H_0}{\overset{H_1}{\gtrless}} \lambda \quad (3.22)$$

where, $\xi_i = \xi_1, \xi_2, \xi_3 \dots \xi_M$ (by considering M CR user in cooperation) are the received test statistic at FC. Equation can be approximated as

$$\sum_{i=0}^{M-1} \log_{10} \left(\exp \left(\frac{-\|\xi_i\|}{\sigma_{w,m}^2 + \sigma_s^2} + \frac{\|\xi_i\|}{\sigma_{w,m}^2} \right) \right) = \|\xi_i\| \left(\frac{\|\sigma_s\|^2}{\sigma_{w,m}^2 (\sigma_{w,m}^2 + \xi_i)} \right) \quad (3.23)$$

Then the detection probability C_d for cooperative sensing can be written as

$$C_d = \sum_{m=0}^{M-1} \xi_m w_m \underset{H_0}{\overset{H_1}{\gtrless}} \lambda \quad (3.24)$$

$$w_m = \frac{\sigma_s^2}{\sigma_{w,m}^2 (\sigma_{w,m}^2 + \xi_m)} \quad (3.25)$$

where, ξ_m is the energy of the m^{th} node, σ_s^2 is the signal variance, $\sigma_{w,m}^2$ is the noise variance at the m^{th} node and C_d is the cooperative decision.

3.8.1.2 Equal gain Soft decision Fusion

Estimating ξ_i at the CR node is a computational burden. Also if the CR nodes have different capabilities then their estimates of ρ_i may also vary. One way to simplify is to set $w_i=1$ in (3.19) which is called Equal Gain soft decision fusion.

3.8.1.3 Weight estimation using Differential Evolution algorithm

The Differential Evolution (DE) algorithm is a stochastic and population based optimisation technique. It was proposed by Storn and Price in 1996 [69]. It belongs to a class of algorithms called metaheuristics, which make no assumptions about the differentiability of the function. DE is able to find an optimum value for intractable problems which are some times non-continuous or noisy. DE algorithm has been applied to estimate the weights in co-operative sensing using entropy as test statistic [53]. In the present work, DE is used for estimating weights using improved energy as test statistic.

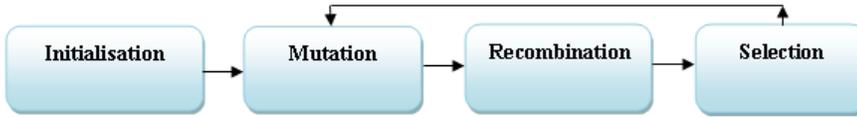


Figure 3.2: The flow of the Differential Evolution algorithm.

DE operates in four steps namely : initialisation, mutation, recombination and selection in an iterative fashion as in Fig 3.2. In this work, DE algorithm is used to estimate the optimum weight values associated with each CR node in the co-operative environment, by maximizing the co-operative probability of detection (3.24). So the weight estimation problem can be defined as :

$$\operatorname{argmax} \left(\sum_{m=0}^{M-1} \xi_m \theta_m \right); \theta_m \in [0, 1] \text{ such that } \sum_{m=0}^{M-1} \theta_m = 1 \quad (3.26)$$

where, ξ_m is the improved energy of the m^{th} CR node.

The detailed steps to estimate the weights using DE algorithm are :

1. Step 1 : Initialisation Initialise Population size P ; Scale factor F ; Dimension D . Initialise $\theta_{i,j}$ for $i = 1, 2, \dots, P$ and $j = 1, 2, \dots, D$ randomly following Uniform distribution within the limits $[0,1]$.
2. Step 2 : Mutation
For each target vector $\theta_{i,G}$ randomly select four distinct indices $i, r1, r2$ and

$r3$ such that $i \neq r1 \neq r2 \neq r3$. Generate a mutant vector as :

$$v_{i,G+1} = \theta_{r1,G} + F(\theta_{r2,G} - \theta_{r3,G}) \quad (3.27)$$

3. Step 3 : Recombination

The parent vector is recombined with the mutant vector to generate a child/trial vector $u_{i,G+1}$. The elements of child vector is:

$$u_{j,i,G+1} = \left\{ \begin{array}{l} v_{j,i,G+1} \ ; \ \text{if}(r_j \leq C_r) \ \text{or} \ j = \text{Irاند} \\ \theta_{j,i,G} \ \ \ \ ; \ \text{if}(r_j > C_r) \ \text{or} \ j \neq \text{Irاند} \end{array} \right\} \quad (3.28)$$

4. Step 4 : Selection

The performance of the child vector is compared with the parent vector and the better one is selected as parent vector for the next generation. All these steps are continued till stopping criteria is met (maximum generation).

$$\theta_{i,G+1} = \left\{ \begin{array}{l} u_{i,G+1} \ ; \ \text{if} \ f(u_{i,G+1}) \leq f(\theta_{i,G}) \\ \theta_{i,G} \ \text{otherwise} \ ; \ i = 1, 2, \dots, P \end{array} \right\} \quad (3.29)$$

5. Step 5 : Report the optimum parent vector (weights), θ_i .

where, $j = 1, 2, \dots, D$; $r_j \in [0, 1]$ is the random number; Crossover constant $CR \in [0, 1]$ and $\text{Irاند} \in [1, 2, \dots, D]$ is a random integer. The performance of the trial vector is compared with the parent vector and better one selected. The simulation is run for 60 generations to arrive at the optimum weights for each node.

3.9 Simulation details

This section presents the set up of for carrying out the simulation. In this work, a burst of BFSK is created for 450 frames. Each frame consists of 450 real numbers, called 'bits', generated randomly. The burst has some initial frames of pure noise followed by the signal with additive noise (Refer Fig 3.3). Towards the end of the burst again it is pure noise. The signal is passed through a flat fading Rayleigh channel and then an additive noise like white, blue or red, is added. The required SNR is created by scaling the noise and then summing it with the signal. The

energy is calculated for each frame. The number of samples per bit is varied to check the detection performance with respect to the sample size.

Energy Detection (ED) threshold is adapted based on the estimated noise power. The noise power is estimated by using the Unbiased estimator (U/b Est) and the Linear Predictor estimator (LP Est). The LP estimator is computed as (3.10). The unbiased estimator is computed over the current and previous 15 bits as (3.12). Performance of ED algorithm is analyzed with these estimators and compared with the ED without estimation (denoted as W/o Est) and tabulated. The simulation parameters of DE algorithms are as follows :

Number of populations $P = 20$; $D = 5$; ; $F = 0.9$; $C_r = 0.9$; maximum number of generations $G = 60$.

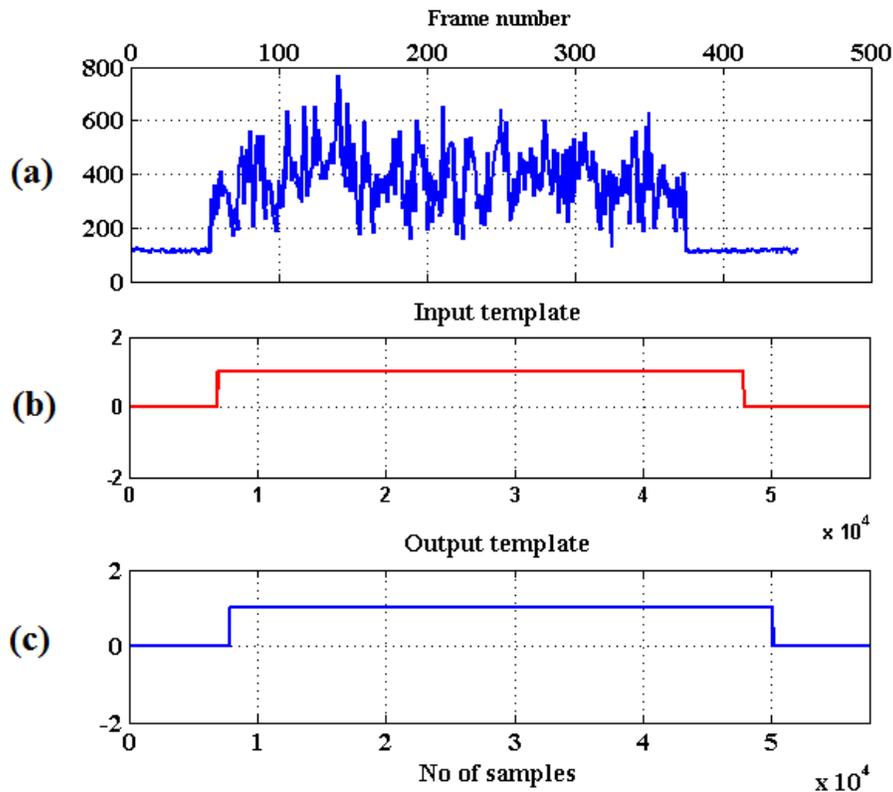
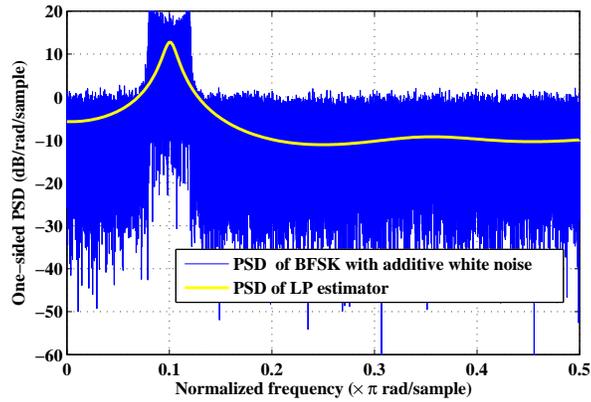
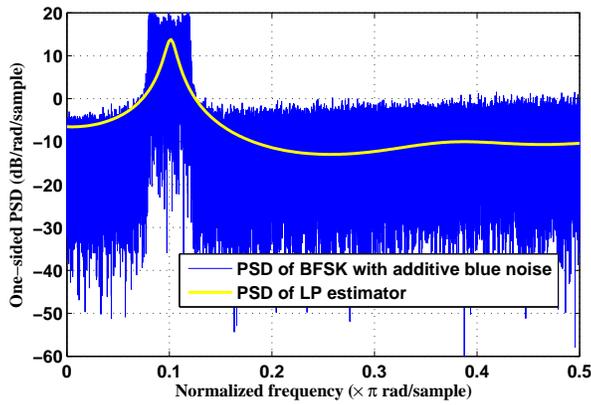


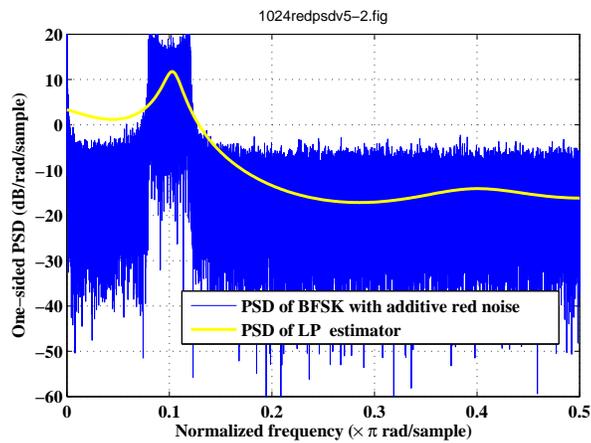
Figure 3.3: (a) Plot of the computed energy. (b) Plot of the input template (c) Plot of the output (detected) template.



(a)



(b)



(c)

Figure 3.4: Plots of the power spectral density for the BFSK waveform with additive *white*, *blue* and *red* noise and the autoregressive linear prediction estimate. The PSD of the LP estimator is accurate for *white* and *blue* noise but is not well matched for the *red* noise case.

3.10 Results and discussions

This section presents a performance comparison of energy detection (single node and multinode) using different noise estimation methods for three types of noise in the single node sensing environment. In the plots that are discussed in this section, LP, U/b and W/o refer to Linear Prediction Estimator, Unbiased estimator and Without estimator respectively. The one-sided power spectral density for the BFSK waveform with additive *white*, *blue* and *red* noise is plotted in Fig 3.4. The Auto Regressive linear prediction estimate is also plotted to demonstrate the accuracy of the linear prediction estimate.

The detection performance of energy detection with and without noise variance estimation for additive *white*, *blue* and *red* noise is plotted in Fig 3.5, 3.6 and 3.7 respectively. From Fig 3.5 it is observed that a probability of detection P_d of 0.9 is achieved at -8, -3 and -2 dB respectively for LP, U/b and W/o estimator for a sample size of 512. It concludes that modifying the detection threshold, (by subtracting the estimated variance of the additive noise), the detection probability improves up to 6 dB in the case of the LP estimator. It is also observed that the Unbiased estimator is giving a marginal improvement of 1 dB as compared to the case of energy detection without noise variance estimation.

With reference to Fig 3.4(a), it is observed that the LP estimator models the peak accurately. Since the LP estimator accurately estimates the noise variance, the detection threshold is adapted correctly thus improving the probability of detection. However, the unbiased estimator gives approximately same detection performance as simple energy detection because it estimates the signal plus noise variance instead of only noise variance.

The detection performance of energy detection, with and without noise variance estimation, is plotted in Fig 3.6 for additive *blue* noise condition. A probability of detection P_d of 0.9 is achieved at -6, -2 and -1 dB respectively for LP, U/b and W/o estimator for a sample size of 512. The values are lower compared to the *white* noise case as shown in Fig 3.5, but the pattern is similar, with a shift towards higher SNR . The Unbiased and W/o estimation plots are similar to that for *white* noise. The detection performance is lower as the noise is *blue* which increases at the rate of 3 dB/octave.

Fig 3.7 is a plot of the detection performance of energy detection, with and without noise variance estimation, for additive *red* noise condition.

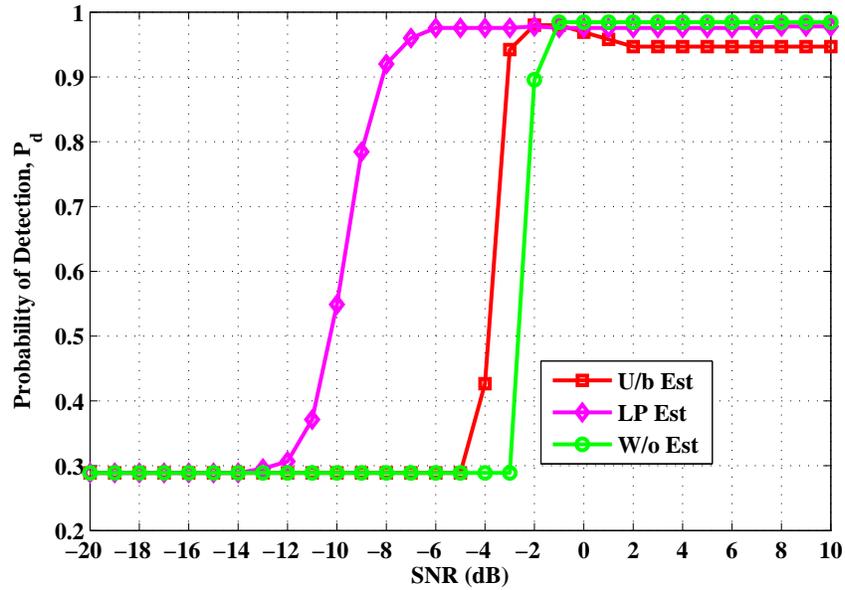


Figure 3.5: The probability of detection vs SNR with and without Noise variance estimators is plotted for additive *white* noise. The BFSK waveform has a doppler shift of 200 Hz.

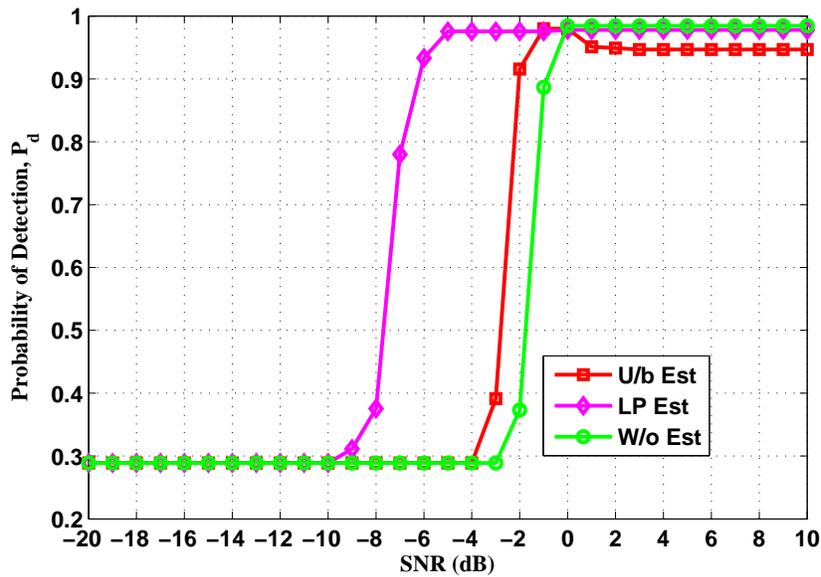


Figure 3.6: The probability of detection vs SNR with and without Noise variance estimators is plotted for additive *blue* noise.

A probability of detection P_d of 0.9 is achieved at +1 dB and +2 dB respec-

tively for U/b and W/o estimator for a sample size of 512. However, there is no detection for the LP estimator with red noise. It is observed that the performance of the U/b and W/o estimator case has reduced by 5 dB and 3 dB respectively as compared to the *white* noise results of Fig 3.5.

Linear prediction assumes that the all-pole filter is excited with white noise to produce a waveform. In our case we are doing the reverse i.e., we are estimating the filter parameters using the received signal autocorrelations, and the resultant error should be white noise. It may be seen from Fig 3.4(b)(c), that the assumption of flat white noise for LP estimator is not met in the case of *blue and red* noise since their PSD are not flat and hence the estimates have an error. This resultant error reduces the performance of the energy detection as compared to the white noise case. However, the error is smaller in the case of *blue* noise. Thus it may be concluded that the LP estimator works well with *white and blue* noise.

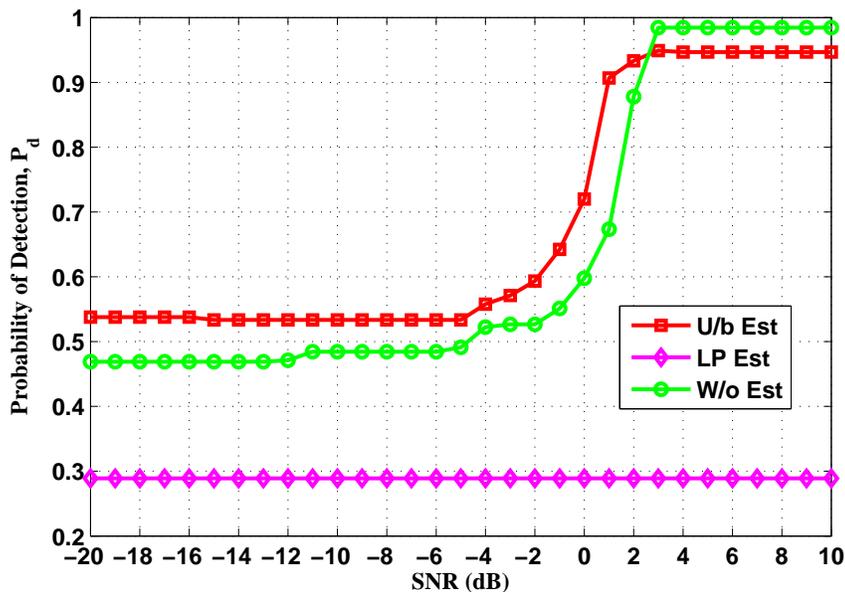


Figure 3.7: The probability of detection vs SNR with and without Noise variance estimators is plotted for additive *red* noise.

From the Figs 3.5, 3.6 and 3.7 it is concluded that the Unbiased estimator works consistently with all types of noise, but the improvement is marginal with respect to the Energy detection without any estimation. However, it is still able to detect the presence of signal in the case of additive *red* noise condition. ED

without noise power estimation gives the same result with all types of noise.

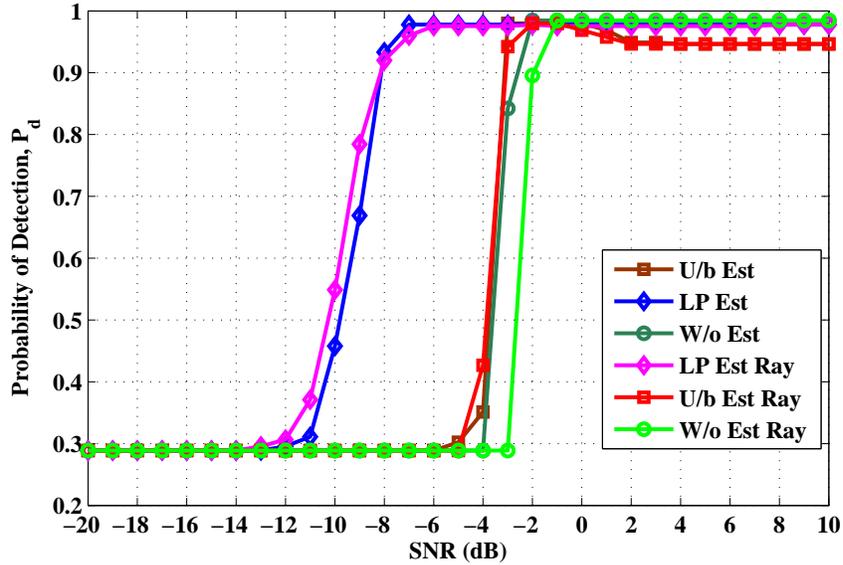


Figure 3.8: Detection for the cases of no fade and flat fading Rayleigh channel

The LP estimator is able to estimate the PSD of the BFSK waveform as it has distinct peaks. However, it also implies that its performance for modulations which do not have distinct peaks may not be satisfactory. N/d : No detection
 The detection performance with respect to sample size for *white*, *blue* and *red noise* is tabulated in Table 3.1. From the table, it is observed that the LP estimator outperforms, Unbiased estimator for sample sizes of 64 and higher. The LP estimator fails to detect for additive *red* noise irrespective of sample size.

The effect of Rayleigh fading on the improved energy detection performance is also investigated. Fig 3.8 shows the results with and without Rayleigh fading for the additive white noise channel. It is observed that the detection performance is not affected with or without fade. LP and Unbiased estimators are giving consistent results with or without fade. The performance of energy detection without noise estimation is less by 1dB with fade. The consistent performance could be attributed to the fact that the simulated Rayleigh channel simulated has a flat fading (single path) channel which introduces only a gradual change in the envelope. The amplitude variations induced by the fading seem to have little impact on the computed energy. It may be concluded that the flat fading Rayleigh

Table 3.1: Detection performance at different sample sizes

Sample size	SNR, White Noise,dB			SNR, Blue Noise,dB			SNR, Red Noise,dB		
	U/b est	LP est	W/o est	U/b est	LP est	W/o est	U/b est	LP est	W/o est
16	+4	N/d	+8	2	N/d	+10	+6	N/d	+9
32	+4	+10	+6	+2	N/d	+6	+4	N/d	+8
64	+2	0	+4	0	+2	+4	+2	N/d	+6
128	0	-2	+2	0	-2	+2	0	N/d	+3
256	-2	-5	0	0	-4	+1	+1	N/d	+3
512	-2	-7	+1	-2	-6	0	+1	N/d	+2
1024	-4	-10	-2	-2	-7	-2	+1	N/d	+2

channel has a small effect on the detection performance.

The detection performance for BFSK signal with different noise variances is plotted in Fig 3.9. From the figure, it is observed that the unbiased estimator fails to detect the signal irrespective of SNR for larger noise variances. It gives the same performance as the detection without estimation. However, the LP estimator outperforms other estimators in detection performance. It shows that noise variance estimation makes energy detection algorithm robust.

The detection performance, for a single node and multi-node cooperative sensing is shown in Fig 3.10. The results are for cooperative sensing with 5 nodes. The results for four combining schemes namely: Log Likelihood Ratio (LLR), Equal Gain Combining (EGC), Weighted Gain Combining (WGC) and Differential Evolution (DE) are plotted. The figure shows two groups of plots marked *with est* and *w/o est* denoting energy detection with and without noise variance estimation respectively. From the plots it is observed, that in both groups the cooperative sensing performance is better than a single node. Secondly, the performance of both single node and cooperative sensing improve with noise variance estimation. The improvement is between 10 to 12 dB. It is also observed that the four com-

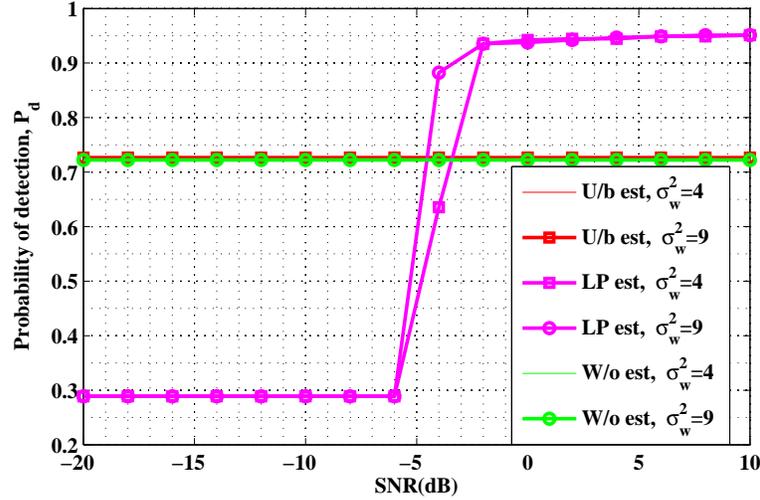


Figure 3.9: Detection performance for BFSK with additive *white* noise for two cases of noise variance. It is observed that the LP estimator is able to detect the BFSK signal with additive noise of different variance whereas the other two methods fail.

binning schemes give similar performance in both groups. It may be due to the low number of nodes in cooperation in sensing. The advantage with DE based weight estimation is that it does not require any noise variance estimation, whereas other techniques require noise variance estimation.

Fig 3.11 is a comparison of single and multinode detection under conditions of constant additive noise and variable additive noise. The required SNR in the simulation can be created either by keeping the additive noise power constant and scaling the signal power or by adding variable noise keeping signal power constant. The first case is encountered in radio frequency bands where the channel noise is constant and the SNR is only a function of signal attenuation from the transmitter to the receiver. The second case is encountered in radio bands with many transmitters like TV and FM stations that introduce variable noise into the CR receiver. From the figure, it is observed that in the case for constant noise, single and multinode sensing perform equally well, whereas in the case of variable noise multinode combining perform better than a single node. This is attributed to better noise variance estimation when noise power is variable and has significant power.

Fig 3.12 is a plot of the Complementary ROC for multinode sensing for two combining methods namely WGC and LLR. The performance of each combining

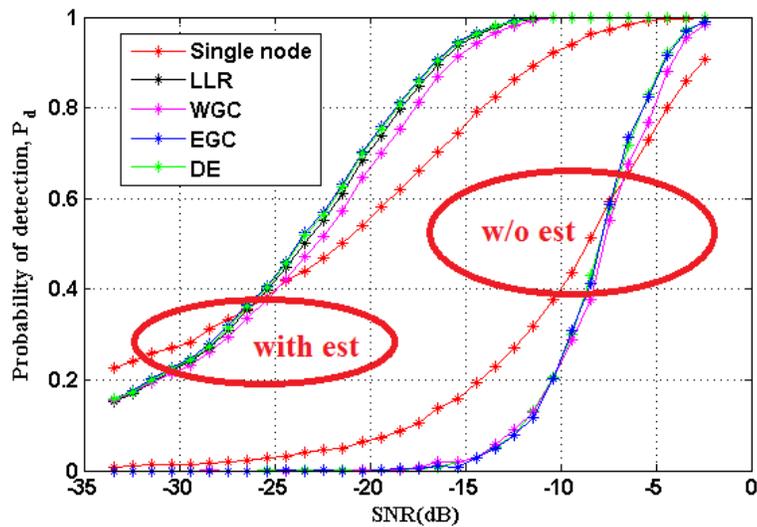


Figure 3.10: Detection performance with respect to received signal SNR for single node and multi-node cooperative sensing with LLR, EGC, DE and WG combining. The plots for the four combining schemes with LP noise variance estimation and without estimation are compared with single node performance.

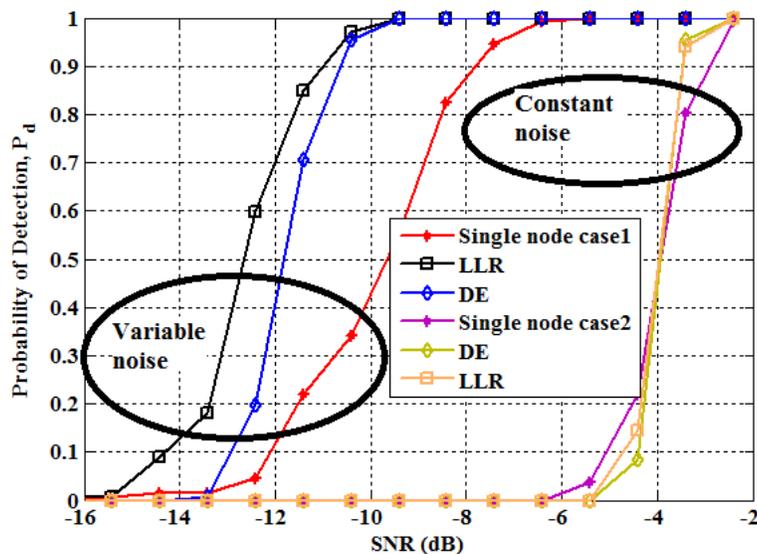


Figure 3.11: The probability of detection for Single node and multinode cooperative sensing. Plots for two methods of fusion namely, LLR and DE are shown. In both cases noise variance is estimated.

method is investigated for two conditions namely with and without LP noise variance estimation. It is observed that at $P_m=0.01$, the probability of false alarm,

is 0.3 and 0.1, in the cases of without and with noise variance estimation respectively. It is noted from the plots that the improvement is roughly 3 dB or better at all P_f greater than 10^{-2} . So it may be concluded that multinode combining with noise variance estimation improves the performance.

Combining with noise variance estimation is plotted in Fig 3.13. The Com-

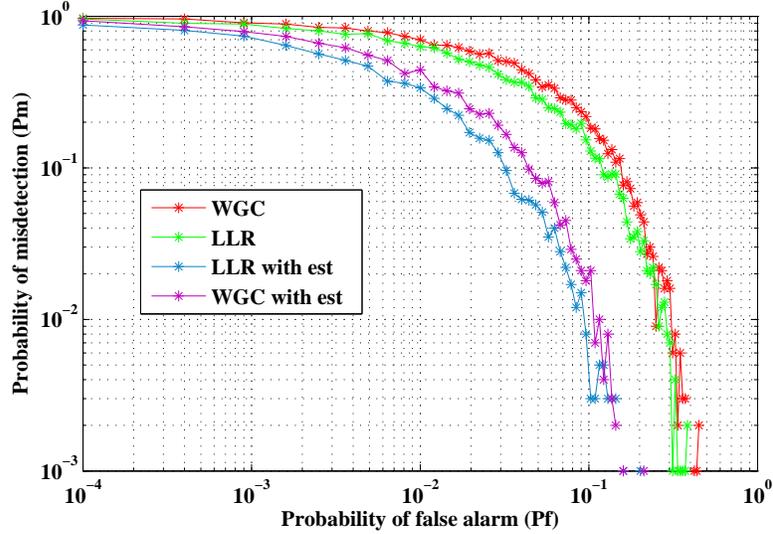


Figure 3.12: Complementary ROC for multi-node cooperative sensing using Weighted Gain (WG) and Log Likelihood Ratio (LLR) combining. The plots for the two combining schemes, WGC and LLR, with and without LP noise variance estimation are shown.

plementary ROC of a single node and multinode sensing. It is observed that multinode sensing outperforms the single node sensing for obvious causes. The simulation is carried out for 5 nodes. It is observed that the plots for LLR and EGC are coinciding. In the case of EGC the weights for all nodes is 0.2. Differential Evolution algorithm is used for evaluating optimum weight. In this case as the number of nodes is low, the optimised weights tend to the EGC value of 0.2. In the case of LLR where the weights are in inverse proportion to the SNR of the received signal, the weights are more optimum thus yielding a better performance. However, the use of DE algorithm eliminates the need for noise variance estimation for determining weights.

Fig 3.14 is a plot of the ROC single and multinode sensing. Three methods of combining are investigated namely EGC, LLR and DE. It is observed that the multinode methods give improved performance as compared to single node. The

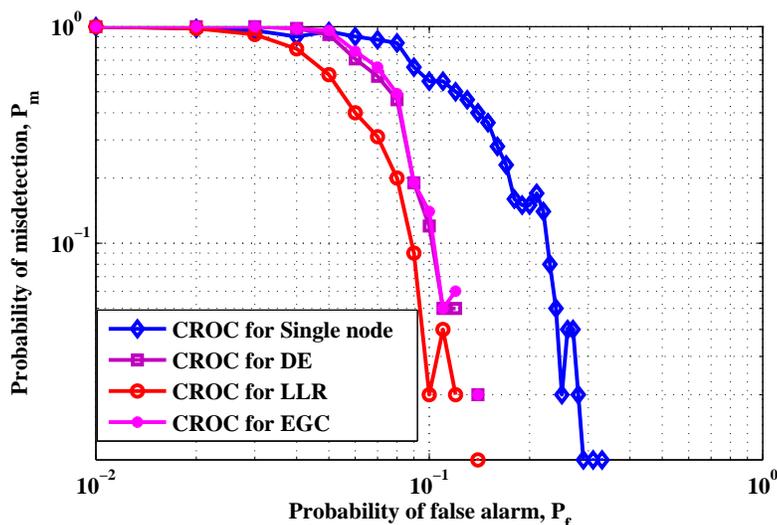


Figure 3.13: The Complementary Receiver Operating Characteristic (CROC) for single node and multi-node cooperative sensing with noise variance estimation. For multinode cooperative sensing three methods of combining are used namely Differential Evolution (DE), Equal gain combining (EGC) and Log Likelihood Ratio (LLR). The SNR is -12 dB.

performance of EGC and DE are almost similar. DE performance is better when the number of nodes is large. In this simulation with 5 nodes DE tends to EGC performance. LLR performance is the best among all the combining methods.

3.11 Conclusions

Two different noise power estimation techniques were been studied to improve the performance of Energy detection. It is shown that the LP estimator gives improved performance for white and blue noise, of the order of 6 dB or more. However its performance is weak in the case of red noise. The Unbiased estimator works uniformly for all types of noise but the improvement over simple energy detection is not significant. It is shown that the LP and unbiased estimators perform satisfactorily even with Rayleigh fade which is expected in the real world. The study indicates that the energy detection with these improvements could be used in situations where the noise type is known apriori to be *white or blue*.

The effect of noise variance estimation on multi-node cooperative sensing has

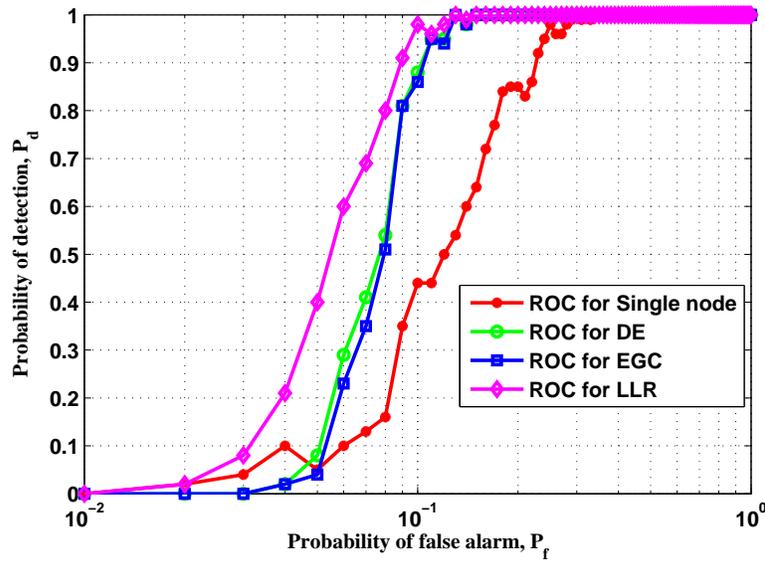


Figure 3.14: Receiver Operating Characteristic (ROC) for single node and multi-node cooperative sensing with noise variance estimation. For multi-node cooperative sensing three methods of combining are used namely Differential Evolution (DE), Equal gain combining (EGC) and Log Likelihood Ratio (LLR). The SNR is -12 dB.

been investigated. It is demonstrated that the detection performance and ROC improve significantly with noise variance estimation, regardless of the combining scheme used at the fusion centre. The performance for single node is compared with the cooperative sensing to demonstrate the improvement. On an average an improvement of 3 dB or better is obtained in multi-node sensing with noise variance estimation.

**This page is intentionally left
blank**

Chapter 4

SNR estimation for spectrum sensing

4.1 Objective

The objective of this chapter is to discuss the importance of SNR estimation in Spectrum sensing. The algorithm for Blind SNR estimation using Covariance technique is elaborated.

4.2 Introduction

SNR estimators play a significant role in wireless receiver functionality, i.e., equalization, signal detection, spectrum sensing, turbo decoding, etc. [70],[71]. Error in SNR estimation degrades the receiver performance. Different techniques for estimating the SNR are reported in [43],[72],[73],[74]. It was demonstrated that the best estimator for a given scenario depends on the number of samples per symbol, block length, type of modulation and the SNR range of interest [43]. In general, SNR estimators are classified as either Data Aided (DA) or Non-Data-Aided (NDA). In the DA scheme, pilot symbols are transmitted along with the data symbols. It decreases the bandwidth efficiency. NDA schemes have been proposed to overcome this drawback [75]. It was suggested that Maximum Likelihood (ML) technique and Moment-based technique are preferred choice for DA and NDA estimators respectively [76],[77],[44]. Moments of the received signal are used to estimate the SNR of the QAM (Quadrature Amplitude Modulated) signal [78]. In [78], a low order moment is used to estimate the SNR whereas a higher order moment is used for minimizing the variance of the estimate. Different ML estimators namely PDA MLE, NDA MLE, Joint Pilot and Data Aided MLE and Estimation using Data Statistics (EDS) were investigated for SNR estimation of Non-Coherent M-ARY FSK signal [79],[80]. The ML estimator is derived for Data

Aided and Data cum Pilot Aided scheme for Rayleigh and Rician fading channels, with an assumption that the receiver system is entirely synchronous [81],[82]. This assumption may not be valid in practical scenarios. The drawbacks of the techniques as mentioned above are that (i) they work for positive and high SNR only (ii) need a large number of samples to obtain a small Normalised Mean Square Error (NMSE) (iii) their performance depends on the rate at which the pilots are repeated (iv) require prior knowledge of the signal characteristics. Eigen value based SNR estimation is proposed with parameters like sample size, number of eigen values optimised using Particle Swarm Optimization (PSO) [83]. The SNR of Linear FM modulated signal is estimated by fitting the Fourier series to the main lobe of the autocorrelation [84].

Most of the reported algorithms successfully estimate the positive SNR only. However, limited studies exist for estimating negative SNR. When specified in dB, the Signal to Noise Ratio (SNR) is said to be negative when it is less than unity.

In tactical communications, M-ARY FSK modulation is used due to its low design complexity. It is not sensitive to non-linearity in power amplifiers. Hence, M-ARY FSK modulation is the preferred modulation choice for low power and low data rate applications. Therefore we have investigated the SNR estimation of an M-ARY FSK signal in Rayleigh and Rician fading channels. The proposed algorithm is a blind SNR (Signal-to-Noise-Ratio) estimation algorithm. The SNR is estimated by comparing the test statistic of the received signal with a calibrated signal. The estimated SNR is the value of the SNR that minimizes the difference between the computed and calibrated test statistics. The test statistic of both the received and calibrated signal is calculated using the Sample Covariance Matrix (SCM). The proposed algorithm first detects the presence of a signal in the channel, and if a signal is detected then, it estimates the SNR. However, if the signal detection is not necessary, then the proposed method can independently estimate the SNR of a signal. The algorithm estimates both positive and negative SNR.

The proposed method determines a test statistic of the received signal by computing its Sample Covariance Matrix [16]. The test statistic is compared with pre-stored test statistic of a calibrated signal to determine the SNR of the received signal. Initially during calibration, an M-ARY FSK signal at different SNR is created by adding White Gaussian Noise (WGN). The test statistic derived from the SCM of this calibrated signal is approximated by an exponential function with three unknown parameters. The unknown parameters are evaluated and stored

in a lookup table. The estimated SNR of the received signal is the SNR that corresponds to the minimum error between the computed and stored test statistic. Further, the proposed method inherently detects the presence of the signal during the SNR estimation process. The advantage of this approach is that it can be used for both signal detection and SNR estimation without extra computational cost.

4.3 SNR estimation algorithm

Consider an M-ARY FSK signal as:

$$y(n) = \alpha(n)x(n) + w(n); \quad (4.1)$$

where, $x(n)$ and $y(n)$ are the transmitted and received signal respectively. $w(n)$ is complex white Gaussian noise with zero mean and unit variance, $\alpha(n)$ is the channel gain with a Rayleigh (zero mean) or Rician (non-zero mean) distribution with unit variance and n is the time index. In the proposed SNR estimation algorithm, the test statistic γ is based on the Covariance Absolute Value (CAV) [16]. It is the ratio of two random variables T_1 and T_2 , where T_1 and T_2 are the sum of all the elements and sum of all the diagonal elements of the covariance matrix of the received signal respectively. The T_1 and T_2 are scaled by a smoothing factor L that defines the number of lags selected for computing the autocorrelation. Equations (2.13-2.20) are applicable. The test statistic $\gamma(k)$ is used to (i) detect the presence of a signal and (ii) estimate the SNR.

4.3.1 Signal detection

The presence of a signal is determined by detecting the *rising* and *falling* edges of the signal burst. The *rising – edge* of the signal burst is determined if $\gamma > \beta$, where β is a threshold defined as (2.20)

$$\beta = \frac{1 + (L - 1)\sqrt{2/N_s\pi}}{1 - Q^{-1}(P_f)\sqrt{2/N_s}} \quad (4.2)$$

where P_f and $Q^{-1}(\cdot)$ define the probability of false alarm (for signal detection) and inverse Q function respectively. For detecting the *falling – edge* of the signal

burst, let us define $\delta(k)$ as

$$\delta(k) = \gamma(\text{peak}) - \gamma(k) \quad (4.3)$$

where $\gamma(\text{peak})$ is the peak value of γ , upto frame k , of the signal burst. A Moving Average (MAV) Filter of window width W is applied to $\delta(k)$ to remove the effect of outliers. The output of filter is

$$\delta_{MAV}(k) = \frac{1}{W} \sum_{q=-(W-1)/2}^{(W-1)/2} \delta(k+q) \quad (4.4)$$

The *falling – edge* is determined when $\delta_{MAV}(k) > \tau$, where $0 < \tau < 0.5$. For a BFSK signal, with signal burst duration from frame number 50 to 250, the relationship between γ , $\gamma(\text{peak})$ and δ_{MAV} is shown in Fig 4.1. From this figure, it is observed that γ starts increasing from frame number 50, at the *rising – edge* of the burst, and starts decreasing from frame number 250, indicating that the *falling – edge* has occurred. Correspondingly δ_{MAV} decreases to a value close to zero after the *rising – edge* and increases in value after the *falling – edge*. In Fig 4.1 it is seen that the value of δ_{MAV} is almost zero after the *rising – edge* till the *falling – edge* of the burst, after which it starts to increase in value. To ensure quick detection of the falling edge the value of τ is set as close as possible to zero. A value of $\beta/100$ is found to be optimal, where β is the threshold calculated as per (4.2). Fig 4.2 plots γ vs. frame number for different burst widths for a BFSK signal at an SNR of -8 dB. From this figure, it is observed that regardless of the width of the signal burst, the envelope of γ follows the same trajectory for a given SNR. The *rising – edge* is the same for all the curves, whereas the *falling – edge* depends on the width of the signal burst. It shows that the envelope of γ for a particular SNR is fixed and can be stored in a *lookup table*.

4.3.2 Approximation of test statistic

The test statistic γ can be approximated to an exponential function with few unknown coefficients as :

$$\hat{\gamma}(p, k) = \gamma(p, 0) + A(p, 0) * \exp(R(p, 0) * k); \quad (4.5)$$

$$\epsilon(p, k) = |\hat{\gamma}(p, k) - \gamma(k)|; \quad (4.6)$$

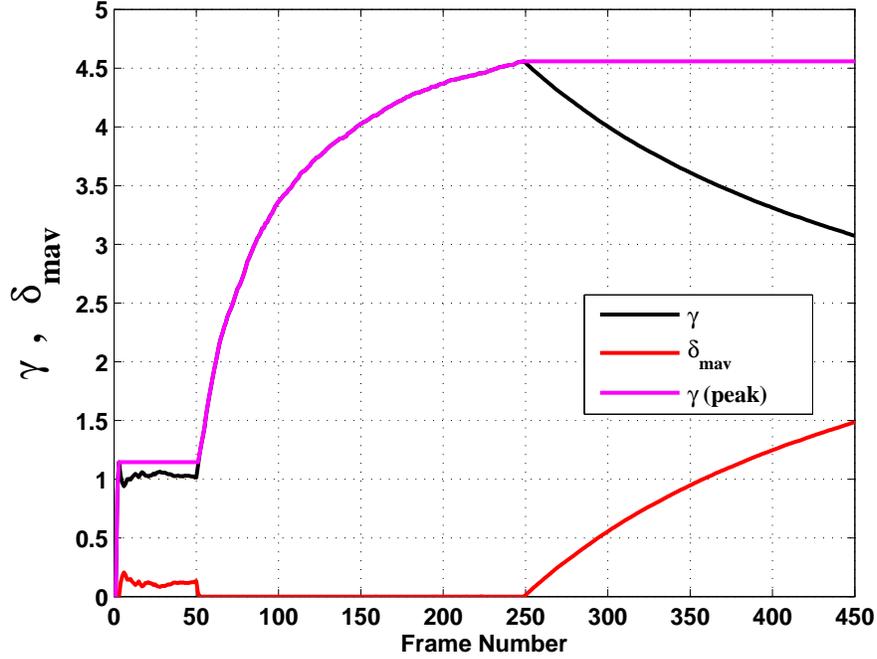


Figure 4.1: Plot of γ , $\gamma(\text{peak})$ and δ_{MAV} for a BFSK burst starting at frame 50 and ending at frame 250. The smoothing factor L is 32, number of samples per frame is 512 and SNR is -8 dB.

where p and k correspond to the index of SNR and frame respectively. $\hat{\gamma}(p, k)$ is the estimated value of the test statistic for the k th frame signal with SNR index p . $\gamma(p, 0)$, $A(p, 0)$ and $R(p, 0)$ are the fitting coefficients for p th SNR of the calibration signal, ϵ is the error between estimated and computed value of γ . The computed test statistic and the exponential curve fit for a BFSK signal of -8 dB SNR is plotted in Fig 4.3. From this figure, it is observed that the approximation is in reasonable agreement with the true value of γ . The coefficient values are stored in a *lookup table* for the p th SNR. The curve fitting statistics for the Non-Linear Curve fit (exponential) are also shown in Table 4.1. The Adjusted R^2 indicates the accuracy of the curve fitting. A value close to unity indicates good agreement between the true and curve fitting value. The Adjusted R^2 values for SNR between -20 dB and +20 dB are shown in Table 4.1. This table is used to estimate the SNR of the received signal.

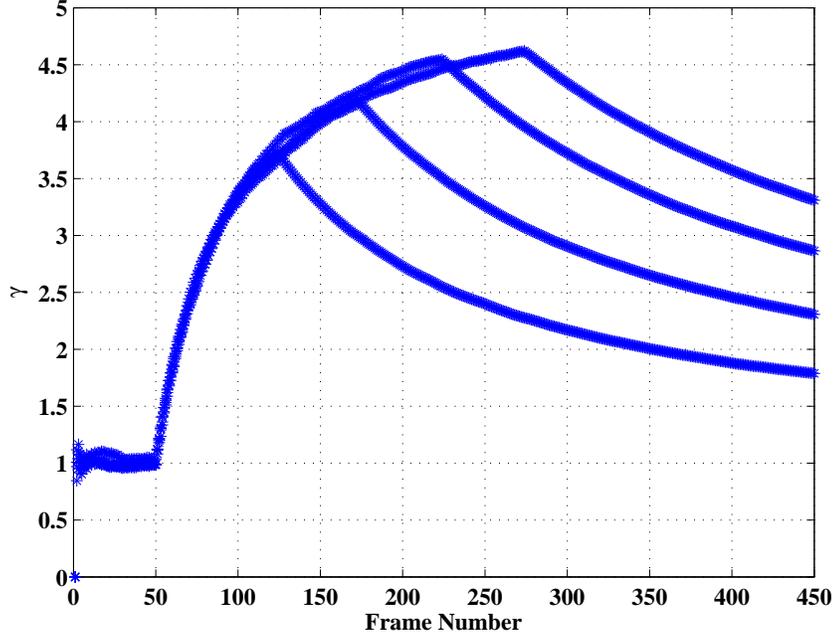


Figure 4.2: Plot of γ for a BFSK burst at -8 dB SNR. The number of samples is 512. The four traces correspond to four signal burst widths. The *rising – edge* is fixed at frame number 50 and the *falling – edges* are at frame numbers 125, 175, 225 and 275 respectively.

4.3.3 System model

The algorithm steps for estimating the SNR of an unknown received signal are shown in *Algorithm 1*. The Normalised Mean Square Error (NMSE) is used as the metric for measuring the accuracy of the estimation algorithm. It is given as

$$NMSE(\hat{\eta}) = \frac{E[(\eta - \hat{\eta})^2]}{\eta^2} \quad (4.7)$$

where η and $\hat{\eta}$ are actual and estimated SNR respectively.

4.4 Partially Data Aided ML estimator for M-ARY FSK

Consider an M-ARY FSK transmission with $M = 2^m$ symbols in the constellation where, m is a positive integer. This signal is corrupted by complex Gaussian noise

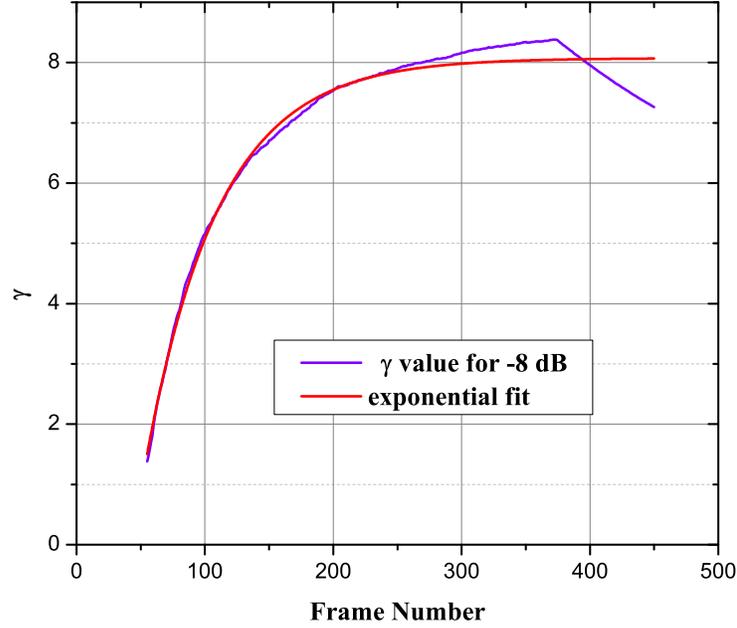


Figure 4.3: Exponential curve fit of γ for -8 dB SNR. The signal burst starts at frame number 54 and ends at frame number 375.

and Rayleigh fading independently. The received signal after matched filtering and squaring is given as [79]

$$y_n = |s_n \alpha_n + w_n|^2 \quad (4.8)$$

In the data aided MLE technique, the SNR is estimated as

$$\hat{\eta}_{DA} = \frac{(M-1) \sum_{n=1}^g y_{1,n} - \sum_{m=2}^M \sum_{n=1}^g y_{m,n}}{\sum_{m=2}^M \sum_{n=1}^g y_{m,n}} \quad (4.9)$$

where $g = [1, 0, \dots, 0, \dots, 0]^T$ and $s = [0, 0, \dots, 1, \dots, 0]^T$ are the pilot and data symbols respectively. The NMSE is calculated as (4.7). The received symbols from M branches are $y(m, n)$ where m and n are the branch and time index respectively. w_n is a random variable that follows Gaussian distribution whereas α_n follows Rayleigh distribution. The performance of the proposed algorithm is compared with this Partially Data Aided ML Estimator.

Input: No. of samples, Number of calibrated SNR values $pmax$, Maximum number of frames for analysis $kmax$, Lookup Table values

$(\gamma(p, 0), A(p, 0), R(p, 0))$

Output: $\hat{\eta}$ as the estimated SNR

Compute the covariance matrix of the k th frame received signal using (2.13)

Compute the test statistic $\gamma(k)$ and $\delta_{MAV}(k)$ as (2.17) and (4.4)

respectively.

Compute β as (4.2)

Initialise $kcount=0$

for $k = 1$ to $kmax$ **do**

for $\gamma(k) > \beta$ & $\delta_{MAV} < \tau$ **do**

$kcount \leftarrow kcount + 1$

for $p = 1$ to $pmax$ **do**

 Compute $\hat{\gamma}(p, k)$ as (4.5), using lookup table

 choose p_{opt} that minimises (4.6)

end

$\rho(k) \leftarrow p_{opt}$

$\rho_{sum} \leftarrow \rho_{sum} + \rho(k)$

end

$\hat{\eta} \leftarrow \rho_{sum}/kcount$

end

return $\hat{\eta}$

Algorithm 1: Proposed SNR estimation algorithm

4.5 Simulation results and discussions

This section presents numerical simulation results and performance analysis of the proposed SNR estimator. The simulations are carried out for Rayleigh and Rician fading channels for $M = 2, 4$ and 8 FSK modulation. A complex baseband M-ARY FSK signal $s(n)$ is generated and its standard deviation is computed as (4.11). $s(n)$ is scaled as (4.10) to generate $x(n)$. The complex Gaussian noise $w(n)$ is added to $x(n)$ obtain the required SNR as (4.1).

$$x(n) = \frac{\sigma_w}{\sigma_s} \sqrt{10^{SNR/10}} s(n) \quad (4.10)$$

$$\sigma_s = \frac{1}{N_s - 1} \sum_{n=1}^{N_s} (s(n) - \mu_s)^2 \quad (4.11)$$

Table 4.1: Exponential Curve fitting statistics

SNR (dB)	Number of Points	Degrees of Freedom	Adj. R^2 M=2	Adj. R^2 M=8
-20	66	63	0.98384	0.95358
-16	66	63	0.99765	0.99674
-12	66	63	0.99756	0.99579
-8	66	63	0.99923	0.99846
-4	66	63	0.99953	0.99848
0	66	63	0.99904	0.99679
4	66	63	0.99676	0.99473
8	66	63	0.98956	0.98939
12	66	63	0.97544	0.98428
16	66	63	0.96204	0.9796
20	66	63	0.96675	0.96517

where SNR is the signal to noise ratio in dB, μ_s , σ_s and σ_w are the sample mean, standard deviation of the signal and noise respectively. The scaled signal $x(n)$ is multiplied with channel gain $\alpha(n)$, which is drawn from a Rayleigh or Rician distribution.

For simulation, 450 frames of M-ARY FSK, each with 512 number of samples (N_s) are generated. The frames slide into the input buffer with time. Initially, the buffer has only zeros, then as the frames are received, the buffer fills up until maximum buffer size is reached. The sample covariance matrix for each frame of the received signal is computed as (2.13). For signal detection, the *rising* and *falling* edges of the signal burst are detected as discussed in subsection 4.3.1. The number of frames between the two detected edges is termed as the *detected – burst – width*. The ratio of *detected – burst – width* to the *actual – burst – width* gives the probability of detection (P_d). For simulation purpose, the threshold β is calculated as (4.2) for $P_f = 0.1$. The moving average filter window width W and τ are chosen as 16 and $\beta/100$ respectively, where β is the threshold calculated as per (4.2). The probability of detection P_d for each SNR is computed using multiple simulation runs and is plotted in Fig 4.4. This figure demonstrates that the proposed method can detect the M-ARY FSK signal up to -8dB SNR with P_d of 0.9, using 512 samples. However, it can detect the signal at much lower SNR if the number of samples are increased.

The Normalised Mean Square Error (NMSE) for a -8dB SNR M-ARY FSK (M=2,4 and 8) signal burst for Rician and Rayleigh flat fading channel are plotted

in Fig 4.5 and Fig 4.6 respectively. From these figures, it is observed that for the complete burst duration, the NMSE is less than 10^{-2} for $M=2$ and 4 whereas it is almost constant, i.e., 10^{-1} for $M=8$. It can be analyzed by referring to Fig 4.7, that plots γ vs. frame number for different values of M for a -8 dB BFSK signal. The exponential curve fitting coefficient values for $M=2$ and 4 are close to each other whereas it is different for $M=8$. From figures 4.5 - 4.7, it can be concluded that the curve fitting technique works satisfactorily for both types of fading in the case of $M=2$ and 4 but not for $M=8$. Hence, the fitting coefficients for $M=8$ need to be re-evaluated for accurately estimating the SNR of an 8FSK signal. The test statistic for $M=8$ is re-evaluated and tabulated in Table 4.2.

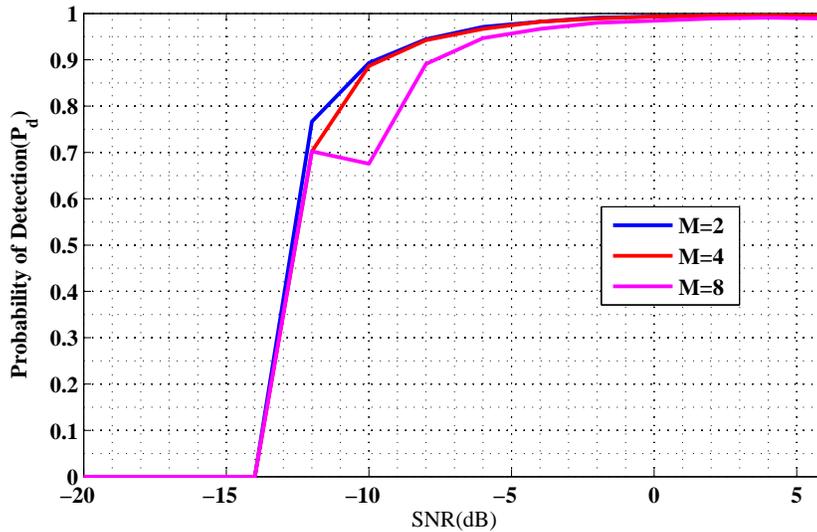


Figure 4.4: P_d vs SNR for M-FSK signal in Rayleigh flat fading channel

The performance of the proposed algorithm is compared with the PDA MLE SNR estimator for an M-ARY FSK ($M=2,4,8$) signal. Fig 4.8 is the PDA-ML estimator result for the signal with Rayleigh fading and a Doppler of 20 % of the frequency deviation for SNR ranging from 0 dB to +25 dB. From this figure, it is evident that the PDA MLE SNR estimator gives satisfactory performance for high and positive SNR. The NMSE is approximately 10^{-2} to 10^{-3} , over a narrow range of SNR. However, the NMSE shows a dip at an SNR of 7 dB whereas it is expected to decrease monotonically with increasing SNR. This may be explained as follows: (4.9) is derived for Rayleigh fading with no Doppler [79]. However, when Doppler is added to Rayleigh fading, the joint pdf of signal and noise is no

Table 4.2: Curve fitting values for M=2 and M= 8

M	SNR(dB)	$\gamma(p, 0)$	$R(p, 0)$	$A(p, 0)$
8	-20	1.107	-0.030	-0.577
2	-20	1.174	-0.046	-2.584
8	-16	1.382	-0.020	-1.155
2	-16	1.540	-0.023	-2.133
8	-12	1.805	-0.025	-3.034
2	-12	2.253	-0.023	-4.637
8	-8	2.898	-0.024	-7.170
2	-8	3.926	-0.027	-12.548
8	-4	4.978	-0.031	-22.067
2	-4	7.362	-0.031	-33.86
8	0	8.470	-0.038	-56.091
2	0	12.960	-0.039	-98.694
8	4	12.465	-0.055	-215.244
2	4	19.277	-0.059	-431.955
8	8	15.844	-0.092	-2051.714
2	8	24.829	-0.092	-3195.228
8	12	18.056	-0.172	-176843.206
2	12	28.434	-0.167	-206061.424
8	16	19.267	-0.344	-2.03E+09
2	16	30.434	-0.361	-8.10E+09
8	20	19.860	-0.685	-2.17E+17
2	20	31.540	-0.735	-5.07E+18

more exponential (Eqn 4 of [79]) and the estimator given by (4.9) is mismatched.

From figures 4.5 - 4.7, it is clear that the fitting coefficients for M=2 are not suitable for estimating the SNR of an 8FSK signal. Thus, we explored the use of M=8 fitting coefficients (Table 4.2) and estimated the SNR of the 8FSK signal using both the fitting coefficient sets, i.e., M=2 and 8. The NMSE for SNR range -20 to +20 dB (flat fading Rayleigh channel) is plotted in Fig 4.9. It is observed that when the SNR of the 8FSK signal is estimated using M=8 fitting coefficients, the NMSE improvement is significant and is consistently below 10^{-2} for the SNR range of -20 to +20 dB.

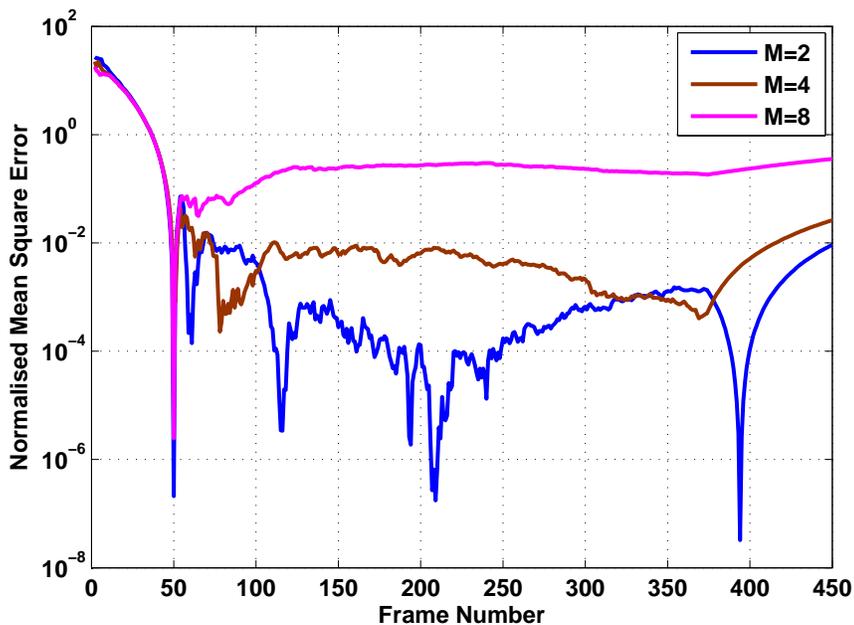


Figure 4.5: Normalised Mean Square Error for M-ARY FSK for $M=2,4,8$ for -8 dB SNR under *Rician* fading with a K factor of 10. The signal burst starts at Frame number 54 and ends at Frame number 375. The number of samples is 512.

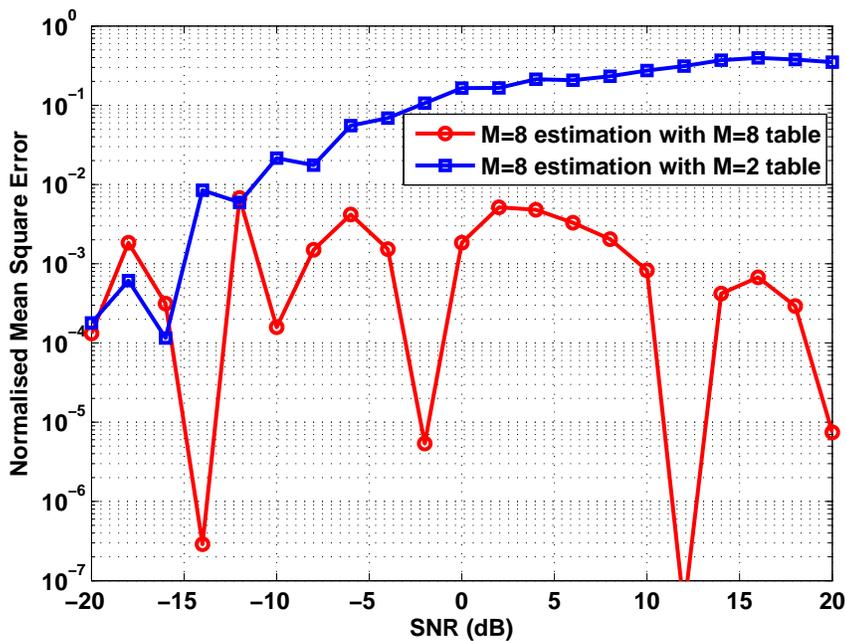


Figure 4.9: Normalised Mean Square Error for 8FSK for Rayleigh flat fading with lookup table generated from $M=2$ and $M=8$ curve fitting.

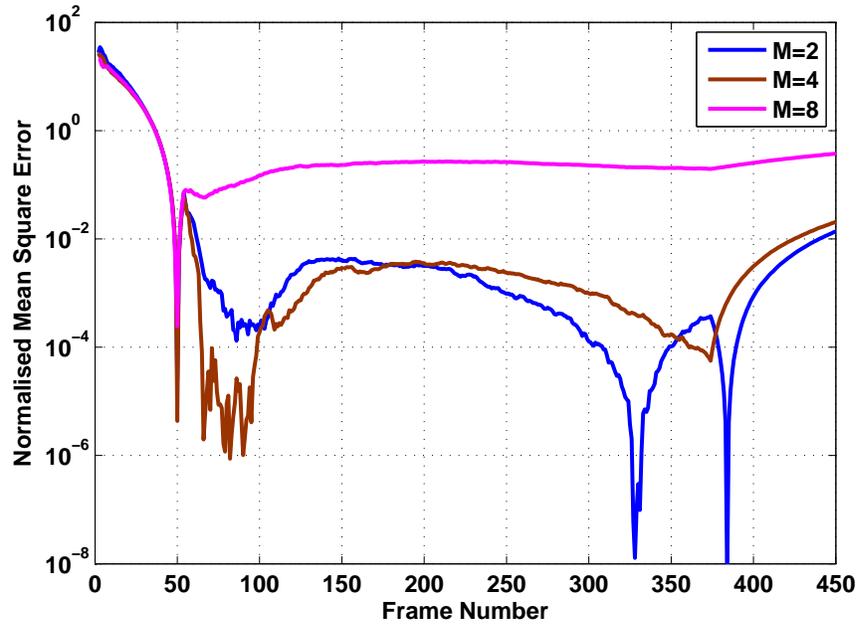


Figure 4.6: Normalised Mean Square Error for M-ARY FSK for $M=2,4,8$ for -8 dB SNR under *Rayleigh* flat fading. The traces for $M=2$ and 4 are less than 10^{-2} for the duration of the pulse. For $M=8$ it is almost constant at 10^{-1} as the γ curve is significantly different.

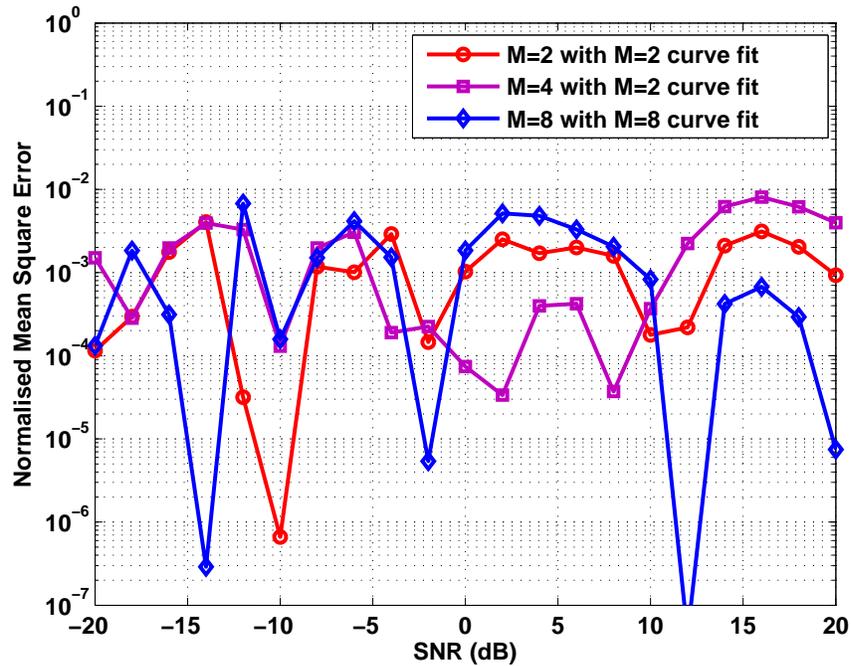


Figure 4.10: Normalised Mean Square Error for $M=2,4$ and 8 for Rayleigh flat fading with lookup table generated from $M=2$ and $M=8$ curve fitting.

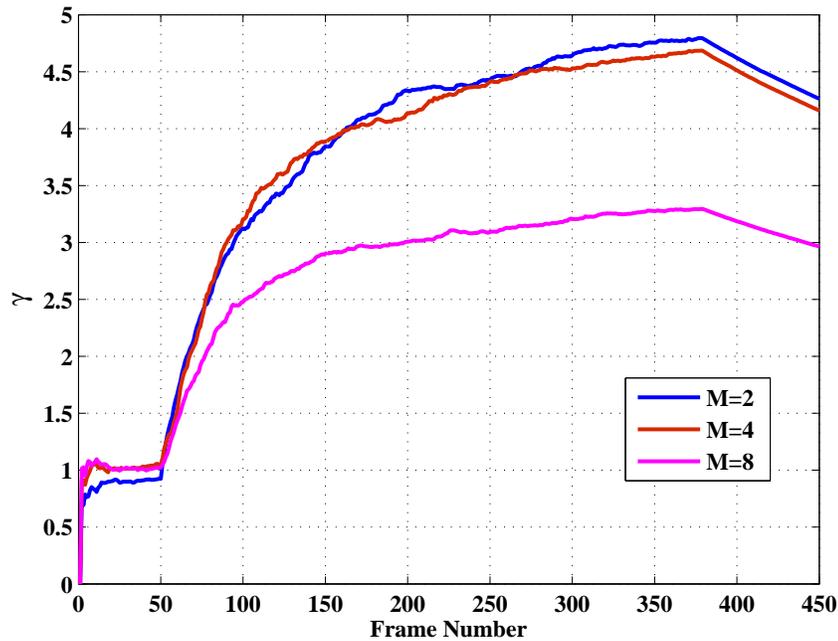


Figure 4.7: Plot of γ for BFSK for -8 dB SNR with *Rician* fading with K factor of 10. It may be noted that the curve for M = 2 and 4 are similar but for M=8 the trace is different. The curve fitting for M=2 holds for M=4 but not for M = 8

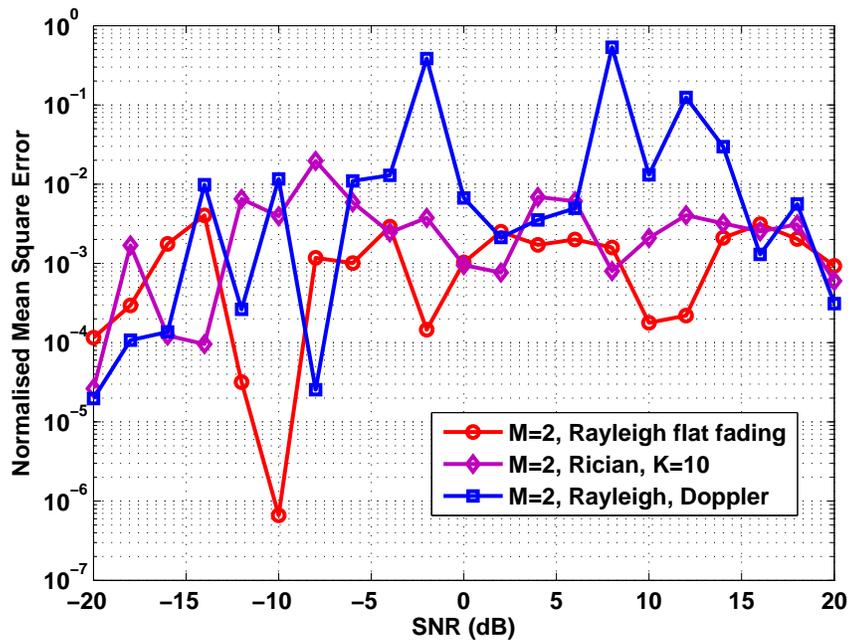


Figure 4.11: Normalised Mean Square Error for BFSK for Rayleigh fading with a Doppler of 20% of frequency deviation and Rician fading with a Doppler of 20% of frequency deviation and K = 10.

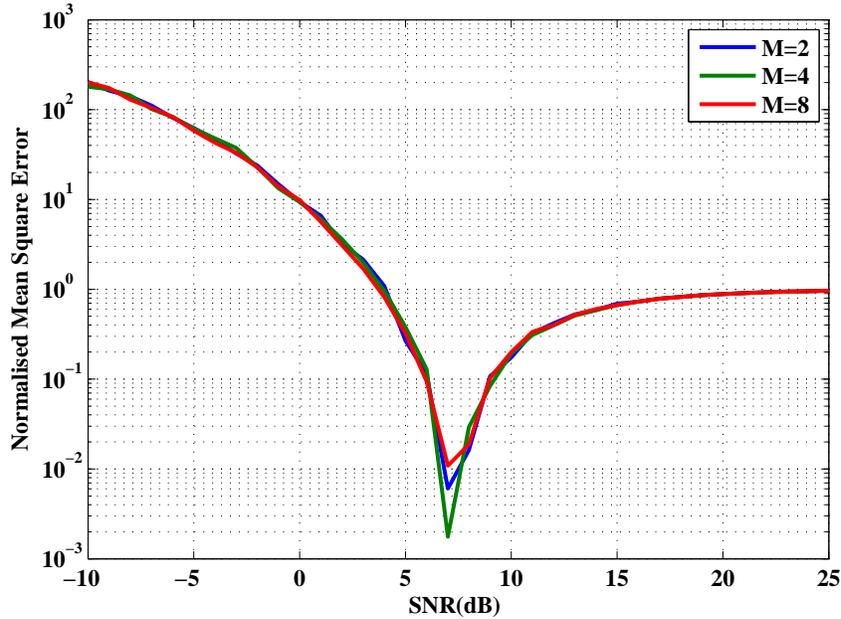


Figure 4.8: Partially Data Aided ML estimation of SNR for M-ARY FSK for $M=2,4,8$ for flat fading Rayleigh channel with a Doppler of 20 % of the frequency deviation.

We also estimated the SNR of the M-ARY FSK signal for $M=2,4$ and 8 using the fitting coefficients of $M=2,2$ and 8 respectively, and the NMSE is plotted in Fig 4.10. This figure attests that in all the three cases, the proposed algorithm performs well in the entire range of SNR. The NMSE is better than 10^{-2} for both negative and positive SNR.

In figures 4.5 - 4.6, the simulation assumed a flat fading Rayleigh channel i.e.; it is frequency flat with no Doppler shift. This shows up in the anomalous result that the performance of Rayleigh channel is as good as for the Rician channel. Fig 4.11 is a plot of the NMSE with Doppler shift of 20% of frequency deviation for Rayleigh and Rician channels ($K=10$). From this figure, it can be observed that for negative SNR, the performance in Rayleigh Channel is almost the same as for the Rician channel. This is because, at negative SNR, the Rician distribution has almost the same pdf (probability density function) as the Rayleigh distribution. However, at positive SNR, the NMSE in Rician fading gives better performance than Rayleigh fading due to the K factor.

4.6 Conclusions

SNR estimation for Rayleigh and Rician fading channels for M-ARY FSK signal with AWGN has been presented. The SNR was estimated using the test statistics of the signal. The proposed algorithm has an inherent feature of signal detection along with SNR estimation. The proposed algorithm gives satisfactory performance compared to PDA MLE for a wide range of SNR. The performance of the algorithm is evaluated under both Rayleigh and Rician channel fading conditions. The proposed method achieves NMSE better than 10^{-2} over an SNR range of -20 to +20 dB for M=2,4 and 8 whereas PDA MLE achieves an NMSE of 10^{-2} to 10^{-3} for positive SNR only.

Chapter 5

Real time implementation of spectrum sensing algorithms on Virtex 6 FPGA

5.1 Objective

Limited studies were reported on the hardware implementation of sensing algorithms. This chapter presents a systematic evaluation methodology for real-time implementation of spectrum sensing algorithms on FPGA. It consists of four steps i.e., floating point simulation, fixed point simulation, hardware-in-loop simulation and real-time test with instruments. The performance of Energy detection and CAV sensing algorithms are evaluated using this methodology. It further details SNR calibration and signal processing at baseband level.

5.2 Introduction

In addition to an enormous effort on the theoretical investigation of spectrum sensing algorithms, limited work on hardware implementation has been reported as well [85],[86],[87],[88]. As the primary function of CR is to detect vacant spectrum bands, it requires reconfigurable hardware platform. Software Defined radio is one such platform that enables implementation of CR functionalities. In SDR, radio communication is controlled by software defined protocols in contrast to hardware implementations [89]. Several works addressed experimental results with different test beds of energy detection and traced the effect of the noise uncertainty [90],[91],[92]. Different type of SDR platforms are used for evaluating the performance of spectrum sensing algorithm [93]. GNU radio along with USRP is the most popular among all [94],[95]. Apart from the SDR platform, other platforms such as the Helio board that houses Cyclone-V FPGA is reported. It executes a

GNU radio application to control the RF switch and it receives the digitized I/Q samples [96]. Berkeley Emulation Engine is being used to check the performance of energy detection method [57]. The block diagram of real-time spectrum sensing architecture is :

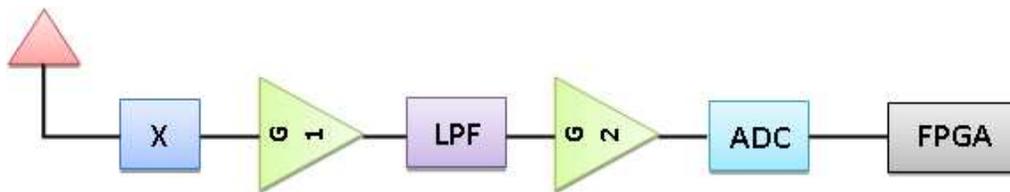


Figure 5.1: Block diagram of spectrum sensing architecture. The first block is a mixer for downconverting to IF, G1 and G2 are the Low Noise Amplifier(LNA) gains, LPF is a low pass filter, ADC is the Analog to Digital convertor and FPGA is a Field Programmable Gate Array.

Apart from the implementation of energy detection algorithm on the USRP platform, covariance based spectrum sensing with maximum eigen value detection is implemented in white space device (WSD) [97]. Power spectral density based sensing algorithm is implemented on the USRP platform [98]. Energy detection based spectrum sensing to sense TV band was implemented using Memece development board that uses Virtex-4 FPGA and Humax F3-Fox-T was used as a TV receiver [87]. Although energy detection algorithm does not work effectively for low SNR signals (where the variation of the noise floor exists) it is a popular method because of its low algorithmic complexity. The present work contributes in real-time implementation and validation of the Covariance Absolute Value (CAV) and energy detection algorithm on a Virtex-6 FPGA. The detection probability is evaluated at different signal to noise ratio (SNR) and reported.

5.3 Algorithm evaluation methodology

It is often encountered that performance predicted by simulation is not realised on actual target hardware. The difference in performance is due to the inadequate modeling of noise variance or finite word effects. Therefore, there is a need to evaluate algorithms in a structured manner so that comparison of efficiency, among competing spectrum sensing algorithms, could be made. This section presents a

detailed methodology for validating sensing algorithm on hardware platform. The methodology involves four different stages that are detailed below.

5.3.1 Step 1: Matlab Simulation

In the first step, the functionality of the sensing algorithm is simulated in floating point environment, using MATLAB tool. The Binary Frequency Shifted Keying (BFSK) modulation technique is used as modulation type. BFSK is a modulation in which the binary numbers are mapped to either a frequency deviation of $+\delta f$ or $-\delta f$. This results in a constant envelope signal which is ideally suited for wireless communication. In BFSK the transmitter amplifier non-linearities do not cause significant degradation in the BER (Bit Error Rate). The test signal is a complex number of the form $a + jb$. Similarly, complex white noise, whose variance can be changed, is generated for the same number of samples. The test signal for sensing is created by gating the BFSK signal with white noise. The gating is done such that BFSK signal is present in between two noise frames at the beginning and end of the frame. The durations of the frame can be varied to create different signals for simulation. This type of signal is created to mimic the transmission of a burst by the Primary User (PU). The noise in the receiver is assumed to be thermal noise given as:

$$W(\text{dBm}) = kTB \quad (5.1)$$

Where, k is the Boltzmann constant, $1.38 * 10^{-23}$ J/K, T is the noise temperature of the receiver and B is the signal bandwidth. Thus the noise is known for a given temperature and bandwidth. For example, the noise power for a temperature of 290 K and 1 MHz bandwidth is -114 dBm. The signal from the transmitter suffers attenuation and fade as it passes through the channel. The path loss P_L is given as

$$P_L(\text{dB}) = 20 \log_{10}(4\pi d/\lambda) \quad (5.2)$$

Where, λ and d are the wavelength and distance in meters respectively. Therefore to create different SNR, the signal level is scaled and added to the noise. The BFSK signal corrupted with additive white Gaussian noise is generated as (5.3) where, $s(n)$ and $x(n)$ are the baseband and scaled baseband signal respectively.

The received signal $y(n)$ is generated as

$$y(n) = x(n) + h(n)w(n) \quad (5.3)$$

where,

$$x(n) = \frac{\sigma_w}{\sigma_s} \sqrt{10^{SNR/10}} s(n) \quad (5.4)$$

$$\sigma_s = \frac{1}{N_s - 1} \sum_{n=1}^{N_s} (s(n) - \mu_s)^2 \quad (5.5)$$

$$\sigma_w = \frac{1}{N_s - 1} \sum_{n=1}^{N_s} (w(n) - \mu_w)^2 \quad (5.6)$$

$$(5.7)$$

Where, SNR is the signal to noise ratio in dB, μ_s , μ_w are the sample mean, σ_s and σ_w are the standard deviation of the signal and noise respectively. The scaled signal $x(n)$ is multiplied with channel gain $h(n)$. The signal with a fixed SNR is generated using Simulink model as shown in Fig 5.2.

The channel gain is a random variable which could be drawn from a Rayleigh or Rician distribution. In this evaluation the channel is assumed flat fading and therefore assumed as unity. During simulation, 450 random length frames, of BFSK modulated signal, is generated. Each frame has 512 samples. The SNR is varied from -20 dB to +10 dB. The algorithms are implemented using floating point precision. This step of simulation allows us to work with large amounts of data and good visualisation to optimise the algorithm to get the desired performance.

5.3.2 Step 2: Simulink simulation with Sysgen blocks (Fixed point simulation)

In the real-time environment, all the signal samples are not available at one time. The signal samples are processed sequentially. Therefore in Simulink the parallel data needs to be serialised to model the arrival of sequential samples and processed using a TDM (Time Division Multiplexing) approach. In this step, a Simulink model is created using Xilinx Sysgen blocks.

The Sysgen blocks are optimised RTL blocks, that gives cycle and bit perfect fixed point outputs. The top-level block diagram of the energy detection algorithm

is shown in Fig. 5.3. The System generator specifies the simulation parameters and system control. The Resource Estimator token is added to provide an estimate of the resources required on the FPGA to implement the model. It gives an estimate of the number of multipliers, flip-flops, block memory (BRAM) and lookup tables (LUT) required. The blocks shown in yellow colour are the Gateway blocks. These are placed between the Simulink and the Sysgen blocks. The Simulink model with sysgen blocks is built for the real-time algorithm. By using sysgen blocks there is greater predictability of the final real-time performance. The Resource estimator block enables to get the hardware resources required to implement the model. The Resource estimation for the energy detection algorithm is tabulated in Table. 5.1.

The Sysgen blocks run the equivalent fixed point simulation of sensing algorithms, whereas MATLAB simulation results are on floating point environment.

5.3.3 Step 3: Hardware Co-Simulation

The sensing is performed using ML605 Xilinx FPGA board [99]. It hosts a high performance Virtex-6 XC6VLX240T-1FFG1156 FPGA, 16 MB flash and 512 MB of DDR3 flash. It communicates with the computer on 10/100/1000 Tri speed Ethernet. The master clock runs at 200 MHz differential. A FMC 150 (FPGA Mezzanine Card) board with ADC and DAC features is plugged into the ML605 board. The FMC board has two channels of 14 bit ADC that can give up to 250 Msps. It has two channels of 16 bit DAC. The specifications of the FMC board are given in [60].

The System generator block is configured for co-simulation mode for the FPGA board ML605. The data flow between Simulink and the board are shown in Fig. 5.4. System generator is run to generate the bit stream for Virtex-6 FPGA using the ISE development tool. The bitstream is downloaded to the board through the JTAG interface. The JTAG interface is used for programming the FPGA and for debugging via the Chipscope analyser tool. The algorithm is now ported on to the FPGA and validated using hardware-in-loop simulation technique. The algorithm now runs on the FPGA with the inputs coming from the Simulink and outputs is viewed on Simulink. The results are compared with the simulation results of step 2.

The 14 bit ADC and 16 bit DAC on the FMC150 daughter card can operate at 250 and 800 Msps respectively. System Generator creates input and output

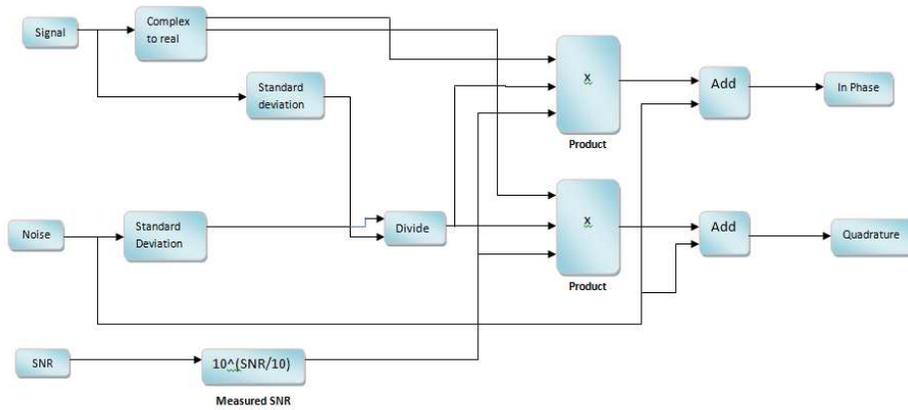


Figure 5.2: Block diagram of creation of Signal to Noise ratio in Xilinx System Generator. The signal is scaled and noise added to both In-phase and Quadrature-phase signals.

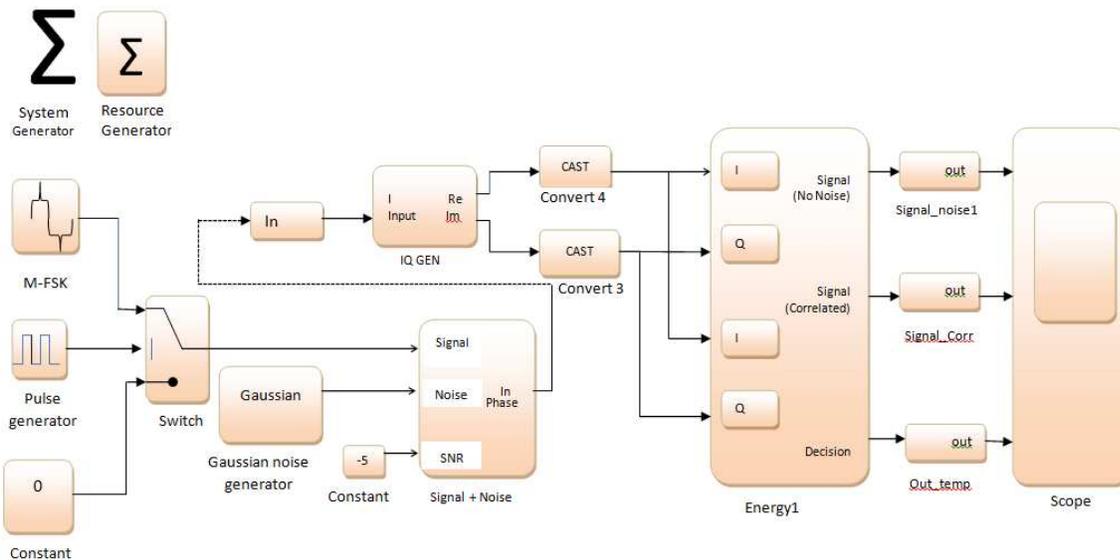


Figure 5.3: High level schematic of ED implementation using SYSGEN blocks.

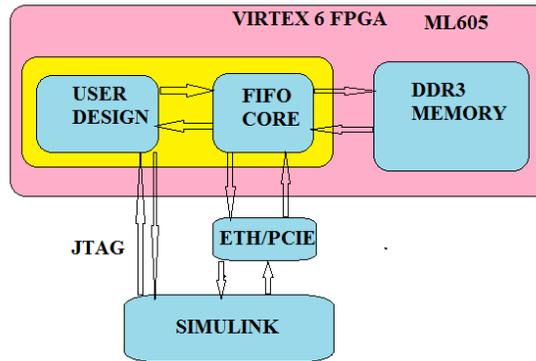


Figure 5.4: Data flows between Simulink and ML605 FPGA board for hardware co-simulation.

Table 5.1: FPGA resource estimation for energy detection

Number of Comparators	Qty	Number of Multiplexers	Qty
12-bit comparator greater	4	12-bit 2-to-1 multiplexer	2
13-bit comparator greater	50	13-bit 2-to-1 multiplexer	50
17-bit comparator greater	2	25-bit 2-to-1 multiplexer	2
18-bit comparator greater	6	27-bit 2-to-1 multiplexer	2
25-bit comparator greater	2	29-bit 2-to-1 multiplexer	2
38-bit comparator lessequal	2	30-bit 2-to-1 multiplexer	2
		38-bit 2-to-1 multiplexer	3
		69-bit 2-to-1 multiplexer	3

16x16-bit multiplier	2
27-bit adder	2
Flip-Flops	21383
16x16-to-33-bit MAC	2

FIFO buffers as the Ethernet/PCIe interface does not have enough bandwidth to transfer the data converter samples, typically in Mega samples/sec, directly to and from the board. Therefore in hardware co-simulation, the data is transferred in burst mode from Simulink to FPGA and back. Samples arriving from the FPGA are first buffered in FIFO and when it is full, the data is transferred to Simulink with a data-valid signal. This may lead to discontinuity in samples, however by implementing large FIFO, 16K words for both receive and transmit, the algorithm can be run for a enough time for evaluation purpose.

5.3.4 Step 4: Real time test with instrumentation

In this stage, the performance of sensing algorithms are tested in real-time using a RF signal generator. Fig. 5.5 is a block diagram of the test setup for real-time evaluation. The RF signal from the signal generator is burst modulated using an external symbol clock. The signal generator is set to pulsed mode, with BFSK modulation, a pulse period of 0.5 ms with a duty cycle of 50%. The required SNR is created by suitably attenuating the noise power in the noise generator as shown in Fig. 5.6. The noisy signal is fed to the FMC150 ADC/DAC board on FMC connector ML605 FPGA board. The ADC samples are processed by the algorithm running on the FPGA. The detection output is a discrete signal whose state is *high* if the signal is present, otherwise, it is *zero*. This discrete signal is connected to Channel *A* of the oscilloscope. The symbol clock is also fed to Channel *B* of the oscilloscope. Therefore Channel 1 and Channel 2 show the *actual burst* and the *detected burst* waveforms respectively. The ratio of the burst durations observed on the oscilloscope is used to calculate the P_d . The experimental setup is shown in Fig. 5.6.

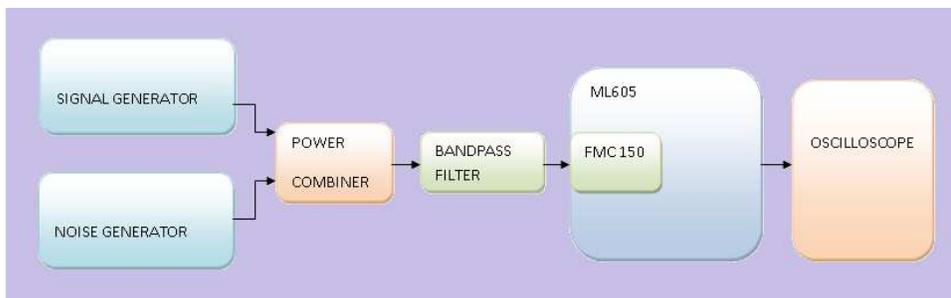


Figure 5.5: Functional setup for Real time performance evaluation.

A BFSK signal of bandwidth 200 KHz centered at 70 MHz is generated using Agilent 4430B signal generator. Random noise of fixed power is generated using a noise generator (RS SMBV 100A). The signal power is combined with the noise power in a power combiner (Pasternak 2000). The signal of the desired SNR is generated by keeping the noise power fixed while varying the signal power to the combiner using an attenuator. The test conditions of the signal and noise are given in Table 5.2.

The resultant signal of desired SNR is input to an analog bandpass filter of centre frequency 70 MHz with a bandwidth of 10 MHz. The filtered signal is input to



Figure 5.6: Realtime Spectrum Sensing algorithm evaluation Setup : (1) Noise Generator RS SMBV 100A, (2) Signal Generator Agilent 4430B, (3) Pasternak 2000 Power Combiner, 70 MHz BPF (4) Xilinx ML605 with Virtex-6 XC6VLX240T-1FFG1156 FPGA and FMC-150 ADC/DAC add-on card, (5) Simulink and ISE running on Laptop

an Add-On FMC 150 Mezzanine card [60] on the ML605 FPGA board [99]. Here it is sampled at 61.44 MHz. This results in an aliased signal at 8.56 MHz. The signal of interest is extracted by applying this to a digital bandpass filter centred at 8.56 MHz with a bandwidth of ± 3 MHz. The rate reduction is obtained by decimating the signal by a factor of 10. Due to the rate reduction the signal is effectively resampled with a sampling frequency of 6.144 MHz thus resulting in the signal getting translated to an Intermediate Frequency (IF) of 2.416 MHz. This input is fed to a mixer with a local oscillator frequency of 2.416 MHz. The local oscillator (for converting the IF signal to baseband) is generated by a high accuracy Direct Digital Synthesizer (DDS). The DDS consists of a phase accumulator which drives a high frequency Digital-to-Analog (DAC) Converter. Sine waves, with micro Hertz precision, can be produced by incrementing the accumulator with small phase steps and resetting it every 2π radians. The resulting baseband signal is low pass filtered with a corner frequency of 200 KHz. It is followed by a rate decimation by a factor of 10 for generating the baseband $I(n)$ and $Q(n)$ samples at 614.4 KHz rate. The signal flow is shown in Fig. 5.8.

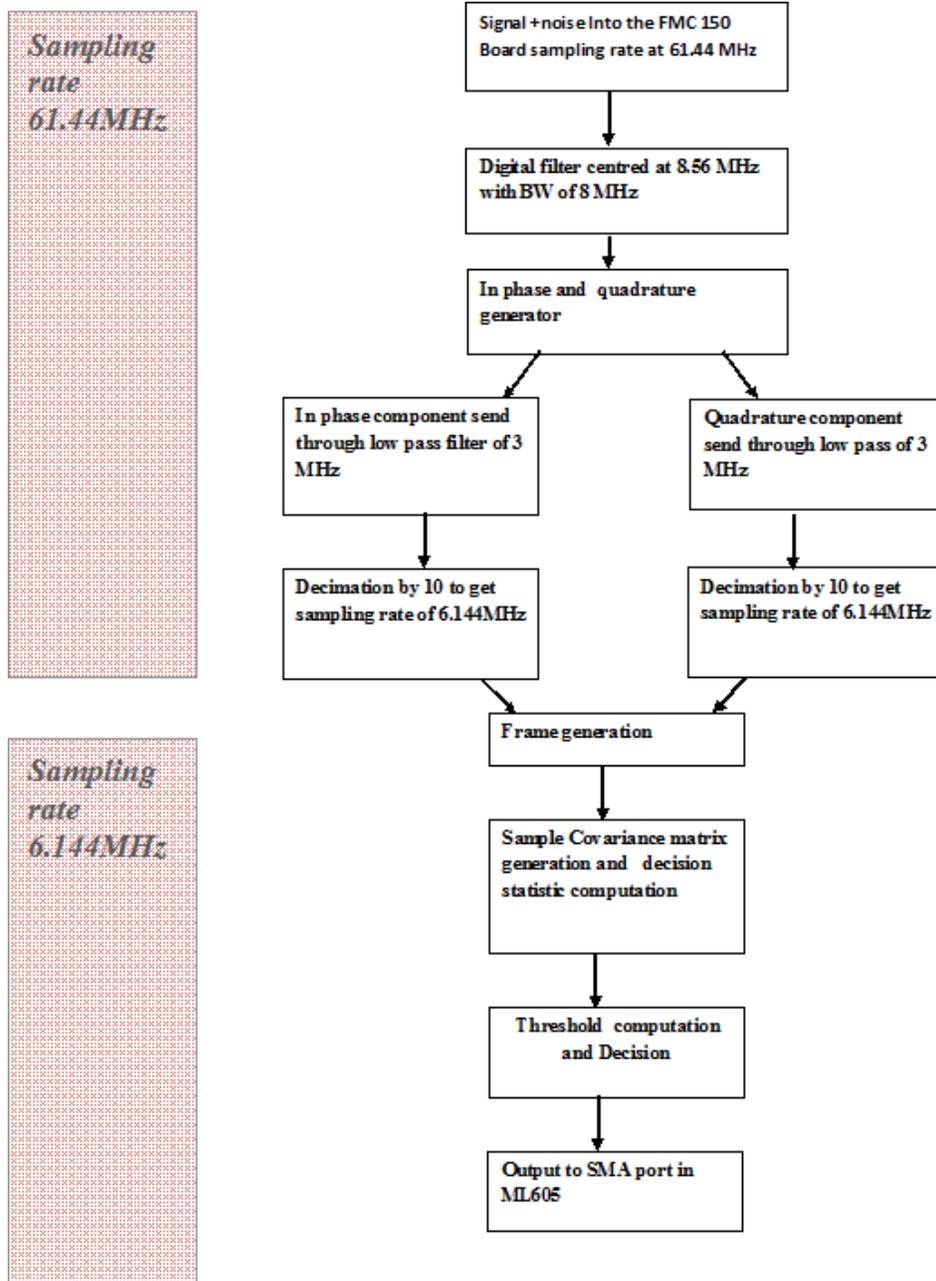


Figure 5.7: Flow chart of CAV implementation on Virtex 6 FPGA

Table 5.2: Signal capture conditions

sl	Description	Conditions
1	Signal frequency	70 MHz
2	Signal power level	-20 dBm
3	Noise Band width	6 MHz
4	70 MHz Bandpass filter	+/- 3 MHz @ - 3dB
5	70MHz BPF loss	-5.17 dB
6	Power Combiner loss	-3 dB
7	FFT	64 K
8	Sample rate	61.44 MHz
9	Aliased frequency	8.56 MHz

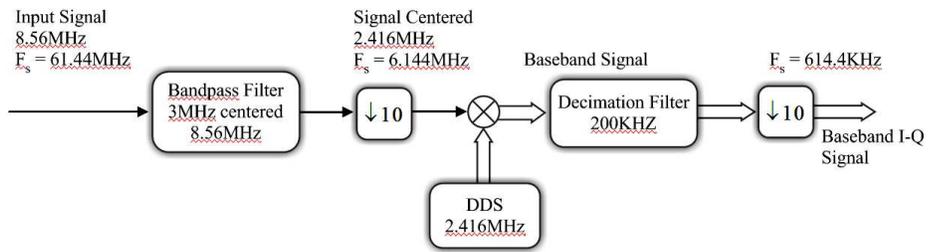


Figure 5.8: Intermediate frequency to Baseband signal processing

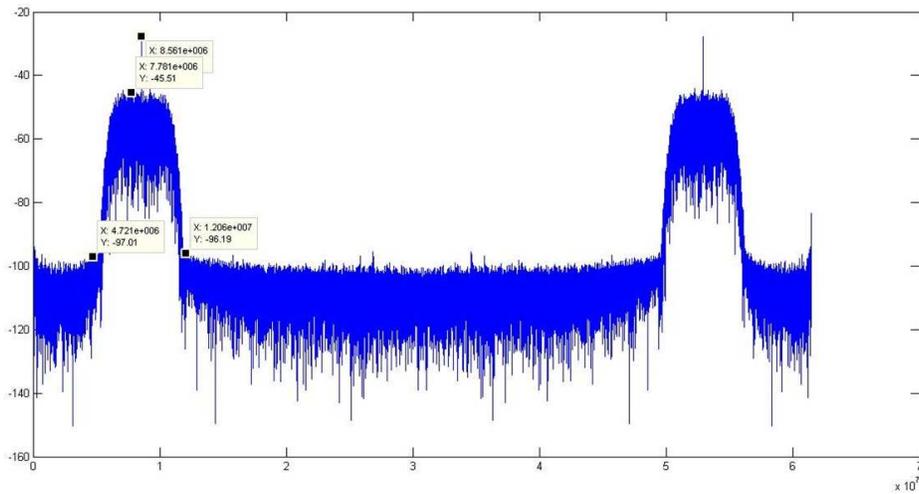


Figure 5.9: 64 K FFT of the noisy signal at an *SNR* of -11.5 dB

Table 5.3: SNR calibration : Actual vs Computed on FPGA

At Power Combiner			After FFT		
Signal Generator power, dBm	Noise Generator power, dBm	Input SNR, dB	Computed Signal Power, dBm	Computed Noise Power, dBm	Computed SNR, dB
-20	-8.27	-11.73	-28.17	-16.69	-11.48
-20	-13.27	-6.73	-28.17	-21.63	-6.54
-20	-18.27	-1.73	-28.17	-26.81	-1.36
-20	-23.27	3.27	-28.17	-31.73	3.56

5.4 SNR calibration

Calibration is carried out to ensure that the noise added to the signal is resulting in the desired SNR . To validate this, a 64K FFT is computed on the sampled data and the SNR is computed. The 64K FFT of the captured signal at SNR of -11.5dB is shown in Fig. 5.9. This figure indicates that the 70 MHz carrier is positioned at 8.56 MHz as expected. The agreement between actual and computed SNR is less than 0.4 dB as listed in Table 5.3.

5.5 Simulation and real-time test results

For algorithm simulation, random burst of a BFSK modulated signal of 450 frames, of 512 samples each, are created. The modulated signal power is normalised to unity. Flat Rayleigh fading is added to the signal, and the necessary SNR is created by adding random noise, of unit variance, to the scaled signal power. The SNR is varied from -20 dB to +10 dB. The $I(n)$ and $Q(n)$ is generated from the received signal, and the test statistics are computed for energy and CAV detection. The leading and trailing edges of the burst are detected. The first and second edges are indicating that the channel is occupied or vacant respectively. The number of frames between the two detected edges is the sensed length. The ratio of sensed length to total burst length gives the probability of detection (P_d). For the CAV simulation, the sample covariance matrix is computed and the ratios of T_1 and T_2 is computed. Rising and falling edges are detected when $T_1/T_2 > 1.1$ and < 0.8 respectively.

Table 5.4: Detection sensitivity vs number of sample per frame

Number of samples	ED	CAV
32	+2	-
64	0	-12
256	-4	-15
512	-4	-16
1024	-6	-16
2048	-8	-16

For the real-time burst generation, the signal generator is set to pulsed mode, with BFSK modulation. The gating clock is set to 1 KHz and this is fed to the external trigger of the signal generator. The output is therefore pulsed with an *ON* time of 0.5 ms and an *OFF* time of 0.5 ms. This gating clock is fed to the Channel A of the oscilloscope. The FPGA executes the sensing algorithm and outputs a discrete which goes high on the rising edge and goes low on the falling edge of the burst. This is fed to Channel B of the Oscilloscope. This is shown in Fig. 5.10. The ratio of the *ON* time of the input pulse and detected pulse is used to calculate the P_d . The detection sensitivity of the two schemes as a function of number of samples is tabulated in Table 5.4. It is observed that beyond 512 samples there is little improvement in sensitivity. Therefore all real-time implementations were carried out for a frame size of 512 samples.

5.6 Results and discussion

The two algorithms namely ED and CAV are implemented on the FPGA and the *SNR* vs P_d results are shown in Fig. 5.11. It is observed that $P_d = 0.9$ is achieved at SNR of -3 and -17 dB for ED and CAV algorithms respectively. For energy detection the threshold is fixed and is a function of required P_f and the number of samples per frame [57]. Therefore the detection performance falls suddenly below -3 dB SNR when the frame energy falls below the threshold. The *SNR Wall* phenomenon is valid for energy (radiometer) detection [30]. The sensing (locking) time for the two algorithms is shown in Fig. 5.12 plots. The ED and CAV take approximately 2-3 ms for execution. The processing time changes sharply and becomes large at the SNR where the algorithm starts failing. 2-3 ms delay is not significant considering that a burst lasts for tens of seconds. This result indicates

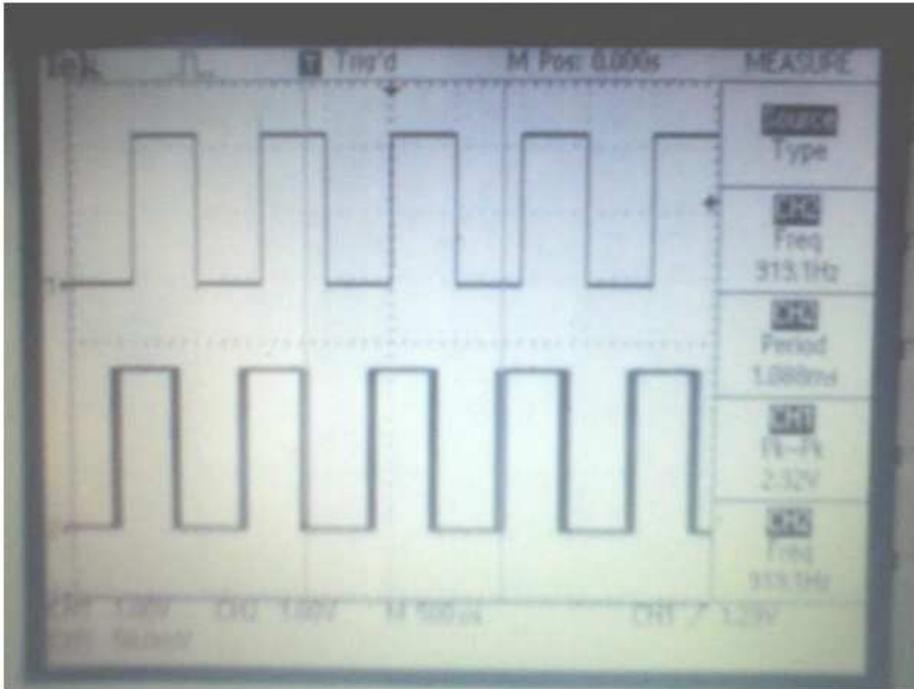


Figure 5.10: Real time spectrum sensing capture on digital oscilloscope

that about 1000 channels can be scanned in the 2 seconds allowed by the IEEE 802.22 standard [100]. The performance of CAV and ED algorithms for Matlab simulation, Simulink with SYSGEN blocks, Hardware Cosimulation and in Real time (RT) are shown in Fig. 5.13.

From this figure it is inferred that $P_d = 0.9$ occurs at SNR of -8 and -18 dB for ED and CAV algorithms respectively. The degradation from simulation to real-time is 5 dB for ED and less than 1 dB for CAV indicating its robustness. It is clear that the degradation in sensitivity from algorithm simulation to real-time can range from 1 to 5 dB. This margin has to be built into the system design to ensure reliable performance. Table 5.5 gives the resource utilization on the FPGA for the real-time implementation of the two algorithms. As expected the ED utilizes the least-resources followed by CAV. Fig. 5.14 shows the performance of ED and CAV algorithms for BFSK and DVBT signals. ED performs identically for BFSK and DVBT whereas CAV detects BFSK but fails to detect the DVBT signal. This can be attributed to the lack of correlation in the DVBT signal.

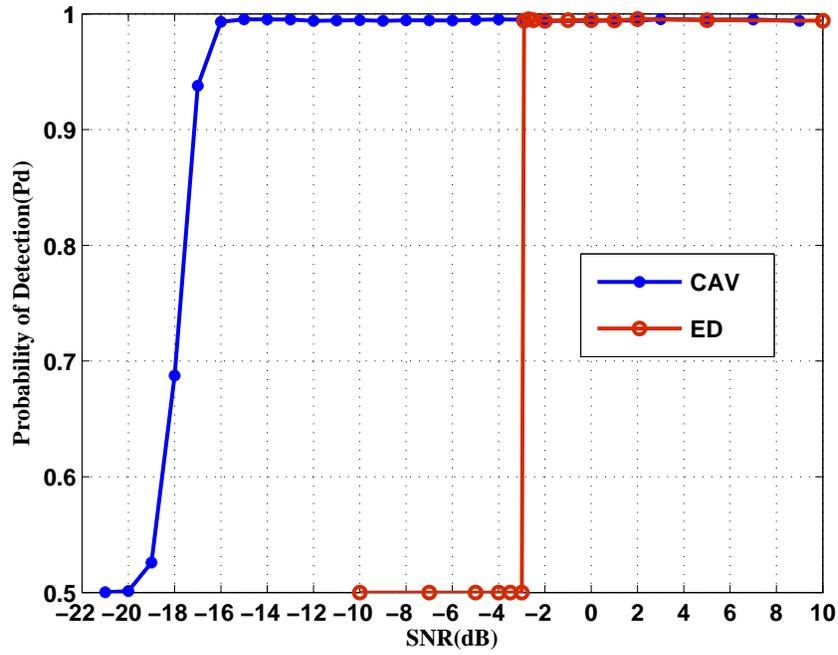


Figure 5.11: Pd vs SNR Comparison of two schemes in real-time on FPGA with pulsed input.

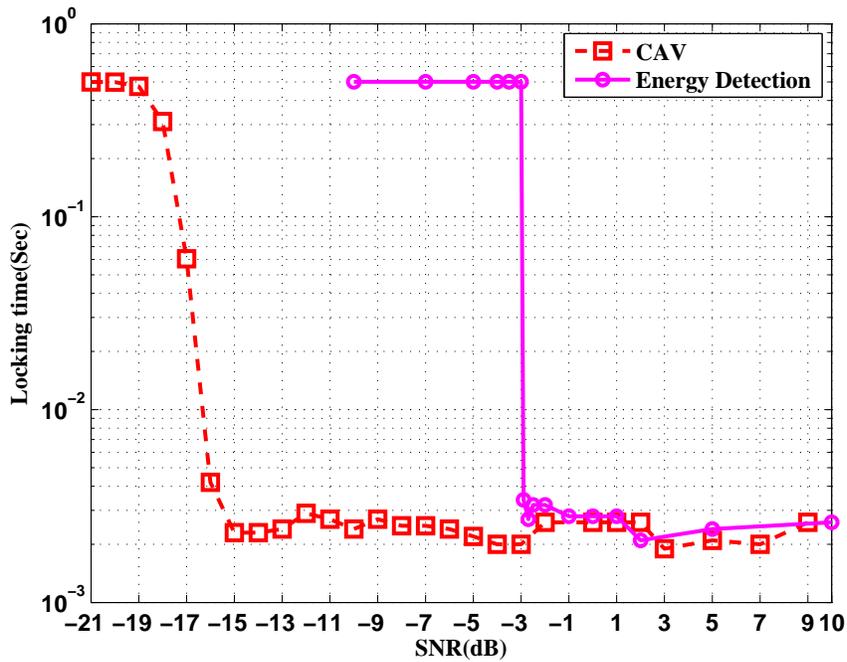


Figure 5.12: Sensing time vs SNR of the pulsed signal in real-time on FPGA

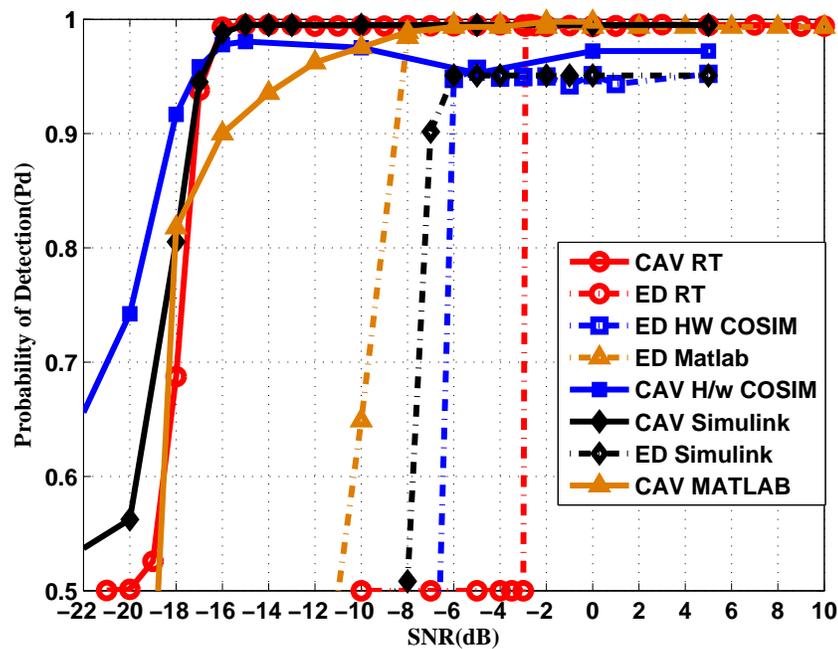


Figure 5.13: Pd vs SNR Comparison of ED and CAV schemes in all four stages

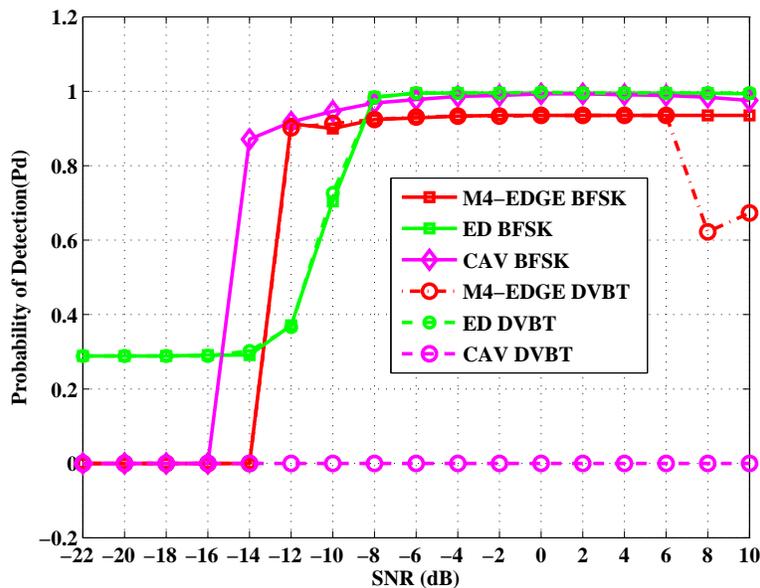


Figure 5.14: Pd vs SNR for ED and CAV for BFSK and DVBT. Note that ED works for both BFSK and DVBT whereas CAV fails in the case of DVBT , due to lack of correlation.

Table 5.5: Resource utilization on FPGA

Resource on Virtex 6 FPGA	Available	Utilization		Percentage	
		CAV	ED	CAV	ED
Number of Slice Registers	301,440	8,442	1800	2	1
Number of Slice LUTs:	150,720	8,502	1401	5	1
Number of occupied Slices:	37,680	2,552	519	6	1
Number of bonded IOBs:	600	76	25	12	4
Number of RAMB36E1/FIFO36E1s	416	0	0	0	0
Number of RAMB18E1/FIFO18E1s:	832	1	1	1	1
Number of DSP48E1s:	768	129	44	16	5

5.7 Conclusions

The CAV and ED algorithms are implemented on Virtex-6 FPGA. From the simulation and implementation results, it is concluded that the sensing time varies between 2 to 4 milliseconds within their SNR range of detection. Simulation on a DVBT signal demonstrates that ED algorithm can achieve SNR_{wall} of -8dB SNR, whereas CAV fails to detect, due to lack of correlation in the signal. However, when the signal is correlated, CAV outperforms the ED algorithm. The FPGA resource utilization is lower for ED compared to CAV. We have implemented two spectrum sensing algorithms on a Virtex-6 FPGA and have demonstrated a calibration method for proper evaluation of spectrum sensing algorithms in real-time. We have shown that real-time results could be many dB off from the simulation for ED whereas CAV performance is consistent. However, CAV would fail if the signal lacks correlation whereas ED algorithm would still perform.

**This page is intentionally left
blank**

Chapter 6

Spectrum sensing with envelope tracking and signal moment

6.1 Objective

The objective of this chapter is to develop an algorithm for improving the detection using a time domain approach. The approach is based on the concept that whenever the PU starts transmitting, a step change occurs in the signal strength. If this could be captured using envelope tracking, then detection is possible. To make the algorithm more robust, the signal moment could be tested against a threshold to confirm the change is signal strength. The detection performance is compared with the ED algorithm. Real-time validation results are also presented.

6.2 Introduction

In the case of Covariance Absolute Value (CAV) algorithm, detection is achieved by exploiting the correlation in the sample covariance matrix [16]. The CAV algorithm performs well for low SNR. However, it gives satisfactory performance only if the signal is correlated. Although the CAV technique works at low SNR , it requires a large number of samples which implies longer sensing time. However, CAV is efficient for sensing correlated signals whereas ED suffers from SNR Wall limitation. Other approaches for spectrum sensing include Discrete Wavelet transform and Cyclostationary technique [19],[101]. These techniques assume apriori knowledge of the signal. Hardware implementation of Energy detector on a wireless testbed is reported in [57] wherein sensing time was varied to achieve a predetermined P_d at low SNR regime. Hardware implementations of spectrum sensing algorithms are also reported on testbeds such as WSD, BEE2, USRP, and SDR [102],[103]. Few studies have focussed on sensing time for a given modulated signal of finite duration. Moment-based sensing algorithms were reported in the liter-

ature for spectrum sensing [43],[44]. In [28] noise variance was estimated using the optimal moment pair for improving the performance of the energy detection technique. The estimators were derived for BPSK and QAM constellations. In [45] a fourth order detector was derived for detection of linearly modulated signals. In [104] the authors assumed that the sum of interference is Gaussian, based on the Central Limit Theorem, whose kurtosis is zero. So the kurtosis is used as a test statistic to detect the signal. Generally, these detectors are tuned for a particular constellation. A p^{th} order detector for detection in independent Laplace noise for non-fading channels is investigated in [105]. A novel time-domain algorithm, named M4-Edge, is proposed to overcome the limitations of CAV and ED algorithms. The proposed algorithm tracks the envelope of the signal burst, of the primary user, in the time domain. The fourth central moment of the envelope is evaluated and compared with a threshold to detect the rising and falling edges of the burst and hence detecting the presence of the signal. Further, the algorithm is implemented on a Xilinx Vertex-6 Field Programmable Gate Array development board for evaluating its real-time performance. In real-time, the performance of the proposed algorithm is compared with ED and CAV algorithm by considering both BFSK and DVBT signal corrupted by Additive White Gaussian Noise (AWGN) and flat fading, as the Primary user signal to be sensed. The probability of detection, sensing time and resource utilisation are used as the metrics for measuring the efficiency of the algorithms.

6.2.1 M4-Edge

The algorithm is based on the idea that the presence of a signal can be declared by detecting the *rising – edge* and *falling – edge* of the signal burst. The algorithm has three steps namely:

6.2.1.1 Envelope tracking

The envelope of the *Mean*, mean tracking, *Upper*, maxima tracking and *Lower*, minimum tracking, of the frame envelope of the signal burst is computed and tracked. They are denoted as U_μ , U_U and U_L envelopes. The envelope tracking should react fast when there is an increase/decrease in the envelope, to reflect the start/end of the signal, and decay/ramp slowly when the signal falls/rises for a small duration, either due to fade or due to outliers in the additive noise[106].

1. The frame envelope is computed as:

$$U(k) = \sum_{n=1}^{N_s} (s(n)) \quad (6.1)$$

Where, k is the frame number, $s(n)$ is the n^{th} sample of the input signal, N_s is the number of samples per frame.

2. Initially the U_m , U_U and U_L are set as:

$$U_U(k) = U_m(k) = U_L(k) = U(k); \quad 0 < k < p; \quad (6.2)$$

3. The mean envelope, μ is given as:

$$U_\mu(k) = \frac{1}{p+1} \sum_{k-p}^k U(k) \quad (6.3)$$

where, p is the number of frames over which the mean is computed. Thus $\mu(k)$ tracks the mean envelope of the received signal.

4. From the $p+1^{\text{th}}$ frame, the value of the *Upper* envelope, U_U is set to latest computed value of the envelope $U(k)$, in case of an *increase* in the envelope compared to the previous value (*fast-attack*). Otherwise, the *Upper* envelope is allowed to *decay* slowly with a recursive low-pass filter of first order as per:

$$\delta(k) = |U_p(k)| - |U_U(k-1)| \quad (6.4)$$

$$|U_U(k)| = |U_p(k)| \quad ; \quad \delta(k) > 0 \quad (6.5)$$

$$U_U(k) = U_U(k-1)\alpha - U_p(k)(1-\alpha) \quad ; \quad \delta(k) \leq 0 \quad ; \quad 0 < \alpha < 1 \quad (6.6)$$

The envelope value, U_U is *censored* for removing outliers induced by noise, as follows:

$$\text{if } |U_U(k)|/|U_U(k-1)| > \beta$$

$$\text{then } |U_U(k)| = \beta U_\mu(k) \quad (6.7)$$

$$\text{else } |U_U(k)| = |U_U(k-1)|; \quad 0 < \beta < 2;$$

where, β is a censoring constant.

5. From the $p+1^{\text{th}}$ frame, the value of the *Lower* envelope, U_L is set to latest computed value of the envelope $U(k)$, in case of an *decrease* in the envelope compared to the previous value (*fast-attack*). Otherwise the *Lower* envelope is allowed to

ramp up slowly with a recursive low-pass filter of first order as per:

$$\gamma(k) = |U_L(k-1)| - |U_p(k)| \quad (6.8)$$

$$|U_L(k)| = |U_p(k)| \quad ; \quad \gamma(k) > 0 \quad (6.9)$$

$$U_L(k) = U_L(k-1)\alpha + U_p(k)(1-\alpha) \quad ; \quad \gamma(k) \leq 0 \quad ; \quad 0 < \alpha < 1 \quad (6.10)$$

6. The envelope value U_L is *censored* for removing outliers induced by noise, as follows:

$$\begin{aligned} & \text{if } |U_L(k-1)|/|U_L(k)| > \beta \\ & \text{then } |U_L(k)| = U_\mu(k)/\beta \\ & \text{else } |U_L(k)| = |U_L(k)| \end{aligned} \quad (6.11)$$

The detailed steps are shown in Algorithm 2. Fig. 6.1 plots the envelope tracking of the M4 EDGE algorithm at an SNR of -10 dB . It is clear that the envelopes are tracked even at low SNR because of the peak holding, first order filtering and censoring. From the figure it may be inferred that though the computed envelope looks entirely random in amplitude , the algorithm is able to track the signal burst.

6.2.1.2 Moments

In general the p^{th} central moment of a random variable y is given by

$$M_p(y) = \mathbb{E} [(y - \mu)^p] \quad (6.12)$$

where, $\mathbb{E}[\cdot]$ and μ denotes expectation value and sample mean respectively. The value of p as two or four corresponds to Variance and 4th Central Moment of the random variable y respectively. Even and odd central moments give information about the broadness and skew of the distribution respectively. In this step the fourth central moments of the *Upper, Lower and Mean* envelopes are computed and denoted as M_{4U} and M_{4L} and $M_{4\mu}$ respectively.

6.2.1.3 Decision

The burst of BFSK signal has a rising and a falling edge. Our aim is to detect the *rising* and *falling* edges and compare it with the transmitted burst. The presence of an edge is declared if the test statistic is greater than a threshold. Since this is a time domain approach for detection, intuitively the threshold must be dynamic

Input: $\alpha, \beta, \rho, \lambda$, Frame envelope U_P , Frame number, SNR, p, k
Output: risingedge, fallingedge

```

for  $SNRcount = 1$  to  $SNR_{max}$  do
  Compute frame envelope as (6.1)
  For first  $p$  frames, set  $Mean(U_\mu), Upper(U_U), Lower(U_L)$  envelopes as
  (6.2)
  Compute the  $Mean(U_\mu)$  as (6.3)
  Compute  $Upper(U_U)$  and apply censoring as (6.6) and (6.7) respectively.
  Compute  $Lower(U_L)$  and apply censoring as (6.10) and (6.11)
  respectively.
  Compute the test statistics  $M_{4U}$  and  $M_{4L}$  and  $M_{4\mu}$  as (6.12)
  Set threshold  $\rho$  as (6.13)
  Initialise  $kcount, edgcount=0$ 
  for  $kcount = 1$  to  $kmax$  do
    for  $M_{4\mu}(k) > \rho$  and  $M_{4U}(k) > \rho$  OR  $M_{4L}(k) > \rho$  do
       $edgcount \leftarrow edgcount + 1$ 
      if  $edgcount > 1$  then
        |  $fallingedge \leftarrow kcount$ 
      else
        |  $risingedge \leftarrow kcount$ 
      end
    end
     $kcount \leftarrow kcount + 1$ 
  end
   $SNRcount \leftarrow SNRcount + 1$ 
end
return  $risingedge, fallingedge$ 

```

Algorithm 2: M4-Edge detection algorithm

and related to the mean of the signal. The threshold ρ is set as:

$$\begin{aligned}
 & \rho = U_\mu(k)\lambda \\
 & \text{if } M_{4\mu}(k)/U_\mu(k) > 1; \text{ then } \lambda = 1 \\
 & \text{else } 0 < \lambda < 1
 \end{aligned} \tag{6.13}$$

where, λ is optimised by simulation.

The following conditions must be satisfied for at least 5 frames for declaring the detection of an edge:

$$\begin{aligned}
 & M_{4\mu}(k) > \rho \\
 & M_{4U}(k) > \rho \text{ OR } M_{4L}(k) > \rho
 \end{aligned} \tag{6.14}$$

Fig. 6.2 shows the input and output templates which are used to compute P_d . The first and second detected edge namely *rising* and *falling* edges of the burst are detected and their difference is denoted as the ‘*detected burst length*’. The ratio of ‘*detected burst length*’ to ‘*actual burst length*’, known apriori during simulation, is termed as Probability of Detection (P_d). The samples for which there is no signal but is detected as present gives the P_f .

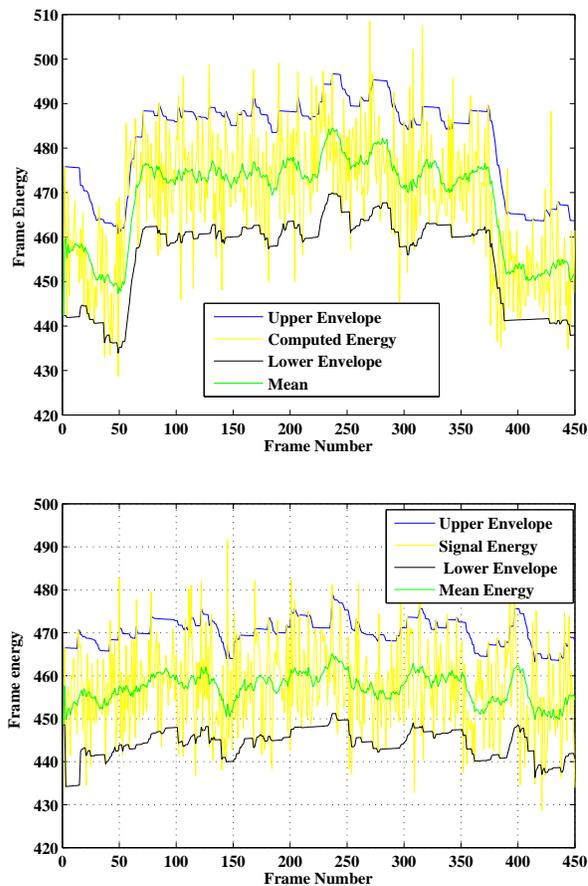


Figure 6.1: Envelope tracking at an SNR of -10 dB (upper figure) and -16 dB (lower figure). The signal burst is between frames 50 and 375. The envelope tracking algorithm is able to track the signal burst at an SNR of -10 dB, but is not able to track at -16 dB.

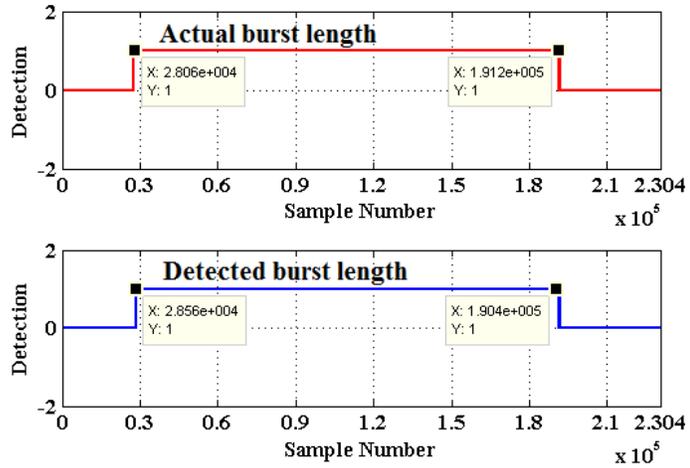


Figure 6.2: Plots of the *actual burst length* and *detected burst length*.

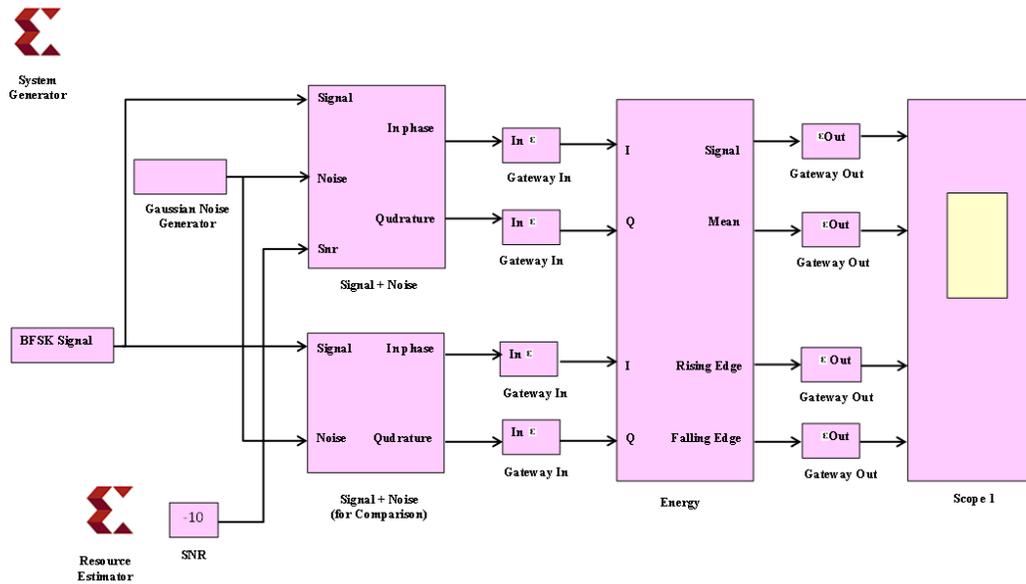


Figure 6.3: High level schematic of M4-Edge implementation using SYSGEN blocks.

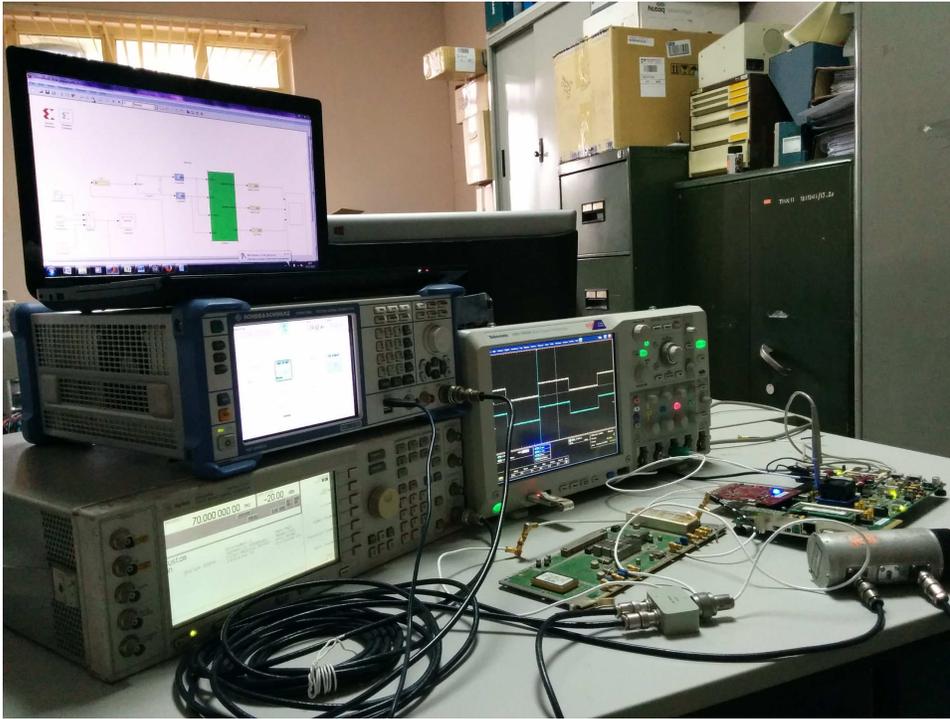


Figure 6.4: Real-time Spectrum Sensing evaluation Setup: (1) Noise Generator RS SMBV 100A, (2) Signal Generator Agilent 4430B, (3) Pasternak 2000 Power Combine, 70 MHz BPF and attenuator, (4) Xilinx ML605 with Virtex-6 XC6VLX240T-1FFG1156 FPGA and FMC-150 ADC/DAC add-on card, (5) ISE running on Laptop

6.3 Hardware details, calibration and signal processing

6.3.1 Hardware details

In this work, the sensing is performed using the ML605 Xilinx FPGA board [99]. It hosts a high performance Virtex-6 XC6VLX240T-1FFG1156 FPGA, 16 MB flash and 512 MB of DDR3 flash. It communicates with the computer on 10/100/1000 Tri-speed Ethernet. The master clock runs at 200 MHz differential. A FMC 150 (FPGA Mezzanine Card) board with ADC and DAC features is plugged into the ML605 board. The FMC board has two channels of 14 bit ADC that can give up to 250 MSPS. It has two channels of 16 bit DAC. The specifications of the FMC board are given in [60].

6.3.2 Simulation and Real-time test results

For simulation, 450 frames of a random burst of a BFSK modulated signal is generated. Each frame consists of 512 samples. The start and end of a burst are detected using the *rising* and *falling* edges. The modulated signal power is normalised to unity. Flat Rayleigh fading is added to the signal and the necessary *SNR* is created by adding random noise, of unit variance, to the scaled signal power. The *SNR* is varied from -20 dB to +10 dB. The constant λ and α is set to 0.2 and 0.99 respectively. β is set to 1.03 and 0.97 for upper and lower envelopes respectively. The high level schematic of M4-Edge implementation using Sysgen block is shown in Fig. 6.3

The test setup for real-time evaluation of M4-Edge algorithm is shown in Fig. 6.4. The CAV and ED algorithms have also been implemented on the ML605 FPGA board, details of which are given in Chapter 5.

6.4 Results and discussions

Fig. 6.5 is a plot of the fourth central moments of the *Upper*, *Mean* and *Lower* envelopes for a BFSK signal for a duration of 450 frames. Each frame has a length of 512 samples. The trace with the caption *Signal Burst* is a plot of the signal presence viz. the signal is present between frame number 50 to 375 and absent elsewhere. From this figure, it is observed that at the start and end of the signal burst, the moments are much greater than the threshold. Secondly, the moments have small values for the rest of the burst. The first and second instances, where the threshold is exceeded, are detected as the *rising* and *falling* edges of the burst respectively. It is also observed that the peak of the moments occurs with delays with respect to the rising and falling edges. The time constants used for envelope tracking introduce these delays. The values of the time constants are optimised to ensure to ensure a low P_f . The trace captioned *threshold* is the decision threshold, in the M4-Edge algorithm, for deciding between the binary hypotheses H_0 , signal absent, and H_1 , signal present. It is observed from the figure that the threshold is not constant but is adapted dynamically. It is also observed that the threshold increases at the *rising* and *falling* edges and elsewhere it is greater than the the moments. The first figure depicts a high *SNR* of + 10 dB wherein the moments are far greater than the threshold. The second one is at an *SNR* of -16 dB wherein

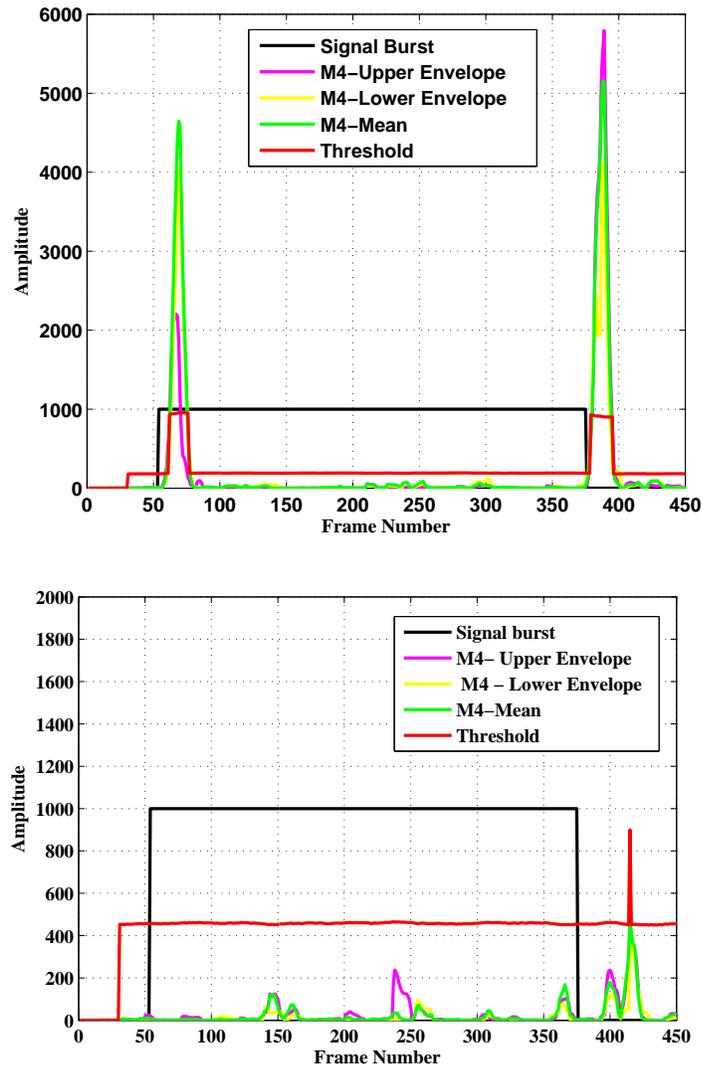


Figure 6.5: Plot of the Fourth Central Moment for SNR of -10 dB and -16 dB shown in upper and lower figures respectively. The signal burst begins at frame number 50 and ends at frame number 375. In the upper figure, note the delay in the M4 peaks from the leading edge and falling edge of the signal burst. This delay is due to the time constants used for envelope tracking.

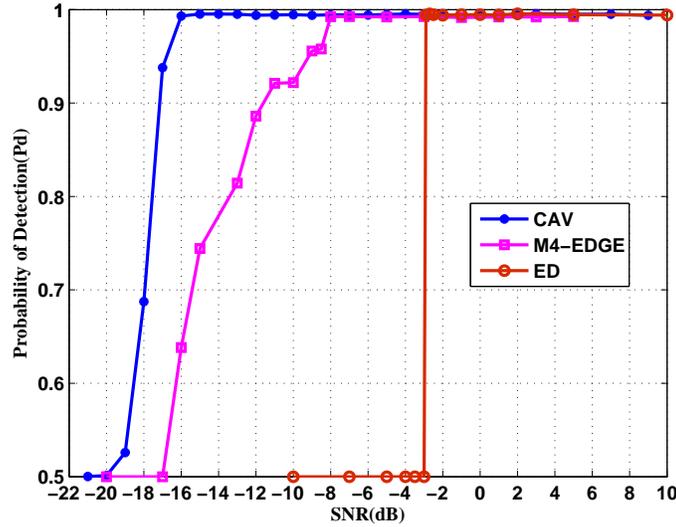


Figure 6.6: P_d vs SNR Comparison of three schemes in real-time on FPGA with pulsed input. $P_f = 0.1$

the moment at the rising edge is absent and the moments at the falling edge are below the threshold.

Three algorithms namely Energy detection, CAV and M4-Edge are implemented on the Virtex-6 FPGA and evaluated their real-time performance. Fig. 6.6 is a plot of the SNR vs P_d for BFSK signal. From this figure, it is observed that the ED, M4-Edge and CAV algorithms reach a P_d of 0.9 at an SNR of -3, -12 and -17 dB respectively.

For energy detection, the threshold is fixed and it is a function of only the number of samples per frame. Therefore the detection performance falls suddenly below -3 dB SNR when the frame energy falls below the threshold. The SNR Wall phenomenon [30] is valid for energy (radiometer) detection. However, in our proposed algorithm the change in the variance at the start and end of the burst is magnified to detect the burst. So this algorithm will work as long as the magnification produces a value significantly above the selected threshold. Although the proposed M4-Edge algorithm detects the signal upto SNR of -12 dB as compared to -17 dB for CAV, it does not require a signal to be correlated. Fig. 6.7 plots the sensing time of all the three considered algorithms. The ED and CAV take about 2-3 mS whereas the M4-Edge averages about 4 mS for faithful

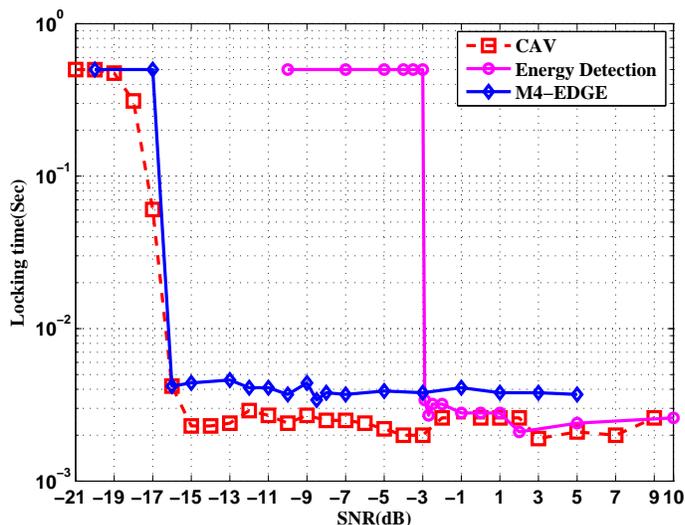


Figure 6.7: Sensing time vs SNR of the pulsed signal in real-time on FPGA. The sensing times are 2- 3 mSec for ED and CAV and 4 mSec for M4-Edge. The signal processing load for M4 edge is more and hence the larger processing delay.

detection. The processing time changes sharply and becomes large at the SNR where the algorithm starts failing. Considering the fact that in normal condition the burst to be detected will last for tens of seconds this delay is not significant. The result indicates that about 1000 channels can be scanned in 2 seconds, the Channel Detection time as per IEEE 802.22 standard.

Fig. 6.8 is a consolidated plot for the three algorithms using the 4 step methodology. The plot shows the results of each algorithm in four environment namely MATLAB, SIMULINK, HW Co-simulation and Real time (RT). Real time refers to the results obtained in real-time using FPGA board. From this figure, it is inferred that ED, M4 EDGE and CAV algorithm can detect SNR of -8, -15 and -18 dB respectively with $P_d = 0.9$ in a simulation environment (MATLAB). The degradation from simulation to real-time is 5 dB for ED, 3 dB for M4 EDGE and less than 1 dB for CAV indicating its robustness. It is clear that the degradation in sensitivity from Matlab simulation to real-time can range from 1 to 5 dB. This margin has to be built into the sensing system to ensure reliable performance. Table 6.1 gives the FPGA resource utilization for the real-time implementation of the three algorithms. As expected the ED utilizes the least-resources followed by CAV and M4-Edge. The resource usage of the M4-Edge can be explained by the need to track the envelopes, implement the censoring and decay equations and

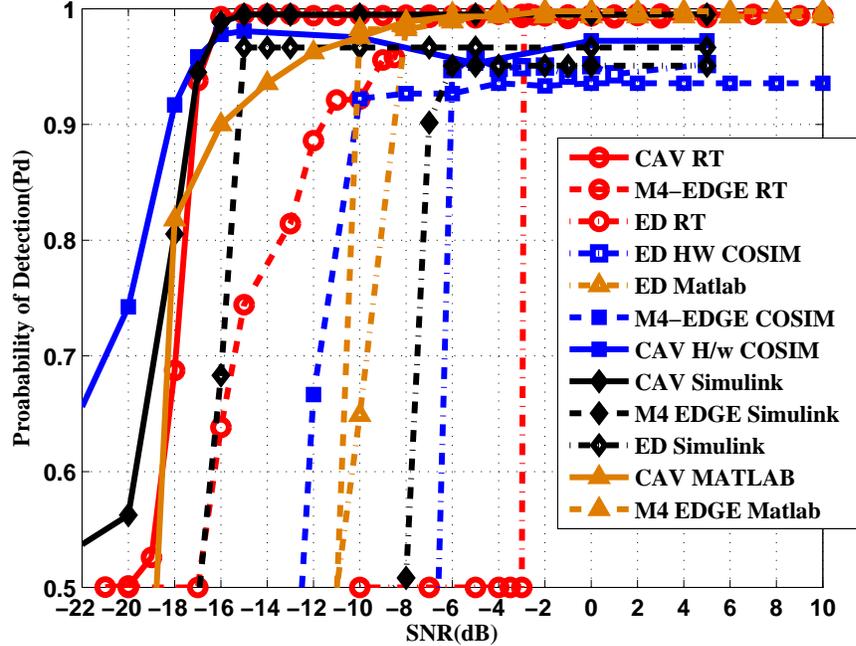


Figure 6.8: Pd vs SNR Comparison of three schemes in all four stages. This plot is combination of four plots. Note the spread of the traces , going from simulation to Real time. $P_f = 0.1$

calculate the moments. Fig. 6.9 is a plot of P_d as a function of SNR for Energy detection, CAV and M4-Edge algorithms for a DVBT (Digital Video Broadcast Terrestrial) signal. DVBT signal is an uncorrelated signal and used for comparing the algorithm efficiency. ED and M4-Edge perform identically for BFSK and DVBT whereas CAV detects BFSK and fails to detect the DVBT signal. It can be attributed to the lack of correlation in the DVBT signal. The CAV algorithm fails to give a good result for an uncorrelated signal (DVBT) as compared to the M4-Edge algorithm. Detection is achieved with a smaller number of samples by M4-Edge as compared to the ED technique (refer Fig.4 of [29]). The fourth central moment gives the sharpest peak and performance is consistent even at -12 dB SNR . We have also investigated the performance of the proposed algorithm with higher moments. However, as the order of the moment increases the peak tends to have a saddle like structure with multiple maxima and minima and thresholding becomes difficult at low SNR .

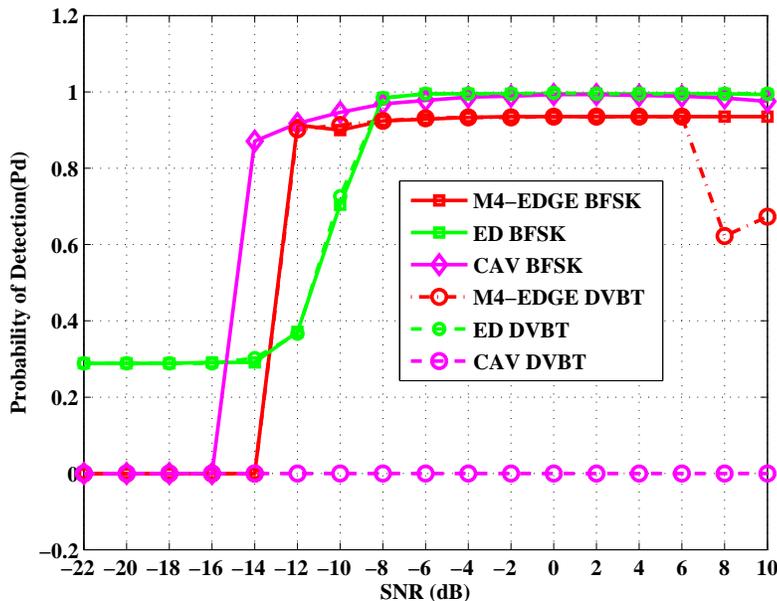


Figure 6.9: Pd vs SNR for ED, M4-Edge and CAV for BFSK and DVBT. Note that ED and M4-Edge work for both BFSK and DVBT whereas CAV fails in the case of DVBT, due to lack of correlation. $P_f = 0.1$

Table 6.1: Resource utilization of different algorithms on FPGA

sl	Resource on Virtex 6 FPGA	Available	Utilization			Percentage		
			CAV	M4 EDGE	ED	CAV	M4 EDGE	ED
1	Number of Slice Registers	301,440	8,442	24,214	1800	2	8	1
2	Number of Slice LUTs:	150,720	8,502	23,723	1401	5	15	1
3	Number of occupied Slices:	37,680	2,552	6,677	519	6	17	1
4	Number of bonded IOBs:	600	76	95	25	12	15	4
5	Number of RAMB36E1/FIFO36E1s	416	0	0	0	0	0	0
6	Number of RAMB18E1/FIFO18E1s:	832	1	1	1	1	1	1
7	Number of DSP48E1s:	768	129	362	44	16	47	5

6.5 Conclusions

The sensing time for all three algorithms varies between 2 to 4 milliseconds within their SNR capability envelopes. The FPGA resource utilization is lowest for ED and highest for the M4-Edge algorithm. The proposed algorithm outperforms ED and has equivalent performance to CAV in Signal-to-Noise-Ratio (SNR) range of -12 to +10 dB. The proposed algorithm has the additional benefit that it performs well when the signal is not correlated. The experimental results reveal that, in the case of the (Digital Video Broadcast Terrestrial) DVBT signal, the proposed

M4-Edge algorithm detects the signal with a probability of detection 0.9 at -12dB SNR , whereas CAV algorithm fails to detect, due to lack of correlation in the signal.

The proposed M4-Edge algorithm has shown performance better than ED and comparable to for correlated signal. However, CAV would fail if the signal lacks correlation whereas the proposed algorithm would still perform.

**This page is intentionally left
blank**

Chapter 7

Conclusions and Future work

7.1 Conclusions

This thesis has focussed on four facets of the spectrum sensing problem namely: improved Energy detection using noise power estimation, SNR estimation using covariance method, detection using envelope tracking and signal moment and real-time evaluation of the ED, CAV and M4-Edge algorithms on a Virtex 6 FPGA.

Two different noise power estimation techniques, namely Unbiased and LP, have been studied for improving the performance of Energy detection. It is shown that the LP estimator gives superior performance for white and blue noise, however its performance is weak in the case of red noise. The Unbiased estimator works uniformly for all types of noise but the improvement over simple energy detection is not significant. We have shown that the LP and unbiased estimators perform satisfactorily with Rayleigh fade which is expected in the real world. The study indicates that energy detection with these improvements could be used in situations where the noise type is known apriori to be white or blue.

The probability of energy detection with LP estimator is 5 and 6 dB better than the Unbiased estimator, and W/o estimation respectively, for additive white noise. The LP estimator outperforms the unbiased estimator at all sample sizes. For white noise, the Unbiased and W/o estimation performance is similar. The detection performance of the LP estimator is better than the Unbiased and W/o estimation for additive blue noise. The PSD for blue noise increases at the rate of 3 dB/octave. The LP estimator implements an inverse filter whose input is the signal autocorrelation and the output should be white. However, in the case of blue noise this condition is not met, and the LP estimate has an error resulting in lower P_d .

LP estimator works well with white and blue noise. However, it does not perform well with red noise. As the power in red noise falls off at the rate of 6 dB/octave, the LP estimator sees a varying noise, unlike the white noise case

Table 7.1: Detection performance at sample size of 512

Estimator	White Noise (dB)	Blue Noise(dB)	Red Noise(dB)	Rayleigh fade(dB)
Unbiased	-2	-2	+1	-4
LP	-7	-6	No detection	-8
ED	+1	0	+2	-2

where the power is the same at all frequencies. The LP estimator is based on the assumption that the all-pole filter produces the waveform when excited with a white noise input. Therefore the noise power estimate is not accurate.

Unbiased estimator works well with all types of noise, but the improvement is marginal with respect to W/o estimation. However, it is still able to detect in the case of additive red noise. ED without noise power estimation gives the same result irrespective of noise types.

The effect of Rayleigh fading has been investigated. Both estimators are consistent for a channel with and without fading. ED performance is marginally less with fade. The consistent performance could be attributed to the fact that the Rayleigh channel simulated is a flat fading (single path) channel, which introduces only a gradual change in the envelope. The amplitude variations induced by the fading have little impact on the computed peak energy.

The LP estimator is able to estimate the PSD in both cases, with and without fade, as the BFSK waveform has distinct peaks. However, it also implies that its performance for modulations that do not have distinct peaks may not be satisfactory. The results are summarised in the table 7.1.

The effect of noise variance estimation on multi-node cooperative sensing was also investigated. It was demonstrated that the detection performance and ROC improve significantly with noise variance estimation, regardless of the combining scheme used at the fusion centre. The performance for single node is compared with the cooperative sensing to demonstrate the improvement. On an average improvement of 3 dB or better was obtained in multinode sensing with noise variance estimation.

SNR estimation for Rayleigh and Rician fading channels for M-ARY FSK signal with AWGN is presented. The proposed algorithm has an inherent feature of signal detection along with SNR estimation. It gives satisfactory performance

compared to PDA MLE for a wide range of SNR. The performance of the algorithm was evaluated under both Rayleigh and Rician channel fading conditions. The proposed method achieved NMSE better than 10^{-2} over an SNR range of -20 to +20 dB for $M=2,4$ and 8 whereas PDA MLE achieves an NMSE of 10^{-2} to 10^{-3} for positive SNR only.

The numerical results demonstrated that the Normalized Mean Square Error (NMSE) of the proposed algorithm is better than the PDA MLE. The NMSE is consistently less than 10^{-2} over the SNR range -20 dB to +20 dB using 512 samples. Further, the algorithm can detect the presence of a signal with a probability of detection 0.9 up to -8 dB SNR without any extra computation. However, the detection performance can be improved by increasing the number of samples. The proposed algorithm can be used for signal detection and SNR estimation for a broad range of SNR.

The performance of the ED and CAV algorithms, was compared, in the real-time testbed, by considering both BFSK and DVBT signal corrupted by Additive White Gaussian Noise (AWGN) and flat fading. The details of the software implementation and firmware are detailed. The implementation results were analysed based on the parameters like sensing time and logic blocks utilisation. Further, the algorithms were implemented on a Xilinx Virtex-6 Field Programmable Gate Array development board for evaluating their real-time performance.

The sensing time for all three algorithms varies between 2 to 4 milliseconds within their *SNR* capability envelopes. The FPGA resource utilization is lowest for ED and highest for the M4-Edge algorithm. The M4-Edge algorithm outperforms ED and has equivalent performance to CAV in Signal-to-Noise-Ratio (*SNR*) range of -12 to +10 dB. It has the additional benefit that it performs well when the signal is not correlated. The experimental results reveal that, in the case of the Digital Video Broadcast Terrestrial (DVBT) signal, the M4-Edge algorithm detects the signal with a probability of detection 0.9 at -12dB *SNR*, whereas CAV algorithm fails to detect, due to lack of correlation in the signal. M4-Edge algorithm has shown performance better than ED and is comparable to CAV algorithm for correlated signal. However, CAV fails if the signal lacks correlation whereas the M4-Edge algorithm still performs. A systematic approach to evaluate the real-time performance of spectrum sensing algorithms is demonstrated.

Two spectrum sensing algorithms were implemented on a Virtex-6 FPGA and a calibration method for proper evaluation of spectrum sensing algorithms in real-

time was implemented. The real-time results are many dB off from the simulation result in case of ED, whereas CAV algorithm gives consistent performance.

The probability of detection, sensing time and resource utilization were used as the metrics for measuring the efficiency of algorithm. It was concluded that ED consumes less resource whereas M4-Edge algorithm detects presence of signal with $P_d=0.9$ up to -12 dB.

7.2 Future work

1. It is required to use off-the-air RF signals to test spectrum sensing algorithms. The signal received suffers attenuation and multipath fading in the channel. The receiver front-end adds noise to the signal. Synchronisation to the carrier frequency is also required. So there is a need to study the impact of these factors on the detection performance of spectrum sensing algorithms.
2. A variety of algorithms are available for spectrum sensing with varied implementation complexity. However, if the channel SNR is known then the most suitable and low complexity algorithm can be used which could improve detection time. So adaptation of spectrum sensing algorithm with respect to channel SNR is an area of study.
3. Compressive sensing is a new area of research for spectrum sensing. It requires a large bandwidth receiver which can analyse a large number of channels in one capture. Since channel occupancy is very low, the resulting matrix has a *sparse* nature. This sparse nature is exploited to reduce the number of samples required for detection. Implementation of compressive sensing is an area of study.
4. Cognitive radio assumes an underlying SDR platform. The cognitive engine decides on the strategy to implement: waveform, power or algorithm, to achieve robust detection. Therefore future study is required for real-time implementation and testing on an SDR platform.

References

- [1] D. George, T. Hatt, and et.al, “Global mobile trends,” tech. rep., GSMA Intelligence, Sep 2017.
- [2] P. Kolodzy, P. Tenhula, L. Van Wazer, M. Marcus, M. McLaughlin, R. Engelman, D. Furth, K. O. Ham, E. Kwerel, K. Larson, C. Murphy, J. Williams, and J. Wong, “Spectrum policy task force report,” tech. rep., FCC, Nov. 2002.
- [3] J. Yang, “Spatial channel characterization for cognitive radios,” Master’s thesis, University of California, Berkeley, 2004.
- [4] M. H. Islam, C. L. Koh, S. W. Oh, X. Qing, Y. Y. Lai, C. Wang, Y. C. Liang, B. E. Toh, F. Chin, G. L. Tan, and W. Toh, “Spectrum survey in singapore: Occupancy measurements and analyses,” in *Proceedings of the 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom 2008)*, pp. 1–7, May 2008.
- [5] V. Valenta, R. Marlek, G. Baudoin, M. Villegas, M. Suarez, and F. Robert, “Survey on spectrum utilization in europe: Measurements, analyses and observations,” in *Proceedings of the Fifth International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pp. 1–5, June 2010.
- [6] A. Agarwal, A. S. Sengar, R. Gangopadhyay, and S. Debnath, “A real time measurement based spectrum occupancy investigation in north-western india for cognitive radio applications,” in *Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 2035–2039, March 2016.
- [7] A. Agarwal, A. Sengar, and R. Gangopadhyay, “Spectrum occupancy prediction for realistic traffic scenarios: Time series versus learning-based models,” *Journal of Communications and Information Networks*, pp. 1–7, April 2018.

- [8] S. Haykin, “Cognitive radio: brain-empowered wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 201–220, Feb 2005.
- [9] J. Mitola and G. Q. Maguire, “Cognitive radio: making software radios more personal,” *IEEE Personal Communications*, vol. 6, pp. 13–18, Aug 1999.
- [10] D. M. Alias and R. G. K, “Cognitive radio networks: A survey,” in *Proceedings of International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 1981–1986, March 2016.
- [11] R. Tandra, S. M. Mishra, and A. Sahai, “What is a spectrum hole and what does it take to recognize one?,” *Proceedings of the IEEE*, vol. 97, pp. 824–848, May 2009.
- [12] H. Urkowitz, “Energy detection of unknown deterministic signals,” *Proceedings of the IEEE*, vol. 55, pp. 523–531, April 1967.
- [13] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Springer, 1994.
- [14] K. Kim, I. A. Akbar, K. K. Bae, J. S. Um, C. M. Spooner, and J. H. Reed, “Cyclostationary approaches to signal detection and classification in cognitive radio,” in *Proceedings of the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 212–215, April 2007.
- [15] S. V. Nagaraj, “Entropy-based spectrum sensing in cognitive radio,” *Signal Processing*, vol. 89, no. 2, pp. 174 – 180, 2009.
- [16] Y. Zeng and Y.-C. Liang, “Spectrum-sensing algorithms for cognitive radio based on statistical covariances,” *IEEE Transactions on Vehicular Technology*, vol. 58, pp. 1804–1815, May 2009.
- [17] T. Yucek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” *IEEE Communications Surveys Tutorials*, vol. 11, pp. 116–130, First 2009.

- [18] R. Rao, Q. Cheng, and P. K. Varshney, "Subspace-based cooperative spectrum sensing for cognitive radios," *IEEE Sensors Journal*, vol. 11, pp. 611–620, March 2011.
- [19] Z. Tian and G. B. Giannakis, "A wavelet approach to wideband spectrum sensing for cognitive radios," in *Proceedings of the 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pp. 1–5, June 2006.
- [20] L. S. Cardoso, M. Debbah, P. Bianchi, and J. Najim, "Cooperative spectrum sensing using random matrix theory," in *Proceedings of the 3rd International Symposium on Wireless Pervasive Computing*, pp. 334–338, May 2008.
- [21] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *Proceedings of the IEEE International Conference on Communications*, vol. 4, pp. 1658–1663, June 2006.
- [22] L. B. Jiang and S. C. Liew, "Improving throughput and fairness by reducing exposed and hidden nodes in 802.11 networks," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 34–49, Jan 2008.
- [23] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *Proceedings of the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*, pp. 131–136, Nov 2005.
- [24] K. Shuaib, E. Barka, N. Al Hussien, M. Abdel-Hafez, and M. Alahmad, "Cognitive radio for smart grid with security considerations," *Computers*, vol. 5, no. 2, 2016.
- [25] X. Chen and S. Nagaraj, "Entropy based spectrum sensing in cognitive radio," in *2008 Wireless Telecommunications Symposium*, pp. 57–61, April 2008.
- [26] F. Digham, M.-S. Alouini, and M. K. Simon, "On the energy detection of unknown signals over fading channels," in *Proceedings of the IEEE International Conference on Communications, ICC '03.*, vol. 5, pp. 3575–3579, May 2003.

- [27] V. Kostylev, “Energy detection of a signal with random amplitude,” in *Proceedings of the IEEE International Conference on Communications, ICC*, vol. 3, pp. 1606–1610, 2002.
- [28] T. Cui, J. Tang, F. Gao, and C. Tellambura, “Moment-based parameter estimation and blind spectrum sensing for quadrature amplitude modulation,” *IEEE Transactions on Communications*, vol. 59, pp. 613–623, February 2011.
- [29] R. Tandra and A. Sahai, “Snr walls for signal detection,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, pp. 4–17, Feb 2008.
- [30] R. Tandra and A. Sahai, “Fundamental limits on detection in low SNR under noise uncertainty,” in *Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 1, pp. 464–469, June 2005.
- [31] H. Tang, “Some physical layer issues of wide-band cognitive radio systems,” in *Proceedings of the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*, pp. 151–159, Nov 2005.
- [32] D. Cabric, S. M. Mishra, and R. W. Brodersen, “Implementation issues in spectrum sensing for cognitive radios,” in *Proceedings of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 772–776 Vol.1, Nov 2004.
- [33] D. He, W. Li, F. Zhu, and W. Lin, “An enhanced covariance spectrum sensing technique based on stochastic resonance in cognitive radio networks,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 818–821, May 2012.
- [34] Y. Zeng and Y. . Liang, “Eigenvalue-based spectrum sensing algorithms for cognitive radio,” *IEEE Transactions on Communications*, vol. 57, pp. 1784–1793, June 2009.
- [35] L. Du, M. Laghate, C. Liu, D. Cabric, and Y. Chen, “Improved eigenvalue-based spectrum sensing via sensor signal overlapping,” in *Proceedings of the 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, pp. 122–126, June 2016.

- [36] R. R. Nadakuditi and A. Edelman, "Sample eigenvalue based detection of high-dimensional signals in white noise using relatively few samples," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 2625–2638, 2008.
- [37] A. Ikram and A. Rashdi, "Complexity analysis of eigenvalue based spectrum sensing techniques in cognitive radio networks," in *Proceedings of the 18th Asia-Pacific Conference on Communications (APCC)*, pp. 290–294, Oct 2012.
- [38] Z. Ye, J. Grosspietsch, and G. Memik, "Spectrum sensing using cyclostationary spectrum density for cognitive radios," in *Proceedings of the IEEE Workshop on Signal Processing Systems*, pp. 1–6, Oct 2007.
- [39] M. Ghozzi, F. Marx, M. Dohler, and J. Palicot, "Cyclostationarity-based test for detection of vacant frequency bands," in *Proceedings of the IEEE Int. Conf. Cognitive Radio Oriented Wireless Networks and Commun. (Crowncom)*, pp. 1 – 5, 07 2006.
- [40] W. A. Gardner, "Exploitation of spectral redundancy in cyclostationary signals," *IEEE Signal Processing Magazine*, vol. 8, pp. 14–36, April 1991.
- [41] S. Narieda, "Computational complexity reduction for signal cyclostationarity detection based spectrum sensing," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, May 2017.
- [42] X. Liu, Z. Zeng, and C. Guo, "Cyclostationary spectrum sensing based channel estimation using complex exponential basis expansion model in cognitive vehicular networks," in *20th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 167–172, Dec 2017.
- [43] D. Pauluzzi and N. Beaulieu, "A comparison of SNR estimation techniques for the awgn channel," *IEEE Transactions on Communications*, vol. 48, pp. 1681–1691, Oct 2000.
- [44] M. Alvarez-Diaz, R. Lopez-Valcarce, and C. Mosquera, "SNR estimation for multilevel constellations using higher-order moments," *IEEE Transactions on Signal Processing*, vol. 58, pp. 1515–1526, March 2010.

- [45] J. Renard, J. Verlant-Chenet, J.-M. Dricot, P. De Doncker, and F. Horlin, "Higher-order cyclostationarity detection for spectrum sensing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, no. 1, pp. 721–695, 2010.
- [46] M. Gupta, G. Verma, and R. K. Dubey, "Cooperative spectrum sensing for cognitive radio based on adaptive threshold," in *Proceedings of the Second International Conference on Computational Intelligence Communication Technology (CICT)*, pp. 444–448, Feb 2016.
- [47] H. Qin, Y. Du, and J. Su, "Fault-tolerant algorithm for distributed primary detection in cognitive radio networks," in *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 1, pp. 353–356, April 2009.
- [48] H. Li, X. Cheng, K. Li, C. Hu, N. Zhang, and W. Xue, "Robust collaborative spectrum sensing schemes for cognitive radio networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 2190–2200, Aug 2014.
- [49] Y. Li, R. Zhang, C. Wang, A. Daneshmand, and X. You, "A distributed relay selection algorithm for cognitive radio ad-hoc networks," in *Proceedings of the International Conference on Network and Service Management*, pp. 439–442, Oct 2010.
- [50] X. Liu, Z. Li, X. Liu, and Y. Hu, "Optimal cooperative spectrum sensing in cognitive radio networks," in *Proceedings of the IET International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1–5, Dec 2012.
- [51] 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, November 2008.
- [52] K. W. Choi, E. Hossain, and D. I. Kim, "Cooperative spectrum sensing under a random geometric primary user network model," *IEEE Transactions on Wireless Communications*, vol. 10, pp. 1932–1944, June 2011.

- [53] Sesham Srinu, “Entropy based Reliable Cooperative Spectrum Sensing for Cognitive Radio Networks,” in *Ph.D Thesis, University of Hyderabad*, Sep 2013.
- [54] Z. Chair and P. K. Varshney, “Optimal data fusion in multiple sensor detection systems,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-22, pp. 98–101, Jan 1986.
- [55] E. C. Y. Peh, Y. C. Liang, Y. L. Guan, and Y. Zeng, “Cooperative spectrum sensing in cognitive radio networks with weighted decision fusion schemes,” *IEEE Transactions on Wireless Communications*, vol. 9, pp. 3838–3847, December 2010.
- [56] S. Nallagonda, Y. R. Kumar, and P. Shilpa, “Analysis of hard-decision and soft-data fusion schemes for cooperative spectrum sensing in rayleigh fading channel,” in *Proceedings of 7th International Advance Computing Conference (IACC)*, pp. 220–225, Jan 2017.
- [57] A. Tkachenko, D. Cabric, and R. Brodersen, “Cognitive radio experiments using reconfigurable bee2,” in *Fortieth Asilomar Conference on Signals, Systems and Computers*, pp. 2041–2045, Oct 2006.
- [58] Xilinx, “ML 605 datasheet.” https://www.xilinx.com/support/documentation/boards_and_kits/ug534.pdf, 2012.
- [59] Pasternak, “PE 2000 Power Combiner datasheet.” <https://www.pasternack.com/2-way-bnc-reactive-power-divider/2-mhz-500-mhz-1-watts-pe2000-p.aspx>, 2012.
- [60] Xilinx, “FMC 150 datasheet.” <http://www.4dsp.com/FMC150.php>, 2012.
- [61] D. Makovoz, “Noise variance estimation in signal processing,” in *Proceedings of the IEEE International Symposium on Signal Processing and Information Technology*, pp. 364–369, Aug 2006.
- [62] D. R. Joshi, D. C. Popescu, and O. A. Dobre, “Adaptive spectrum sensing with noise variance estimation for dynamic cognitive radio systems,” in *Proceedings of the 44th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–5, March 2010.

- [63] A. Ahmed, Y. F. Hu, and J. M. Noras, “Noise variance estimation for spectrum sensing in cognitive radio networks,” *Proceedings of the AASRI Conference on Circuit and Signal Processing (CSP)*, vol. 9, pp. 37–43, 2014.
- [64] J. G. Proakis, C. L. Nikias, C. M. Rader, F. Ling, M. Moonen, and I. K. Proudler, *Algorithms for Statistical Signal Processing*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1st ed., 2001.
- [65] K. K. Paliwal, “Estimation of noise variance from the noisy ar signal and its application in speech enhancement,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 292–294, Feb 1988.
- [66] J. P. Burg, “The relationship between maximum entropy spectra and maximum likelihood spectra,” *GEOPHYSICS*, vol. 37, no. 2, pp. 375–376, 1972.
- [67] A. Mariani, A. Giorgetti, and M. Chiani, “Effects of noise power estimation on energy detection for cognitive radio applications,” *IEEE Transactions on Communications*, vol. 59, pp. 3410–3420, December 2011.
- [68] M. Barkat, *Signal detection and estimation*. Artech House, 1991.
- [69] R. Storn and K. Price, “Differential evolution : A simple and efficient heuristic for global optimization over continuous spaces,” *J. of Global Optimization*, vol. 11, pp. 341–359, Dec. 1997.
- [70] Y. Huang and J. A. Ritcey, “Exit chart analysis of BICM-ID over AWGN channels with SNR mismatch,” *Communications Letters*, vol. 8, pp. 532–534, Aug 2004.
- [71] T. A. Summers and S. G. Wilson, “SNR mismatch and online estimation in turbo decoding,” *IEEE Transactions on Communications*, vol. 46, pp. 421–423, Apr 1998.
- [72] J. Hua, L. Meng, Z. Xu, and G. Li, “An adaptive signal-to-noise ratio estimator in mobile communication channels,” *Digital Signal Processing*, vol. 20, no. 3, pp. 692 – 698, 2010.
- [73] K. Wang and X. Zhang, “Blind noise variance and SNR estimation for OFDM systems based on information theoretic criteria,” *Signal Processing*, vol. 90, no. 9, pp. 2766 – 2772, 2010.

- [74] N. Yuanfei, G. Jianhua, and W. Yong, "Iterative SNR estimation using apriori information," *Digital Signal Processing*, vol. 19, no. 2, pp. 278 – 286, 2009.
- [75] F. X. Socheleau, A. Aissa-El-Bey, and S. Houcke, "Non data-aided SNR estimation of OFDM signals," *IEEE Communications Letters*, vol. 12, pp. 813–815, November 2008.
- [76] F. Bellili, R. Meftahi, S. Affes, and A. Stephenne, "Maximum likelihood SNR estimation of linearly-modulated signals over time-varying flat-fading SIMO channels," *IEEE Transactions on Signal Processing*, vol. 63, pp. 441–456, Jan 2015.
- [77] T. Benedict and T. Soong, "The joint estimation of signal and noise from the sum envelope," *IEEE Transactions on Information Theory*, vol. 13, pp. 447–454, Jul 1967.
- [78] T. Cui, J. Tang, F. Gao, and C. Tellambura, "Moment-based parameter estimation and blind spectrum sensing for quadrature amplitude modulation," *IEEE Transactions on Communications*, vol. 59, pp. 613–623, February 2011.
- [79] S. A. Hassan and M. A. Ingram, "SNR estimation for M-ARY non-coherent frequency shift keying systems," *IEEE Transactions on Communications*, vol. 59, pp. 2786–2795, October 2011.
- [80] U. Javed and S. A. Hassan, "SNR estimation for non-coherent M-FSK receivers in Rice fading environment," *IEEE Communications Letters*, vol. 17, pp. 1786–1789, September 2013.
- [81] S. A. Hassan and M. A. Ingram, "SNR estimation for a non-coherent M-FSK receiver in a Rayleigh fading environment," in *IEEE International Conference on Communications ICC*, pp. 1–5, May 2010.
- [82] S. Dan and G. Lindong, "A blind SNR estimator for digital bandpass signals," in *Proceedings of 6th International Conference on ITS Telecommunications*, pp. 1334–1337, June 2006.

- [83] M. R. Manesh, A. Quadri, S. Subramaniam, and N. Kaabouch, “An optimized SNR estimation technique using particle swarm optimization algorithm,” in *Proceedings of 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1–6, Jan 2017.
- [84] W. Jiang, W. Gao, J. Wang, D. Lu, and Q. Zhou, “SNR estimation of lfm signal based on the auto-correlation’s main lobe in time-domain fitted by fourier series,” in *Proceedings of Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 179–184, Oct 2016.
- [85] Zhe Chen and Nan Guo and Qiu, R.C., “Demonstration of Real-Time Spectrum Sensing for Cognitive Radio,” *IEEE Communications Letters*, vol. 14, no. 10, pp. 915–917, 2010.
- [86] D. Cabric, “Cognitive radios: System design perspective,” in *Ph.D Thesis, University of California, Berkeley*, 2007.
- [87] Povalac, K. and Marsalek, R. and Baudoin, G. and Sramek, P., “Real-time implementation of periodogram based spectrum sensing detector in TV bands,” in *Proceedings of the 20th International Conference, Radioelektronika*, pp. 1–4, April, 2010.
- [88] Nilanjan Byabarta, “Realization of Some Cognitive Radio Aspects Using WARP: A Reconfigurable FPGA Based Hardware Platform,” in *Masters Thesis, Jadavpur University*, pp. 33–46, May 2012.
- [89] Z. Tong, M. S. Arifianto, and C. F. Liau, “Wireless transmission using universal software radio peripheral,” in *Proceedings of the International Conference on Space Science and Communication*, pp. 19–23, Oct 2009.
- [90] S. Koley, V. Mirza, S. Islam, and D. Mitra, “Gradient-based real-time spectrum sensing at low snr,” *Communications Letters, IEEE*, vol. 19, pp. 391–394, March 2015.
- [91] Azza, M.A. and El Moussati, A. and Barrak, R., “Implementation of Cognitive Radio applications on a Software Defined Radio platform,” in *Proceeding of the International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 1037–1041, April 2014.

- [92] G. Chaitanya, P. Rajalakshmi, and U. Desai, "Real time hardware implementable spectrum sensor for cognitive radio applications," in *Proceedings of the International Conference on Signal Processing and Communications (SPCOM)*, pp. 1–5, July 2012.
- [93] M. Nayak, U. Bhanja, D. Parida, D. Dash, and K. D. Sa, "A real time implementation of spectrum sensing system using software defined radio," in *Proceedings of International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)*, pp. 603–607, July 2017.
- [94] M. A. Sarijari, A. Marwanto, N. Fisal, S. K. S. Yusof, R. A. Rashid, and M. H. Satria, "Energy detection sensing based on GNU radio and USRP: An analysis study," in *Proceedings of the IEEE 9th Malaysia International Conference on Communications (MICC)*, pp. 338–342, Dec 2009.
- [95] Y. Gao and Z. Qin, "Implementation of compressive sensing with real-time signals over tv white space spectrum in cognitive radio," in *84th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, Sept 2016.
- [96] D. Mazzer, M. M. Bontempo, A. M. Alberti, and A. Cerqueira Sodr, "Low-cost software-defined wireless cognitive network based on real-time multi-sector spectrum sensing and reconfigurable antenna array," *Microwave and Optical Technology Letters*, vol. 58, no. 8, pp. 1929–1934.
- [97] S.W.Oh, T. Le, W.Zhang, S. Ahmed, Y.Zeng, and K.J.M.Kua, "TV white-space sensing prototype," in *Wireless Communication and Mobile Computing*, vol. 9, pp. 1543–1551, Dec 2008.
- [98] O. Mian, R. Zhou, X. Li, S. Hong, and Z. Wu, "A software-defined radio based cognitive radio demonstration over fm band," in *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, IWCMC '09*, (New York, NY, USA), pp. 495–499, ACM, 2009.
- [99] Xilinx, "UG534 (v1.8)," ML605 Hardware User Guide," 2012.
- [100] "IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 22: Cognitive wireless ran medium access control MAC and physical layer PHY specifications: Policies and

- procedures for operation in the TV bands,” *IEEE Std 802.22-2011*, pp. 1–680, July 2011.
- [101] K. Kim, I. Akbar, K. Bae, J.-S. Um, C. Spooner, and J. Reed, “Cyclostationary approaches to signal detection and classification in cognitive radio,” in *Proceedings of the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 212–215, April 2007.
- [102] S. W. Oh, A. Naveen, Y. Zeng, V. P. Kumar, T. P. C. Le, K. Kua, and W. Zhang, “White-space sensing device for detecting vacant channels in TV bands,” in *Proceedings of the 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pp. 1–6, May 2008.
- [103] A. Mate, K.-H. Lee, and I.-T. Lu, “Spectrum sensing based on time covariance matrix using GNU radio and USRP for cognitive radio,” in *Proceedings of IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1–6, May 2011.
- [104] S. Suresh, S. Prakriya, and M. R. Bhatnagar, “Kurtosis based spectrum sensing in cognitive radio,” *Physical Communication*, vol. 5, no. 3, pp. 230 – 239, 2012.
- [105] X. Zhu, Y. Zhu, Y. Bao, and W. Zhu, “A p-th order moment based spectrum sensing for cognitive radio in the presence of independent or weakly correlated laplace noise,” *Signal Processing*, vol. 137, Jan 2017.
- [106] M. Marzinzik and B. Kollmeier, “Speech pause detection for noise spectrum estimation by tracking power envelope dynamics,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, pp. 109–118, Feb 2002.