NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

HARDWARE EFFICIENT ALGORITHMS AND ARCHITECTURES FOR BURST COMMUNICATIONS IN COGNITIVE RADIOS

SYED NAVEEN ALTAF AHMED

SCHOOL OF COMPUTER ENGINEERING

2018

Page Intentionally
Left Blank

HARDWARE EFFICIENT ALGORITHMS AND ARCHITECTURES FOR BURST COMMUNICATIONS IN COGNITIVE RADIOS

SYED NAVEEN ALTAF AHMED

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial in fulfilment of the requirement for the degree of
Doctor of Philosophy

2018

SYED NAVEEN ALTAF AHMED

Page Intentionally
Left Blank

Acknowledgement

This research work has been a long journey for me. I have learnt a number of things while on this path, both technical and personal lessons which have become part of me for good. I am glad I took this path and have been fortunate to have a number of people aid and motivate me reach this point. I would like to forward them my words of appreciation here.

First and foremost I would like to thank my supervisors, Assoc. Prof A.S. Madhukumar, Prof. Vinod A Prasad and Dr. Pramod Kumar Meher, to whom I will forever be in debt for accepting me as their student. Assoc, Prof. Vinod A Prasad and Dr. Pramod Kumar Meher have provided me with timely technical guidance and moral support over the years. Even with their multiple work and personal commitments they had unbelievable patience and have given me constant encouragement, without which I would no doubt have given up on this work, along the way to here.

Secondly, I would like to thank my family, my parents, Sayed Moosa Ahamed and Fathima Ahamed, my wife, Munasira Naveen, and my lovely kids Misha and Aiza. I feel deeply grateful to my parents who have been encouraging me to take this path for a long time now. My wife, who was very supportive of me during this period, putting up with my mood swings and had to be on her own many a days, when I had to stay late at work or attend classes in the evenings. And my kids, who do not yet understand why I had to

Page Intentionally
Left Blank

go to work even on weekends to catch-up for lost time, but still ask why I have office on weekends, when they have holiday at school.

Thirdly, I want to thank my brothers, sisters and cousins, especially Syed Mubeen, for all the encouragement over our weekend chats. And my paternal and maternal uncles and aunts, who live back in India, who by causally enquiring about my progress whenever I visited them, reminded me to keep going with this work.

Fourthly, I want to thank my colleague, friend and lunch buddy, Dr. Raymond Jayaraj, who has been reminding me, how close I am to the finish line and it is not over till I have crossed it. His approach towards his work and life as a whole helped me realize things which make a difference in the long-run.

At times, it had been difficult at work, especially when we had secured a few industry funded projects with challenging delivery schedules. In this regard, I would like to thank my seniors and friends at work Dr. Ashok Kumar Marath and Mr. Santhosh K. Pilakkat for their understanding and encouragement to complete this work.

Lastly I want to thank also a few of my colleagues and friends at work and from back in India, for the camaraderie, constant encouragement and at times help with few things which I tend to momentarily forget: namely, Dr. Nasimuddin, Dr. Sivanand, Dr. Faeyaz, Dr Ishtiaq Rasool, Mr. Jin Bo, Mr. Zhang Weiqiang, Ms. May Thu Zin, Dr. Lee Kee, Dr. Peter John Green, Mr. Simon Chen Jian, Ms. Indira, Ms. Meena, Mr. Saravanan Assoc. Prof. Rambabu, Mr. M. Ramesh and last but not least Mr. Badrinarayanan.

Page Intentionally
Left Blank

Table of Contents

Page Intentionally
Left Blank

List of Figures

List of Tables

Page Intentionally
Left Blank

List of Acronyms

ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AGC	Automatic Gain Control
AoA	Angle of Arrival
ATSC	Advanced Television Systems Committee
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CAF	Cyclic Autocorrelation Function
CAV	Covariance Absolute Value
CAZAC	Constant Amplitude Zero Autocorrelation
CFO	Carrier Frequency Offset
CFR	Channel Frequency Response
CIR	Channel Impulse Response
CP	Cyclic Prefix
CR	Cognitive Radio
CRLB	Cramer Rao Lower Bound
CRN	Cognitive Radio Network
DAC	Digital to Analog Converter
DARPA	Defense Advanced Research Project Agency
DC	Direct Current
DDC	Digital Down Converter
DFT	Discrete Fourier Transform
DFTFB	DFT Filter Bank

DLL	Delay Lock Loop
DSA	Dynamic Spectrum Access
DSP	Digital Signal Processor
DSSS	Direct Sequence Spread Spectrum
DTV	Digital Television
DVB-T	Digital Video Broadcast - Terrestrial
ED	Energy Detection
EGC	Equal Gain Combining
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
FBCM	Forward Backward Correlation Matrix
FCC	Federal Communications Commission
FFT	Fast Fourier Transform
FH	Frequency Hopping
FPGA	Field Programmable Gate Arrays
FM	Frequency Modulation
GFDM	Generalized Frequency Division Multiplexing
GLRT	Generalized Likelihood Ratio Test
HAP	High Altitude Platform
ICI	Inter-Carrier Interference
IF	Intermediate Frequency
ISDV	Intra-Segment Decimated Vector
ISI	Inter-Symbol Interference
ISM	Industrial Scientific and Medical Frequency Spectrum
LNA	Low Noise Amplifier
LP	Long Preamble
LTE	Long Term Evolution
LTE-EPA	LTE Extended Pedestrian A Model
LTE-ETU	LTE Extended Typical Urban Model
LTE-EVA	LTE Extended Vehicular A Model
M2M	Machine to Machine Communications
MANET	Mobile Adhoc Mesh Network
MCAS	Maximum Cyclic Autocorrelation Selection

MCP	Minimum Clock Period
MRC	Maximum Ratio Combining
MRFB	Multi-Resolution Filter Bank
MUSIC	Multiple Signal Classification Algorithm
NLOS	Non Line of Sight
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PA	Power Amplifier
PAL-B	Phase Alternating Line System-B
PAPR	Peak to Average Power Ratio
PD	Probability of Detection
PFA	Probability of False Alarm
PN	Pseudo Noise
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RMS	Root Mean Square
SC	Selection Combining
SDR	Software Defined Radio
SNR	Signal to Noise Ratio
SP	Short Preamble
TDD	Time Domain Duplexing
TDMA	Time Division Multiple Access
TDoA	Time Difference of Arrival
TLS-ESPRIT	Total Least Square - ESPRIT
ToA	Time of Arrival
UAV	Unmanned Aerial Vehicle
UHF	Ultra High Frequency band
ULA	Uniform Linear Array
UWB	Ultra-Wide Band
VHDL	VHSIC Hardware Description Language
VHF	Very High Frequency band
WiMAX	Worldwide Interoperability for Microwave Access

WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
ZC	Zadoff-Chu

Abstract

Wireless communications systems over the years have evolved to provide ubiquitous wireless connectivity for users who are increasingly mobile. More recently apart from the communication needs of the humans, data traffic originating and terminating from/to machines, machine to machine (M2M) communications, has increased. Furthermore, positioning and localization information have also been derived using the wireless communication signals. These diverse usage scenarios make the wireless modems necessary to operate in different channel environments meeting the ever increasing quality of service (QoS) requirements and with the decreasing availability of dedicated frequency spectrum. Burst communications with reconfigurable modems based on technologies such as Cognitive Radios (CR) and Cognitive Radio Networks (CRN) could satisfy these diverse systems requirements by transmitting information in short intervals, detecting the available frequency bands for transmission and accurately estimating the burst signal's time-of-arrival at the receiver so that the receiver is synchronized for effective computation of the positioning information and demodulation of the transmitted data.

For this purpose, there are many spectrum sensing, time of arrival and synchronization algorithms which have been proposed in literature. Most methods are based on covariance and/or correlation of the received signal with known features of the transmitted signal.

However, since these algorithms operate on received signal samples, even before any receiver corrections such as carrier frequency offset or channel equalization are performed, they need to be robust against different impairments and also be hardware efficient for low complexity implementation and low power consumption.

This thesis proposes low complexity schemes and architectures for spectrum sensing, time synchronization, fast acquisition and time of arrival estimation which address the issues of robustness and hardware efficient implementation for low power consumption. The proposed schemes take advantage of the pilot signals and repetitive preamble features which are part of the burst communication signals. The proposed algorithms are based on segmented processing of the received signal samples, processing only a derived signal or a single preamble out of the repetitive preambles. An analytical formulation is presented to explain how the proposed schemes are able to provide sufficient robustness and simplification of the hardware implementation. Further, performance of the proposed segmented processing schemes are evaluated through simulations over different propagation channel environments and comparisons with existing methods are presented. To evaluate the proposed segmented schemes in terms of implementation hardware complexity, FPGA synthesis is conducted and results are presented.

1 Introduction

1.1 Background & Motivation

New services are being introduced everyday based on wireless communication systems. The limited frequency spectrum bandwidth available for such services necessitates either careful frequency planning of these wireless deployments and could require reuse of frequencies to tackle the spectral scarcity problems. More recently, apart from the communication needs of the humans, wireless data traffic from machine to machine (M2M) communications and from services such as wireless positioning/localization has increased. These asynchronous data communication's requirements are characterized by their low latency requirement, burst nature of the data traffic and the need for information and communications security of the communication link. Figure 1.1 illustrates typical wireless system deployment scenarios.

In these scenarios, opportunistic and dynamic spectrum access based systems would serve well as compared to conventional fixed frequency spectrum based wireless deployments. Further for such deployments, use of Cognitive Radios (CRs) would become inevitable. The TV White Space initiative by the Federal Communications Commission (FCC) approves the use of certain bands of licensed TV spectrum opportunistically by unlicensed radios. This has provided a framework for spectrum reuse

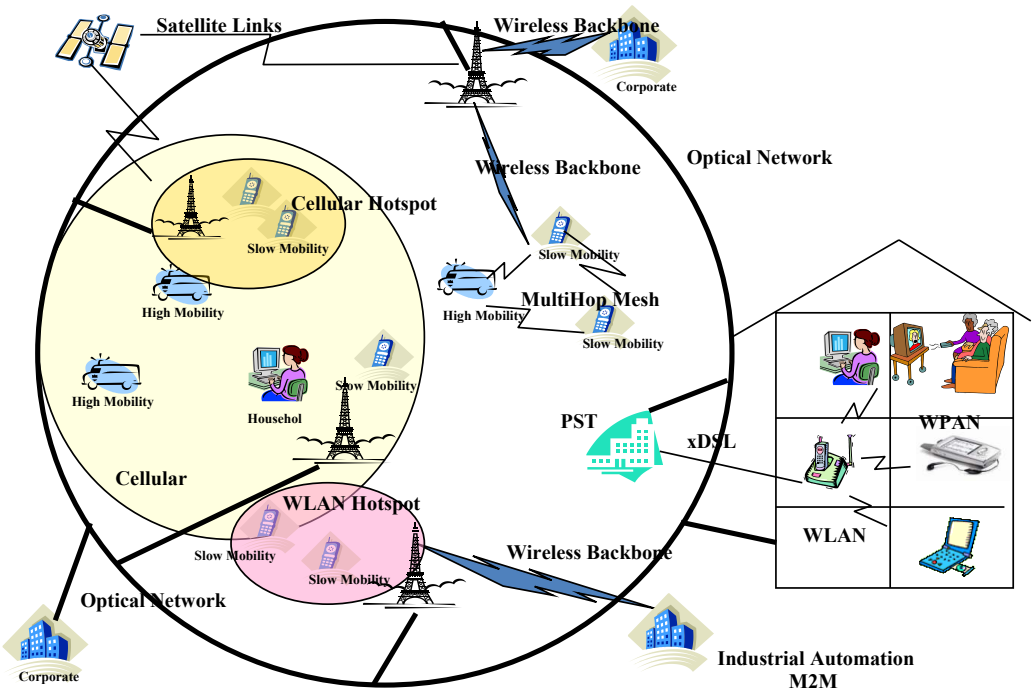


Figure 1.1: Different wireless system deployments.

and as such enhanced the adoption rate of cognitive radios (CRs) in commercial applications.

1.1.1 Cognitive Radios

Cognitive Radio (CR) is basically a reconfigurable transceiver which is referred to as an intelligent software defined radio (SDR) which autonomously and continuously monitors the different aspects of the transceiver operation, such as the operating environment, channel condition, communication parameters, communication link performance, etc., and recalibrates its operating communications parameters so that the desired link quality and throughput is maintained in varying operating conditions. In the process, it also dynamically detects the availability of a frequency band for setting up and maintaining a communication link. This could also be referred to as Dynamic Spectrum Access (DSA). Though the above description could suite a full cognitive radio, a restrictive approach which only requires the frequency band of operation to be changed

based on the spectrum availability has also been explored. These are referred to as spectrum sensing cognitive radios [1].

The main functions in a CR, apart from the conventional reconfigurable communications transceiver functions, are spectrum sensing, spectrum management and cognition cycle. The cognition cycle monitors the different operating parameters and accordingly determines and configures the communications transceiver with suitable communication parameters in order to maintain the required quality of service in the communication link.

The concept of software defined radios and cognitive radios were first proposed by J.Mitola [1, 2]. The concept has been extended to cognitive radio networks (CRN) which mandate spectrum sensing and DSA for maintaining seamless wireless connectivity. Different studies have been carried out and shown that though the frequency spectrum availability is getting scarce, the usage of available frequency spectrum is not efficient, i.e. spectrum is underutilized. This has prompted the opening up of specific licensed frequency bands for unlicensed users. In such cases, spectrum sensing cognitive radios which allow dynamic spectrum switching and management can be used to exploit the licensed spectrum by unlicensed (secondary) users. It also allows for opportunistic spectrum usage among the secondary users of the licensed spectrum.

CRs require frequency agile radio front-end and reconfigurable signal processing blocks to make opportunistic and dynamic spectrum access possible. Spectrum sensing is one of the primary functionalities in CR. It allows determination of occupancy of a particular frequency spectrum and hence it’s availability for opportunistic use, thus allowing for realizing the opportunistic use of available spectrum and dynamic spectrum access.

1.1.2 Dynamic Spectrum Access

Dynamic spectrum access (DSA) is envisaged to alleviate the spectrum shortage issues, allowing for introduction of new services in the available spectrum by enhancing spectrum usage efficiency.

Cognitive radios which allow for different communication parameters to be changed dynamically are primarily suited for implementing DSA. However, apart from the economic and regulatory challenges, certain technical challenges such as interference management, link adaptation, bandwidth management, performance criteria for spectrum opportunity detection, multi-user detection, burst signal detection, etc., still remain and are being actively investigated.

One of the main challenges, spectrum opportunity identification, detection and tracking is that it requires suitable spectrum sensing mechanisms. This is achieved either through intelligent and cooperative signal processing networking techniques (cooperative sensing), which rely on efficient signal processing for enhanced spectrum sensing. Several spectrum sensing algorithms have been proposed in published literature. However, there is always a need for a low complexity algorithm for spectrum sensing which can be implemented effectively.

1.1.3 Burst Communications

Wireless communications nowadays not only link stationary or slowly moving mobile devices, but are also widely used to connect moving cars, train carriages and unmanned aerial vehicles (UAVs). The wireless communications between fast moving objects and high altitude platforms (HAP) encounter a combination of high Doppler and delay spread in propagation channel. In the design of wireless links for these applications frame-based burst communications are generally used. Here we refer burst communications as

transmissions over a short period of time, which are of relatively high-bandwidth and are of intermittent or predetermined in time of transmission. Especially in CRNs, where the channel frequencies are opportunistically reused and the transmissions will have to switch operating frequencies depending on the frequency spectrum availability, which is determined through spectrum sensing. In such systems, the data transmission takes place in short frames and the channel effects encountered by the data burst are corrected within the frame and in the worst-case the corrupted frame is discarded altogether.

DSA enhances the reuse of frequency spectrum similarly multiple users can also use the same frequency spectrum at different times, Time Division Multiple Access (TDMA). This further optimizes the use of frequency spectrum. Moreover, the asynchronous nature of data traffic in evolving services has led to use of burst signals, which allow for TDMA multiplexing of users and time domain duplexing (TDD) of wireless communications over a given frequency spectrum. This fits in well with the opportunistic use of the available frequency spectrum. In some cases, the high data rate requirement of certain scenarios is achieved through the use of higher order modulation and the use of Orthogonal Frequency Division Multiplexing (OFDM) modulation techniques. However, for proper demodulation and decoding of data from burst signals, the receiver needs to effectively time synchronize with the incoming burst before it can perform suitable signal processing, such as frequency offset estimation and correction, channel estimation and equalization and data error correction. Given that the receiver needs to achieve time synchronization before any kind of signal processing can be applied on the received signal, the time synchronization algorithm will need to be robust against impairments such as carrier frequency offset, propagation channel and environment. Moreover, since synchronization is carried out on the signal's sample interval, rather than the symbol interval, the synchronization algorithm will need to be of low complexity and suited for

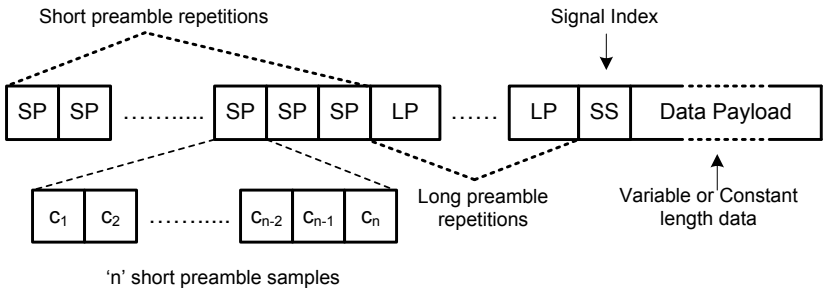


Figure 1.2: Burst signal structure.

hardware efficient implementation leading to lower power consumption. Time synchronization is achieved through detection of the preamble signals which are transmitted at the start of each burst. Typically, each transmit burst contains multiple short preambles and a few long preambles followed by the signal index and data payload, as shown in Figure 1.2. Short preambles (SP) are used for achieving coarse time synchronization. Once coarse time synchronization is achieved, long preambles (LP) are used for determining the fine time synchronization and estimation of the impairments which could be corrected through suitable signal processing. Signal Index (SS) contains the meta-data related to the constant or variable length data payload which follows it. Most of the synchronization algorithms achieve time synchronization through correlation of the received signals, either with a delayed version of itself or with a stored replica of the transmitted signal. Robust synchronization algorithms are essential to tackle the varied scenarios, which is one of the objectives of the work presented in this thesis. A number of algorithms have already been proposed for this purpose and in this work a solution based on segmented synchronization has been proposed.

More recently, wireless data communication signals have been used for determining the position of wireless devices relative to other devices. With the over-the-air travelled distance of a radio frequency (RF) signal directly proportional to time of travel, most of the techniques derive position information based on time-of-flight of the signal from

multiple anchor node transmitters to the wireless receiver. Depending on the user environment, multiple replicas of a transmitted signal arrive at the receiver due to reflected signals. In such instances, along with the timing synchronization accuracy between multiple anchor node transmitters, accuracy of the estimated time-of-arrival of the transmitted signal at the receiver determines the accuracy of the estimated position information.

One of the most important aspects of wireless communication is the mechanism how the link is secured. Wireless security can be classified into Information security, security of the actual data transmitted over the wireless link, and Communications security, security of the wireless signals itself.

Information security is usually achieved through data encryption techniques such as Advanced Encryption Standard (AES), and communications security is achieved either through frequency hopping (FH) or direct sequence spread spectrum (DSSS) techniques. Apart from communications security, FH also provides frequency diversity and hence is a preferred technique. In FH, the transmit frequency is hopped based on a pseudo noise (PN) sequence generator which determines the next hop frequency of the burst. When the PN sequence length and the number of frequencies in the FH set are large, the time to acquire the FH signal is also large. However, once a single burst is correctly received and its data demodulated, the FH sequence is known from the demodulated data. However, during the acquisition process of the FH signal multiple transmitted bursts could be lost and delay the acquisition of the FH signal. In this direction, a scheme for fast acquisition and synchronization of FH burst signals is proposed in this thesis.

1.2 Objectives

The main objective of the work presented in this thesis is to develop hardware efficient algorithms and architectures for burst communication's use in cognitive radios. Towards this objective, the main focus is on developing solutions for spectrum sensing and time synchronization functional processing modules. The proposed solutions should provide robust performance, especially in low SNR scenarios and in different propagation channel environments, which would improve the overall performance of CRs in diverse wireless applications. The proposed architectures should be able to translate into hardware implementations which are either low in complexity or efficient in terms of hardware resource utilization. The specific objectives of the work presented in this thesis are:

1) To propose a spectrum sensing algorithm and architecture which is suitable for both continuous and burst signal sensing in CR, especially for achieving good sensing performance with reduced hardware implementation complexity. Most of the current spectrum sensing solutions are either suited for continuous signals or have high computational complexity.
2) To propose a time synchronization scheme for burst signals which is suitable to provide robust timing synchronization in different channel environments, especially at low SNR scenarios. It should also be able to translate into an efficient hardware implementation with reduced hardware resource utilization. The proposed timing synchronization's timing estimate should also be able to aid in estimating the position / localization of the receiver.
3) To evaluate the hardware implementation complexity of proposed schemes with the aid of FPGA synthesis.

1.3 Contributions

The key focus is to develop low complexity, hardware efficient schemes for spectrum sensing, time synchronization, fast acquisition and fractional time delay estimation in wireless communication systems.

1) Low Complexity Spectrum Sensing Using Segmentation and Decimation.

A low complexity spectrum sensing technique based on correlation of intra-segment decimated vectors has been proposed. The primary signal to be detected is sampled at a particular frequency, and the digitized samples are segmented and decimated by a suitable decimation index. The resulting vector's arithmetic means are correlated to detect the presence of the signal. MATLAB simulations of the proposed technique have been carried out to validate the proposed technique. Single-tone, Multi-tone signal at a 25MHz IF frequency, Advanced Television Systems Committee for digital television transmission over terrestrial, cable, and satellite networks (ATSC) Digital Television (DTV) signal, Phase Alternating Line System B (PAL-B) Analog TV signal and a QPSK modulated burst signal have been used to validate the proposed detection technique. The proposed scheme is able to detect the presence of a signal effectively and its complexity is only nearly 1/10th of the complexity (for vector length n = 10) of the simplest of the existing correlation based technique and has much higher hardware gains over FFT based detection schemes. The proposed spectrum sensing technique also allows for varying the performance of the spectrum sensing by trading off the implementation complexity. Using the proposed reconfigurable implementation architecture, it is possible to dynamically trade-off detection performance of the proposed spectrum sensing technique with the hardware computations required for detection.

2) Efficient Cross-correlation Algorithm and Architecture for Robust Synchronization.

A time synchronization algorithm based on preamble sequences, which are generally used for time and frequency synchronization has been proposed. Recently Zadoff-Chu (ZC) sequences have gained popularity for this purpose. The proposed synchronization method is an efficient cross-correlation based algorithm. Further an implementation architecture for robust synchronization in systems with multiple preambles has also been shown. The proposed method is well suited for systems experiencing carrier frequency offsets and operating in high-mobility environments. Synchronization is achieved by cross-correlation of local transmit preamble replica with the segmented and decimated samples of the received preamble. The proposed architecture has been synthesized and implemented on Xilinx FPGA platform for validation and performance evaluation. The system level MATLAB simulation under Additive white Gaussian noise (AWGN), in Long-Term Evolution (LTE) channels with frequency offsets and Doppler shifts have also been carried out to find the efficacy of the proposed approach. The inherent parallelism of the proposed scheme results in a fast and energy efficient implementation. One such implementation of the correlator structure in Xilinx FPGA is presented and is shown to have 28% reduction in power consumption, capable of clocking 1.5x higher clock speed with 73% reduction in the usage of FPGA hardware resources such as DSP blocks and logic resulting in an overall reduction in energy consumption by 57%, when compared to full parallel cross-correlator based synchronization.

3) ***Fractional Time Delay ToA Estimation Using Short Preambles For OFDM-WLAN.***

The proposed time synchronization algorithm has been extended for fractional time delay Time-of-Arrival (ToA) estimation in time domain. The proposed ToA estimation approach has been applied to OFDM based WLAN signals. The proposed approach makes use of OFDM-WLAN's short preamble repetitions to estimate fractional time delay ToA. Estimation is done by oversampling, segmenting, re-arranging and cross-correlating received preambles with a stored replica of short preamble. It has been extended for receivers with multiple receive antennas. Simulation results show the improvement in ToA estimation error over both low and high SNR scenarios as compared to single antenna receivers.

4) ***Fast Acquisition and Time Synchronization of Frequency Hopping Burst Signals.***

Further, the proposed synchronization algorithm has been applied for successful detection and demodulation of frequency hopping (FH) signals, which is dependent on proper tuning of the front-end to the transmit frequency and time synchronization of the incoming burst. The sequence of FH hop frequencies is generally determined by a Pseudo-Noise (PN) sequence and time synchronization is achieved using synchronization preambles in the transmit burst. Successful acquisition of the hop frequency sequence could be achieved when at least a single burst's data is successfully decoded at the receiver. In this thesis, a low complexity, two-level acquisition based scheme for fast acquisition and time synchronization of the frequency hopping burst based on the Zadoff-Chu synchronization preambles has been presented. Simulation results detailing the proposed scheme's performance and a low complexity hardware implementation architecture are presented. Simulations show that

the single IF channel synchronization detection performance is above 99% for SNRs more than -20dB and wideband multiple digital IF detection performance is similar to single digital IF channel performance for SNRs more than -5dB. The simulations have been carried out for characterizing the performance in different propagation channel environments such as AWGN, LTE-EVA, LTE-ETU and 6-path Rician propagation channels. The performance degradation due to the low complexity hardware is seen to be less than 1% for SNRs above -20dB and the synchronization performance difference reduces to less than 0.1% for SNRs above -5dB.

1.4 Thesis Outline

The rest of this thesis is organized as follows. In Chapter 2, a review of related published works has been carried out. These include:

- Concepts of CR, significance of spectrum sensing and various techniques that have been proposed for spectrum sensing.
- Time synchronization algorithms that have been proposed.
- Techniques and approaches that are being used for ToA estimation for determining the position information.
- Fast acquisition and synchronization algorithms that have been proposed for FH signals.

Chapter 3 presents the proposed method for spectrum sensing, performance simulation results and demonstration of the low complexity nature of the proposed approach.

In Chapter 4, an efficient cross-correlation algorithm for robust synchronization in frame-based burst communication systems has been presented. The proposed method's performance is simulated and compared with few existing algorithms. Implementation

architecture along with its hardware implementation and resource efficiency of the proposed algorithm are shown.

Chapter 5 presents two adaptations of the proposed time synchronization algorithm for fractional time-delay ToA estimation and synchronization, and fast acquisition and time synchronization for FH burst signals. Performance simulation results of both extensions along with the low complexity hardware implementation for fast acquisition of FH burst signals is presented.

Finally, Chapter 6 concludes the thesis with a summary of works presented and suggestions for future work.

Page Intentionally
Left Blank

2 Literature Review

In this chapter, a review of published research related to Spectrum sensing and Time Synchronization of burst signals has been carried out.

2.1 Spectrum Sensing

CRs use different transmission parameters for a communication link to satisfy the link's set requirements. Their ability to dynamically change the operating frequency is primarily suited for implementing DSA.

DSA can be broadly categorized into the following three user models [3]

1. Dynamic Exclusive Use Model
2. Open Sharing Model
3. Hierarchical Access Model

Dynamic exclusive use model assigns exclusive spectrum use to specific services by way of spectrum property rights and dynamic spectrum allocation. The main intention of this model is to allow for greater spectrum usage efficiency. In this model, the spectrum is either dynamically allocated depending on the usage pattern or by users with proprietary rights spectrum, who choose to allow use of it by other users.

Open sharing model is similar to the unlicensed industrial, scientific and medical (ISM) frequency spectrum bands. In this model, different spectrum sharing strategies have been investigated to accommodate the co-existence of different wireless services in ISM frequency bands.

Hierarchical access model adopts a primary user - secondary user structure. This model opens up the licensed spectrum of the primary users for use by secondary users and limiting the interference of the secondary users as seen by the primary users. For this purpose, generally two different approaches have been considered, Spectrum Underlay and Spectrum Overlay.

In the spectrum underlay, the secondary users' transmissions are constrained so that they operate only below the noise floor of the primary users. This approach does not need to detect the operation of primary users as the constraints are derived based on the assumption that the primary users are in operation all the time. Spectrum overlay approach was first proposed by J. Mitola [4] as spectrum pooling and was further investigated by United States Defense Advanced Research Project Agency (DARPA) as opportunistic spectrum access in the XG program [5]. Unlike the spectrum underlay approach, spectrum overlay approach restricts the use of frequency spectrum by secondary users only when the primary users are not in operation. This notion of primary users not being in operation is exploited in both spatial and temporal spectrum white spaces for use by secondary users.

The different DSA models present varied requirements on the different spectrum sensing parameters such as sensing time, probability of detection (P_d), probability of false alarm (P_{fa}), etc…

CRs determine the availability of a radio channel frequency for establishing a link through spectrum sensing as well as through global database [6] for TV White Space usage or through other networking techniques also known as cooperative sensing [7]. Cooperative sensing still relies on the individual nodes sensing the available frequency spectrum at their respective location. These local observations are shared with other nodes in the network. Sharing could be through a centralized node or through individual reporting over the network. Cooperative sensing technique is likely to reduce the sensing performance degradation due to fading, hidden node issues, etc., The sensing performance of a node/user is improved in cooperative sensing due to the fact that all nodes/users in a network experience different operating environments and have different level of sensing confidence, as they are likely to have different SNRs with respect to the primary user signal. The overall performance of such collaborative sensing is dependent on the individual node's capability to sense the spectrum. The popular spectrum sensing algorithms used at the individual nodes are

1. Energy detection
2. Matched Filter detection
3. Wavelet-based detection
4. Feature detection (Cyclo-stationary based & Covariance-based detection)

There has been extensive research to develop hardware efficient spectrum sensing algorithms which will reduce the implementation complexity and improve the sensing sensitivity. The sensing sensitivity metric is used to quantify the performance of a sensing algorithm in terms of its Probability of Detection (P_d), Probability of False Alarm (P_{fa}) and Signal to Noise Ratio (SNR). The spectrum sensing algorithm's hardware efficiency is primarily of significance to devices intended for consumer use. The implementation

complexity of spectrum sensing is a major factor in order to keep the computing resource requirements and hence the cost of devices to the minimum. Different spectrum sensing algorithms have been proposed in the literature and a summary is presented below.

2.1.1 Energy Detection

Energy detector based spectrum sensing is the most simplest to implement with least complexity. It has been extensively used to determine the presence of primary users. Basically, in energy detector, samples of the received band-limited signal within a specific integration interval are used to reconstruct the signal. The integrated power over the integration interval is compared to a set threshold and the presence or absence of the signal is declared depending on a chosen threshold. One of the main advantages of energy detection approach is that it does not require any prior knowledge of signal and noise in the intended frequency spectrum band. The implementation complexity of these algorithms is modest, but their performance suffers heavily in presence of higher or uncertain noise power. Different techniques have been proposed to reduce the impact of noise on the performance of energy based detection [8-10].

The integrated power is directly proportional to the signal power and noise power present in the integration interval. Setting a fixed threshold in the energy detector for different SNR scenarios affects the performance of the energy detector. In [8], an adaptive threshold technique has been suggested in which the energy detector's threshold for detection of the primary user is determined based on the SNR estimate to determine the presence or absence of the signal. Similarly in [9], noise power uncertainty is introduced into threshold setting by combining with inputs from other users in a network to perform collaborative sensing. This suggests an improvement in performance of conventional energy detection in the presence of uncertain noise power due to collaborative gain, which

is realized through combining multiple inputs. The overall performance of the energy detector is also improved by optimally combining the received signal samples from different antennas. The method proposed in [10] blindly combines the received signal samples in space and time based on the principle of maximizing the signal-to-noise ratio (SNR) and after combining, energy detection (ED) is used.

Further, channelized energy detectors have also been proposed to reduce the impact of out-of-band noise on the decision and to improve energy detection in desired frequency bands. Channelized energy detectors have different types of channelizers to limit the bandwidth and the integrating interval. Discrete Fourier Transform Filter Bank (DFTFB) are generally used in channelized energy detectors [11, 12] which offer fixed sensing resolution. In such energy detectors, usually the outputs of several band energies are combined to produce the final output. In [13], multi-resolution filter banks (MRFB) have been used to provide multiple sensing resolutions by efficiently selecting the varying resolution sub-bands without much of a hardware re-implementation. Further, an improved tree-structured DFT filter bank based spectrum sensor is presented in [14]. The outputs of several branches of the DFT are combined to determine the edge frequencies of the channel.

2.1.2 Matched Filter detection

Matched filter detection [15, 16] is also known as coherent sensing or detection. Matched filters maximize the received signal SNR. Therefore they are more suited for reducing the impact of noise uncertainty in stationary noise environments for signal detection. With prior knowledge about features of the signal to be detected, such as the type of modulation, symbol rate etc., the matched-filter based algorithms offer better performance at the cost of higher computational complexity. Generally matched filter

based detection is suitable when the primary user's signal has certain templates such as pilots, preambles, etc., which can be used to match with the received signal. Since a known template is used for detection, the sensing time required for a given performance metric is shorter.

The received signal needs to be demodulated in most cases in order to extract the template information. With a variety of waveforms being used by primary users and in CRN, the waveforms itself being chosen dynamically to suit the environments, this particular requirement increases the complexity of the spectrum sensing receiver and makes its practicality very limited in cognitive radios. Moreover, even in certain limited applications where the primary user is clearly defined, as in [16], when the derived template is corrupted, the performance of matched filter detection degrades significantly. This also places constraints on the secondary user's waveforms which should avoid using similar templates as being used by primary users.

2.1.3 Wavelet-based Detection

Wavelet transforms have time-frequency localization properties and have been used as a tool to analyze local spectral structure to identify singularities and edges of the spectrum with low computational complexity. These are mostly useful in wideband spectrum sensing where it is necessary to determine the edges of the frequency band in use. One of the issues with wavelet based detection is that it is difficult to isolate false detection due to isolated impulses, spike, very-narrowband interference and white noise [17]. In [18], a first-order derivative scheme is used to detect frequency band edges using wavelets. The first-order derivative scheme detects frequency boundaries accurately at high SNR but has degraded performance at low SNR. To circumvent this issue, Hilbert transforms have been proposed in [19] to determine the edges of channel frequency band. Generally,

Hilbert transforms are used to analyze the harmonic functions. The spectrum of interest is divided into multiple sub-bands and the average power spectral density of each sub-band is used to determine the occupied sub-bands.

2.1.4 Feature Detection

Most feature detection based algorithms make use of cyclostationarity characteristic [20] of the signals to determine the presence of the signal. The cyclostationary features in the signal exist mainly due to the periodicity in the transmitted signal. Cyclostationary features are, to some extent, in the signal statistics such as mean, autocorrelation, covariance, etc. The performance of feature detection based spectrum sensing is generally better than energy detection based sensing because usually noise modelled as Additive White Gaussian Noise (AWGN), and it has no cyclostationary features. However, the modulated primary user signals are with cyclostationary features which can be used to effectively differentiate against noise. Cyclostationarity based methods are advantageous over energy detection based spectrum sensing only when noise power is not known [21].

Cyclic autocorrelation functions (CAF) and statistical covariance (CAV) are generally used to determine the cyclostationarity present in the signal. In [22] and [23], multi-cycle cyclostationary detectors based on the classical cyclic autocorrelation estimator and its asymptotic properties have been proposed. An improved sub-optimal multi-cycle sum detector is proposed in [23]. In [23], it is shown that by proper scaling of test statistics of different candidate cycle frequencies, sensing performance is increased. These algorithms make an assumption that the knowledge of at least one cyclic frequency of the primary user's signal is known. However, there are other approaches which make use of the distinctive characteristics of waveforms such as OFDM which are recently being used. For OFDM waveforms, usually it is the Cyclic Prefix (CP) - induced cyclostationarity

which is exploited for spectrum sensing. In [24], Cyclic autocorrelation function of OFDM and Generalized Frequency Division Multiplexing (GFDM) signals is shown and Generalized Likelihood Ratio Test (GLRT) is applied for one non-zero cyclic frequency, which is characterizing the OFDM/GFDM CAF structure and in differentiating it from noise. A binary tree classifier for WiMAX (Worldwide Interoperability for Microwave Access) and LTE-OFDM signal's detection and classification is presented in [25-27]. It makes use of CP-induced cyclostationarity in OFDM signals, preamble-induced cyclostationarity of the mobile WiMAX OFDM-based signals and the distinctive Reference Signal-induced (RS-induced) cyclostationarity of the LTE OFDM-based signals. It has generally been noted that the CP- induced cyclostationarity based features are stronger than preamble, and RS-induced cyclostationarity. Apart from the CP-induced cyclostationarity, preamble, and RS-induced cyclostationarity in OFDM system, signal is also identified using correlated pilot subcarriers, if present. The OFDM system could be uniquely determined by the frequency location of the pilots, thus enabling precise signal identification. The detection algorithm proposed in [28] makes full use of the cyclostationary specificities of the target signal, IEEE WLAN 802.11a [29] signal. The cyclic signatures induced by the correlated pilots are applied to determine the primary signal and this method is shown to be robust to frequency-selective channel fading.

The statistical covariance and Maximum-Minimum Eigen-value based covariance absolute value (CAV) algorithms are based on auto-correlation estimation, which compute the covariance matrix of received signal to derive the detection parameters. CAV algorithms work well but tend to be of higher computational cost. The statistical covariances of the received signal and noise are usually different and they are used to

differentiate noise from the signal and determine the presence of the signal. Two different test statistics are extracted from the covariance matrix in [30] and the presence of the signal is determined by comparing extracted statistics. The test statistic derived out of the covariance matrix, as proposed in [31], are the Eigen values of the covariance matrix of the received signal. The detection hypothesis is based on the ratio of the maximum eigenvalue to minimum eigenvalue and ratio of the average eigenvalue to the minimum eigenvalue.

The autocorrelation based algorithm, Maximum Cyclic Autocorrelation Selection (MCAS) proposed in [32] improves the performance of the CAV algorithm presented in [30, 31] by incorporating the assumption that the primary user signal is a low-pass signal with symmetric spectrum and complex valued with independent real and imaginary signal components. The decision statistic is different from that proposed in [30, 31] and allows for scanning the frequency band to determine the center frequency of the captured primary signal. In order to improve the performance of CAV based detection and energy based detection, a hybrid detection mechanism is proposed in [33]. The hybrid detection uses energy detection in low correlation scenarios and CAV based detection in high correlation scenarios. The final result to use is determined by a test statistic which is based on the received signal SNR. It is shown that the hybrid detection methods perform better in time-varying fading conditions.

The autocorrelation based algorithms presented in [34] perform correlation coefficients (CCE) based detection and nonparametric autocorrelation based detection. Cyclostationarity-based spectrum sensing with subspace projection as presented in [35], it makes use of cyclic autocorrelations for different delays at the same frequency for the detected primary user. Correlation and summing algorithm proposed in [36] makes use of

difference in autocorrelation envelopes of band-pass noise and band-pass signal to detect the presence of the signal. The maximum cyclic autocorrelation selection (MCAS) algorithm presented in [32] compares the peak and non-peak values of the cyclic autocorrelation function to determine whether the primary signal is present or not. The computational complexity of MCAS is shown to be relatively low. In [37], a constant false alarm rate detection algorithm based on autocorrelation of received samples is presented. The autocorrelation algorithm performance is improved by incorporating the assumption that the primary signal is low-pass and complex-valued with independent real and imaginary components. Spatially uncorrelated samples from multiple antennas are correlated to detect the presence of the signal in [38].

Most of the algorithms discussed above have been proven through simulation studies and shown to have satisfactory performance. Their implementation complexity is directly dependent on the multiple autocorrelations and computation of covariance matrices of the received signal. But, these algorithms seldom take advantage of the fact that most of the analog radio frequency receivers have an output intermediate frequency which is band limited and usually downconverted and digitized to generate baseband samples for further processing. The possibility of changing the sampling frequency dynamically to digitize the down-converted received signal, and exploiting this to reduce the computation requirements of the spectrum sensing algorithms has been hardly explored in the existing literature. In this thesis a simplified correlation based sensing method is proposed in Chapter 3.

2.2 Time Synchronization of Burst Signals

Frame-based communications of any kind require frequency and timing synchronization between two nodes to correctly receive, demodulate, and decode the transmitted data. Here we refer burst signals as transmissions over a short period of time, which are of relatively high-bandwidth and are of intermittent or predetermined in time of transmission. For transmissions in high mobility environments, which experience high Doppler and delay spread, OFDM systems are popularly used mainly due to their large symbol times and the high data-rate they offer. Efficient demodulation and decoding for OFDM systems depends on time and frequency synchronization which maintains the orthogonality between the OFDM sub-carriers and reduces Inter-Carrier Interference (ICI). Once timing synchronization is achieved, powerful channel estimation/correction and channel coding schemes are being used to increase the Bit-Error-Rate (BER) performance of the data link. Hence achieving timing synchronization with least error is important even at lower Signal-to-Noise ratios (SNR), this is especially true in CRNs and in Mobile Adhoc mesh NETwork (MANET) systems, where individual node's communication link could employ different channel coding schemes for increasing the overall network performance.

Frequency Hopping (FH) is used in many of these systems as it provides frequency diversity, robustness against jamming and interfering signals [39, 40]. Acquisition of FH sequence is the initial detection of current hop and determination of subsequent hop frequencies. Till the FH sequence is acquired, all burst data is lost and leads to possible late entry into a network of nodes, delaying the overall network formation in a MANET. Acquisition time can be large depending on the length of the PN-sequence used to generate the FH sequence and the number of discrete hop frequencies in the FH frequency

set. A receiver which can be tuned to the hop frequencies by effectively time synchronizing with the incoming burst can demodulate and decode the incoming bursts.

Apart from demodulation and decoding of the data, the estimated time offset and time-of-arrival (ToA) of the burst signal can also be used for ranging. Based on ToA of the burst at multiple anchor nodes, whose position information is known apriori, the transmitter's location could be determined. Knowledge of location information can be effectively used to improve the wireless communication link efficiency and also efficient frequency reuse in CRs through directed transmissions [41, 43].

Preambles and pilot symbols are commonly used for timing and frequency synchronization. In one of the widely used OFDM systems, IEEE 802.11a WLAN standard [29], repetitive preamble sequences have been defined. More recently in DirectNet waveform [44], which is being developed for highly mobile, directional, high-data-rate, adhoc, mesh communications networks, multiple and repetitive preambles are defined for AGC, Doppler compensation and equalizer training. In high mobility environments, multipath propagation attenuates and distorts the preamble sequence further making the time synchronization task difficult. The preamble detection is also affected by carrier frequency offset, which can be due to reference clock offset or in high mobility environments where fixed Doppler frequency presents itself as carrier frequency offset, which biases the timing offset estimate. In such scenarios, multiple preamble symbols are used. Generally, in a receiver, the preambles are used to first estimate the appropriate receiver front-end gain (Automatic Gain Control, AGC), and then estimate the time and frequency offset and correct it. Unlike in wired OFDM systems, in wireless systems, short bursts prohibit long preamble sequences. Preamble sequences need to be

sufficiently long to precisely estimate both time and frequency offsets and in some cases are also used for the initial estimate of the propagation channel.

Estimation of time offset for synchronization through correlation over multiple periodic or repetitive preamble / training symbol lengths consumes extra computation resources in the receiver. Hence it needs to be performed with reduced complexity for efficient hardware implementation. Generally correlation (either autocorrelation or cross correlation) is used for achieving synchronization. Different algorithms for timing synchronization have been proposed [48, 73]. In OFDM systems, post-FFT, channel estimation and equalization are performed in the receiver [47]. Though the equalizer is capable of estimating the fine time offset, a pre-FFT coarse synchronization is still needed. By adopting the schemes used for pre-FFT synchronization, a novel post-FFT method has been proposed in [45]. However, this introduces large delay in fine time offset estimation, whose effect can be easily compensated in an equalizer [99]. However, in burst communications, the pilot data is desired to be kept to the minimum so as to maximize the information data and this could compromise the performance of the equalizer. Hence the post-FFT method in [45] will not be suitable for burst mode transmissions where the synchronization has to be achieved much faster to avoid straining the equalizer. But it is valid for continuous transmissions such as DVB-T broadcast systems [46] which have sufficient pilot data for equalizer to perform optimally. Hence, correlation based synchronization schemes are preferred for burst communication systems as they perform time synchronization before the FFT processing in OFDM systems.

2.2.1 Burst Synchronization through Correlation

Synchronization is proposed to be achieved through specific frequency tones modulated along with data in OFDM/FM mobile radio communications in [48]. These

tones were detected based on frequency domain correlation in the receiver. A similar frequency domain correlation approach is detailed in [49] with specific pilot tones used for synchronization. Subsequently as communications evolved into burst signal communications, more sophisticated pilot, training and preamble structures have been proposed, thus leading to development of equally sophisticated time domain and frequency domain correlation techniques for synchronization.

Correlation based synchronization can be classified broadly into two schemes:

1. Autocorrelation based synchronization
2. Cross-correlation based synchronization

In autocorrelation based synchronization, the received signal is correlated with a time shifted version of itself. With repetitive preambles or training symbols, or in some cases cyclic-prefixes, this leads to an output which results in a plateau-like timing metric [50, 51]. In cross-correlation based synchronization, a stored replica of the transmitted preamble or training symbol is correlated with the received signal. The cross-correlation of the periodic or repetitive symbols results in multiple peaks [52, 53].

A number of synchronization schemes have been proposed for frame synchronization in OFDM Systems for different operating scenarios. In [50], Schmidl and Cox have proposed to determine the start of frame, and carrier frequency offset (CFO) by correlating the training sequence of two symbols lengths. Each of the training symbols are made up of pseudo-noise sequences, with the first training symbol divided into two identical halves and transmitted over all the subcarriers of the OFDM signal. Synchronization is achieved by looking for two identical halves of the first training symbol and subsequently by correlating with the second training symbol, which is transmitted only on odd-subcarriers, the frequency offset is corrected. The timing metric

is a plateau, but still allows for robust frame synchronization for OFDM symbol timing estimate. However, in low SNR scenarios, synchronization becomes less reliable and the mean square error is large [54]. Moreover, possible DC-offset in receiver front end which will increase the autocorrelation magnitude could also limit the algorithm's performance. This coupled with fluctuating received signal level due to carrier frequency offset and unsettled AGC in receiver front end presents relatively high uncertainty on the timing offset estimation. To address the issue of plateau like timing metric arising out of the method presented in [50], a modified preamble has been proposed in [55]. They have proposed a preamble with positive samples in the first half and negative samples for the second half of the preamble. This eliminates the plateau like timing metric but the timing estimation error in inter-symbol interference (ISI) channel environments is still high. For this purpose, in [54], another modified preamble structure with symmetric alternate preambles for upto four preamble symbols is proposed. This leads to a single timing metric, but the correlation length is increased to twice the preamble length. In order to reduce the false alarm probability, segments of high correlated preamble symbols are combined linearly to determine the timing synchronization in [56]. This method takes into account the difference in power of data and preamble symbols from different users as 802.16 is an Orthogonal Frequency Division Multiple Access (OFDMA) based standard. This method is also dependent on the segments of the preamble which have high correlation, which in turn is dependent on multipath scenarios. Segmented processing of preambles is also proposed in [57], which uses different preambles sequences to distinguish different users and achieve timing synchronization through segmented moving autocorrelation of time domain preambles. Autocorrelation for Coarse time synchronization and cross-correlation for symbol time synchronization has been proposed

in [58]. This increases the complexity of the overall hardware, hence a reduced bit-width hardware implementation has also been proposed in [58]. However, issues due to multipath and carrier frequency offset, possibility of diminished cross-correlation peak and plateau like timing metric due to auto-correlation still remain. Further to reduce the complexity in synchronization, a data partitioning approach along with a dynamic threshold setting for detection of the symbol time offset is proposed in [59]. The dynamic threshold is proposed based on the channel condition and is derived as a function of auto-correlation across multiple preamble symbols. The data partitioning approach reduces the number of samples used for correlation. However, as mentioned in [59] this is more suitable for UWB systems where the redundancy is generally high.

In [60], the threshold for detection of the symbol timing offset estimator is selected based on channel conditions, but instead of dynamic threshold setting, the threshold is selected based on the Gaussian distribution's mean and variance for AWGN and subsequently apply a strategy for detection declaration in multipath simulations. Though this is possible for simulations, in the case of actual implementation, differentiating AWGN and multipath scenarios is difficult, which would result in poor or overly cautious detection performance.

Instead of estimating the symbol time offset and correcting the offset, in [61], after coarse synchronization, the problem is converted to estimating the channel's first path delay. The path delay estimation is performed after FFT and a Delay Locked Loop (DLL) is used to correct the clock by tracking the pilot sub-carrier in OFDM systems. This requires the use of dedicated pilot sub-carriers in the OFDM waveform design and cannot be applied to other burst signal modulation formats.

Maximum Likelihood (ML) estimation of the time offset is presented in [62 - 64]. In [62], a joint ML time and frequency offset estimator is presented for OFDM systems. Cyclic prefix (CP) information is used for estimating both time and frequency offset. This method would not be suitable for burst systems other than OFDM, which may not have CP like structure within the signal. The joint ML estimator presented in [63] also correlates the adjacent repetition patterns or specific patterns which are present in the burst signal. It is also suggested to separate the estimation of time and frequency offsets in order to reduce the implementation complexity. The ML estimator presented in [64] makes use of the antenna diversity gain to improve the estimator's performance. This method also uses the CP information as in [62]. Antenna diversity is used for time offset estimation in [65]. The method is valid for both single carrier burst and OFDM signals. In this method, it is assumed that the pilot symbol consists of two periods of a chirp signal. These can be considered as preamble symbols, as in rest of the proposed techniques. The preambles (chirp signals) are correlated in multiple antenna paths and are combined to determine the timing offset which is similar to that in other single antenna systems. The effects of different antenna combining techniques such as Equal Gain Combining (EGC) or Maximum Ratio Combining (MRC) or Selection Combining (SC) have also been discussed in [65]. An ML estimator for synchronization is also presented in [66]. In this, the frequency offset estimation is extended to up to half of the useful bandwidth. It is also proposed to reduce the complexity by using only the sign-bit of the quadrature signals. Though this reduces the complexity, it is not suitable for all types of preamble sequences.

Over frequency selective fading channels, Speth et. al in [67] have shown that any error in frame synchronization will affect channel estimation performance and the overall performance of the link. They suggest that since synchronization time is not important,

the metrics used for frame synchronization can be averaged over multiple frames to achieve frame synchronization. However, this will increase the time required to achieve synchronization and delay the entry of the node into a network. To improve frame synchronization, specific preamble structures such as the sandwiched preambles were exploited for timing synchronization by Muller et. al in [68]. Though this reduces the overall training sequence overhead because sandwiched preambles can also be used as training sequences for channel estimation, performance in high multipath and Doppler environments is doubtful and it has not been characterized. In [69], the frequency domain pilots in an OFDM system are interpreted in time domain for the purpose of synchronization. This method results in different approaches depending on the propagation channel environment. One method for when the channel length is comparable to or longer than the cyclic prefix length and a different method when channel length is less than the cyclic prefix length, Here, the issue is that the channel length is not known prior to actual synchronization and channel estimation. In [70], for frequency selective channels, synchronization in achieved by determining the effects of symbol offset errors on the decoded OFDM symbol constellation. For this purpose specific reference symbol structure has been proposed in [70].

The effects of timing synchronization errors on OFDM systems in high mobility environments have been well quantified in [71]. In this work, synchronization is achieved with inputs from channel estimators which are aided by pilot signals. Specific training symbols which have desirable timing metrics such as steep roll off have been proposed to aid in synchronization. When channel estimation is carried out with such specifically designed training symbols, fine timing estimate and robustness against frequency offsets is also achieved [55]. In [72], a two-step approach to synchronization has been presented.

In the first step, a coarse synchronization is achieved through correlation of the long preamble of 802.16d and then followed by fine synchronization, which is accomplished through correlation of the second OFDM symbol of the long preamble in the second step. Though, this is likely to produce a sharp impulse like timing metric, it is not suitable for high mobility environments as it still relies on the method proposed in [50].

Specialized time domain sequences such as constant envelop preambles have been proposed in [73]. These are Constant Amplitude Zero Autocorrelation (CAZAC) sequences, which have two identical halves. These are very useful as preamble sequences as they are nearly immune to any Peak to Average Power Ratio (PAPR) issues at the transmitter. However, the timing estimate performed in [73] is based on the methods presented in [50, 54, 55] has the same limitations as that in these methods.

Zadoff-Chu (ZC) sequences discussed in [74], which are Constant Amplitude Zero Auto Correlation (CAZAC) sequences, have recently been used as preambles in LTE and LTE-Advanced Standards [75]. With ZC sequences as preambles, Cyclic Prefix (CP) and Cyclic Suffix (CS) are added and cross-correlation over multiple preamble symbols is performed to effectively estimate the time and frequency offset [52, 53]. The accuracy in estimating he fine time offset is still low and the reported timing estimation error is large, ~>100 samples, for SNRs of -10dB and below.

In cross-correlator based synchronization when multiple repetitive preambles are used, there are multiple peaks at the output of cross-correlator. The timing metric to determine the correct timing is by minimizing the metric which calculates the squared average distance between the peaks [50]. The magnitudes of the peaks are dependent not only on the signal and noise power but also on the encountered multipath scenario encountered. Varying peak magnitudes create ambiguity in detection of the peak and detection is

dependent on the set threshold. A simple solution for this problem is inverting successive transmitted preamble sequences, which give negative peaks after cross-correlation as has been proposed in [55] and [75]. However as pointed out in [55], this could also lead to high false detection probability and a new normalized timing metric generalized over multiple symbols is proposed in [55]. But still, the correlation over the entire multiple preamble symbol sequence needs to be carried out and multiple peaks will need to be detected, which is very challenging in a multipath environment. This is especially true when there is uncorrected carrier frequency offset, which will also vary the magnitude of the correlation peaks. The Symmetrical cross-correlation approach proposed in [77] produces pronounced cross-correlation peaks however at the cost of extra computation and still experiences high false detection probability as reported in [55]. The derived estimate of the timing offset is further affected by the presence of any carrier frequency offset as presented in [52, 53].

These issues present an opportunity to develop computationally efficient synchronization techniques. Taking advantage of the repetitive preamble symbols, a novel time synchronization technique is presented in Chapter 4 of this thesis.

2.2.2 Acquisition of Frequency Hopping Burst Signals

Different FH acquisition mechanisms have been discussed and early studies [78, 79] have concluded that serial search schemes which step through the different possible frequency bins are simple to implement but have large acquisition times, especially when the number of discrete frequencies in the FH frequency set is large. Matched filter schemes have the advantage of faster acquisition time but the implementation hardware needed could be complex depending on the number of matched filters used. Combining the advantages of these two schemes is the two-level acquisition scheme [80], which has

the advantage of the fast acquisition time and the simpler implementation. However, when the number of active correlators is increased for better acquisition reliability, the hardware complexity also increases.

Acquisition of FH signals in time, frequency and hop sequence can also be accomplished through frame synchronization [81], for which specific preamble structures and waveform features have been introduced in the transmit bursts which are then correlated at the receiver to achieve time synchronization [50]. In this thesis, in Chapter 5, an acquisition FH acquisition scheme is presented, which makes use of the proposed time synchronization method.

2.2.3 Fractional Time Delay ToA Estimation

Time synchronization of received preamble sequences in burst signals is required for demodulation and recovery of transmitted data. For this purpose, it is sufficient if the fine time estimate is within the guard interval (GI) time provisioned in the burst or within a few symbol times. This for OFDM systems is well analyzed in [82] and shown in Figure 2.1. Due to the cyclic nature of the channel, this symbol time offset results in phase rotation for the OFDM sub-carriers with minimal inter-symbol interference (ISI) and inter-carrier interference (ICI). With further signal processing, such as with equalizers, the fine time offset is estimated, corrected and transmitted data recovered with minimal errors.

However, in order to utilize the synchronization time offset estimate for ranging and localization, the received burst time offset estimate has to be better than that required for demodulation and data recovery. For this purpose, various ToA estimation algorithms have been proposed in literature. A number of high resolution time delay estimation algorithms have been discussed in published literature. In [83], multipath time delay for

broadband random signals is estimated through a maximum-likelihood estimator and a linear predictor estimator. These are used for two-path and three-path scenarios respectively.

In indoor environments, the number of multipath is likely to be much longer. For time delay estimation of pulse-like signals, neural network based time delay estimator has been used in [84]. The time delay is estimated by relating received signal energy to a cost function of the time delay estimation. For large time-bandwidth product signals, time delay estimation has been estimated in time domain and frequency domain in [85]. In time domain, different multipath signals are modeled as signals with different attenuation and delays, these are correlated to derive the covariance matrix and MUSIC algorithm is applied for time delay estimation.

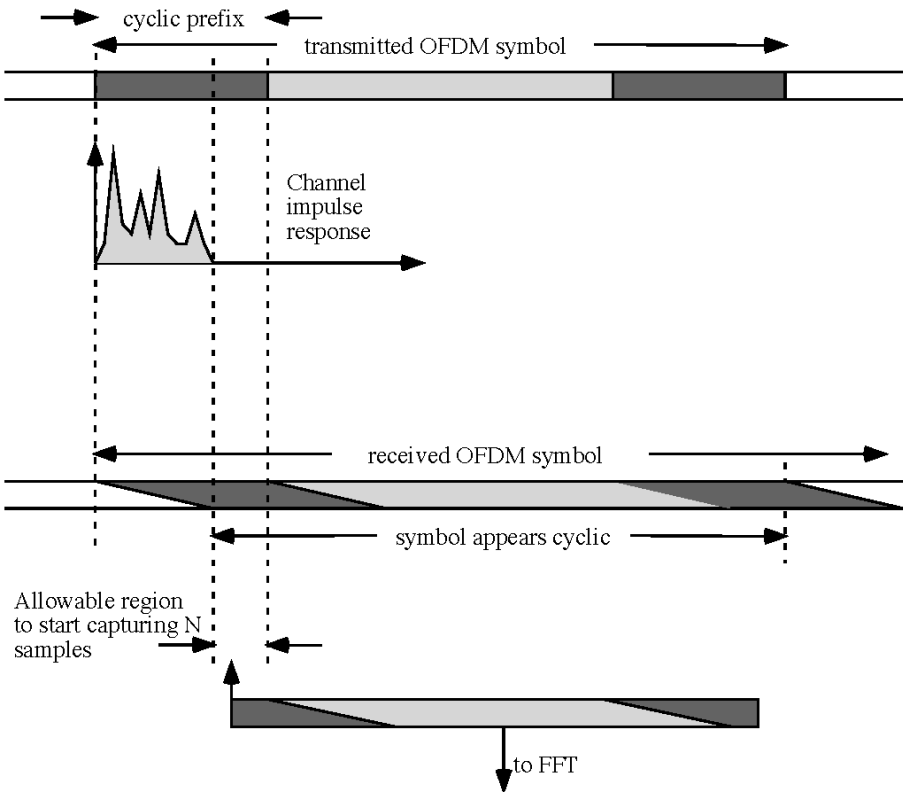


Figure 2.1: Symbol time offset requirement in OFDM systems [82].

In frequency domain approach, the time delays are estimated after resolving the different path's complex gains through matched filter, Fourier transform and successive deconvolution. Root multiple signal classification (Root-MUSIC) has been used for time delay estimation for a spread spectrum based localization system in [86]. Total least square-estimation of signal parameters via rotational invariance techniques (TLS-ESPRIT) has been used in [87] for estimating the time delay. The problem of delay estimation has been converted to a problem of estimation of frequencies of complex sinusoids in a white nonstationary noise and TLS-ESPRIT has then been applied. In [88], for cases with less data samples, forward backward correlation matrix (FBCM) along with TLS-ESPRIT has been used for time delay estimation. In [89], after converting the estimating problem to the problem of complex sinusoid frequency estimation, wavelets are used to de-noise and then TLS-ESPRIT is applied for time delay estimation. The super-resolution ToA estimation algorithm proposed in [90] is a frequency domain algorithm based on MUSIC algorithm, which is applied on measurement data from a indoor radio propagation channel measurement campaign which effectively represents the indoor multipath channel. Performance gain with use of diversity techniques is also discussed with the introduction of Correlation Matrix Based Diversity Combining Scheme (CMDCS). Different diversity samples, time, space and frequency diverse samples, are combined to generate a single correlation matrix before eigenvalue decomposition in [90]. Though these super-resolution techniques can increase time-domain resolution, these also increase complexity for a system implementation. Moreover, for OFDM-WLAN signals, different super-resolution algorithms have been evaluated in [91] and it has been reported that the ToA estimation accuracy leads to a spatial location resolution of 3 to 5 meters. In [92], three different approaches for ToA

estimation of OFDM-WLAN signals have been presented. In the first method, ToA is estimated based on Maximum Peak to Leakage Ratio (MPLR) of the channel impulse response (CIR). In the second approach, reconstruction of the Channel Frequency Response (CFR) is carried and the time delays are estimated based on the first CIR component and in the third method, time delay is estimated based on the real peak of first arrived signal path. These frequency domain ToA estimation methods assume that coarse symbol time synchronization has been carried out, which is followed by a fractional delay search after estimating the channel frequency response based on known pilots tones. A cross-correlation based approach wherein the long preambles of the OFDM-WLAN burst are cross-correlated with the locally stored long preamble replica has been proposed in [95]. After the coarse time delay estimation through cross-correlation, fine time delay adjustment is estimated based on the phase delay of the received signal. Received signal phase delay is estimated based on the gradient of a linear fit to the phase difference between the transmitted and received sub-carriers. This requires cross-correlation over the entire long preamble sequence, which will increase the number of computations and subsequently the hardware implementation complexity. In order to improve time resolution of the estimated time delay, high sampling rates have been used in [96]. Though this improves the time resolution, the number of samples used for cross-correlation become large, which leads to complex hardware implementation and the performance deteriorates rapidly in multipath scenarios, requiring high SNR for reliable ToA estimates.

None of the works published in literature make use of the multiple antennas, which are already introduced in the IEEE 802.11 standards, and provide a low complexity solution for hardware implementation. In this thesis, in Chapter 6, a low complexity algorithm is

presented to estimate fractional time-delay ToA using short preambles of the OFDM-WLAN signal burst. This algorithm is an extension of the proposed timing synchronization algorithm presented in Chapter 4 and it is extended to the scenario where multiple antennas are available in the implementation.

2.3 Conclusions

In this chapter, the different spectrum sensing methods for detecting a primary signal have been presented. The most popular spectrum sensing algorithms are broadly classified under the following four categories: Energy detection, Matched filter detection, Wavelet-based detection and Feature detection (Cyclo-stationary based & Covariance-based detection). Different algorithms available in literature have been briefly reviewed. These algorithms report to achieve different levels of sensing performance in different operational scenarios. Most of the algorithms which have better performance also tend to have higher implementation complexity. It would have been ideal if the reviewed algorithms could provide a mechanism to dynamically trade-off implementation complexity for detection performance. Moreover, the performance of the algorithms is not studied for burst signals, which are more are likely to be used in CR applications, where opportunistic use of spectrum is envisaged. Given that the sensing time is related to the sensing detection performance, for burst signals, it is a challenging task to achieve reliable spectrum sensing detection performance within the short duration of the burst signal. These conflicting requirements of lowering implementation complexity, reduced sensing time and high signal detection performance present an opportunity for development of novel spectrum sensing algorithms. In this direction, a low complexity spectrum sensing algorithm is presented in Chapter 3.

Apart from the review of spectrum sensing algorithms, this chapter also reviewed the different time synchronization algorithms and techniques used for estimating the start-of-frame time of the burst signal and fractional time-delay ToA of the signal. Time synchronization is usually achieved through auto-correlation or cross-correlation of the received signal. Different techniques involving transmission of multiple preamble symbols at the start of the burst and the use of different kinds of unique preamble symbols have been reviewed. Since time synchronization of the incoming burst is the first functional module in a receiver, it needs to be robust for operation in different propagation channel scenarios and against any receiver impairments such as carrier frequency offset or incorrect receiver gain. Along with robust time offset estimation, as time synchronization is performed at sample rate, the hardware implementation needs to be efficient in terms of power consumption, use of hardware resources and maximum clock speed that can be used in the implementation, again, these are conflicting requirements. In Chapter 4, a hardware efficient time synchronization algorithm has been presented along with efficient cross-correlator architecture. The proposed time synchronization is further adapted for fast acquisition of FH signals and for estimation of fractional time-delay ToA of OFDM-WLAN in Chapter 5 of this thesis.

3 Low Complexity Spectrum Sensing based on Segmentation and Decimation

3.1 Overview

In this chapter, a simplified correlation based sensing method is proposed. It takes advantage of the band limited nature of the digitized received signals in the sensing receiver. The sensing receiver is assumed to be frequency agile, which can be tuned to different frequencies [1-4], and the digitizer sampling frequency is programmable.

The spectrum sensing technique is based on the cross-correlation approach. In conventional correlation techniques, multiple correlations are performed before the presence of the signal is determined. Most of these correlation operations are required to resolve the multiple peaks which occur at multiple cyclic frequencies present in the signal to be detected. In the proposed technique, the signal samples are segmented and decimated before performing the correlation. This eliminates multiple peaks at the output of the correlation function and thus reduces the complexity. Further, in the proposed technique, the segmented vectors are averaged over a period to average out the AWGN noise.

The proposed spectrum sensing technique can be used to detect burst signals and individual frequency components present in the primary signal. For example, in the case of detecting the Advanced Television Systems Committee Digital Television (ATSC DTV) signal, the presence of its pilot carrier is detected first by tuning the center frequency of the sensing receiver to the pilot carrier frequency and then the detection of other frequency components present in the DTV signal bandwidth can be carried out. Similarly, for detecting the analog TV signals, the sensing receiver's center frequency can be tuned to either detect video carrier and/or audio carrier. The tuning of the sensing receiver can be out either by changing the RF Front-End local oscillator or by changing the sampling clock of the Analog to Digital Converter (ADC). In the proposed receiver, the RF local oscillator is tuned to down-convert the primary signal's bandwidth and the ADC sampling clock is varied to place the desired frequency component at the center of the digitized passband signal. This is further explained in Section 3.2. In order to validate the performance for burst signals, a Quadrature Phase Shift Keying (QPSK) modulated narrowband signal is used in the simulations. During the detection process, at first, the received signal is digitized with a specific sampling frequency, and then the digitized samples are segmented to generate a set of vectors of digitized input samples. The segmented vectors are then decimated and averaged over a period of time. Cross-correlation between two averaged vectors is performed, and the presence of primary signal is detected from the occurrence of peaks in the magnitude of correlation output. Further, by using FFT of the cross-correlator output, the performance is shown to be improved. In the proposed method, the correlation performed between two decimated vectors and the FFT are the major parts of the total computation. The rest of the computation is for segmentation and averaging of the digitized samples.

3.2 Spectrum Sensing Receiver

A simplified block diagram of an ideal CR transceiver is shown in Figure 3.1. In the transmit path, a wideband power amplifier is connected to a broadband antenna. A digital processing block generates the required transmit samples.. In the receive path a broadband low noise amplifier (LNA) amplifies the RF signals from broadband antenna and the signal is digitized by a broadband analog to digital converter (ADC), with the digitized data being processed by a baseband processor to extract the information bits. There are several practical device limitations for realization of such an ideal architecture.

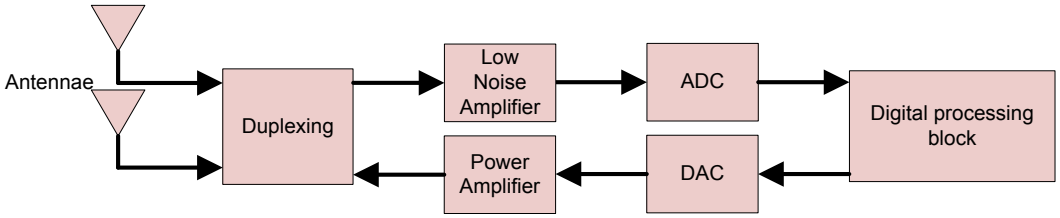


Figure 3.1: Cognitive radio transceiver block diagram [93].

Conventional broadband super heterodyne architectures with parameterized control, such as the one shown in Figure 3.2 are preferred in reconfigurable receivers.

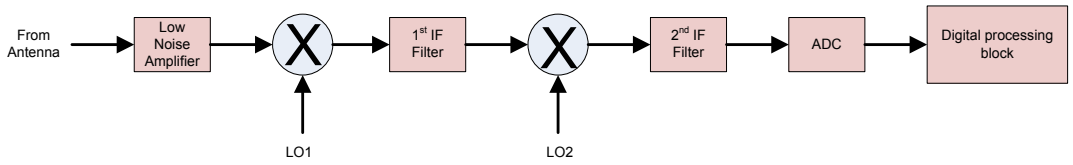


Figure 3.2: Super heterodyne receiver block diagram [94].

In such receivers, the available signal to any spectrum sensing implementation is limited by the intermediate frequency (IF) bandwidth of the IF filters used in the RF receiver. The sampling frequency used in the digitization at the ADC is also determined by the IF filter bandwidth to avoid aliasing in the digital samples.

A simplified block diagram of a CR receiver assumed in the proposed spectrum sensing technique is shown in Figure 3.3. The received RF signal with center frequency 'f_c' and bandwidth 'B' is down converted in an analog RF downconverter to a passband signal of IF frequency 'f_{IF}'. The local oscillator signal of frequency 'f_{osc}' used for frequency down conversion is generated with a reference oscillator of frequency 'f_{ref}'.

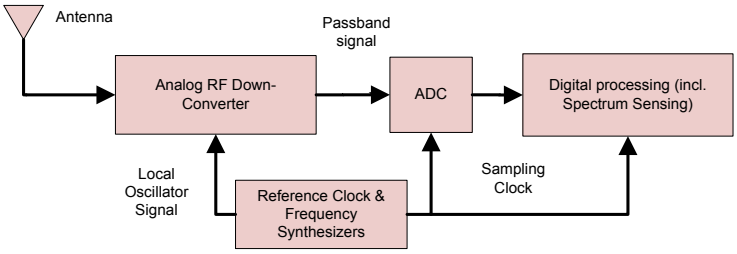


Figure 3.3: Block diagram of proposed CR receiver.

Depending on the received channel frequency of interest 'f_c', the local oscillator is tuned to a particular frequency 'f_{osc}' based on the receiver's IF frequency 'f_{IF}'.

$$f_{IF} = f_c \pm f_{osc} \qquad (3.1)$$

The analog IF passband signal is bandlimited by an IF filter inside the analog RF downconverter. The filtered passband IF signal is digitized by an ADC with a sampling clock of frequency 'f_s'. The sampling clock is also generated from the same reference clock as the local oscillator signal. This sampling clock is varied dynamically to select the frequency of the spectrum which is intended to be detected in the proposed spectrum sensing technique. The sampling clock of frequency 'f_s' used to digitize the down-converted passband signal is chosen such that it satisfies the Nyquist criterion.

$$f_s \geq 2(f_{IF} + B/2) \qquad (3.2)$$

The digitized passband digital IF signal, is centered around 'f_{dif}', which is dependent on the sampling frequency (f_s) and the IF frequency (f_{IF}). For $\left(f_{IF} + \frac{B}{2}\right) = kB$ with 'k'

being an odd integer and $f_s = 2B$, the digitized signal is a prefect down converted IF signal along with perfectly interweaved shifted spectral replicas [39]. This is shown in Figure 3.4(a). Further, as shown in Figure 3.4(b), when 'k' is even, the perfectly interweaved shifted spectral replicas are present but the spectrum is flipped. In the more general case when $\left(f_{IF} + \frac{B}{2}\right) \neq kB$ the sampling frequency should meet the criteria in (3.3) to avoid any aliasing [39].

$$\left(\frac{2\left(f_{IF}+\frac{B}{2}\right)}{k} \leq f_s \leq \frac{2\left(f_{IF}-\frac{B}{2}\right)}{k-1}\right) and \left(k \leq \frac{\left(f_{IF}+\frac{B}{2}\right)}{B}\right) \qquad (3.3)$$

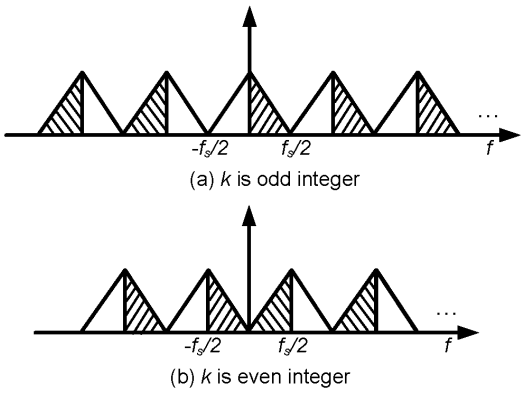


Figure 3.4: Bandpass sampling spectral replicas [39].

To tune the spectrum sensing receiver for detecting different frequency components in the received signal, the sampling clock frequency (f_s), which satisfies (3.3), could be changed dynamically by tuning the frequency synthesizer that generates the sampling clock for the ADC. The same can also be achieved by fixing the sampling clock frequency to a fixed frequency and tuning the local oscillator frequency 'f_{osc}' such that the different frequency components of the receiver's RF input signal are mixed with the local oscillator signal to be downconverted to the desired IF signal.

The digitized samples 'x_n' of the IF signal which represent the passband signal are centered around the IF (f_{IF}). These samples are processed to demodulate and decode the

transmitted information. Though the modulated information is with a definite passband bandwidth, on the received signal is randomized, the IF over which it is downconverted to is periodic in nature. The periodicity of the digitized IF is exploited for signal detection in the proposed scheme. However, since the periodicity is embedded in the randomized modulated signal, the proposed scheme extracts the hidden periodicity and detects the presence of signal. The sensing result is used to determine if the frequency band is occupied or free for use by a CR.

3.3 Proposed Low-Complexity Spectrum Sensing Technique

The digitized passband samples (x_n) represent the sum of all the time domain samples of all the frequency components in it. To extract the digital samples representing a particular frequency component one could use a narrowband filter. But to realize a sharp cut-off narrowband filter, a high-order digital filter would be required. This will increase the hardware resource needed for implementation. In this section a novel segmentation and decimation technique is proposed to extract the samples of a single frequency component and use those for detecting the presence of that frequency component.

3.3.1 Segmentation and Decimation

The digitized passband samples 'x_n' are a continuous stream of data samples which are oversampled at the rate 'O_s' and are the input to the spectrum sensing algorithm. The proposed algorithm, arranges the continuous stream of data samples into segments, each of length 'i'. These segmented vectors are represented as '$\boldsymbol{Y}_u$'.

$$\boldsymbol{Y}_u = \left[x_{n+1}\ x_{n+2}\ x_{n+3} \ldots\ x_{n+(i-2)}\ x_{n+(i-1)}\ x_{n+i} \right] \quad (3.4)$$

where $u = \{1, 2, \ldots i\}$, $n = i(u-1)$, $i = 2^t O_s$, and 't' is zero or any positive integer. Exactly 'i' such segmented passband sample vectors ($\boldsymbol{Y}_u$, where $u = \{1, 2, \ldots i\}$) are grouped to form a block of vectors 'B_v'. 'v' such blocks, each of 'i' vectors are formed.

$$B_v = \begin{bmatrix} \boldsymbol{Y}_1 \\ \boldsymbol{Y}_2 \\ . \\ . \\ . \\ \boldsymbol{Y}_{i-1} \\ \boldsymbol{Y}_i \end{bmatrix} = \begin{bmatrix} x_1\ x_2\ x_3 \ldots\ldots\ldots . x_{i-2}\ x_{i-1}\ x_i \\ x_{i+1}\ x_{i+2}\ x_{i+3} \ldots\ldots\ldots\ x_{i+(i-2)}\ x_{i+(i-1)}\ x_{2i} \\ . \\ . \\ . \\ x_{(i-2)i+1}\ x_{(i-2)i+2}\ \ldots\ x_{(i-2)i+(i-1)}\ x_{(i-1)i} \\ x_{(i-1)i+1}\ x_{(i-1)i+2} \ldots\ldots\ x_{(i-1)i+(i-1)}\ x_{i^2} \end{bmatrix} \quad (3.5)$$

The diagonal elements of this '$i \times i$' passband sample matrix are extracted to form a vector '$\boldsymbol{S}_v$'. This '$i \times i$' sample matrix could also be viewed as a segmented vector of digitized passband samples of length 'i^2', which are then decimated to generate the vector '$\boldsymbol{S}_v$'. Hence, these vectors are termed "intra-segment decimated vectors" (ISDV). The elements of a sequence of samples starting with sample 'x_1' are shown encircled in (3.6) for the ISDV '$\boldsymbol{S}_1$' given in (3.7).

$$\begin{bmatrix} (x_1) & x_2 & \ldots\ldots\ldots & x_i \\ x_{i+1} & (x_{i+2}) & \ldots\ldots\ldots & x_{2i} \\ \ldots & \ldots & \ldots\ldots\ldots & \ldots \\ x_{i(i-1)+1} & x_{i(i-1)+2} & \ldots\ldots\ldots & (x_{i^2}) \\ (x_{i^2+1}) & x_{i^2+2} & \ldots\ldots\ldots & x_{ni^2+i} \\ x_{i^2+i+1} & (x_{i^2+i+2}) & \ldots\ldots\ldots & x_{i^2+2i} \\ \ldots & \ldots & \ldots\ldots\ldots & \ldots \\ x_{i^2+i(i-1)+1} & x_{i^2+i(i-1)+2} & \ldots\ldots\ldots & (x_{2i^2}) \\ \ldots & \ldots & \ldots\ldots\ldots & \ldots \end{bmatrix} \quad (3.6)$$

$$\boldsymbol{S}_1 = [x_1\ \ x_{ni+2}\ \ x_{2i+3}\ \ \ldots\ \ x_{i^2}] \quad (3.7)$$

and

$$\boldsymbol{S}_2 = [x_{i^2+1}\ \ x_{ni^2+i+2} \ldots\ldots\ldots\ \ x_{2i^2}] \quad (3.8)$$

Similarly subsequent vectors $\boldsymbol{S}_3, \boldsymbol{S}_4, \ldots, \boldsymbol{S}_v$ are computed.

The sample index of the samples in each segmented vector '$\boldsymbol{S}_v$' can be computed based on the length of each vector 'i' and the total number of generated ISDVs 'm'. '$\boldsymbol{S}_{v,e}$', the 'e^{th}' sample in the 'v^{th}' ISDV is given by

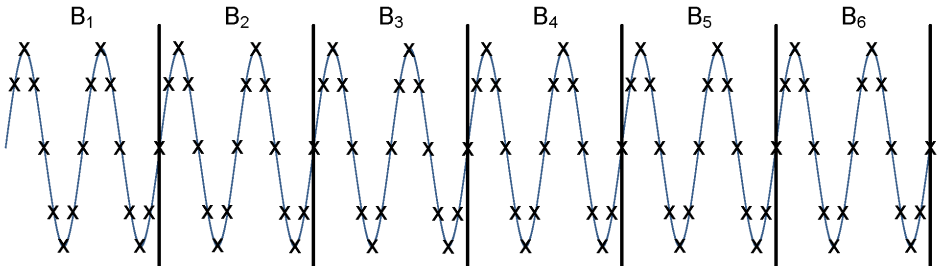
$$\boldsymbol{S}_{v,e} = x_n, \text{ where } n = 1 + i^2(v-1) + (i+1)(e-1) \qquad (3.9)$$

$e = \{1, 2, \dots, i\}$ is the element index within each vector of length 'i' and $m = \{1, 2, \dots, i\}$, is the vector index within the set of 'm' vectors.

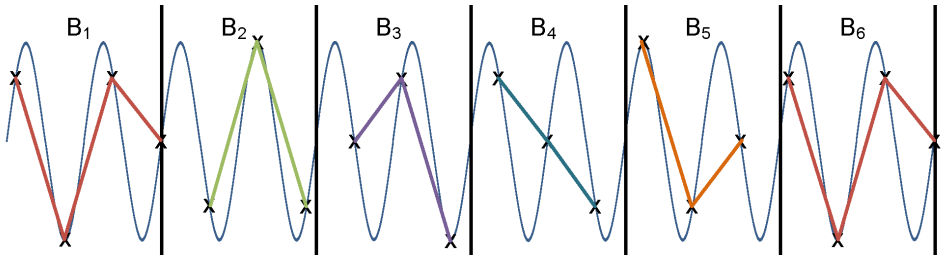
This process of generating the ISDVs by segmenting and decimating the digitized samples is quite different from conventional decimation. In conventional decimation, the decimated sample after sample 'x_{i^2}' would have been 'x_{i^2+i+1}'. In the proposed scheme, it is the start of another ISDV with sample 'x_{i^2+1}'. The physical significance of this segmentation and decimation is shown in Figure 3.5.

When the digital samples of a sinewave which has been oversampled by 8x, as shown in Figure 3.5(a), is segmented with '$i = 4$', the segmented blocks '$B_1, B_2, B_3, \dots$' each of sample length '$i^2 = 16$' are shown. If this digitized sinewave is visualized to be decimated with '$(i + 1) = 5$', then the samples in each segment, '$\boldsymbol{S}_1$'are shown in Figure 3.5(b), Figure 3.5(c) and Figure 3.5(d). The decimation factor of 5 is for this particular illustration's visualization only. The proposed scheme performs the segmentation and decimation by creating sample vectors, blocks out of those vectors and selecting the diagonal elements of the block for decimation as in (3.5), (3.6) and (3.7).

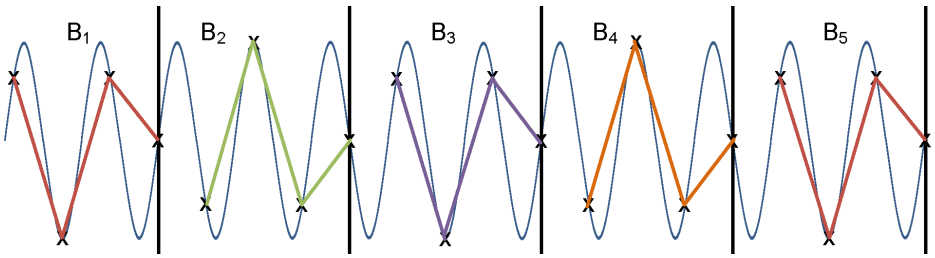
The segments shown in Figure 3.5(b) are generated by performing segmentation first on the digital samples and followed by conventional decimation, as can been these segments are not identical to each other. However, Figure 3.5(c) shows the samples which are segmented, each of segment length '$i = 4$'. These segments are generated by performing segmentation after conventional decimation. It can be seen in Figure 3.5(c) that the time-domain envelop of the samples in each segment is not the same.



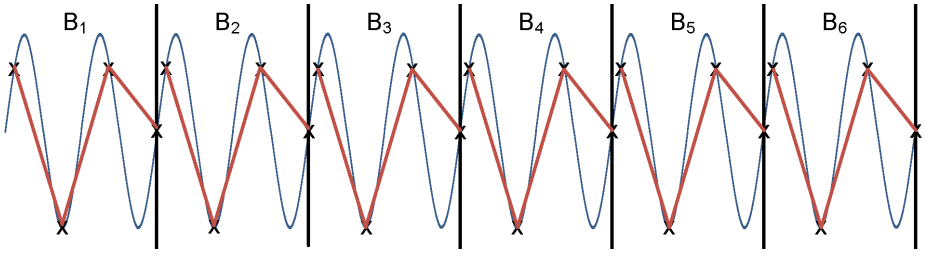
(a) Sinewave sampled at 8x and segmented into blocks of 16 samples



(b) Samples in each segment after conventional decimation



(c) Samples in each segment, segmentation after conventional decimation



(d) Sample in each segment, proposed segmentation and decimation

Figure 3.5: Conventional and proposed segmentation and decimation, for *i*= 4.

They are inverse of the time-domain envelop of the samples in the adjacent segment.. Figure 3.5(d) shows the samples in each segment as decimated by the proposed segmentation and decimation process. It can be seen that the proposed method produces identical segments in Figure 3.5(d). The identical nature of segmented and decimated samples (ISDV samples) is used to determine the presence of a signal.

Each element of the segmented vector '$\boldsymbol{S}_{v,e}$' is made up of the desired signal component and a noise component. Since the additive noise has zero correlation over a period of time, for a fixed threshold, increasing the number of samples 'i' in each ISDV vector, which is akin to increasing the oversampling ratio and oversampling gain, also improves the noise immunity of the proposed sensing technique.

3.3.2 Averaging

For this the sequence of generated ISDVs are grouped into 'k' blocks, with each block containing 'm' ISDVs. This block of 'm' ISDVs are averaged to generate an averaged ISDV denoted as '$\boldsymbol{A}_k$'.

$$\boldsymbol{A}_k = \frac{1}{m} \sum_{v=(k-1)m+1}^{(k-1)m+m} \left(\boldsymbol{S}_{v,e}\right), e = \{1, 2, \dots, i\} \; and \; k = \{1, 2, \dots\} \quad (3.10)$$

Increasing the number of ISDV vectors 'm' in each block for averaging, also improves the noise performance of the proposed sensing technique as the additive noise is averaged out over longer period of time.

3.3.3 Cross-Correlation

The presence of the signal to be detected is determined by cross-correlating any two averaged vectors ($\boldsymbol{A}_p$ and $\boldsymbol{A}_q$) and the output vector, $\boldsymbol{R}_{pq}$, is used for determining the occupancy of the passband channel.

$$R_{pq}(j) = \sum_{l=0}^{i-1} (A_p(l) A_q(l+j)), \qquad \text{where } p \neq q \ \& \ p \ \& \ q = \{1, 2, \dots, k\} \quad (3.11)$$

Substituting (3.10) in (3.11) we have

$$R_{pq}(j) = \frac{1}{m^2} \sum_{l=0}^{i-1} \left(\left(\sum_{v=(l-1)m+1}^{(l-1)m+m} \mathbf{S}_{v,e}^{p} \right) \left(\sum_{v=(l+j-1)m+1}^{(l+j-1)m+m} \mathbf{S}_{v,e}^{q} \right) \right) \quad (3.12)$$

$$\text{where } p \neq q \ \& \ p \ \& \ q = \{1, 2, \dots, k\}$$

3.3.4 Threshold and Decision

The magnitude of the cross-correlator output $\left|R_{pq}(j)\right|$ is an indicator of the presence of the signal. This is compared against a threshold to determine the occupancy. The magnitude is maximum when the two averaged ISDV vectors are identical. In such a case, depending on the peak value of the cross-correlator output, it is determined that the signal is present. However determining this threshold for comparison is difficult and is dependent on the noise power. Hence the FFT of the cross-correlator is taken.

$$T_{mag} = \left|R_{pq}(j)\right|, \ \text{where } p \neq q \ \& \ p, q = \{1, 2, \dots, k\} \qquad (3.13)$$

$$T = FFT(\left|R_{pq}(j)\right|), \qquad \text{where } p \neq q \ \& \ p, q = \{1, 2, \dots, k\} \qquad (3.14)$$

For an ideal scenario, in the presence of signal with high SNR, $\boldsymbol{A}_p$ and $\boldsymbol{A}_q$ are identical and (3.12) yields a single peak at a frequency bin in FFT output dependent on the number of samples 'i' in the ISDV vector. The presence of the signal is determined using this property.

3.4 Complexity Analysis

The proposed sensing method has four major computational steps. Segmentation and decimation, averaging, cross-correlation and the FFT in the decision making process. The computational complexity of the proposed ISDV cross-correlation based sensing technique is presented in Table 3.1. The number of real multiplications and additions are listed for each sub-operation, and are presented in terms of the number of samples per vector 'i', and the number vectors 'm' in each block, which are averaged before cross-correlation.

Table 3.1: Proposed spectrum sensing's computational complexity

Computation Step	Real Multiplications	Real Additions
Segmentation and Decimation	$4i$	$2m(i^2+2)$
Normalization and Averaging	$2m+3i+2mi+1$	$(2i-1)(2m+1)+2i(2m-1)+2m(2n-1)$
Correlation	$i(i-1)$	$(i-1)$
FFT	$ilog(i)$	$ilog(i)$
Total	$i^2+8m+4i+ilog(i)+1$	$6i^2m+8m+ilog(i)+2$

The computations involved in the segmentation and decimation process for deriving the index of the individual samples are the major part of complexity of the proposed technique. The determination of sample index can be performed using a simple control circuit for segmentation and decimation at the front-end of digital processing, with very low complexity. The total computational complexity when a control circuit is used, would amount to $[i^2+2(m+i)+2mi+ilog(i)+1]$ multiplications and $[12mi+i-4m+ilog(i)-2]$ additions, which are the computations required only for normalization, averaging, cross-correlation and FFT as given in Table 3.1. The number of computations increases with the number of vectors 'm' and the number of samples per vector 'i'. This allows for trading off the performance for complexity by appropriately selecting m and i. The simplest known autocorrelation-based detection scheme of [32] computes the

decision variables for detection of frequency component based on the sum of n auto-correlation vectors for detection of each frequency component. The complexity of proposed technique is only $\cong 1/i$ times of the method in [32], as only one cross-correlation is performed for deriving correlation vector which is used for detection of each frequency component.

3.5 Simulation Studies

The proposed sensing technique based on ISDV cross-correlation has been validated with the help of MATLAB simulations. The simulations are based on computer generated signals and captured data samples. As the proposed technique is different from conventional approaches to spectrum sensing, the ISDV correlation technique was first validated with single tone signal, and then with multi-tone signals generated through computer simulations. Captured signals and a QPSK modulated narrowband burst signal were used to further validate the proposed technique. The captured real world signals used were conforming to ATSC DTV signal format [97] and Analog TV signal format [98]. ATSC DTV signal is a 6MHz bandwidth, 8-VSB modulated signal with flat spectrum except for the band edges. The spectral components of the ATSC signal are shown in Figure 3.6. At the suppressed carrier, 310KHz from the lower frequency band edge a pilot is inserted. The pilot carrier is detected in the proposed sensing method. The analog TV signal used in the simulation is of PAL-B format with 8MHz bandwidth spectrum. The frequency components of PAL-B signal are shown in Figure 3.7.

The PAL-B signal contains a video carrier, audio carrier and color-subcarrier signals along with FM sound, chroma and vision sidebands. The proposed method is used to sense the video and audio carrier signals. The QPSK modulated burst signal with a bandwidth of 50KHz and burst duration 10msec has been generated at digital IF of 1MHz

and 2MHz. The sample rate used was 40MHz and different sensing durations over different multipath channels were used for detection.

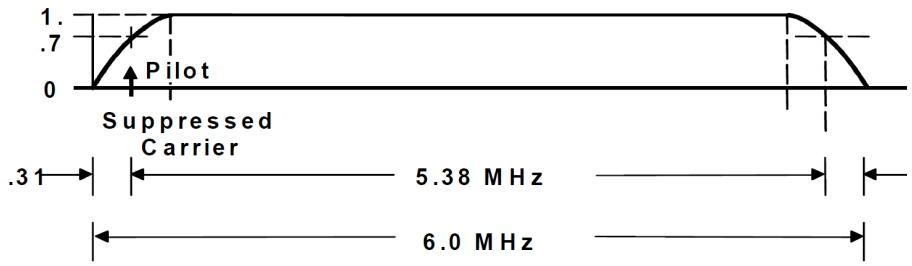


Figure 3.6: ATSC DTV 8-VSB spectrum [97].

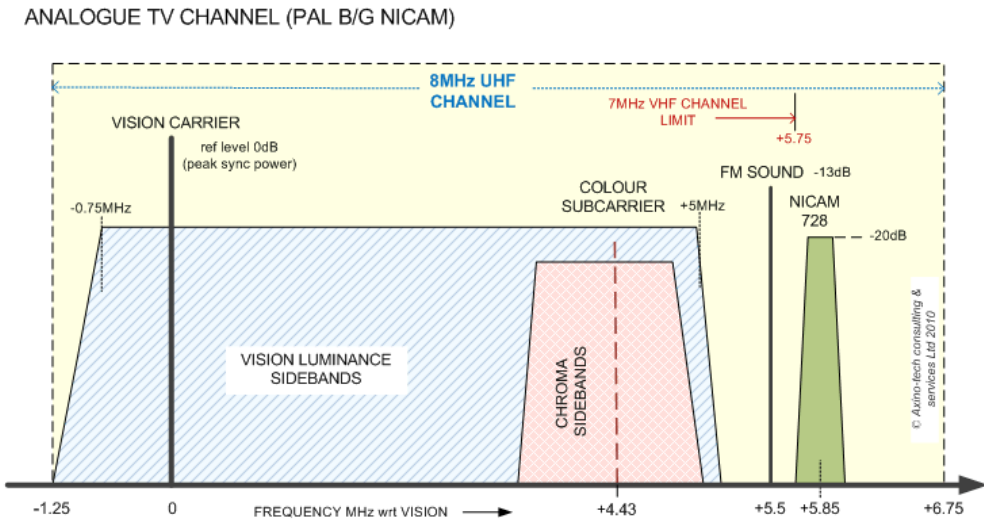


Figure 3.7: PAL-B Analog TV spectrum [98].

A single-tone signal with a frequency of 25MHz was generated at a sampling frequency of 250MHz. A Multi-tone signal with 5 frequency components at 21MHz, 23MHz, 25MHz, 27MHz and 29MHz, was generated with a sampling frequency of 250MHz. The 25MHz signal was considered as the desired signal component whose presence needs to be detected. The input spectrum, resultant averaged ISDV vectors and the cross-correlator output for single-tone and multi-tone signals are shown in Figure 3.8.

In the simulation 'i' is set as 10. Figure 3.8(a) shows the averaged ISDV and cross-correlator output for single tone input and Figure 3.8(b) shows the averaged ISDV and

cross-correlator output for multi-tone signal. The peak of the cross-correlator output is at the sample index 'i', as can be seen in Figure 3.8.

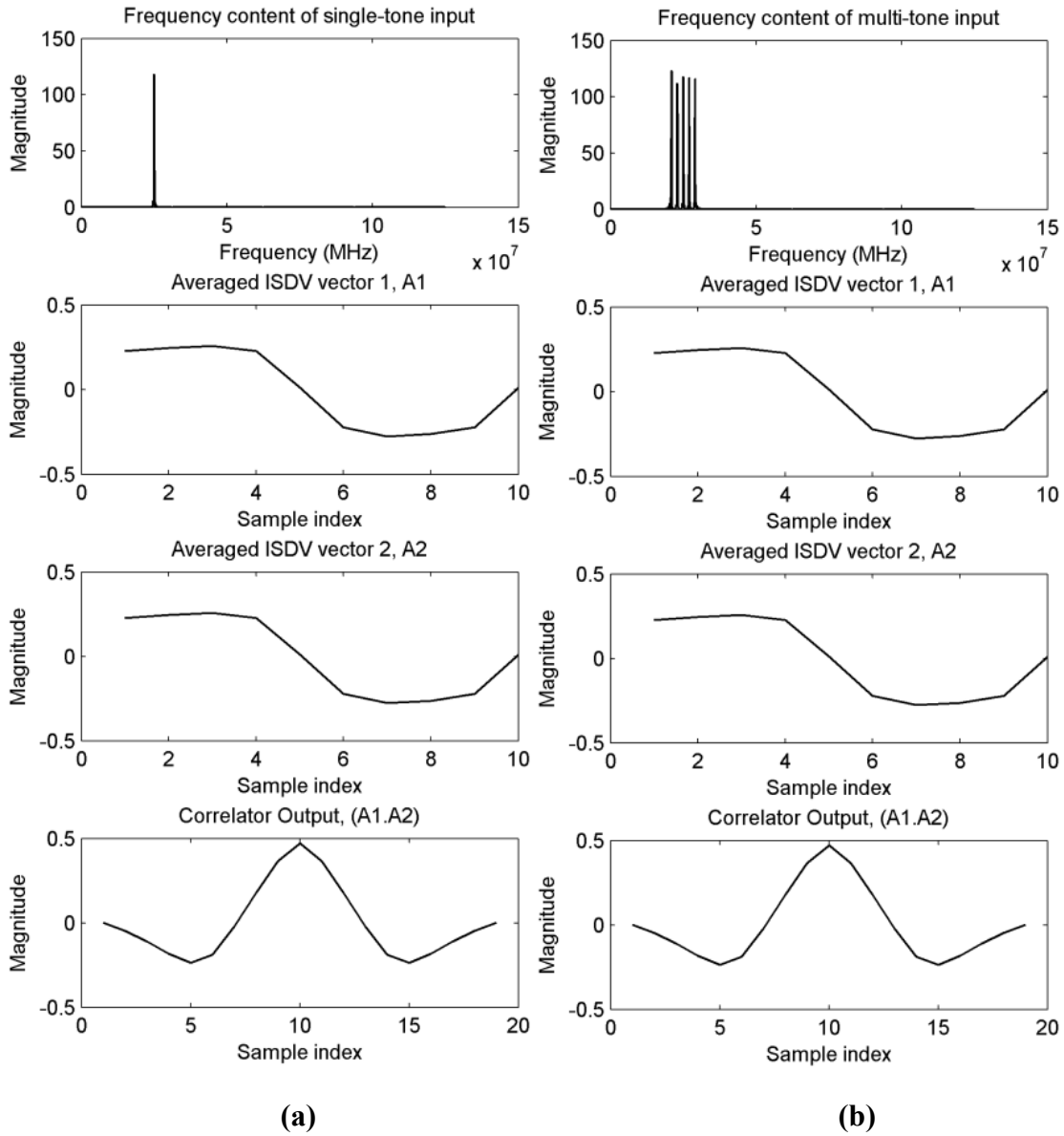


Figure 3.8: (a) Single-tone and (b) Multi-tone signal.

The presence of the desired signal is determined by comparing the magnitude of sample index 'i' of the cross-correlator output vector against a set threshold. Ideally, the threshold is proportional to 'i' for normalized vectors. However, the threshold depends on the required noise performance, especially since correlation distribution of non-

cyclostationary signal, such as AWGN, could be approximated by a Gaussian distribution for a sufficiently large 'm' and would be irrelevant to cyclic frequency correlation peak.

The proposed sensing technique was further validated with Digital TV and Analog TV signals. The ATSC Digital TV signal with an analog bandwidth (BW) of 6MHz was generated at Ultra High Frequency (UHF) channel 14 at 470MHz, using Agilent's Signal Studio for Digital Video and E4438B Signal Generator for RF up-conversion. The PAL-B Analog TV signal of BW 8MHz was generated using Taylor CCIR4 AUVR TV Channel Modulator. The experimental setup used to capture the ATSC DTV and PAL-B Analog TV signal data is shown in Figure 3.9.

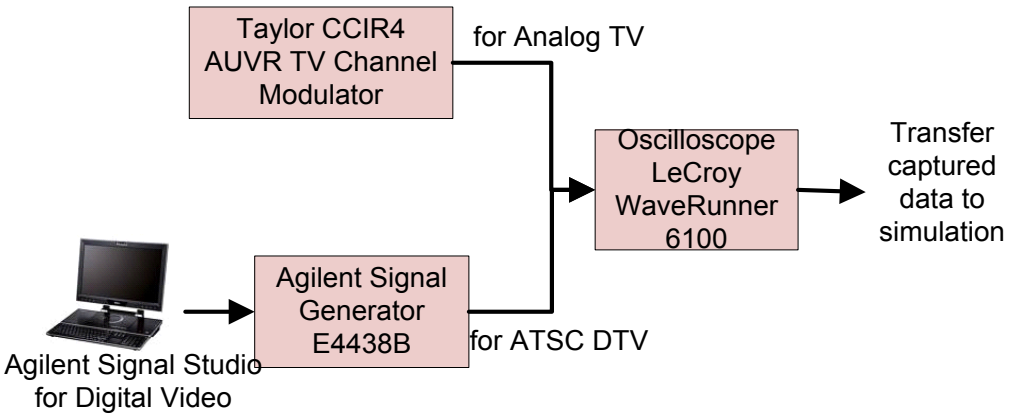


Figure 3.9: Setup used to capture the ATSC DTV and Analog TV signals.

All the analog signals were digitized with a sampling frequency of 1GHz using a LeCroy WaveRunner 6100 Oscilloscope. The digitized DTV and ATV signals were input to the MATLAB simulation model and pre-processed by a digital down converter (DDC) before the proposed spectrum sensing algorithm is implemented.

In the MATLAB model, the ATSC DTV signal is down converted by the DDC such that the DTV's pilot signal was centered at a frequency of 25MHz digital IF. Similarly, the analog TV signal was down converted twice, once with the ATV's video carrier

centered at 25MHz digital IF and the other with ATV's audio carrier set at a frequency of 25MHz digital IF.

The input spectrum of the DTV and ATV signals, the resultant ISDV vectors and the cross-correlator output for ATSC DTV and PAL-B Analog TV captured signals are shown in Figure 3.10. In the simulation, the parameters where set as, $i = 40$, $m = i$ and SNR = 0dB. Figure 3.10(a) shows the results for the input ATSC DTV signal, with its pilot frequency set at 25MHz IF. The averaged ISDV and the corresponding correlator output are shown. Similarly, the results of PAL-B ATV signal, which has its video carrier set to 25MHz is shown in Figure 3.10(b) along with its ISDVs and their cross-correlator output. Figure 3.10(c) shows the PAL-B ATV signal which has its audio carrier set to 25MHz. As can be seen in Figure 3.10, the magnitude of the correlator output has a peak at sample 'i' for all the three signals.

The presence of the desired 25MHz signal is determined by comparing the correlator output vector's sample index 'i' to a set threshold. Monte-Carlo simulations were carried out by varying the SNR. The probability of detection (P_d) for different SNRs is determined and the results are shown in Figure 3.12, Figure 3.13, Figure 3.14.

The MATLAB simulation setup for proposed spectrum sensing process is shown in Figure 3.11. The digitized input in preprocessed in the digital down converter to the appropriate bandwidth, followed by addition of Additive White Gaussian Noise (AWGN) to set a particular SNR for the signal. The noise added signal samples are first segmented and decimated to generate the blocks of ISDVs. These are then averaged and cross-correlated. The cross-correlator output is used to determine the detection of the signal.

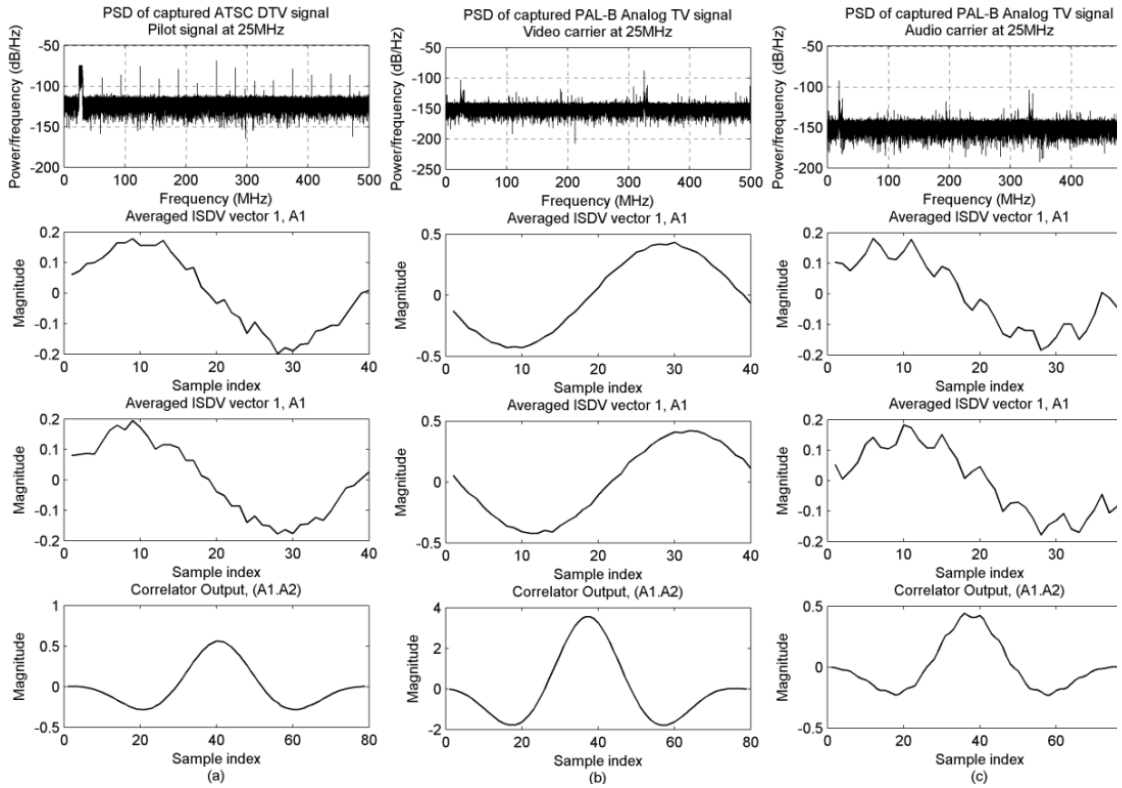


Figure 3.10: ATSC DTV and PAL-B Analog TV detection results.

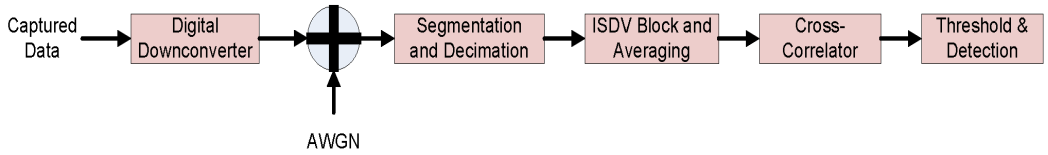


Figure 3.11: MATLAB simulation setup for proposed spectrum sensing.

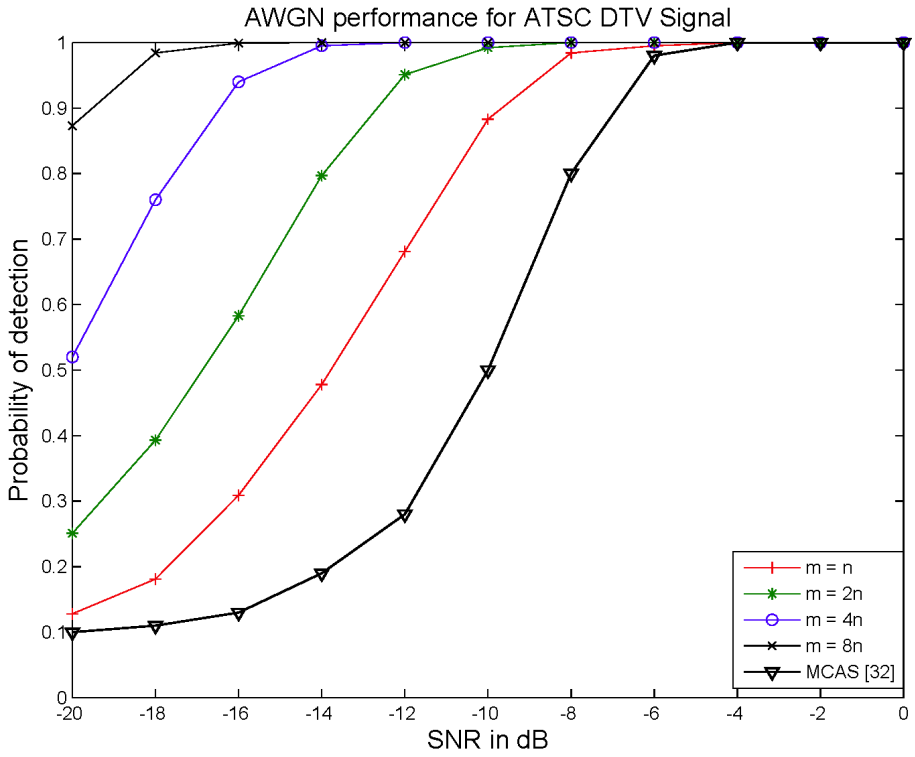
Figure 3.12 gives the probability of detection (P_d) and probability of false alarm for ATSC DTV signal for various 'm'. The ISDV vector length 'i' is set to 40, the oversampling rate. It is seen from Figure 3.12(a) that the probability of detection (P_d) improves significantly with increase in 'm', particularly for very low SNR scenarios. Therefore, to have higher probability of detection (P_d) at very low SNR, higher number of ISDV vectors could be used, which are averaged to generate the averaged out vectors. This results in marginal increase of the computational complexity resulting from

averaging as given in Table 3.1. When compared to the performance results of the Maximum Cyclic Autocorrelation Selection technique as presented in [32], the proposed sensing technique has higher probability of detection in low SNR scenarios. However, the probability of false alarm as shown in Figure 3.12(b) is below 0.05 for all 'm'.

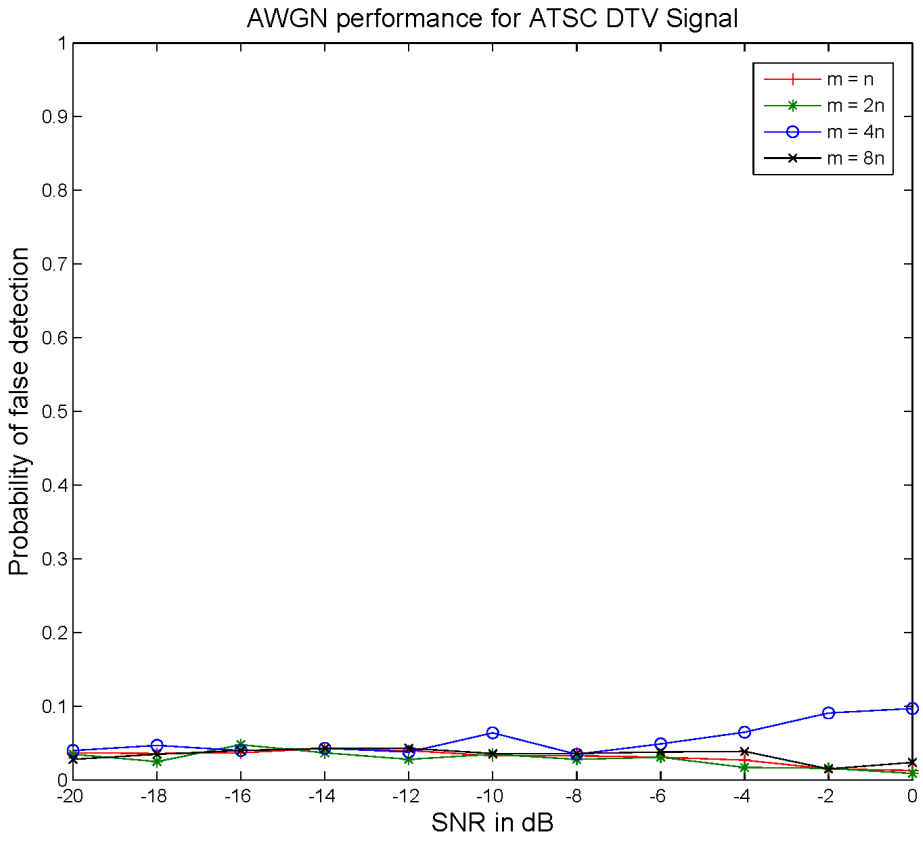
The probability of detection (P_d) performance for PAL-B analog TV signals in AWGN scenarios is shown in Figure 3.13. First, the audio carrier is detected, which is set to 25MHz digital IF. The number of samples in the ISDV (i) is set to 40, the oversampling ratio. The probability of detection (P_d) as shown in Figure 3.13(a) is 100% for SNRs above 6dB for '$m = i$' and the performance improves significantly with increase in m and it is 100% at 12dB with $m = 8i$. The probability of false alarm shown in Figure 3.13(b) remains at ~5% for all m for all simulated SNRs upto -20dB.

The probability of detection (P_d) performance improves significantly when video carrier is used to detect the PAL-B analog TV signal. This is mainly due to the fact that the video carrier is 20dB higher than audio carrier. Figure 3.14(a) shows the probability of detection (P_d) for different m and SNRs. The ISDV sample length was same as for audio carrier, $i = 40$. The probability of false alarm is also found to be ~5% for video carrier detection. Figure 3.14(b) shows the probability of false alarm for different m and SNRs from -40dB to 0dB.

One aspect to note with respect to simulations related to probability of false alarm for ATSC DTV detection, PAL-B TV signal's video and audio carrier detection is that they all produce the same results. This is because the random noise generated with the same seed was used in place of actual signal and the threshold for detection, peak of the FFT output, is the same for all the three signal detection scenarios.

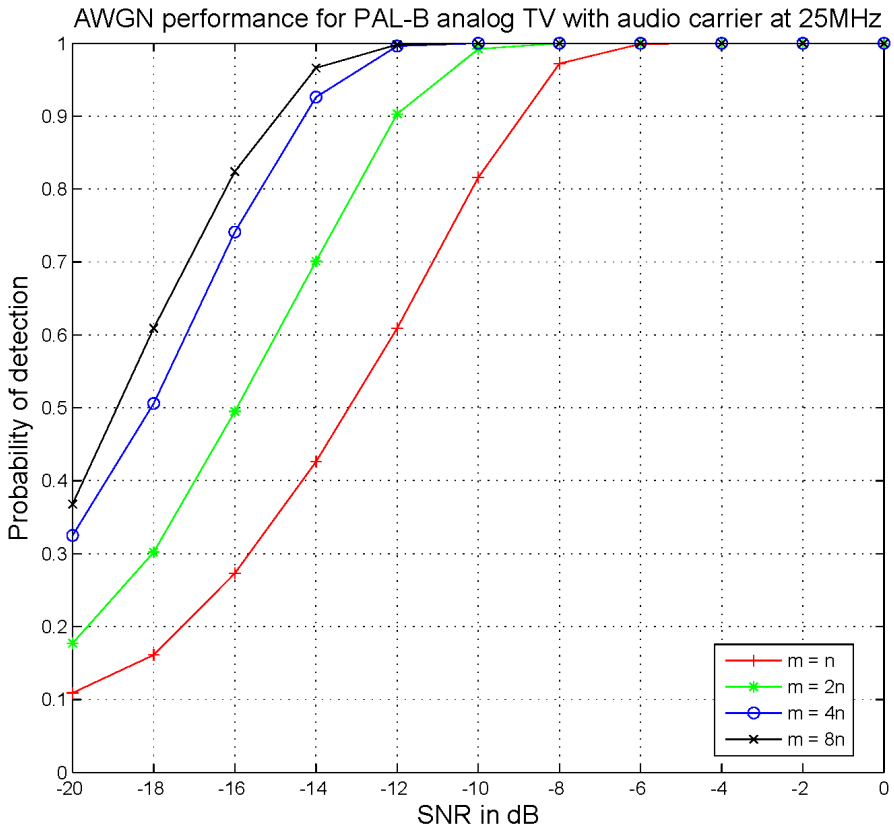


(a) Probability of detection

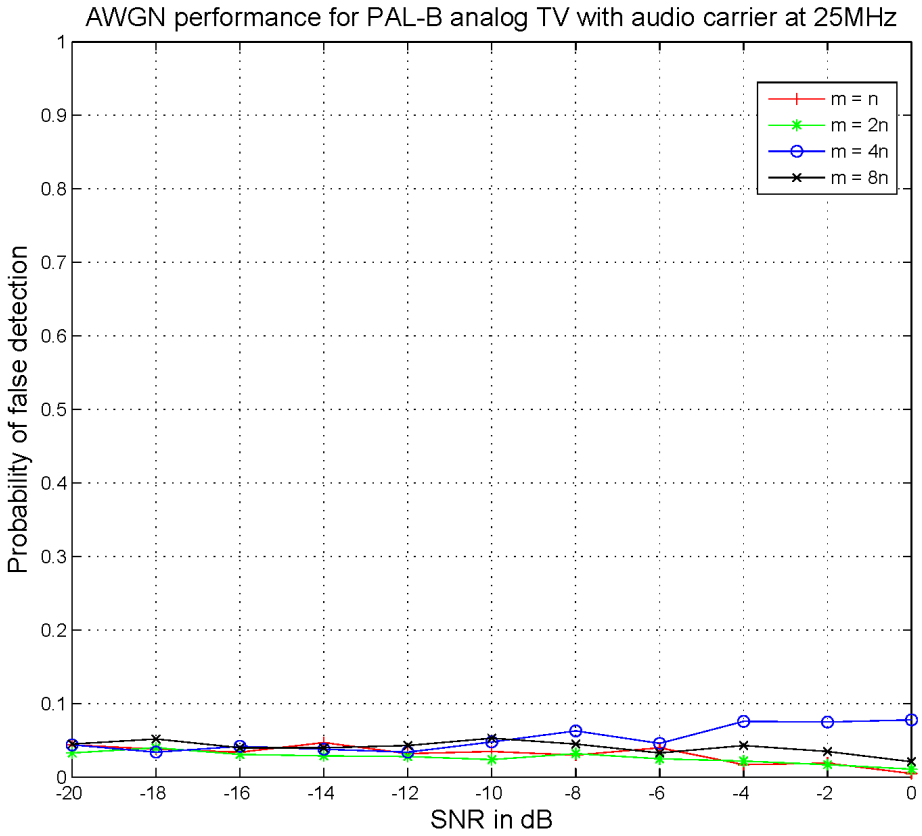


(b) Prpbability of false alarm

Figure 3.12: ATSC DTV signal detection performance for AWGN.

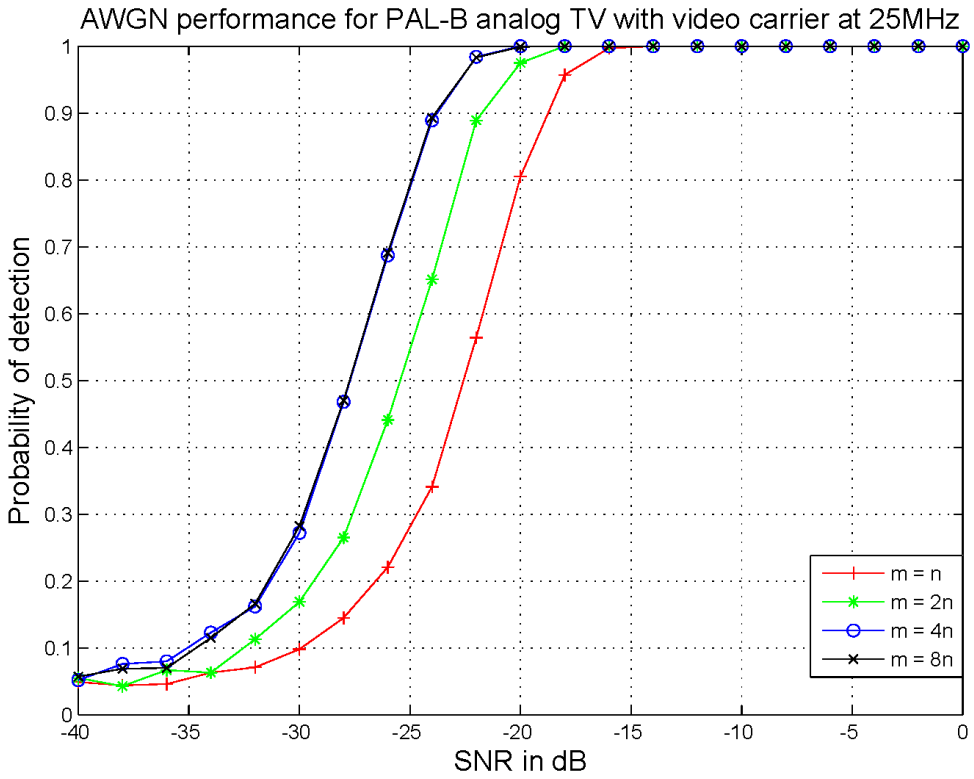


(a) Probability of detection

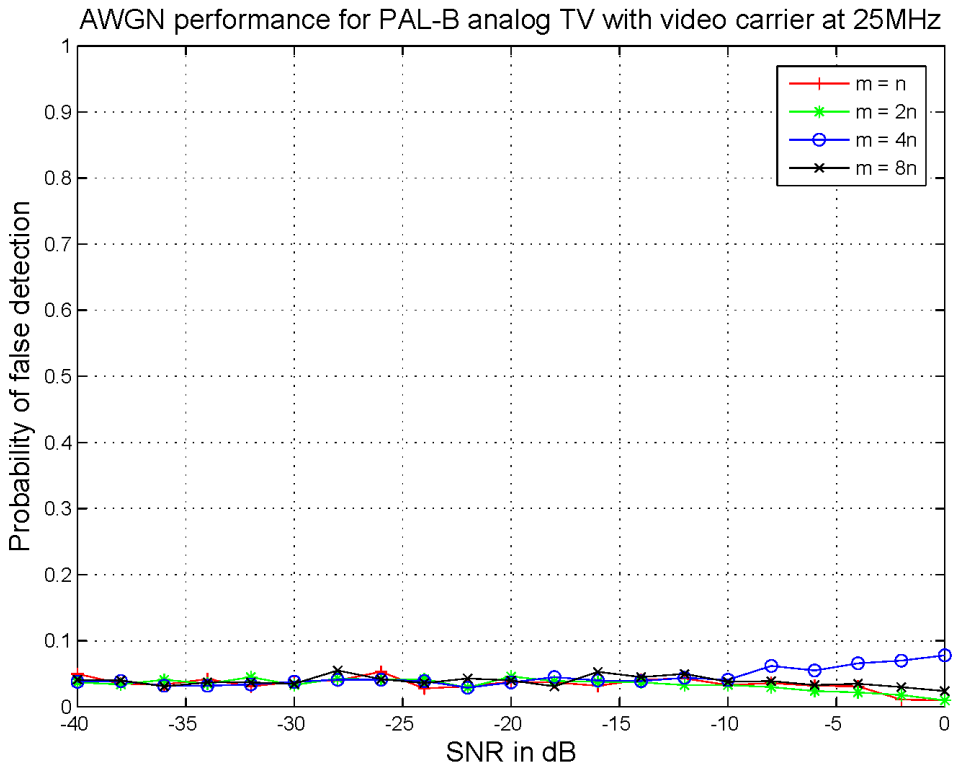


(b) Prpbability of false alarm

Figure 3.13: PAL-B analog TV audio carrier detection performance for AWGN.



(a) Probability of detection



(b) Probability of false alarm

Figure 3.14: PAL-B analog TV video carrier detection performance in AWGN.

For validating the proposed spectrum sensing technique's ability to detect burst signals at multiple channel frequencies, burst signals of 10msec duration, 50KHz bandwidth and centered around 1MHz and 2MHz digital IF have been generated and used. The length of the ISDV vector 'm' has again been set to 40 and the number of vectors 'm' which are averaged has been see to 40, 80 and 160. At the set sample rate of 40MHz, these 'm' correspond to a sensing duration of 3.2msec, 6.4msec and 12.8msec respectively. For 2MHz digital IF channel, the 'm' has been set to 20 to simulate 1.6msec sensing duration. The performance simulations were carried out for AWGN and 6-path Rician (k_factor=79) and Rayleigh fading channels with a maximum Doppler frequency of 300Hz. The gain vectors for the Rician and Rayleigh channels are {0, -1.0, -9.0, -10.0, -15.0, -20.0} and {0, -1.5, -1.4, -3.6, -0.6, -9.1, -7.0, -12.0, -16.9} respectively. The delay vector for the simulated channels are {1.0e-8*[0 120 280 430 685 1000]} and {1.0e-8* [0 30 150 310 370 710 1090 1730 2510]} for Rician and Rayleigh respectively. These were derived from the LTE [102] channel models and the delay has been modified to account for a narrower 50KHz signal bandwidth when compared to LTE signal bandwidth.

Figure 3.15 shows the probability of detection (P_d) in AWGN for the FFT based detection, magnitude based threshold for detection, as in (3.12) and proposed scheme (3.14). It can be seen that the probability of detection (P_d) for magnitude based threshold for detection has poor performance in all sensing durations. Further, It can be seen that for a sensing duration 't' within the burst duration the proposed sensing technique performs better. However when the sensing duration is more than the burst duration the performance falls below that of the FFT based detection. For the FFT based detection, as the sensing duration is increased, there are large numbers of sample points which are used for performing the FFT, this leads to better performance. It should however be noted that since there are more number of samples, the number of computations are also larger.

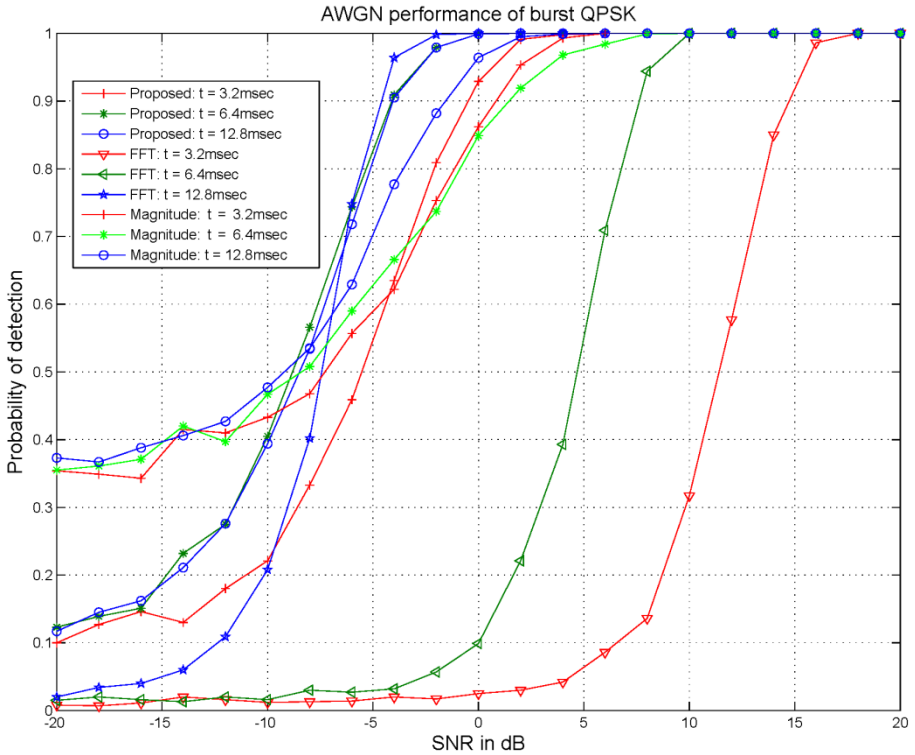


Figure 3.15: Probability of detection in AWGN for proposed, FFT and Magnitude approach.

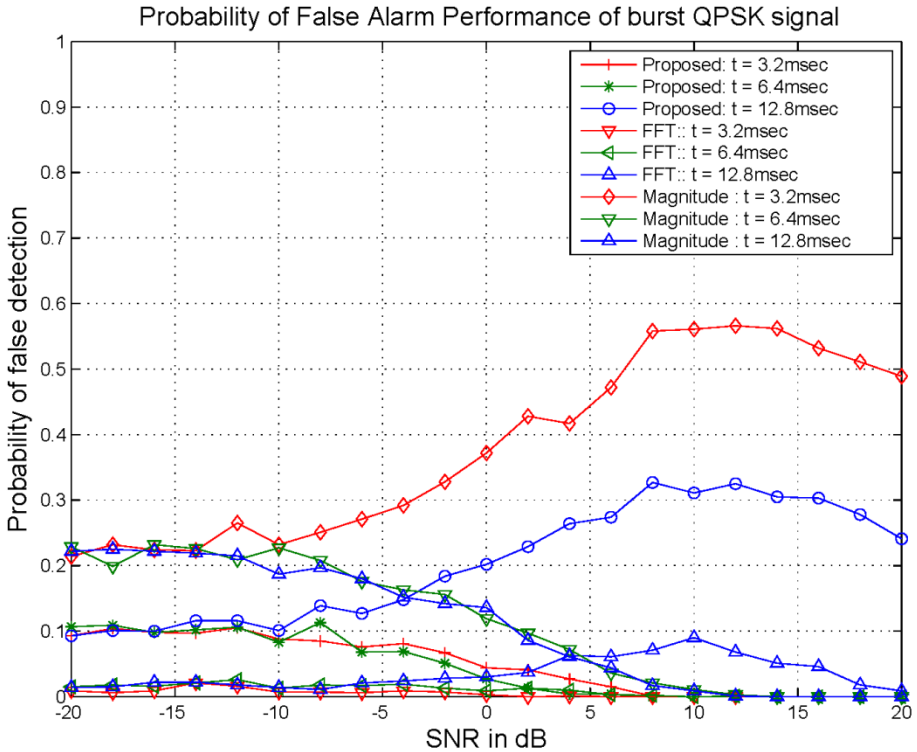


Figure 3.16: Probability of false alarm in AWGN for proposed , FFT and Magnitude approach.

The probability of false alarm (P_{fa}) performance is shown in Figure 3.16. It is seen that for the set threshold, the proposed sensing technique much lower P_{fa} than the magnitude based threshold set for detection. The proposed modification lowers the P_{fa} to 5% at 0dB SNR, but FFT performs better at lower SNR. It can be seen from Figure 3.16 that the P_{fa} is much larger for all the three schemes when the sensing duration is larger than the burst duration.

When two bursts at different channel frequencies are present, the Pd performance is identical for both the channels. However, as shown in Figure 3.17, the performance is better for the signal with higher digital IF channel frequency. This is because for the same set of parameters 'i' and 'm', the number of cycles of the digital IF in a set sensing duration 't' are more. In this current scenario, digital IF frequencies of 1MHz and 2 MHz, the number of cycles is double. Hence the results are shown by further reducing the sensing time for 2MHz digital IF frequency channel.

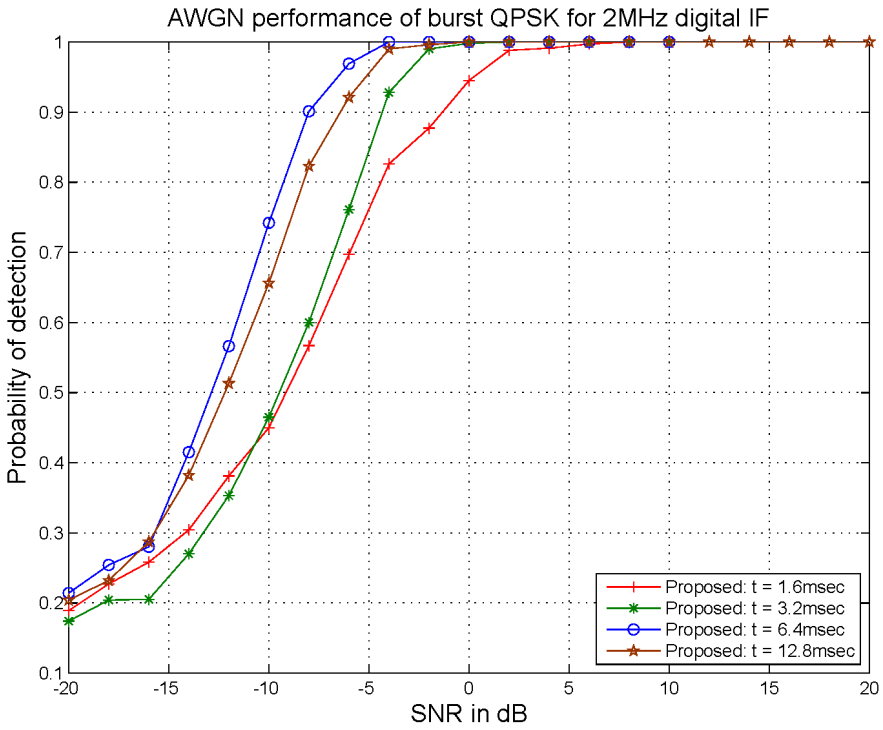


Figure 3.17: Probability of detection performance for an adjacent channel at 2MHz digital IF.

The Probability of detection (Pd) for the proposed technique with FFT approach in Rician and Rayleigh fading channel conditions, with 300Hz maximum Doppler shift are shown in Figure 3.18 and in Figure 3.19 respectively. The proposed sensing technique performs similar to the AWGN channel conditions (as shown in Figure 3.15) for Rician and Rayleigh fading channels given in Figure 3.18 and Figure 3.19. Even for the higher digital IF of 2MHz, there is no change in probability of detection (P_d) performance achieved under Rician and Rayleigh fading channel conditions (given in Figure 3.18 and Figure 3.19) when compared to probability of detection (P_d) achieved in AWGN channel conditions (as given in Figure 3.17). However, when the sensing duration is larger than the burst duration, P_d deterioration is larger than in AWGN.

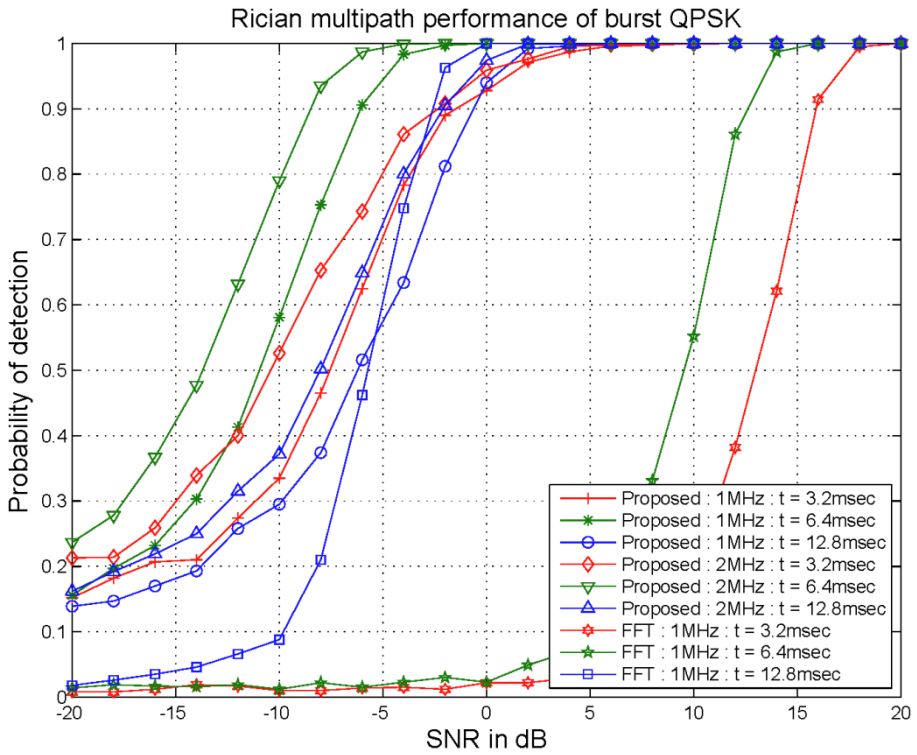


Figure 3.18: Probability of detection performance under Rician channel.

3.6 Hardware Architecture

A reconfigurable hardware architecture for implementing the proposed low complexity spectrum sensing technique is proposed in this section. The architecture for sensing multiple channels by implementing parallel realizations of sensing module for each

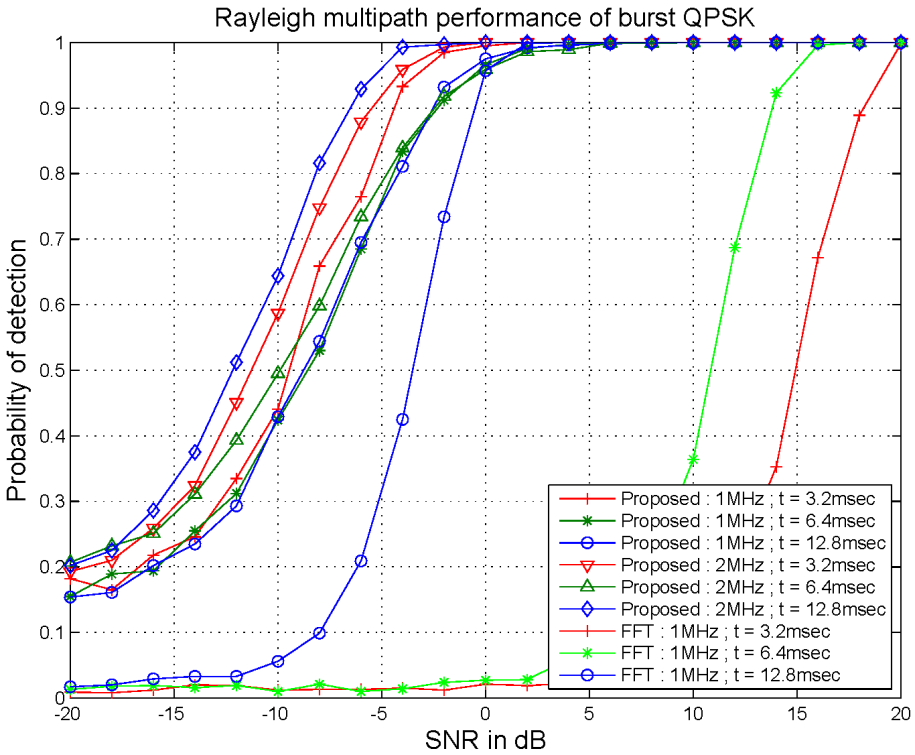


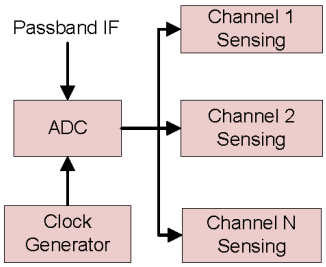
Figure 3.19: Probability of detection performance under Rayleigh channel.

channel is shown in Figure 3.20(a). In the proposed sensing technique, the most computationally intensive task is segmentation and decimation. It can be implemented using reconfigurable hardware circuitry as shown in Figure 3.20(b). The proposed architecture block diagram is intended to show how different sensing durations can be realized in a hardware implementation by changing the decimation counter and segmentation counter widths. The ISDV shuttle memory stores a number of samples depending on the Segmentation Counter and Decimation counter inputs.

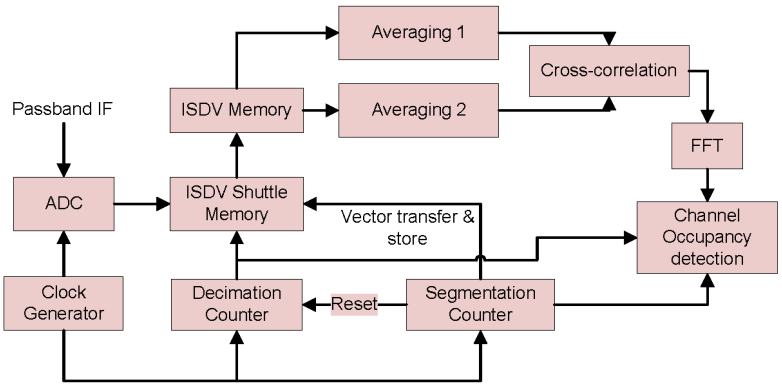
These count values determine the total number of samples and the length of the ISDV vector. The segmentation counter decides the total number of samples in each segment and the decimation counter selects the samples which form the ISDV vector and stored in the ISDV memory.

The alternate ISDV vectors (odd and even ISDV vectors) are averaged in separate modules. The averaged ISDV vectors are cross-correlated. The number of samples used for sensing is determined by programming the appropriate segmentation and decimation counter values, and the total number of ISDV vectors which are averaged,. This is directly related to the total sensing duration.

FFT is performed on the output of the cross-correlator, which is then used to determine the occupancy of the channel by determining if the peak output bin of the FFT output is corresponding to the ISDV length set in the segmentation counter and the number of samples in the decimation counter.



(a) High-level architecture for multiple channel sensing.



(b) Channel Sensing implementation architecture.

Figure 3.20: Hardware architecture for spectrum sensing.

Depending on the implementation scenario, sensing could be carried out sequentially one channel frequency at a time by dynamically programming the decimation and segmentation counters or by realizing independent channels with dedicated parallel hardware.

The proposed architecture for single channel has been implemented on a FMCOMMS2/3 [110] + Xilinx ZC706 FPGA evaluation board platform. The sampling frequency is set at 40MHz and the input sample bit-width for the implementation is 16-bits. The input digital IF signal frequency is 1MHz, the segmentation counter and decimation counter are each set to 40. A total of 80 ISDV vectors were averaged in 2 blocks of 40 each.

The implementation's resource utilization with full-parallel correlators, excluding FFT (because a Xilinx IP core optimized for power of 2 length was used), is 52 DSP48E1s, which is 5% of the device (Xilinx ZC706 FPGA) and the slice LUT utilization is ~33K, which is 15% of the device. The minimum clock period (MCP) for un-constrained implementation is reported at 19.683nsec.

3.7 Summary

In this chapter, a low complexity spectrum sensing technique which is based on cross-correlation of segmented and decimated sample vectors has been proposed. The proposed method requires nearly $1/i^{th}$ of the number of samples which are used for correlation after segmentation and decimation, for a vector length of 'i', of the simplest known autocorrelation-based detection scheme described in [32]. MATLAB simulations have been carried out to validate the proposed method. In the simulations, signals such as, single-tone, multi-tone, ATSC digital TV, PAL-B analog TV and burst QPSK modulated signals have been used for quantifying the detection performance. It has been shown that

the performance of detection improves with the number of ISDV vectors 'm' used for averaging and cross-correlation, except for in burst signal cases, where an increase in 'm' makes the sensing duration more than the burst duration. A simple reconfigurable hardware architecture for implementing the low complexity spectrum sensing technique has been presented.

With the availability of the frequency spectrum determined through the proposed technique, information is transmitted over the frequency band in bursts. One of the reasons burst transmissions are being used is to have the flexibility to cut short the transmission and switch to another transmission frequency when the current one becomes unavailable. For proper data demodulation and information recovery of such burst transmissions, it is imperative to time synchronize with the incoming RF burst at the receiver. In the following chapter a mechanism for efficient time synchronization of incoming burst signals is presented.

4 Efficient Cross-Correlation Algorithm and Architecture for Robust Synchronization

4.1 Overview

Wireless communication in high-mobility environments is usually frame-based burst communication. Short burst communications are also suitable for CR applications where the available frequency spectrum is shared and opportunistically used by secondary users. For correct detection and demodulation of data, the receiver must synchronize, in both time and frequency, with the received burst signals. Generally, preamble sequences are used for time and frequency synchronization. Recently, Zadoff-Chu sequences, which have very good correlation properties, have gained popularity for this purpose. In this chapter, an efficient cross-correlation based algorithm and its implementation architecture is proposed for robust synchronization in systems with multiple repetitive preambles. The proposed method is well suited for systems experiencing carrier frequency offsets and operating in high-mobility environments. Synchronization is achieved by cross-correlation of locally stored preamble replica with a preamble derived from the received signal samples. The derived preamble is generated by segmenting and

decimating/rearranging the signal samples of the received preamble. The hardware implementation architecture for the synchronization technique is also presented. The proposed architecture has been synthesized and implemented on a Xilinx FPGA platform for validation and performance evaluation. The system level simulation under Additive white Gaussian noise (AWGN), in Long-Term Evolution (LTE) channels with frequency offsets and Doppler shifts has also been carried out to find the efficacy of the proposed approach. The inherent parallelism of the proposed synchronization scheme results in fast and energy efficient hardware implementations. One such implementation of the correlator structure in Xilinx FPGA is presented.

With the system model presented in Section 4.2, the proposed technique is presented in Section 4.3. The advantages of the proposed technique are discussed in Section 4.4. In Section 4.5, the performance results of the proposed synchronization technique generated using MATLAB based computer simulations are presented. The performance of proposed method under different Doppler and carrier frequency offsets are also evaluated. In Section 4.6, the details of the proposed segmentation, decimation and cross-correlation technique along with the proposed architecture are discussed. The FPGA implementation of proposed design is presented and the estimated hardware and time complexity are evaluated. Section 4.7 concludes the chapter.

4.2 System Model

In OFDM systems and in other frame-based communication systems, one of the prime requirements for correct demodulation of the transmit OFDM symbol or burst at the receiver is to time synchronize the receiver to the transmitter. In almost all systems, timing synchronization between transmitter and receiver is achieved at the beginning of each OFDM symbol or burst through the use of preamble sequences. Preamble sequences

are predefined patterns which are known apriori at the receiver. The receiver usually searches for the known preamble sequences to determine the start of the OFDM symbol or the burst frame. Time synchronization is of much higher significance in OFDM systems, where an error in estimation of the start time of the OFDM symbol or late synchronization, will result in incorrect decoding of the received symbol due to breaking of orthogonality of the sub-carriers in the OFDM symbol resulting in ICI and incorrect channel estimation [71]. However, early synchronization does not have any major effects as long as it is time synchronized within the cyclic-prefix period. Generally it is possible to get-away with early synchronization by using longer cyclic prefix or a longer guard interval [75, 99]. However this leads to reduced air time for transmission of useful data which results in lowering of overall data throughput.

Preamble sequences are time-domain samples transmitted at the beginning of each transmission burst. Each sequence is repeated multiple times depending on the preamble design. Figure 4.1 illustrates a transmit frame structure.

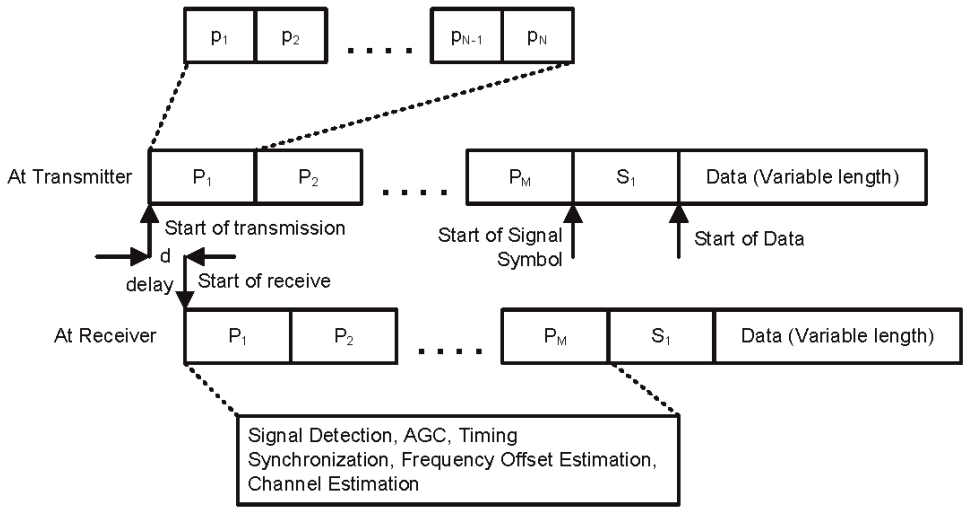


Figure 4.1: Illustration of start of burst and preambles in a frame.

A single frame burst consists of 'M' multiple preambles $\boldsymbol{P}_i$ followed by the Signal Symbol $\boldsymbol{S}_i$ giving details about the following data burst and the receiver makes use of

these preambles for frame detection, AGC, frame timing synchronization, frequency offset estimation and in some cases an initial estimate of the channel. Let p(i) be the i^{th} sample of a 'N' sample preamble sequence which is repeated 'M' times at the beginning of each transmission burst. This sequence is then appended with other data symbols to form a transmit signal. At the receiver, the transmitted signal's n^{th} baseband sample, x$[n]$ is received after a delay of 'd' samples. The received signal's n^{th} baseband sample can be expressed as [99]

$$\mathrm{y}[n] = e^{j2\pi n \mathbf{f}_{CFO}} \sum_{l=0}^{L-1} \mathrm{h}[l]\, \mathrm{x}[n-d-l] + \mathrm{u}[n] \qquad (4.1)$$

where h$[l]$ represents the l^{th} path of a L path wireless channel, u$[n]$ is the AWGN noise component for the n^{th} sample and f_{CFO} is the normalized carrier frequency offset at the receiver. When a single preamble sequence is used for estimating the timing offset 'd', the received signal is cross-correlated with the stored replica of the transmitted preamble sequence

$$r[k] = \sum_{n=0}^{N-1} \mathrm{y}[n+k]\mathrm{p}^{*}[n] \qquad (4.2)$$

where k is the lag of the correlation and * is the complex conjugate operator. The estimated timing offset d_{est} can be determined as

$$d_{est} = arg \max_{k} |r[k]| \qquad (4.3)$$

where $|.|$ denotes the absolute value. The delay due to multipath 'l' can be ignored if it is less than the guard interval between symbols.

4.3 Proposed Timing Synchronization

In the preamble design, when repetitions of a preamble sequence are used, there are multiple d_{est} for each repetition of the preamble. This complicates timing offset estimation as maximum of $|r[k]|$ varies due to variations in the wireless channel as well as any carrier frequency offset. Here, segmentation and decimation of the received signal before it is cross-correlated with the local copy of the preamble sequence is proposed.

4.3.1 Segmentation

Considering a simplified case, let us assume the preamble sequence consists of an even number of N samples and has M number of repetitions. The received signal $\mathrm{y}[n]$ is divided into multiple numbers of segments S and $S = M$ the number of preamble sequence repetitions. However, it can also be shown that it is not necessary for S to be equal to M. The segmented received signal is represented as

$$\mathbf{y}_s = \{\mathrm{y}[sN], \mathrm{y}[sN+1], \ldots, \mathrm{y}[sN+N-1]\} \qquad (4.4)$$

where $s = 0, 1, \ldots, S-1$.

Equation (4.4) gives a representation of the samples in a segment '$\mathbf{y}_s$'. The index s is the segment index which is used to differentiate multiple segments of $\mathbf{y}_s$, such as $\mathbf{y}_0$, $\mathbf{y}_1$, $\mathbf{y}_2$, …, $\mathbf{y}_{S-1}$. Each of these segments $\mathbf{y}_s$ is a vector of N samples. The sample index is represented as, sN, $sN+1$, $sN+2$, $sN+3$, …, $sN+N-1$.

4.3.2 Decimation

Now, to arrive at a single received preamble sequence $\mathbf{y}_p$ from the received signal $\mathrm{y}[n]$, the received signal segments $\mathbf{y}_s$ are decimated with decimation factor of M. The decimation is carried out such that the first decimated sample for segment s is dependent

on the segment index s, $\mathrm{y}[s(N+1)]$. The decimated samples from all the received signal segments are rearranged to obtain $\mathbf{y}_p$, which is then used for cross-correlation with the stored local sequence. The decimated segment can be expressed as

$$\mathrm{y}_p[nM+m] = \mathrm{y}[nM+m(N+1)] \qquad (4.5)$$

where $0 \le n \le \left(\frac{N}{M}-1\right)$ and $0 \le m \le (M-1)$.

This $\mathbf{y}_p$ contains only one preamble sequence and the rest of the preamble repetition samples are rearranged.

An illustration of segmentation, decimation and derivation of the received preamble sequence is shown in the Table 4.1 below for $N = 6$ and $M, S = 3$.

Table 4.1: Illustration of Segmentation, Decimation and Derivation of Preamble Sequence

Received Samples	$\mathbf{y} = \{\mathrm{y}[0], \mathrm{y}[1], \mathrm{y}[2], \dots, \mathrm{y}[16], \mathrm{y}[17], \dots\}$
Segments of received samples for $N = 6$ and $M, S = 3$	$\mathbf{y}_0 = \{\mathrm{y}[0], \mathrm{y}[1], \mathrm{y}[2], \mathrm{y}[3], \mathrm{y}[4], \mathrm{y}[5]\}$ $\mathbf{y}_1 = \{\mathrm{y}[6], \mathrm{y}[7], \mathrm{y}[8], \mathrm{y}[9], \mathrm{y}[10], \mathrm{y}[11]\}$ $\mathbf{y}_2 = \{\mathrm{y}[12], \mathrm{y}[13], \mathrm{y}[14], \mathrm{y}[15], \mathrm{y}[16], \mathrm{y}[17]\}$
Decimated samples of segments	$\mathbf{y}_0 = \{\mathrm{y}[0], \mathrm{y}[3]\}$ $\mathbf{y}_1 = \{\mathrm{y}[7], \mathrm{y}[10]\}$ $\mathbf{y}_2 = \{\mathrm{y}[14], \mathrm{y}[17]\}$
Derived preamble sequence by rearranging the decimated segment samples	$\mathbf{y}_p = \{\mathrm{y}[0], \mathrm{y}[7], \mathrm{y}[14], \mathrm{y}[3], \mathrm{y}[10], \mathrm{y}[17]\}$

In Table 4.1 the received samples $\mathbf{y}$ is a continuous stream of preamble samples. Let us assume the preamble length is 6 ($N = 6$) and has 3 (M = 3) repetitions. The continuous stream of preamable samples, $\mathrm{y}[0], \mathrm{y}[1], \mathrm{y}[2], \dots$ are segmented into vectors of 6 samples

each. Each segmented vector $\mathbf{y}_s$ is made up of N number of samples as represented in (4.4) and there are $S = M$ number of segments. For $N = 6$ and S = 3, there are 3 segmented vectors, $\mathbf{y}_0$, $\mathbf{y}_1$, and $\mathbf{y}_2$ each with 6 samples, representing one preamble sequence each. The samples in each segmented vector are derived as in (4.4) and are shown in row 2 of Table 4.1. Further each of these segmented vectors are decimated and rearranged into a single vector $\mathbf{y}_p$ as in (4.5) with $M = 3$. The decimated vectors, each of 2 samples are shown in row 3 of Table 4.1.This resulting rearranged vector, which represents a single preamble sequence is given in row 4 of Table 4.1.

4.3.3 Timing Synchronization

The time offset between the received preamble sequence and the transmitted preamble sequence is determined by cross-correlation of the derived preamble sequence given in (4.5) with the locally stored replica of the transmitted preamble sequence. The cross-correlator output $r[k]$ is given by

$$r[k] = \sum_{n=0}^{N-1} \mathrm{y}_p[n+k]\mathrm{p}^*[n] \qquad (4.6)$$

With the derived preamble sequence represented by (4.5), (4.6) can be re-written as

$$r[k] = \sum_{m=0}^{M-1} \sum_{n=0}^{\frac{N}{M}-1} \mathrm{y}[nM + m(N+1) + k]\mathrm{p}^*[nM + m] \qquad (4.7)$$

The above described proposed segmentation and decimation has been pictorially illustrated in Figure 4.2.

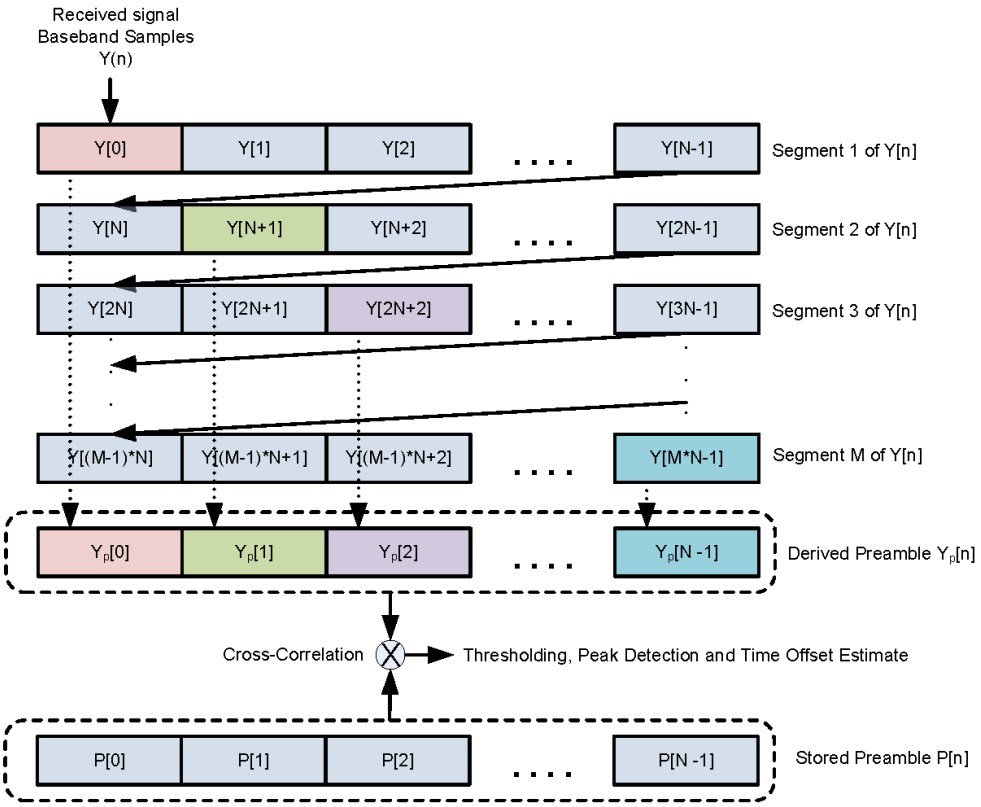


Figure 4.2: Illustration of proposed segmentation, decimation and timing synchronization.

Taking into account the different multipath signals and ignoring the AWGN noise effect on the cross-correlation magnitude, $r[k]$ can be written as

$$r[k] = \sum_{m=0}^{M-1} \sum_{n=0}^{\frac{N}{M}-1} \left[e^{j2\pi[nM+m(N+1)+k]\mathbf{f}_{CFO}} \left(\sum_{l=0}^{L-1} \mathrm{h}[l]\, \mathrm{x}[nM + m(N+1) + k - (d+l)]\mathrm{p}^*[nM+m] \right) \right] \quad (4.8)$$

The timing offset estimate is still determined by (4.3). However, since the derived preamble sequence (4.5) is from samples across multiple preamble duration, the timing offset estimate is determined by the highest $\mathrm{h}[l]$ across the whole repetitive preamble duration of (MN) samples.

In (4.8), for a flat fading channel, where $\sum_{l=0}^{L-1} h[l] = h[0]$, $k = d$ and since the transmitted preamble is a repetitive preamble, (4.8) is re-written as

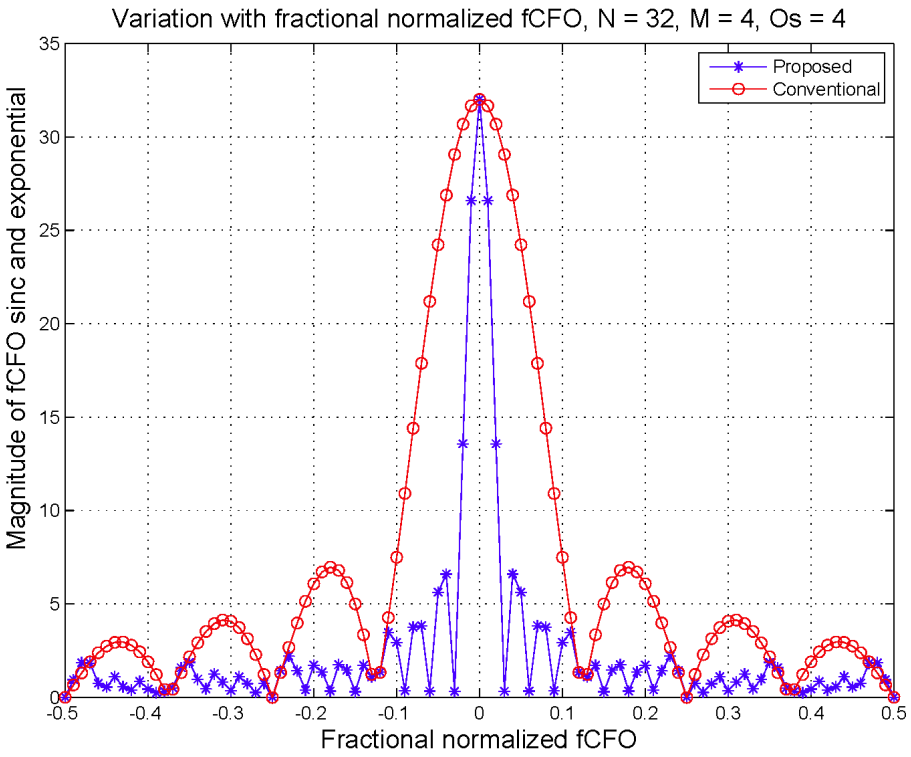
$$r[k] = \sum_{m=0}^{M-1} \sum_{n=0}^{\frac{N}{M}-1} \left[e^{j2\pi[nM+m(N+1)+k]\mathbf{f}_{CFO}} \left(\sum_{i=0}^{N-1} h[0]\, x[i] p^*[i] \right) \right] \qquad (4.9)$$

In (4.9), the term in parenthesis will be a constant, 'C' and the variation in timing estimate due to the presence of carrier frequency offset is largely determined by the exponential term. The exponential summation in (4.9) can be simplified as in [101] to a form

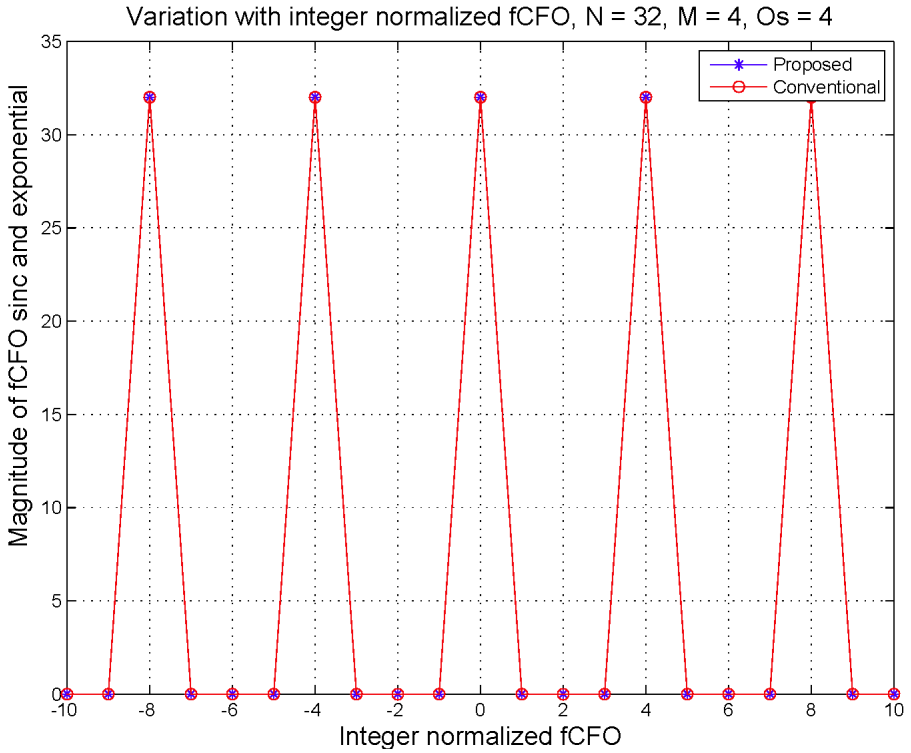
$$r[k] = \left(e^{jz\left(k+(NM-1)\right)} \right) \left(\frac{\sin\left(\frac{1}{2} zM(N+1)\right) \sin\left(\frac{1}{2} zN\right)}{\sin\left(\frac{1}{2} z(N+1)\right) \sin\left(\frac{1}{2} zM\right)} \right) (C) \qquad (4.10)$$

where $z = 2\pi \mathbf{f}_{CFO}$

The normalized carrier frequency offset can either be an integer or a fraction. The magnitude variation of the sinc and exponential components in (4.10) will affect the overall magnitude of the cross-correlation $r[k]$ and determine the error in timing offset estimate due to carrier frequency offset. The proposed algorithm reduces this error. Figure 4.3 compares the magnitude of these terms with respect to fractional and integer normalized carrier frequency offsets. It is seen in Figure 4.3(a) that the magnitude variation of the sine and exponential function in (4.10) is small as compared to conventional approach for fractional normalized $\mathbf{f}_{CFO}$. Magnitude variation with integer normalized $\mathbf{f}_{CFO}$ as shown in Figure 4.3(b) is same. This robustness against fractional normalized carrier frequency offsets is further verified through simulation in Section 4.



(a) Fractional normalized carrier frequency offset.



(b) Integer normalized carrier frequency offset.

Figure 4.3: Magnitude variation of cross-correlation due to fractional and integer carrier frequency offset.

4.3.4 Multiple Peaks in Cross-Correlation

The excellent correlation properties of certain preamble sequences may lead to distinct multiple peaks at the output of the cross-correlator, $|r[k]|$. Hence, it is proposed to determine the timing synchronization in conjunction with the multiple peaks. Once a peak is detected, another peak of lower magnitude (MN) samples ahead and past is looked for. When a peak is determined at either of those locations, the frame timing is determined by the peak which occurs first in time sequence. The occurrence of the peak based on the mean value of $|r[k]|$ over a certain number of samples proportional to (MN) is determined.

4.4 Advantages of Proposed Technique

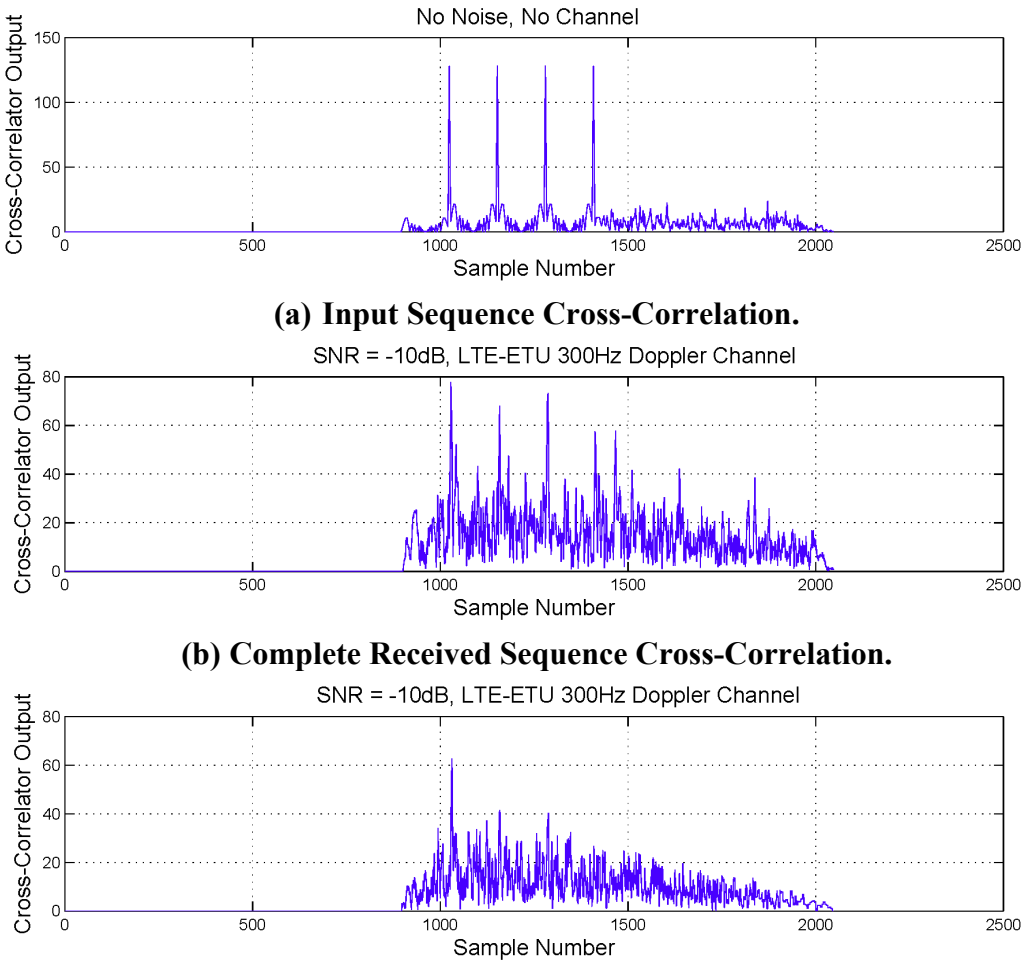
When multiple preamble sequences are used for timing synchronization, either autocorrelation [50] or cross-correlation is performed. A search for a plateau at the autocorrelation output gives the timing metric for synchronization. Though it is robust against carrier frequency offset and fading, it presents high uncertainty on start timing of burst/symbol, mainly due to the plateau nature of the timing metric. In cross-correlation based approach, timing synchronization functionality in a receiver is executed on every digitized sample at the receiver. This results in multiple peaks at the cross-correlation output due to repetition of the preamble sequences, when cross-correlated over a single preamble duration. Alternatively, a single pronounced peak and multiple peaks of smaller amplitude are obtained when cross-correlation is performed with the total duration of the repeated preamble. Cross-correlator output when performed over the received sample sequence and the proposed segmented and derived preamble sequence is shown in Figure 4.4 for one simulation run. Figure 4.4(a) shows the output of cross-correlation of the input

preamble sequence with the local copy of the preamble sequence. It is assumed that there is no additive noise and no multipath channel effects are added. Four distinctive peaks corresponding to four repetitions of the preamble sequence can be seen. Figure 4.4(b) gives the output of the cross-correlation with additive noise being set to give a SNR of -10dB in a LTE-ETU channel with 300Hz Doppler. It can be seen that the distinctive peaks which were present in the no channel case (as in Figure 4.4(a)) are now obscured with multiple false peaks. When the distance-based algorithms suggested in [52, 53, 77] encounter these multiple false peaks, it could lead to incorrect timing estimate. As shown in Figure 4.4(c), for the same SNR and channel conditions, the proposed algorithm produces a pronounced single peak which reduces the timing uncertainty in synchronization. However as mentioned earlier, apart from the single peak, in certain conditions there could be subsequent peaks in the proposed segmented scheme as well, which are again used to reduce the timing uncertainty in synchronization.

The received power level of the signal burst/OFDM symbol is initially unknown at the receiver. Therefore, the first few preambles sequences are used for determining the received signal level and tuning the analog RF front end gain of the receiver for an optimum signal input level. Since the segmented cross-correlation is performed on the derived preamble sequence from the received multiple preamble sequences, the signal level fluctuation during the gain tuning process does not affect the cross-correlation output substantially. The multiple peaks, when they exist, are used to further aid in performing the timing synchronization.

The segmented nature for the proposed algorithm enables hardware efficient implementation of the cross-correlation. The repetitive modular nature of the algorithm helps to derive a hardware-efficient design and supports increased frequency of operation of the synchronization module. This is further elaborated in the hardware implementation

section. Due to the inherently parallel nature of the algorithm, it also allows it to be implemented in a highly parallel architecture for high-throughput implementation.



(a) Input Sequence Cross-Correlation.

(b) Complete Received Sequence Cross-Correlation.

(c) Segmented Sequence Cross-Correlation

Figure 4.4: Comparison of output of cross-correlation under LTE-ETU channel with 300Hz Doppler at -10dB SNR.

4.5 Simulation Results

The proposed timing synchronization technique was simulated in MATLAB for two different preamble structures. In the first structure a ZC sequence of length 32 was used as preamble. The preamble was repeated four times at a sample rate of 1.28MSPS. An oversampling factor of four was used at the receiver. The synchronization performance is

compared with the cross-correlation method in which the whole preamble duration is cross-correlated. The other sequence is a 128 symbol ZC sequence which is repeated twice and oversampled at 8x rate. The second sequence has been used to compare the performance of the proposed scheme with distance-based synchronization methods of [52, 53, 77] with multiple peaks, which determine the timing instant based on the sample distance between correlation peaks. The evaluation of the timing synchronization performance is based on the Synchronization Ratio, which gives the number of bursts in which the synchronization was performed within two preamble symbols of the correct time instant. The simulation model is required to determine the first sample after the last sample of the preamble sequence.

The synchronization is considered to be successful when the determined time sample is any one of the oversampled samples within two preamble symbols of the preamble sequence. In carrying out the simulation of the proposed synchronization method, different channel models, such as AWGN, 6-path Rician with gain and delay vector of [0 -1.0 -9.0 -10.0 -15.0 -20.0] and 1.0e-9 * [0 310 710 1090 1730 2530] respectively and LTE channel models stated in [102] are used. The different LTE channel models are Extended Pedestrian A (EPA) with 7-paths and maximum excess delay of 410nsec, Extended Vehicular A (EVA) with 9-paths and maximum excess delay of 2510nsec and Extended Typical Urban (ETU) with 9-paths and maximum excess delay of 5000nsec, together these channels cover a wide range of operating environments from pedestrian to Urban mobile.

Figure 4.5 shows the performance of the proposed method under AWGN, 6-path Rician and LTE channel models. The 32-sample ZC sequence was used as the preamble with number of repetitions and oversampling factor, $O_s = 4$. It shows more than 99% synchronization rate performance for SNR over 0dB. Synchronization rate is deteriorated

for SNR below -5dB, especially in LTE-EVA, ETU and 6-path Rician channels as compared to AWGN and LTE-EPA which are relatively friendly channels as the former ones present the worst multipath (9-paths) delay profile, with a maximum path delay of 5μsec. When compared to the performance with cross-correlation over entire preamble sequence length, including repetitions, the performance is better for SNRs above -12dB.

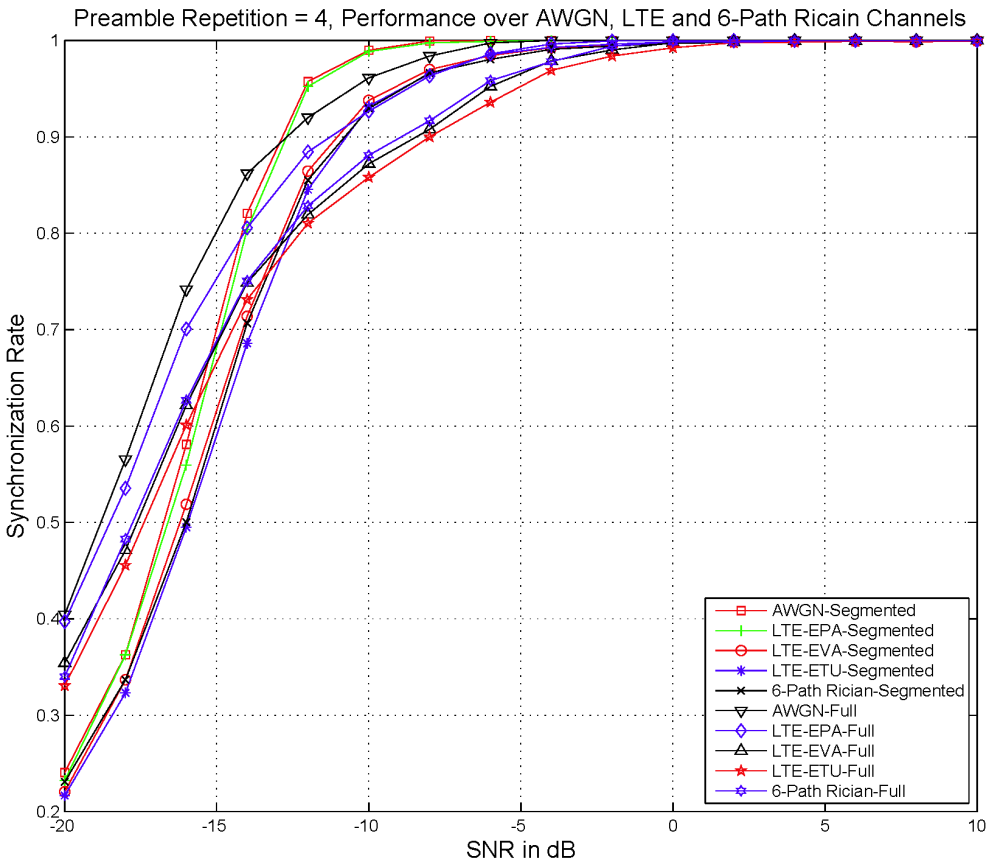
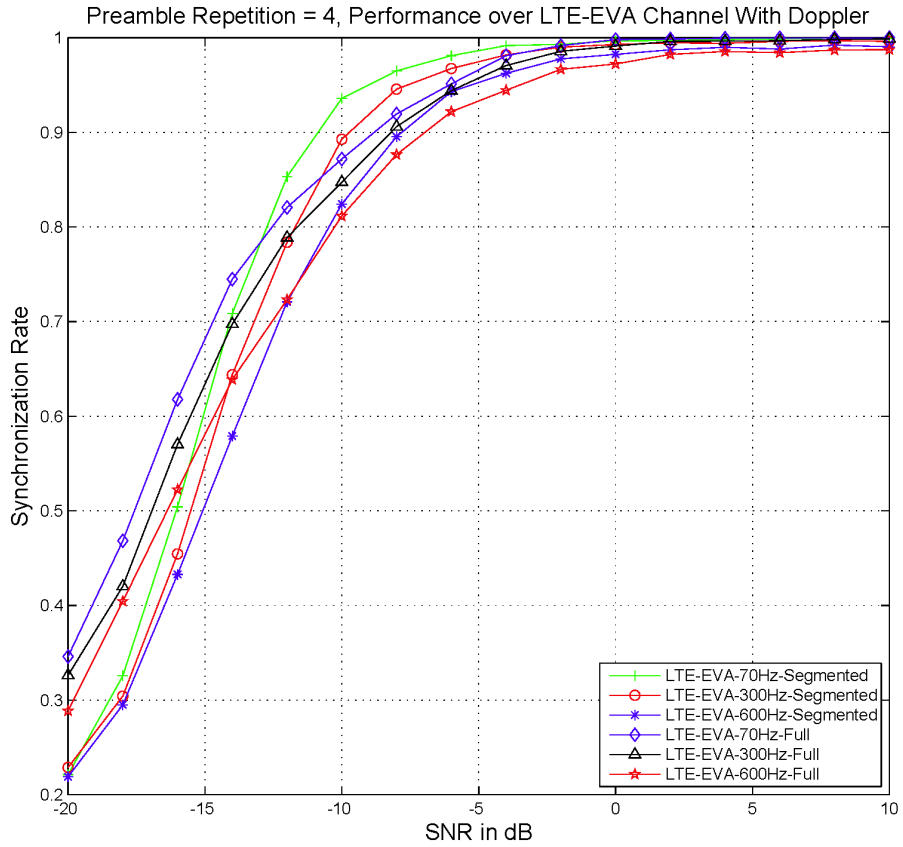
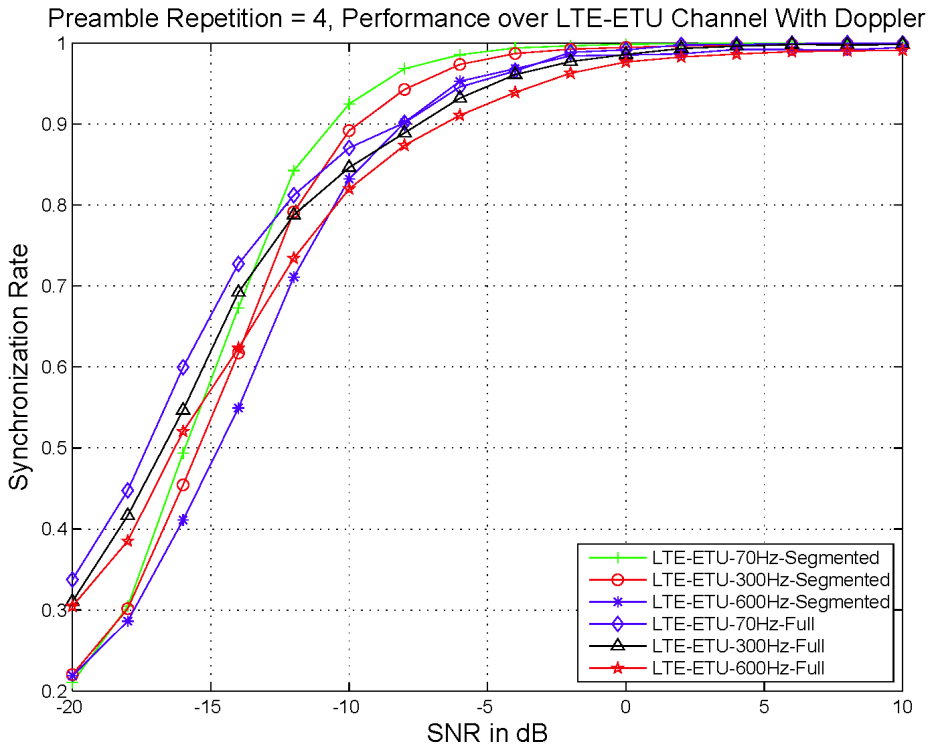


Figure 4.5: Performance over AWGN, 6-Path Rician and LTE channel models.

To these channels Doppler shift is added and the simulation performance results are shown in Figure 4.6. Though LTE channel models specify Doppler frequency of up to 300Hz, a 600Hz Doppler frequency is introduced, which amounts to moving platforms with speeds of ~600Km/Hr for operating carrier frequency of 500MHz or ~135Km/Hr operating at 2.4GHz carrier frequency. For LTE-EVA channels the synchronization rate performance, as shown in Figure 4.6(a), is better than the full cross-correlation



(a) LTE-EVA channel with different Doppler.



(b) LTE-ETU channel with different Doppler.

Figure 4.6: Performance over LTE channel models.

performance for SNR above -12dB. The LTE-EVA channel model has a strong direct path (with 0dB attenuation) and 8 other multipath paths with varying delays. At low SNRs, the direct path present in the LTE-EVA channel model results in better performance of the full-cross-correlation scheme. However at higher SNRs, the presence of 8 multipath paths in the LTE-EVA channel and segmentation and decimation of the received samples in the proposed scheme results in better performance as compared to full cross-correlation approach.

The synchronization rate over LTE-ETU channels (Figure 4.6(b)) is higher for SNR above -12dB SNR. Performance is least affected in high SNR scenarios even after the introduction of Doppler shift along with the multipath propagation channels. The performance for 600Hz Doppler is found to be slightly deteriorated, but still better than the full cross-correlation performance.

For a system operating in UHF frequency range (500MHz, which is also part of TV White Space frequency range), with a reference oscillator crystal accuracy of ±10ppm, the receiver is likely to experience a maximum frequency offset of 10KHz. It has been simulated to determine the maximum timing estimation error and synchronization rate performance of the proposed synchronization method for frequency offsets up to 10KHz under AWGN conditions and the results are shown in Figures 4.7 and 4.8, respectively.

It can be seen in Figure 4.7 that there is no error for SNR above -4dB and the maximum error is 16 samples for SNR less than -4dB. For a 4x oversampled signal this 16-sample error is within 4 preamble samples and it is much less than the mean timing estimation error reported in [53]. Figure 4.8 shows that different carrier frequency offsets up to 10KHz have little or no impact on the performance of the proposed synchronization.

The proposed scheme's performance and its comparison with the symmetric cross-correlation [77] and Distance-based algorithm [52, 53] is shown in Figure 4.9. The

comparison is carried out with four preamble repetitions, as stated in [77], and with preamble length 'N' of 32 and 64-samples. The oversampling ratio (O_s) of 4 and 8 has been used. Under the most demanding channel environment, LTE-ETU, the performance of all the three schemes improves with increase in N and O_s, however, the proposed scheme performs better over the other two in similar conditions.

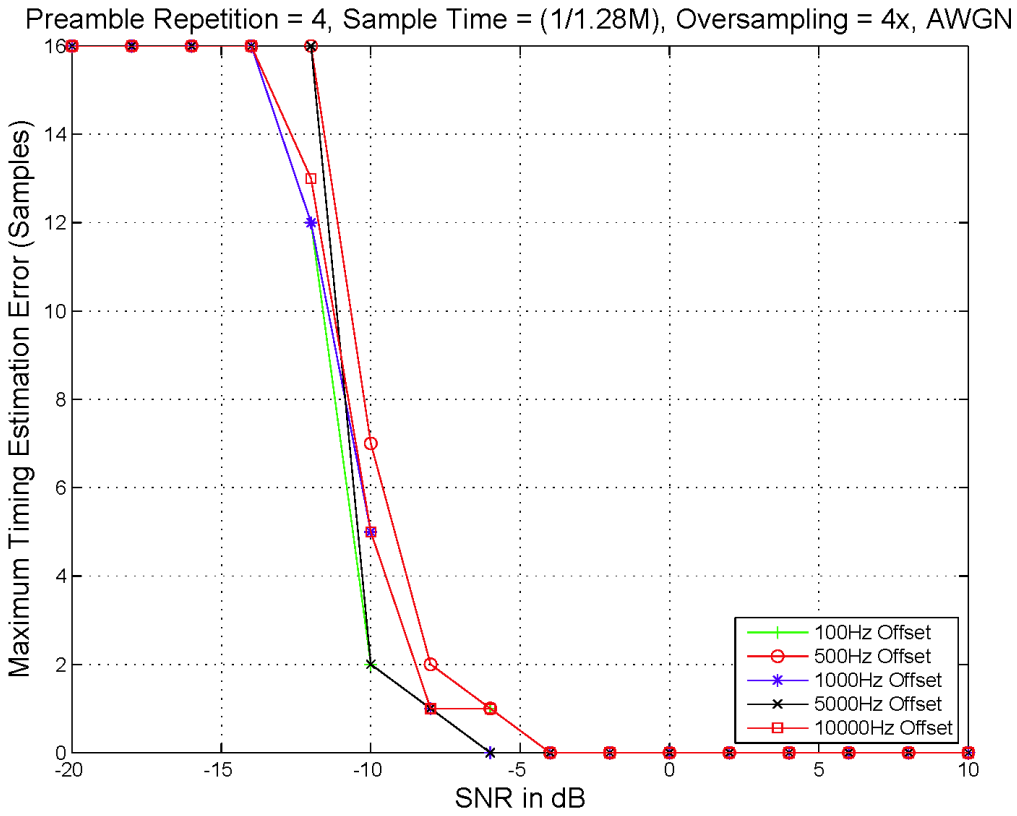


Figure 4.7: Maximum timing estimation error under different frequency offsets.

Figure 4.10 and Figure 4.11 show the comparison of performance of distance-based algorithm [52, 53] with the proposed algorithm. In Figure 4.10, a preamble with a 128-sample ZC sequence (N), 2 preamble repetitions (M) and oversampling ratio (O_s) of 8 is used. The simulations were carried for different channel models. It can be seen from Figure 4.10 that the performance of the proposed synchronization method is better than the peak sample distance-based algorithm with ZC sequence as preambles. Further, Figure 4.11 compares the performance for different N and O_s for LTE-ETU channel

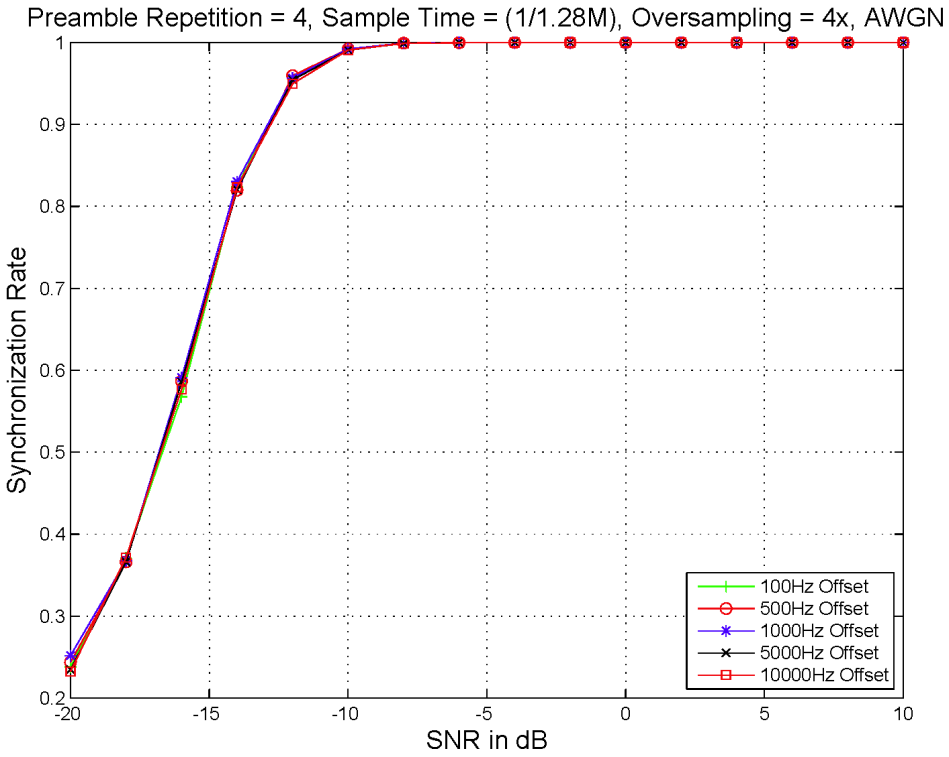


Figure 4.8: Performance under AWGN with frequency offset.

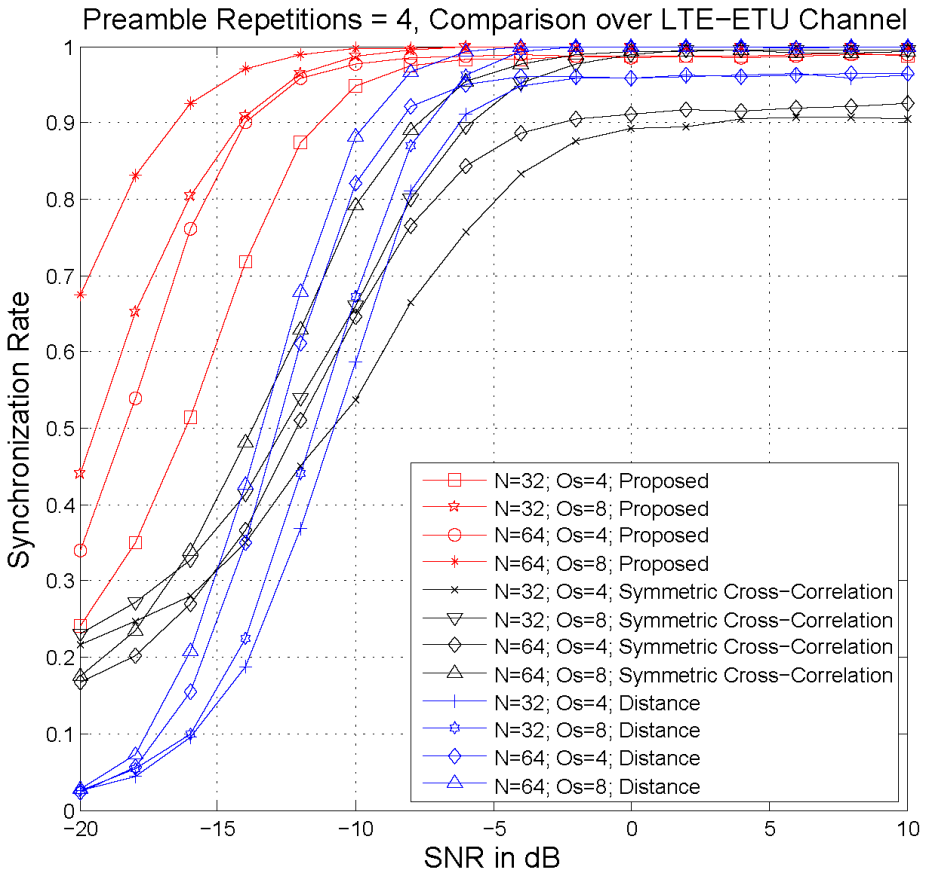


Figure 4.9: Performance comparison with symmetric cross-correlation and distance based timing synchronization.

environment. The improved performance of the proposed scheme is seen, this is especially so at lower SNR. Though [52, 53] introduce a correction factor based on CFO, in the current simulation CFO is set to Zero.

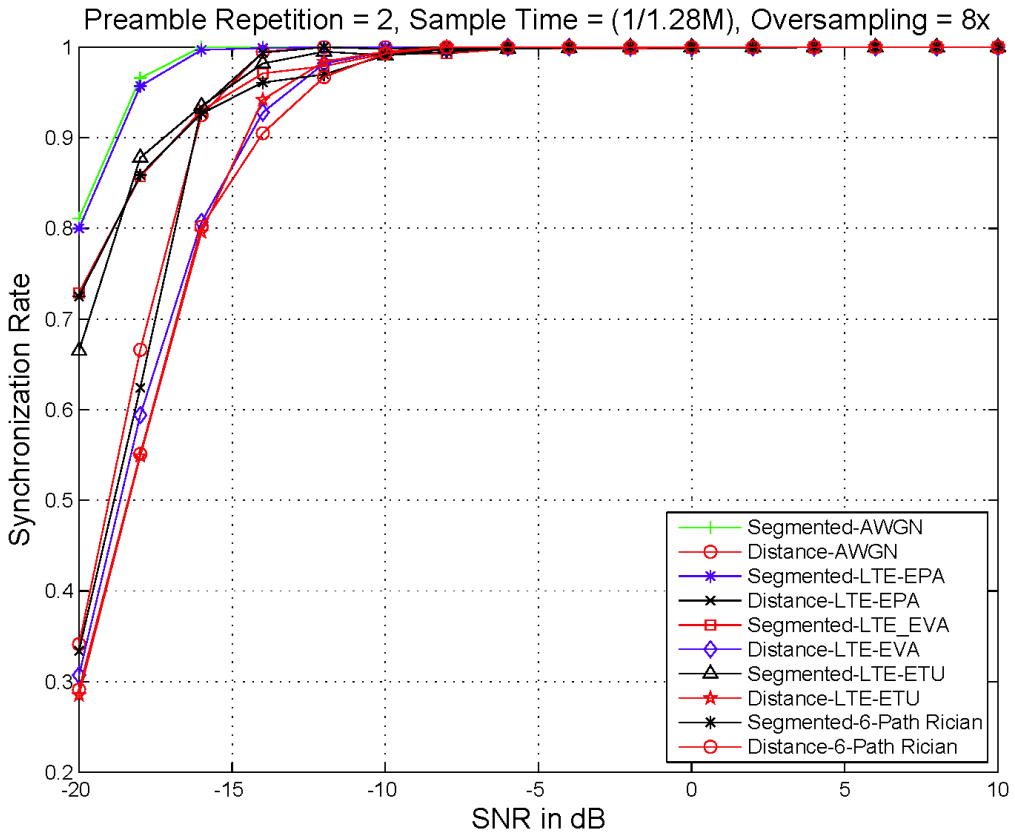


Figure 4.10: Performance comparison with distance based timing synchronization for all channels.

The performance of the proposed synchronization under different preamble repetitions M and oversampling ratio O_s is given in Figure 4.12 and Figure 4.13 for LTE-EVA and LTE-ETU channel environments. A 32-sample preamble at an oversampling ratio of 4 has been used for the results given in Figure 4.12. It can be seen that there is minimal change in performance for different preamble repetitions, this is mainly because, a single preamble from multiple repetitions for cross-correlation is derived. However, in Figure 4.13 the improvement in performance with increase in oversampling ratio (O_s) for a fixed 32-sample preamble with two repetitions can be clearly seen.

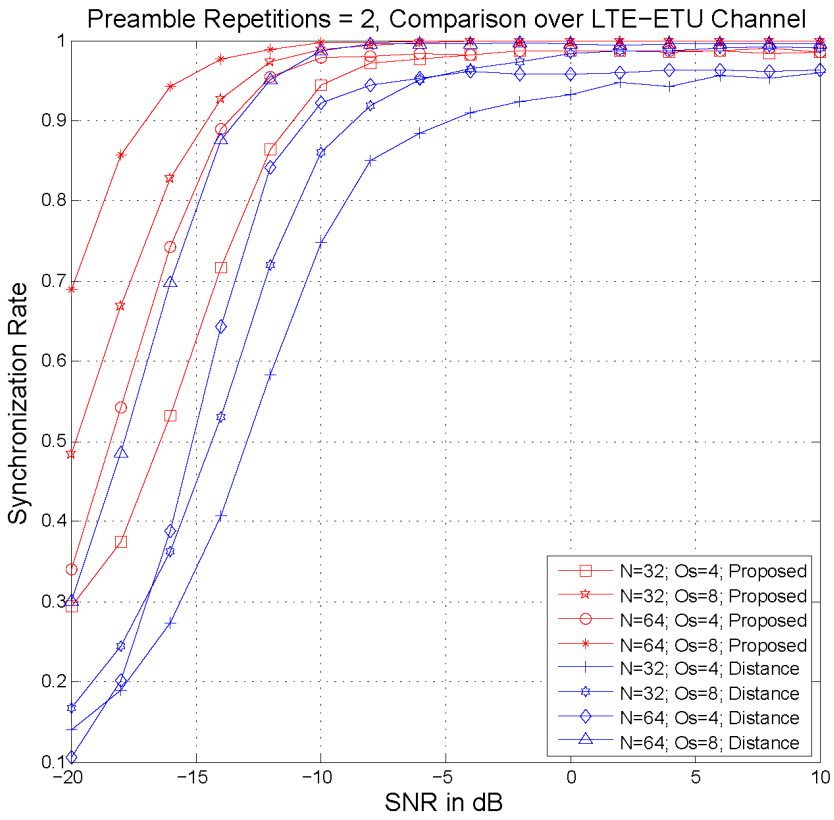


Figure 4.11: Performance comparison with distance based timing synchronization for different N and O_s.

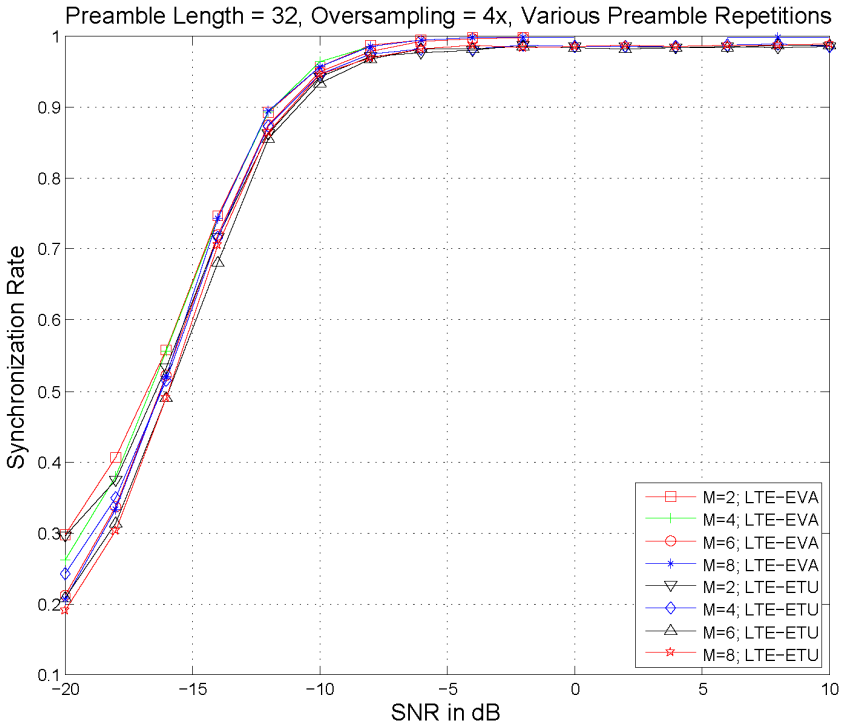


Figure 4.12: Proposed synchronization's performance for different preamble repetitions.

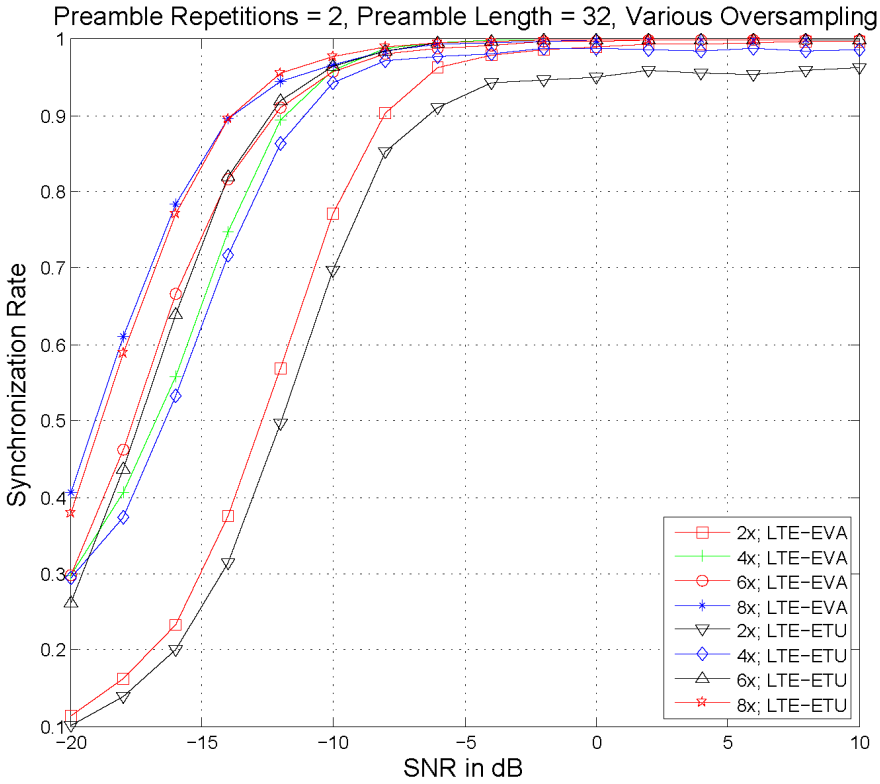


Figure 4.13: Proposed synchronization's performance for different oversampling ratio.

4.6 Proposed Implementation Architecture

4.6.1 Computational Steps and Complexity Estimation

The sequence of computational steps in the proposed timing synchronization scheme are given below in order to estimate the computational complexity.

Step 1: Segmentation of the received samples.

Based on the number of segments S and length of each segment (NO_s) of the received samples, it has been decided to use a circular register of size (SNO_s), where the received samples are shifted sequentially in.

Step 2: Decimation of the segmented and stored samples.

The decimation of the received samples is achieved by simply computing the sample indices of the decimated and derived preamble sequence elements as detailed in Section 4.2.

Step 3: Cross-correlation between the derived preamble sequence and the stored preamble sequence.

Cross-correlation between the stored preamble sequence and the derived preamble sequence are the most computationally intensive tasks in the proposed technique. Cross-correlation can be computed by directly evaluating the cross-correlation sum or efficiently computing via Fast Fourier Transforms (FFT) [103, 104] with suitable zero-padding for avoiding aliasing, sometimes referred to as Fast Correlation. However, the timing synchronization metric as in (4.3) is dependent on integer k and $k \ll N$, the number of samples of the preamble sequence. This needs that the fast correlation is evaluated for a number of different lags and the number of lags can be equal to N. In this case, it is efficient to evaluate the cross-correlation by simply evaluating the sum (4.7). The sum indicated in (4.7) need to be evaluated on every sample. Table 4.2 shows the number of multiplications required in the computation of the cross-correlation for synchronization as compared to other schemes. Symmetric cross-correlation [77] performs two independent and parallel cross-correlations and multiplies the output, hence the number of multiplications is double the proposed scheme. Distance based algorithm [52, 53] performs cross-correlation over the length of the preamble to get two peaks at cross-correlator output, for $M = 2$. However when $M > 2$, the stored preamble which is used for cross-correlation should also be increased so as to get two peaks at the cross-correlator, hence required number of multiplications increase with M. In the proposed scheme, the multiplications are independent of the number of repetitions as cross-correlation is performed over the derived preamble, as in (4.6).

Table 4.2: Comparison of number of multiplications in cross-correlation for timing synchronization.

Algorithm	Number of Multiplications
Proposed	$(NO_s)^2$
Distance-based [52, 53]	$\frac{M}{2}(NO_s)^2$
Symmetric Cross-correlation [77]	$2(NO_s)^2 + 1$

The segmented nature of the cross-correlation in the proposed scheme allows for optimized implementation of cross-correlation. Figure 4.14 shows the architecture of the pipelined cross-correlator module.

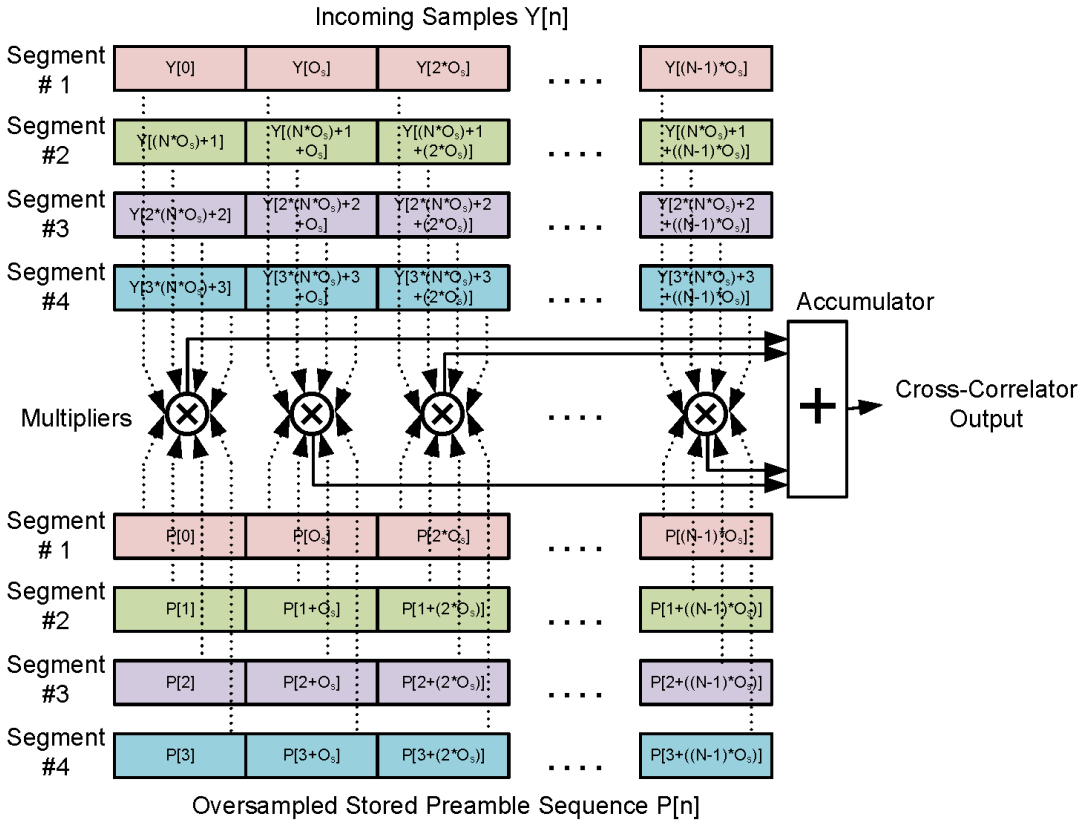


Figure 4.14: Implementation architecture of pipelined cross-correlator.

The number of segments, M, in the cross-correlation module is the same as the number of segments, S, in the circular register buffer used for storing the incoming samples. In the

proposed implementation, M and S are set to O_s the oversampling ratio. The pipelined cross-correlation has the advantage that only fewer number of multiplier units, logic and signal resources are used for the cross-correlator implementation as compared to conventional cross-correlator realization [103]. Fast multiplier architectures such as Booth Recoded Wallace Tree multipliers [105, 106] can be considered for fast multiplication. Apart from fast multipliers, a fast accumulator is required to complete the multiplier accumulations over M segments. To satisfy the speed requirements one could use multi-operand adder architecture for the cross-correlator′s accumulator.

Step 4: Thresholding and Peak Detection of the cross-correlation output.

The peak value in the absolute square of the cross-correlation output is determined by thresholding and searching for the peak after the threshold has been exceeded. The threshold is dynamically determined using the windowed moving average of absolute square of cross-correlation output. However, for certain preamble repetitions which result in multiple peaks, another peak of lower magnitude at (MNO_s) samples ahead and before the current peak is looked for and it is used in determining the frame start sample index. The threshold values are determined based on the mean value of cross-correlator output over a certain number of samples proportional to (MNO_s).

Step 5: Estimation of timing offset.

The sample index with the maximum value of absolute square of the cross-correlation output gives the timing offset with respect to the received preamble sequence when no secondary peaks are present at either of those locations mentioned above, else the frame start sample index is determined by the sample index of the peak which occurs first in the time sequence. This sample index is used to determine the received sample data which is used for further processing.

4.6.2 Hardware Implementation

4.6.2.1 Architecture for Proposed Synchronization

Figure 4.15 shows the high-level architecture for the implementation of the proposed synchronization technique.

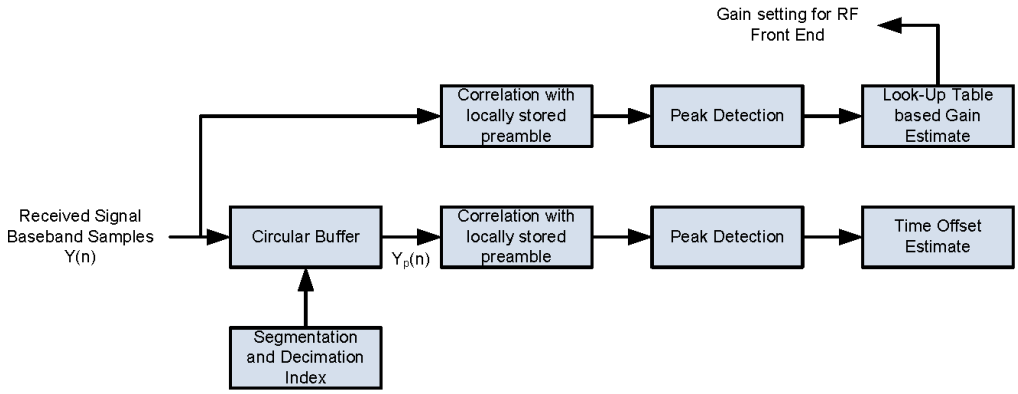


Figure 4.15: High-level architecture of the proposed timing synchronization.

The proposed synchronization process needs to derive the preamble sequence elements by segmentation and decimation. For this purpose, a dedicated address generator circuit is used to generate the address of the register location of elements of the derived preamble sequence. The derived preamble y_p is cross-correlated with the locally stored preamble in the cross-correlation module. Peaks in the output of the cross-correlator are determined. The offset of the peak with respect to the locally stored preamble is the estimated timing synchronization offset, which is used to compute the start of frame sample index time. The required RF gain for Automatic Gain Control (AGC) is determined by comparing peak amplitude of the cross-correlator output to a look-up table. The look-up table maps different range of peak outputs to different gain values. The gain is computed for each repetition of the preamble sequence and is applied till the received peak amplitude is in the required range. The cross-correlator module used for Gain Computation and Time

Synchronization could be multiplexed for reducing the hardware resources of the implementation, albeit with the use of higher operating clock frequency.

The derived preamble sequence element's addresses $e[n]$ is given by (4.11)

$$e[n] = n + \big((NO_s)[(n-1) - (MM_s)]\big) + (D_s - 1)M_d \qquad (4.11)$$

where n is the derived preamble sequence sample number, N is the number of samples in the preamble sequence, M is the number of repetitions of the preamble sequence, O_s is the oversampling factor at signal digitization, D_s is the decimation factor, M_s is the preamble repetition number and M_d is the preamble repetition number after the received samples are decimated by a factor of D_s. It has been noted that, a decimation factor of one $(D_s = 1)$ gives better synchronization performance. Hence, throughout the simulations and implementation, the decimation factor D_s is set to 1.

Figure 4.16 shows the hardware architecture of the pipelined cross-correlator implementation.

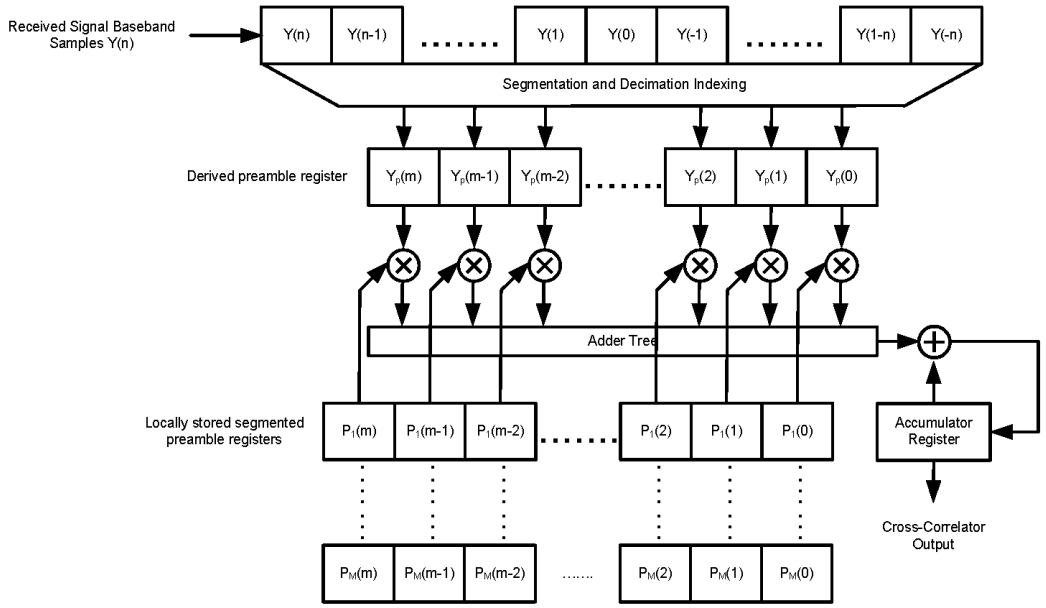


Figure 4.16: Hardware architecture of segmented cross-correlation implementation.

The derived preamble register and the stored preamble registers are clocked at higher frequency (number of segmented vectors, Mx) than the sampling frequency. This considerably reduces the overall hardware resource requirement and power consumption of the cross-correlator implementation.

4.6.2.2 FPGA Implementation

The proposed synchronization scheme including segmentation, decimation and timing synchronization technique has been coded in VHDL and implemented on a Xilinx′ s ZC706 Evaluation Kit for Zynq 7000 FPGA. Zadoff-Chu (ZC) sequences of length 32 (N) are transmitted with repetition of 4 (M). At the receiver, the baseband samples were over sampled by 4 (O_s). The segmentation of the received samples was carried out by arranging the incoming input baseband samples into a *M* segment circular register buffer of total length (NMO_s) The derived preamble is obtained by addressing the circular buffer based on the location as per the address generator given by (4.11). In the current implementation Zynq device′s DSP48 slices were used for the implementation of cross-correlator. In order to compare the gain in hardware resources and power consumption while using the pipelined implementation of the cross-correlator, the cross-correlator is implemented in different ways, as a conventional full-parallel cross-correlator implementation [103] and also conventional synchronization which cross-correlates over the complete duration of the preamble.

4.6.2.3 Hardware and Time Complexity Considerations

The cross-correlation operation takes up most of the hardware and determines the time complexity of synchronization. The comparison of hardware requirement and the power consumption of the proposed segmented algorithm's full-parallel and pipelined cross-

correlator implementations with conventional synchronization are given in Table 4.3. It should be noted that the cross-correlator implementation of the proposed segmented algorithm's full-parallel implementation and the conventional synchronization are the same, except that the number of cross-correlated samples are different. The proposed pipelined implementation involves a shorter critical path and can run at minimum clock period (MCP) of 14.78ns. The proposed design is capable of clocking at ~1.5 times higher speed than the full-parallel implementation (which requires a MCP of 23.48ns). In the FPGA implementation, a 40MHz processing clock has been used. Further, the pipelined cross-correlator implementation requires at least 3 times less hardware resources such as DSP48s, logic and signals as compared to the full-parallel/conventional implementation of the cross-correlator.

The total power consumption of the pipelined cross-correlator module implementation is 1.3x less than the full-parallel/conventional implementation of the cross-correlator when run at their respective maximum usable frequencies. Most of the power savings is from the dynamic power, which is from the reduced use of hardware, more particularly, the clock and signal resources for pipelined implementation. The energy consumption per MCP cycle for cross-correlator implementation in the proposed synchronization using pipelined cross-correlator implementation is 55% less compared to that of both full-parallel and conventional synchronization implementation. Here it needs to be noted that the energy consumption is dependent on how long the cross-correlator is kept active looking for the synchronization pattern.

The energy consumption is computed based on the number of samples used for cross-correlation at 40MHz processing clock of the FPGA implementation and the total static power over the entire duration of the preamble. Over the total duration of the implemented preamble sequence, which is 51.2us, based on 32 symbol preamble, with 4

repetitions at 2.5MHz bandwidth, the pipelined implementation's energy consumption is 14% less compared to that of full-parallel implementation and 57% less as compared to the conventional synchronization implementation such as those presented in [52, 53, 77].

Table 4.3: Comparison of proposed segmented synchronization's full-parallel and pipelined implementation with conventional cross-correlator synchronization.

Implementation / Parameter	Proposed segmentation synchronization			Conventional Synchronization	Overall reduction
	Full-parallel	Pipelined	Reduction		
Number of samples cross-correlated	128	128	0	512	73%
Number of DSP48s	520	136	~73%	520	~73%
Logic LUTs	10241	2490	~75%	10241	~75%
MCP (ns)	23.48	14.78	~37%	23.48	~37%
Static Power (mW)	213.88	212.50	<1%	213.88	<1%
Dynamic Power (mW)	433.34	249.92	~42%	433.34	~42%
Total Power (mW) @ 40MHz	647.22	462.41	~28%	647.22	~28%
Energy Consumption @ 40MHz over duration of preamble (uJ)	4.12	3.51	~14%	8.28	~57%

4.7 Summary

An efficient cross-correlation algorithm for robust synchronization for frame based burst communication systems with multiple repetitions of preambles has been proposed. The proposed synchronization is achieved by cross-correlation of local preamble replica with the segmented and decimated samples of the copies of the received preamble. The suitability of this synchronization approach for systems operating in high-mobility

environments has been shown through MATLAB simulations over different propagation channels such as AWGN and various LTE channel models. Performance of the proposed synchronization has been simulated over different oversampling ratio, number of preamble repetitions and preamble length. Robustness of the proposed algorithm has been demonstrated by adding different frequency offsets and Doppler shifts to these channel models with limited to no-change in the timing estimation performance. It has been noted that the maximum timing estimate error is reduced compared to other synchronization approaches, especially in low SNR scenarios. The segmented nature of the algorithm allows for computationally efficient hardware implementation of the proposed algorithm. The proposed approach has been implemented on Xilinx FPGA platform for functional validation and performance evaluation. The proposed pipelined implementation of the cross-correlator has resulted in nearly 37% reduction in minimum usable clock period, 28% reduction in power consumption and 73% reduction in usage of FPGA hardware resources such as DSP blocks, logic and signals over the conventional correlation. It is also found to offer nearly 55% lower per cycle energy consumption over the full-parallel implementation and 57% reduction as compared to the conventional cross-correlator based synchronization. Overall, the proposed synchronization approach has shown to be suitable for efficient hardware implementation with improved synchronization performance.

Given the proposed synchronization approach's hardware efficiency and minimal timing offset error, in the next chapter it will be shown how these advantages aid in the overall system. To demonstrate these, 1) Fast Synchronization and acquisition of frequency hopping signals and 2) Fractional time delay estimation using multiple antennas, which can be used for ranging and localization purposes, have been proposed.

Page Intentionally
Left Blank

5 Adaptation of Proposed Time Synchronization for FH Acquisition and Fractional Time-Delay ToA Estimation

5.1 Overview

In Chapter 4, an efficient cross-correlation algorithm for robust time synchronization of burst signals has been presented. The proposed method provides a resource efficient hardware implementation of the cross-correlation based time synchronization functionality in a CR receiver. In this chapter, taking advantage of the resource efficiency and robustness of the proposed synchronization scheme, two adaptations of the method are presented. First, it is applied to realize a fast acquisition and time synchronization scheme for frequency hopping (FH) signals which is presented in Section 5.1. In Section 5.2, the proposed synchronization scheme is used for determining the fractional time-delay time-of-arrival (ToA) estimation of OFDM-WLAN signals, which aids in reducing wireless positioning error by improving the ToA estimation accuracy of wireless burst signals.

5.2 Fast Acquisition and Time Synchronization of Frequency Hopping Burst Signals

5.2.1 Overview

Frequency hopping (FH) is used in different communications systems due to its robustness by providing frequency diversity against jamming and interfering signals [39, 40]. Successful detection and demodulation of a frequency hopping signal is dependent on proper tuning to the receiver to the transmit frequency and time synchronization of the receiver with the received burst signal. An important aspect of successfully receiving FH bursts is the initial acquisition and determination of subsequent hop frequencies. The total time for initial acquisition of the FH sequence is important in burst systems, since one lost burst amounts to loss of user data and possibly late entry into a network of nodes, delaying the overall network formation.

In this chapter, based on the synchronization technique proposed in Chapter 4, a scheme for fast acquisition and time synchronization of FH burst signals is proposed. In the proposed approach, acquisition and time synchronization are achieved simultaneously. In this two–level scheme, the received signal is first down-converted to a digital IF frequency using a broadband digital down-converter. The broadband digital IF consists of multiple FH frequencies from the FH frequency set, each at different frequency offsets. Cross-correlation of the received broadband digital IF is performed with a modified replica of the reference preamble sequence, individually for each of the digital IF frequency offsets, for determining the presence of the desired signal in that particular frequency offset. The modified reference ZC preamble sequence is generated by modulating it to the center frequency of FH bin before cross-correlation. If performed conventionally this will increase the total number of multipliers needed for cross-

correlation. However, when the synchronization technique proposed in Chapter 4 is used and by using a signum based cross-correlation, where only the sign-bit of each sample is used for cross-correlation, the hardware resources required for the cross-correlation are reduced. Especially, use of multipliers is eliminated, as multiplication in signum cross-correlator can be implemented with XNOR logic gates.

A two-level acquisition based scheme for fast acquisition of the frequency hopping and time synchronization of FH burst signals with Zadoff-Chu (ZC) based synchronization preambles is presented in this Section. The simulation results detailing the proposed scheme's performance and a low complexity hardware implementation architecture are presented.

In sub-section 5.1.2, the system model is presented and the proposed solution is described in sub-section 5.1.3. The advantages and disadvantages of the proposed scheme in comparison with conventional schemes are discussed in sub-section 5.1.4. In sub-section 5.1.5, the computer simulation setup and results for timing synchronization performance are presented. Implementation architecture of the proposed scheme is presented in sub-section 5.1.6 and sub-section 5.1.7 concludes the above work.

5.2.2 System Model

The signal burst in each of the transmitted FH frequency has a ZC sequence preamble with multiple repetitions followed by a data payload which contains information on the FH frequency set and the FH sequence. The signal structure is shown in Figure 5.1.

A transmit burst will be transmitted in any of the frequency hop bins, $f_1, f_2, f_3, \dots, f_b$ at hop times $h_1, h_2, h_3, \dots, h_a$. The sequence of hop frequencies is generally determined by a Pseudo-Noise (PN) sequence and the hop duration is determined by the total number of frequency hops per second. Time synchronization is achieved using synchronization

preambles in the transmit burst. The transmit packets contain a preamble section, which is used for time synchronization of the incoming burst and a data section as shown in Figure 5.1. The data section contains the user data, along with information on the FH sequence. Successful acquisition of the hop frequency sequence could be achieved when at least a single burst's data is successfully decoded at the receiver.

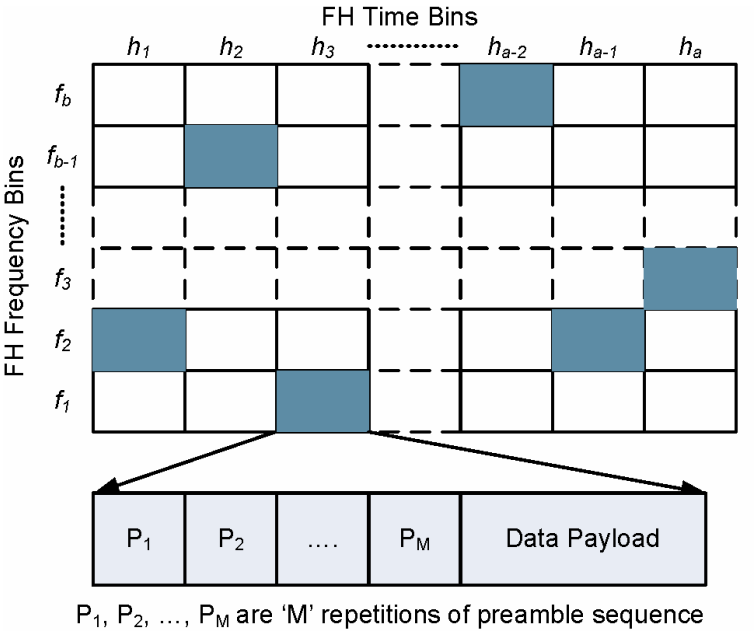


Figure 5.1: Frequency hopping burst structure [107].

The transmit FH burst consists of a ZC sequence of length *'N'*. The transmit preamble ZC is generated with root index *'u'* and it's i^{th} sample $x[i]$ is

$$x[i] = e^{j\frac{\pi}{L}ui^2} \qquad 0 \leq i \leq N-1 \qquad (5.1)$$

The above sequence is repeated 'M' times and the remainder of the hop time 'T_h' is the data payload. The preamble ZC sequence, $\boldsymbol{x}$, is a $(N-1)$ length vector

$$\boldsymbol{x} = (x(0), \quad \ldots, x(N-1))^T \qquad (5.2)$$

Where $(\,)^T$ represents the transpose of the vector.

The 'M' times repeated transmit preamble, 'X', is a simple concatenation of vectors '$\boldsymbol{x}$' at the transmitter of length 'MN'.

$$\boldsymbol{X} = [\boldsymbol{x}^T\ \boldsymbol{x}^T\ \boldsymbol{x}^T\ \ldots.\boldsymbol{x}^T]^T \qquad (5.3)$$

In the receiver, after digital down conversion, the received signal $y[n]$ is

$$y[n] = e^{j\frac{2\pi}{N}nf_i}\, X[n-d] + w[n] \qquad (5.4)$$

where 'd' is the time offset which needs to be estimated to achieve time synchronization with the incoming burst signal and $w[n]$ is the Additive White Gaussian Noise (AWGN) associated with the n^{th} sample.

After baseband conversion of the received signal, 'f_i' would have been the carrier frequency offset (f_{CFO}) as seen at the receiver due to transmitter and receiver frequency mismatch. However, in the proposed scheme, 'f_i' will also include the passband frequency offset f_{ci} for that particular channel from the digital IF frequency of 'f_{IF}'.

$$f_i = f_{IF} \pm (f_{ci} + f_{CFO}) \qquad (5.5)$$

Unlike the carrier frequency offset (f_{CFO}), an unknown parameter at initial acquisition, the frequency offset (f_{ci}) is fixed at design time and is at least twice the bandwidth of the transmitted signal in order to avoid the overlapping of signals from two adjacent channels, even though both the channels may not be occupied at the same hop time 'h_i'.

Conventionally, to estimate 'd', the received signal time offset representing the start of the burst, one could auto-correlate or cross-correlate [50, 53, 55, 68, 77] the received signal '$y[n]$' with its own delayed version or with the locally stored replica of the transmit preamble sequence. The cross-correlation is given as

$$r[k] = \sum_{n=0}^{MN-1} y[n+k]\,(X^*[n]) \qquad (5.6)$$

where $|.|$ denotes the absolute value and * is the complex conjugate operator.

Since, the preamble ZC sequence is repeated 'M' times, the above cross-correlation has multiple peaks, ideally *'M'*, one for each repetition of the preamble ZC sequence. The detection of the preamble sequence and an estimate of 'd', is obtained based on the distance between the peaks (of $r[k]$).

$$r[k] = \sum_{n=0}^{MN-1} e^{j\frac{2\pi}{N}(n+k)f_i} X[n-d+k]\,(X^*[n]) + \sum_{n=0}^{MN-1} w[n+k] \quad (X^*[n]) \tag{5.7}$$

Since $w[n]$ is AWGN, one can ignore the contribution of the 2nd summation in (5.7). However, $y[n]$ has a large frequency offset f_i, due to the passband frequency offset f_{ci}, the cross-correlator output magnitude produces a large variation depending on f_i in (5.5). Also since the cross-correlation is performed at the digital IF, the number of samples required for cross-correlation of the preamble sequence is large. This increases the overall complexity of the hardware required for such an implementation.

5.2.3 Proposed Acquisition and Synchronization

The synchronization mechanism described in Chapter 4 is used to derive preamble sequence from the received samples, $\boldsymbol{y}_p$.

$$y_p[nM+m] = y[nM + M(N^1+1)] \tag{5.8}$$

where $N^1 = (O_s * N)$; $0 \le n \le \left(\frac{N^1}{M} - 1\right)$ and $0 \le m \le (M-1)$; N is the number of samples in the preamble, O_s is the sampling factor and M is the number of repetitions in the preamble sequence. From (5.4), $\boldsymbol{y}_p$ can be re-written as

$$y_p[q] = e^{j\frac{2\pi}{N}qf_i} X[q-d] + w[q];\ 0 \le q \le MN-1 \qquad (5.9)$$

where $q = nM + m$ is derived as in (5.8). This segmentation and decimation of $y[n]$ to derive $y_p[q]$ reduces the number of samples required for cross-correlation and also reduces the multiple peaks at the cross-correlator output as shown in sub-section 4.3. However, the variation of the cross-correlator magnitude due to 'f_i' remains.

In order to compensate for the effect of f_i, instead of using a conjugate of the baseband ZC sequence $x[i]$, a conjugate of $x_{ref}[i]$ is used, which is an up-converted version of $x[i]$, up-converted to $f_{IF} \pm f_{ci}$ with the same sampling factor O_s. Since this frequency offset ($f_{IF} \pm f_{ci}$) is known at design time, this up-converted sequence $x_{ref}[i]$ is precomputed and stored at the receiver. It is given as

$$x_{ref}[i] = e^{j\frac{2\pi}{N}m(f_{IF}\pm f_{ci})} X[m]\ \ 0 \le m \le MN-1 \qquad (5.10)$$

With the above modified reference ZC sequence, the cross-correlator output $r_{if}[k]$, ignoring the noise term in (5.7) can be written as

$$r_{if}[k] = \sum_{q=0}^{MN-1} e^{j\frac{2\pi}{N}(q+k)f_i} X[q-d+k]\,(x_{ref}{}^*[q]) \qquad (5.11)$$

After substituting for f_i and $x_{ref}[q]$ from (5.5) and (5.10) in (5.11), we have

$$r_{if}[k] = e^{\pm j\frac{2\pi}{N}kf_i} \sum_{q=0}^{MN-1} e^{\pm j\frac{2\pi}{N}qf_{CFO}} X[q-d+k]\,(X^*[q]) \qquad (5.12)$$

In (5.12), the overall effect of f_{IF} on $r_{if}[k]$ is removed. But due to the excellent cross-correlation properties of ZC sequences, there are still multiple peaks (as mentioned in sub-section 4.3) but of relatively lower magnitude as compared to that obtained at the

output of the cross-correlator, when cross-correlating the complete repetitions of the ZC sequence preamble. The cross-correlator output $r_{if}[k]$ of the incoming preamble sequence with the up-converted ZC sequence exhibits a single peak, which is used to estimate 'd' the time delay, 'd_{est}'.

$$d_{est} = arg \max_{k} |r_{if}[k]| \qquad (5.13)$$

In (5.12), computation of $r_{if}[k]$ is performed on samples at passband frequency $(f_{IF} \pm f_{ci})$. The number of samples that need to be cross-correlated is larger than that of a baseband cross-correlator. This increases the total number of multiplications required to compute the cross-correlation and subsequently the hardware resources needed for the implementation. In order to reduce the number of computations, the constant amplitude and excellent cross-correlation properties of the ZC sequences [107] are exploited and a signum based cross-correlator [66, 109] is implemented. Each complex real and imaginary sample of the sequence being cross-correlated is converted based on the signum function

$$sgn(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases} \qquad (5.14)$$

By using the signum, the computation of the complex multiplication is easily realized using XNOR functionality, thereby reducing the overall hardware implementation complexity.

A simplified block diagram of the proposed acquisition and time synchronization scheme is shown in Figure 5.2. The incoming analog signal from RF frontend is digitized and down converted on to a digital IF, f_{IF} by a broadband Digital IF downconverter.

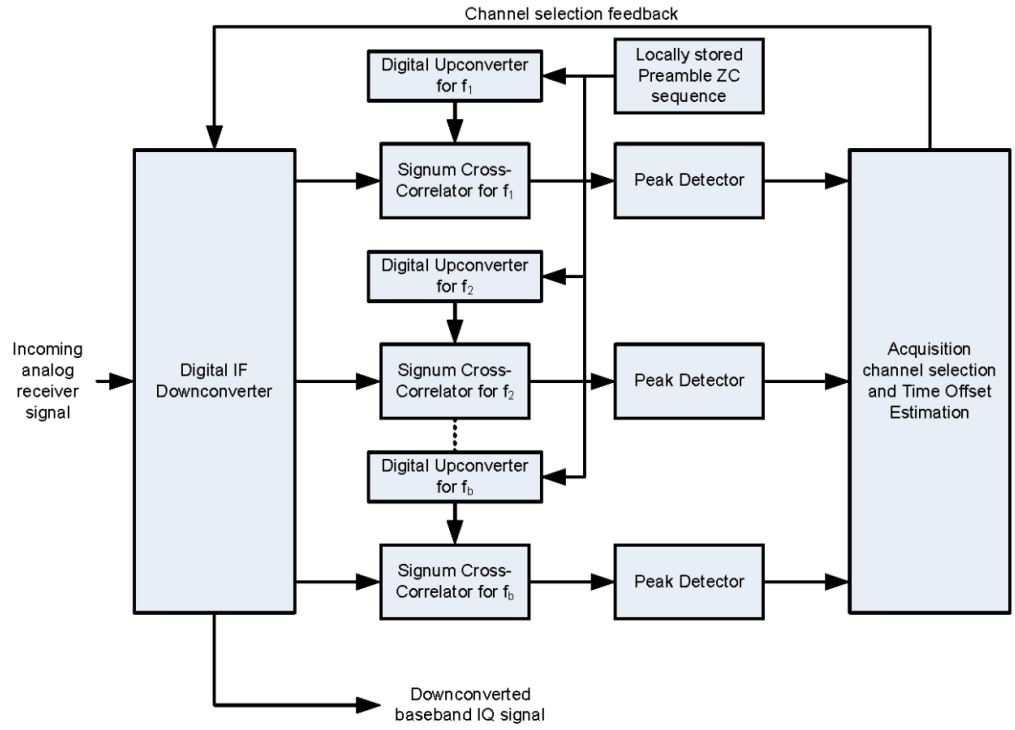


Figure 5.2: Block diagram of the proposed scheme.

The broadband digital IF contains multiple narrowband channels positioned at offset, $\pm f_{ci}$. Each of the down converted digital IF channel is cross-correlated with a locally stored preamble sequence which is up converted to its corresponding digital IF frequency, $f_{IF} \pm f_{ci}$. Each of the cross-correlator's peaks is detected and the acquisition decision along with the estimate of the time offset is made. The detected channel frequency information is fed back to the digital IF downconverter which retunes its digital IF frequency so that a down converted baseband IQ signal is available at its output.

The baseband IQ signal is appropriately filtered, demodulated and its data is decoded. The decoded data of the burst provides further information on the next FH bin, to which the digital IF is tuned to. Since the channel is searched over multiple frequency bins simultaneously, the overall acquisition time of the FH signal is reduced as it reduces the search region by the order of the number of parallel digital IF channels.

5.2.4 Advantages and Disadvantages

The main advantages of the proposed scheme for frequency hopping (FH) signals are

1) Reduced FH acquisition time

With a number of parallel cross-correlators operating on multiple digital IF channels simultaneously, the total time for FH sequence acquisition is reduced by the number of parallel cross-correlators used.

2) Reduced hardware resources for implementation

In the proposed scheme, the derived preamble sequence (as discussed in Chapter 4) from the received signal for cross-correlation is used. Also signum based cross-correlators are used for implementation. These are implemented on FPGA with minimal number of multipliers (DSP48E slices). This results in reduced complexity and hardware resource requirement for implementation.

The main disadvantage of the proposed scheme is that it is implemented on digital IF. This requires wideband digitizers, however the increased availability of integrated wideband RF receiver chipsets [110] makes it easier to implement. Another disadvantage is the larger number of samples needed for cross-correlation, compared to baseband cross-correlation based synchronization implementation. Though the number of samples are reduced with derived preamble implementation, they can be reduced further with appropriate sub-nyquist sampling.

5.2.5 Simulation and Results

To validate the proposed scheme and to demonstrate its effectiveness by quantifying the performance of the proposed scheme, MATLAB Monte Carlo simulations have been carried out on four digital IF channels. The sample rate is set at 56MHz with different IF channels placed at 3MHz, 6MHz, 9MHz and 12MHz. These frequencies were chosen to

specifically demonstrate the validity of the proposed scheme on harmonic multiple frequencies. The transmit preamble sequence used is a ZC sequence of 128 samples, which was repeated 4 times, creating a total preamble length of 512 samples. The bandwidth of each channel was set at 1MHz with time domain ZC preamble sequences.

The proposed scheme's performance was evaluated for AWGN and LTE-EPA, LTE-EVA, LTE-ETU [111] and 6-path Rician multipath channel models. All the performances have been evaluated based on synchronization rate. The synchronization rate is the ratio of received burst preambles which are detected to the total number of transmitted burst preambles. The received burst preambles are declared as detected when the time offset to the start of the incoming preamble sequence is correctly estimated. The channel has been declared as synchronized and detected with time offset estimation completed when preamble is detected and the estimated time is within one baseband sample of the transmitted preamble sequence.

Figure 5.3 shows the synchronization performance of channel-1, a 3MHz IF channel, when detected through a wideband digital IF with respect to the synchronization performance when detected as a single IF channel. In the wideband digital IF all the four channels were injected with different offsets of the same preamble sequence but modulated to different IF frequencies (3MHz, 6MHz, 9MHz and 12MHz). It can be seen that the wideband digital IF detection is similar to single IF channel performance for up to -5dB SNR. Below -15dB SNR, the performance of wideband digital IF scheme starts to degrade for LTE-EVA and LTE-ETU channels, mainly due to multipath. It is still within the 99% for up to -20dB SNR.

The detection performance of different IF channels is shown in Figure 5.4. For this simulation, the same preamble sequence was introduced in all the four channels but with

different time offsets. It can be seen that the performance of all IF channels is similar and is within 99% for above -20dB SNR.

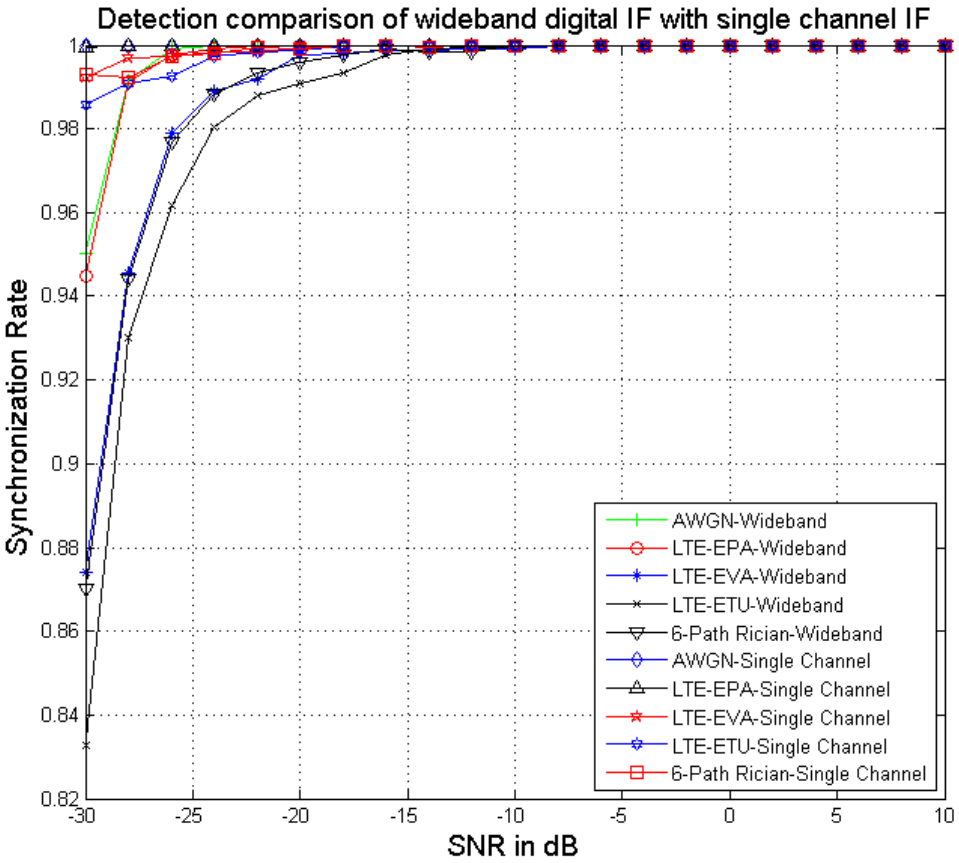


Figure 5.3: Performance of channel one (3MHz) when detected through a wideband digital IF as compared to it's detection as a single IF channel.

The reduction in detection performance when using signum cross-correlators with respect to full-precision correlators has been simulated and given in Figure 5.5. It can be seen that the difference is within 0.01 when the SNR is greater than -20dB for all the four digital IF channels over AWGN, LTE-EVA and LTE-ETU propagation channels.

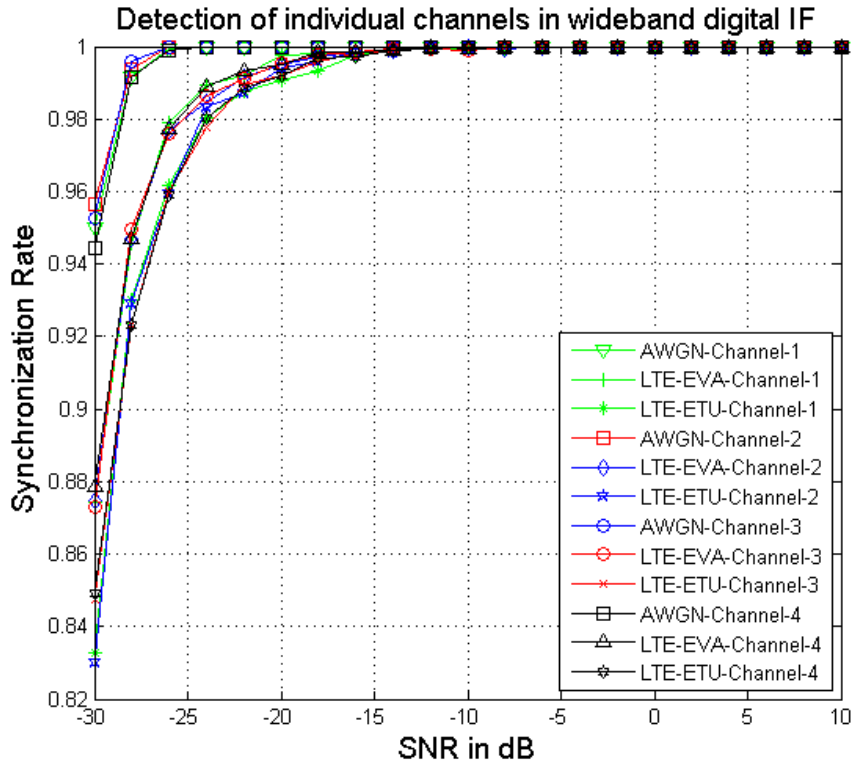


Figure 5.4: Performance of different channels when detected through a wideband digital IF.

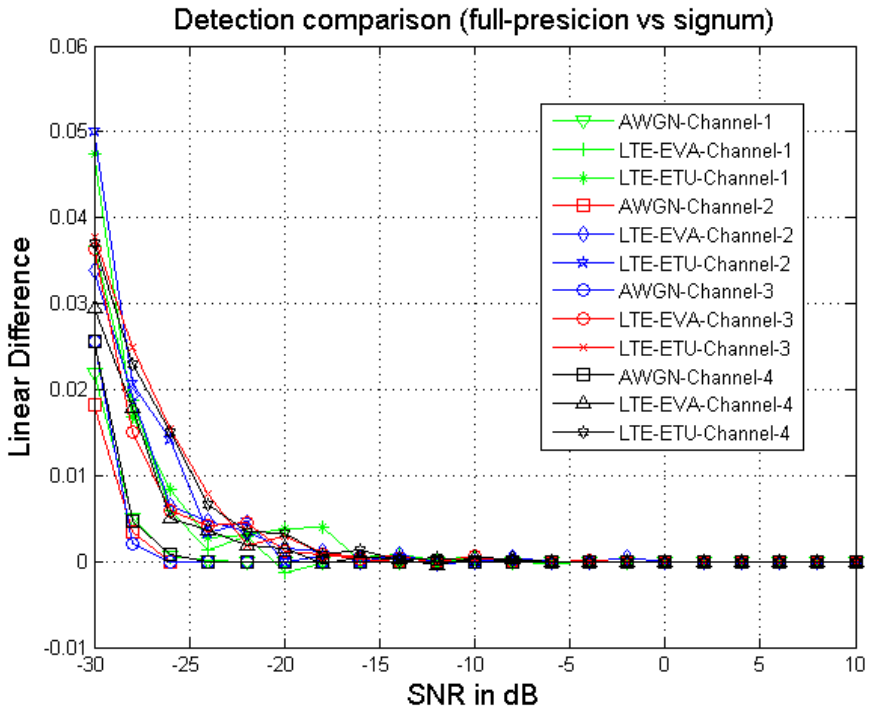


Figure 5.5: Difference in performance when signum correlators are used instead of full-precision correlators.

5.2.6 Implementation Architecture

Receiver hardware implementation architecture for the proposed synchronization is given in Figure 5.6. The hardware is mapped to a platform consisting of Analog Devices FMCOMMS3 [110] based analog front end, which is connected to a Xilinx ZC706 Zync FPGA evaluation board. The FMCOMMS3 is a wideband analog front end which provides a tuning range from 70MHz to 6GHz, based on the AD9361 Wideband RF Transceiver chipset. In the proposed scheme, the sample rate (the ADC sampling rate is higher) at the output of the chipset is set at 56MHz with a bandwidth of 28MHz centered around 14MHz digital IF. The Xilinx ZC706 evaluation board which carries the FMCOMMS3 on one of its FMC sites, has a Zynq-7000 XC7Z045 SoC.

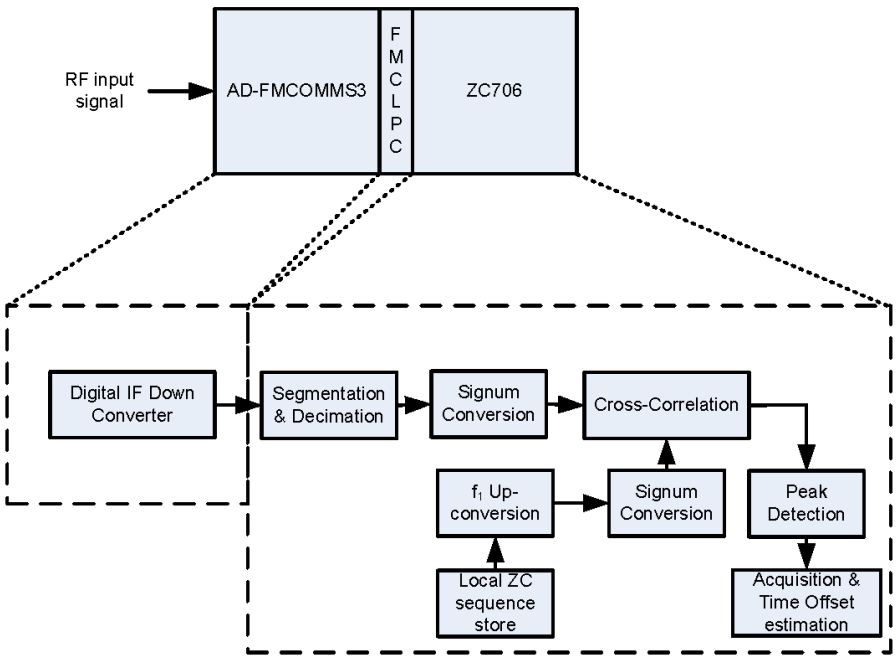


Figure 5.6: High-level implementation architecture mapping of the receiver.

The FH burst is of 1MHz bandwidth with a ZC preamble of 128 samples, which are repeated 4 times. With the set sampling rate of 56MHz, it is required to cross-correlate a sequence of length 7168 samples for each IF channel at the receiver. This will lead to a

complex synchronizer. However, since the proposed implementation is based on signum correlators, it uses only two DSP48E multiplier slices (used in computing the magnitude) of the FPGA, thereby reducing the complexity of the overall FPGA design.

5.2.7 Summary

The delay in acquiring the FH signal is undesirable as it leads to data packets loss in a communication system. Timing synchronization of the received data payload is also important for proper demodulation and decoding of the received data. A scheme and a high-level hardware implementation architecture is presented, which addresses these two issues. The proposed scheme acquires FH signals by searching for the preamble sequences simultaneously over multiple FH frequency bins and determines the time offset estimate based on the detection of the preamble sequence. The segmentation and decimation technique proposed in Chapter 4 is utilized for reducing the overall complexity of the proposed scheme. The performance of the proposed scheme over AWGN, LTE multipath propagation channels and a 6-path Rician propagation channel has been quantified through computer simulations. The synchronization performance of the proposed scheme over wideband digital IF is seen to be similar to conventional baseband synchronization and over single IF channel. This along with the simplified hardware resource requirement due to the use of signum cross-correlators makes the proposed scheme competitive over conventional baseband approaches for acquisition and synchronization of FH burst signals.

5.3 Fractional Time Delay ToA Estimation Using Short Preambles For OFDM-WLAN

5.3.1 Overview

Wireless signals, apart from providing data connectivity, have also been used for determination of position of wireless devices relative to other devices. WirelessLAN (WiFi) hotspots, of which OFDM-WLAN (802.11a/g/n/ac) [29] access points (AP) offering higher data rates, are increasingly being deployed in both indoor and outdoor environments. They provide wireless data connectivity and consequently are being used to reduce internet traffic load on traditional cellular networks. With most mobile terminals being equipped with OFDM-WLAN capability, WLAN APs based localization is becoming the preferred choice for positioning. With the over-the-air travelled distance of a radio frequency (RF) signal directly proportional to time of travel, most of the techniques derive position information based on time-of-flight of the signal from multiple anchor node transmitters to the wireless receiver. Several localization approaches such as Received Signal Strength (RSS), Time of Arrival (ToA), Time Difference of Arrival (TDoA) [111], Four-way ToA [112], Angle of Arrival (AoA) [113] and Two-way Ranging are available for deriving the position information with WLAN APs or a combination of any of these are used to estimate the location of the mobile terminal. Performance of these ranging and localization schemes is dependent on the accuracy of Time of Arrival (ToA) estimation of the incoming OFDM-WLAN signal. It is mentioned in [114-116] that the Cramer Rao Lower Bound (CRLB) for ToA estimation accuracy is dependent on the highest operating frequency, operating bandwidth, observation time duration and the transmit power.

Apart from the timing synchronization accuracy across multiple anchor node transmitters [117], accuracy of the estimated time-of-arrival of the transmitted signal at the receiver determines the accuracy of the estimated position information. With these different algorithms already presented in the literature, none of them make use of the multiple antennas, which are already introduced in the IEEE 802.11 standards and provide a low complexity solution for hardware implementation for ToA estimation. In this Section, based on the synchronization technique given in Chapter 4, a novel time domain fractional time delay ToA estimation method is proposed. The estimation is derived based on cross-correlation of an oversampled and rearranged received signal with a stored SP replica. The preprocessing of rearranging the received signal is similar to that proposed in Chapter 4. The fractional time delay ToA is estimated based on the use of multiple receiver antennas to reduce the error in the fractional time delay estimate. In the proposed method, the ToA estimates at individual antennas are combined to determine the fractional time delay estimate. Two different cases, when the direction of arrival of the OFDM-WLAN is known and when it is unknown are both discussed.

In the remainder of this section, in sub-section 5.2.2, the system model with OFDM-WLAN's preamble structure is presented. In Sub-section 5.2.3, description of the proposed fractional time delay estimation solution and its extension to multiple antenna receiver configurations is presented. The performance results of the proposed algorithm derived through computer simulation are given in sub-section 5.2.4 and a brief summary with conclusions are provided in sub-section 5.2.5.

5.3.2 System Model

The preamble sequence of OFDM-WLAN shown in Figure 5.7 has 10 short preambles (SP), each of 0.8usec duration. These are BPSK-OFDM symbols which occupy 12

subcarriers. In time domain each SP is of 16 chips length. GI is the OFDM symbol guard interval and LP the long preamble [29].

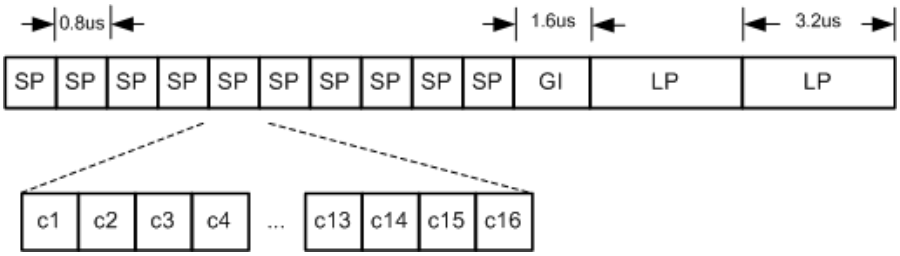


Figure 5.7: Preamble structure of OFDM-WLAN [29].

The transmit OFDM-WLAN preamble '**x**' has M = 10 repetitions of the short preamble '$\boldsymbol{x}$'.

$$\mathbf{x} = [\boldsymbol{x}^{(1)}\ \boldsymbol{x}^{(2)}\ \boldsymbol{x}^{(3)}\ \ldots\ \boldsymbol{x}^{(M)}] \tag{5.15}$$

In time domain, each of the identical short preamble '$x^{(m)}$' is a complex time sequence of N = 16 chips which can be represented as

$$\boldsymbol{x}^{(m)} = [x_1^{(m)}\ x_2^{(m)}\ x_3^{(m)}\ \ldots\ x_N^{(m)}] \tag{5.16}$$

The received OFDM-WLAN short preamble samples can be expressed as

$$y[t] = e^{j2\pi\varepsilon t}\sum_{l=0}^{L-1}\alpha(l)\,X\big(t-\tau(l)\big) + U(t) \tag{5.17}$$

Let $y[n]$ be the sampled equivalent of $y[t]$. With 'ε' being the carrier frequency offset, '$\alpha(l)$' the channel coefficient of the 'l^{th}' path with propagation delay '$\tau(l)$', and $U(n)$ the n^{th} sample's additive white Gaussian noise. At the receiver, the received signal is oversampled by a factor of 'O_s'. Let '$O_s = M$', the number of SP repetitions, 'T_b' the transmit signal sample period and the oversampled sample period, '$T_s = \frac{T_b}{v}$'.

The propagation delay '$\tau(l)$' is a summation of integer delay '$\tau_i(l)$' and fractional delay '$\tau_f(l)$'. Conventionally in OFDM-WLAN, SPs have been auto-correlated to

achieve symbol time synchronization. This approach is sufficient to estimate $\tau(l)$ within the variance of the OFDM symbol's guard time [113]. However, for positioning applications, $\tau_f(l)$ is of importance. More specifically, the propagation delay of the direct path $\tau(0)$, including the fractional delay $\tau_f(0)$ is to be estimated. When '$T_s \rightarrow \tau(0)$', the estimate is accurate.

(5.17) can be re-written by separating direct path and multipath components as

$$y[nT_s] = D[nT_s] + I(nT_s) + U(nT_s) \tag{5.18}$$

where '$D(nT_s)$' is the desired direct path and '$I(nT_s)$' the multipath components.

$$D[nT_s] = e^{j2\pi\varepsilon nT_s}\alpha(0)\, x\big(nT_s - \tau(0)\big) \tag{5.19}$$

$$I[nT_s] = e^{j2\pi\varepsilon nT_s}\sum_{l=1}^{L-1}\alpha(l)\, x\big(nT_s - \tau(l)\big) \tag{5.20}$$

For OFDM-WLAN, cross-correlation output '$r[k]$' of the received signal with the stored preamble replica will be

$$|r[k]| = \left|\sum_{n=0}^{O_s NM-1} y[(n+k)T_s]\, x^*(nT_s)\right| \tag{5.21}$$

where $|.|$ denotes the absolute value and * is the complex conjugate operator.

However, depending on the number of samples that are being cross-correlated and due to the multiple repetitions of SP in OFDM-WLAN, the efficiency to detect and synchronize the received signal is poor. This is mainly due to multiple peaks in '$|r[k]|$', especially in multipath environments and at low to mid Signal to Noise Ratio as shown in Chapter 4 and mentioned in [118].

When multiple antennas, 'A' are used at the receiver, (5.21) can be determined at each antenna, '$r_a[k]$' being the cross-correlator output at each antenna with '$z_a[nT_s]$' each antenna's received signal, '$a = 0, 1, 2, \dots, A-1$'. Assuming a uniform linear array (ULA), the array output can be written as

$$Y_a[nT_s] = a(\theta)\, y[nT_s] + U_a(nT_s) \tag{5.22}$$

$$a(\theta) = \left[1 \exp\left(j\frac{2\pi}{\lambda} d_1 \cos(\theta)\right) \dots \exp(j\frac{2\pi}{\lambda} d_{A-1}\cos(\theta))\right]^T \tag{5.23}$$

and

$$z_a[nT_s] = \exp\left(j\tfrac{2\pi}{\lambda} d_a \cos(\theta)\right)\, y[nT_s] + u_a(nT_s) \tag{5.24}$$

where 'θ' is the angle of arrival, 'd_a' distance between antenna 'a' and the first antenna, 'and 'U_a' and 'u_a' representing the additive white Gaussian noise at antenna 'a'.

5.3.3 Proposed Fractional Time Delay Estimation Scheme

In the proposed method to estimate the overall propagation time delay of the direct path $\tau(0)$, cross-correlation is performed over only certain received samples of 'z_a'.

$$\boldsymbol{w}^{(m)} = \left[x_{11}^{(m)}\, x_{12}^{(m)} \dots x_{1O_s}^{(m)}\, x_{21}^{(m)}\, x_{22}^{(m)} \dots x_{2O_s}^{(m)} \dots x_{N1}^{(m)}\, x_{N2}^{(m)} \dots x_{NO_s}^{(m)}\right] \tag{5.25}$$

In order to determine the set of samples '$\boldsymbol{P}$' of the received signal that will be cross-correlated, one shall look at the time domain structure of transmitted OFDM-WLAN SPs, '$x^{(m)}$'. If it had been oversampled by a factor of 'O_s' and assuming sample repetition oversampling, from (5.16), Since SPs are repetitions,

$$x_{ij}^{(m)} = x_{ik}^{(p)} \tag{5.26}$$

for $\{m, p \in [1,2,\dots M]\}, \{i \in [1,2,\dots N]\}$ and $\{j, k \in [1,2,\dots O_s]\}$.

'$\boldsymbol{P}$' is formed such that, the samples are drawn from each SP repetition,

$$\boldsymbol{P} = \left\{x_{11}^{(1)}\ x_{12}^{(2)}\ \dots\ x_{1O_s}^{(M)} x_{21}^{(1)}\ x_{22}^{(2)}\ \dots\ x_{2O_s}^{(M)}\ \dots x_{N1}^{(1)}\ x_{N2}^{(2)}\ \dots\ x_{NO_s}^{(M)}\right\} \qquad (5.27)$$

Since (5.26) is valid, $\boldsymbol{P} = \boldsymbol{x}^{(m)}$. However, when a similar transformation is applied to the continuous received signal '$y[nT_s]$', only a single valid SP sequence remains in '$\hat{y}[nT_s]$', the transformed received signal, out of 'M' repeated SPs remains. Cross-correlating '$\hat{y}[nT_s]$' with '$\boldsymbol{P}$' should yield a single peak in '$|r[k]|$'.

$$r[k] = \sum_{n \in V} \left(\hat{y}[(n+k)T_s]\ x^*(nT_s)\right) \qquad (5.28)$$

where '$r[k]$' is the cross-correlation output, *is the complex conjugate operator and '$n \in V$' is the index of samples representing the transformation leading to (5.27). The size of 'V' is 'MN'. This is a reduction in the number of samples cross-correlated as compared to conventional cross-correlation. It reduces the range of 'k' over which (5.28) needs to be evaluated, which decreases the complexity.

For a 'a' antenna ULA, at each antenna the cross-correlator out is

$$r_a[k] = \sum_{n \in V} \left(\widehat{z_a}[(n+k)T_s]\ x^*(nT_s)\right) \qquad (5.29)$$

$$\boldsymbol{R}_a[k] = \left[r_0[k]\ r_1[k]\ r_2[k]\ \dots\ r_{A-1}[k]\right] \qquad (5.30)$$

where '$\boldsymbol{R}_a$' is formed by arranging 'a' columns of 'r_a', each being a '$k \times 1$' vector.

The cross-correlation sample offset 'k_{peak}' at which the peak magnitude of (5.29) occurs would be directly proportional to the ToA estimate. Note that, due to oversampling

the time resolution of 'k' is a fraction of 'T_b', $\frac{T_b}{O_s}$. However, the offset 'k_{peak}' is dependent not only on the direct path but also the multipath components at each antenna and the direction of arrival 'θ' of the OFDM-WLAN signal. To reduce the contribution of multi-path component delays '$\tau_m(l)$' on the estimation of 'k_{peak}', 'Q' a '$k \times k$' matrix computed from 'A' antennas '$r_a[k]$' is determined as

$$|Q| = |\boldsymbol{R}_a \, \boldsymbol{R}_a^H| \tag{5.31}$$

The signal's angle of arrival is required to compute (5.31), however when 'θ' is not available, the diagonal elements of (5.31) which are independent of 'θ' are computed and are given by

$$Q_{diag}[k] = \left|\sum_{a=0}^{A-1}(r_a[k]\, r^*{}_a[k])\right| \tag{5.32}$$

'k_{peak}' is determined as the offset at which (18) achieves maximum. Though multipath components contribute to 'k_{peak}', it is largely dependent on '$\alpha(0)$', direct paths magnitude, as determined in [115]. To further reiterate this, the nature of '$\alpha(l)$' is analyzed. If the channel is assumed to be a fast fading channel or a Non-Line of Sight (NLOS) channel, as likely in an outdoor/indoor environment, then the channel coefficients $\alpha(l)$, except for the direct path channel coefficient '$\alpha(0)$', can be assumed to be Rayleigh distributed random variables with a median ('r_{median}'), which is directly proportional to the received signal voltage [99]. This leads to the following three conditions:

1) When ($|\alpha(0)| > |r_{median}|$), the cross-correlation peak offset 'k_{peak}' is determined by the direct path and the estimated time delay is '$\tau(0)$'.

2) When ($|\alpha(0)| \leq |r_{median}|$), the cross-correlation peak offset 'k_{peak}' is determined by the phase changes in the received time samples due to carrier frequency offset 'ε'. It has to noted here that the perceived phase change across the cross-correlated time samples is randomized due to oversampling and rearrangement of the received samples as in (5.25). This leads to reduced cross-correlation magnitude due to multipath signals as they don't combine additively.
3) When ($|\alpha(0)| \ll |r_{median}|$), the estimated ToA is still within the received OFDM symbol's guard time and is good for time synchronizing and decoding the received signal.

5.3.4 Simulation and Results

To demonstrate the effectiveness and quantify the performance of the proposed scheme, MATLAB Monte Carlo simulations have been carried out. The channel models developed by IEEE 802.11 technical group for Wireless LAN and defined in [119] are used in this simulation. Channel Model B, C & F representing NLOS scenarios with maximum delay spread of up to 150nsec and multi-path taps of up to 40-taps have been used to evaluate the proposed scheme.

In the simulations, the time domain I & Q preamble sequences of OFDM-WLAN were used with over sample rate $O_s = 10$, same as the number of SP repetitions. The signal bandwidth was set to 20MHz at carrier frequency of 2.4GHz. In order to determine the Root-Mean-Square (RMS) error in estimating the ToA, random timing offsets were introduced, in the range of 1 to 20 samples, to the received signal. The RMS error has been normalized to the number of time offset samples.

Figure 5.8 shows the RMS error of the proposed fractional delay estimation in AWGN and IEEE802.11 channel models. It is seen that the performance of the proposed scheme

is comparable to the performance when the complete 10 repetitions of SP (total short preamble) are cross-correlated. The RMS error largely remains same for all SNR above -5dB. The RMS error of the proposed scheme as well as the total cross-correlation is ~8 oversampled samples for channel Model F, which is a harsh channel with up to 40 multi-path taps and a maximum rms delay of 150nsec.

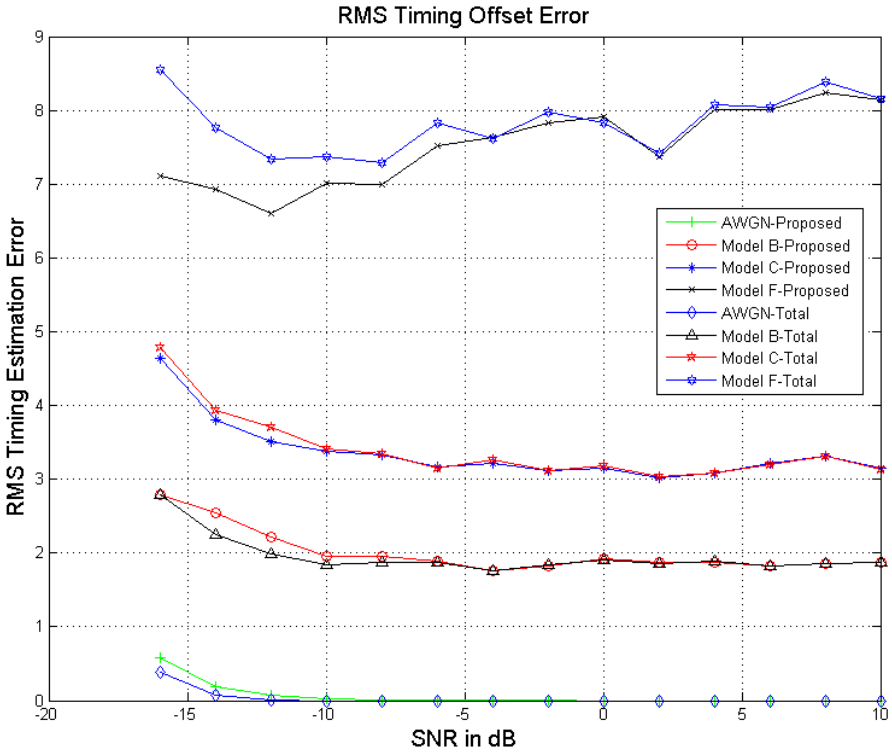


Figure 5.8: RMS Error in timing estimate compared to full length cross-correlation.

With the increase in the number of antennas from one to up to six, and assuming the direction of arrival is not known, with ToA estimate based on (5.32), the normalized RMS error for 5000 simulation runs of the timing estimate is given in Figure 5.9. It is seen that the RMS error reduces by approximately one sample for Model B and Model C channels over the simulated SNR range of -10dB to 10dB. However, for the harsh Model F channel there is very minimal improvement.

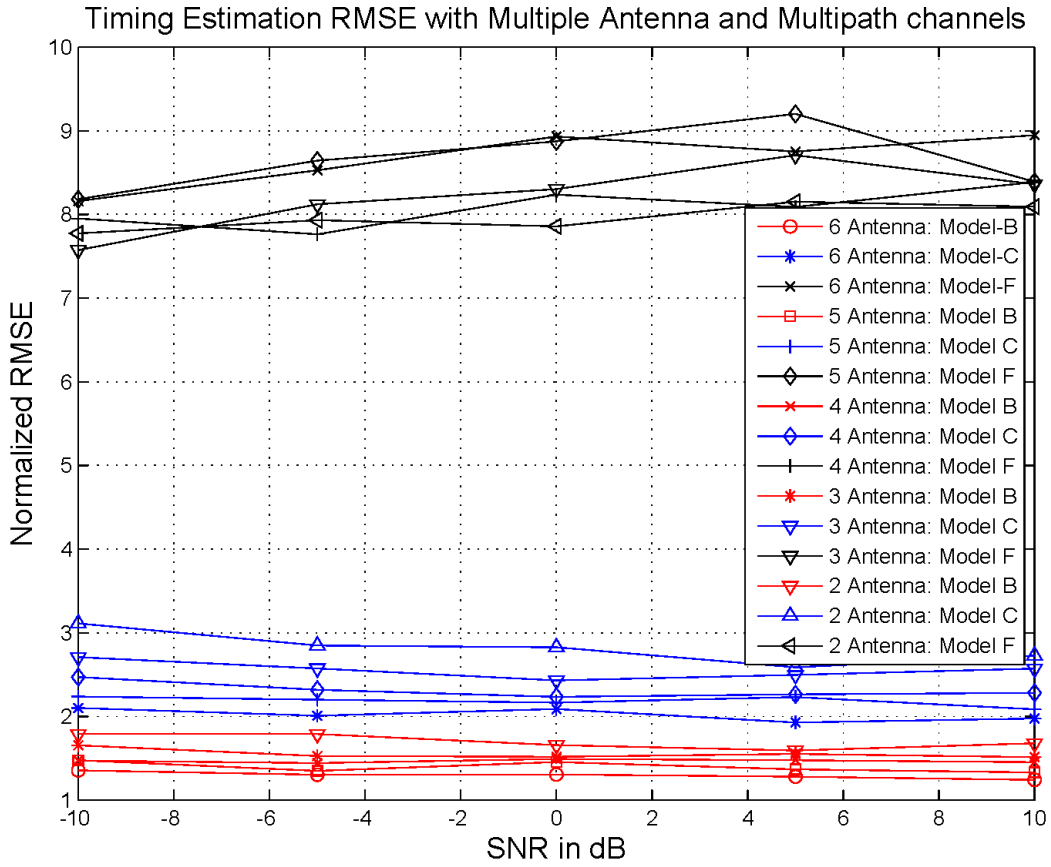


Figure 5.9: RMSE in timing estimate with multiple antennas and unknown DoA.

When using multiple antennas (2 to 6 antennas) and assuming that the DoA of the OFDM-WLAN signal is known when evaluating (5.31), then the normalized RMS error in ToA estimated for different SNR is given in Figure 5.10. The DoA was set to 45 degrees for this simulation. It is seen that the RMS error has improvement for both known and unknown DoA scenario across all the simulated SNRs with respect to the single antenna scenario shown in Figure 5.8. . The additional information of DoA does not contribute to improvement in RMS error for all the three simulated channel models (Model-B, Model-C and Model-F). This is because the DoA information at an antenna array is indirectly related to the time-of-arrival information of the signal at individual antennas. The positioning error is already derived based on the ToA information. Hence the availability of DoA information does not improve the RMS error. On the other hand,

since non line-of-sight signals or multipath signals destructively combine with the line-of-sight signals for ToA estimation, the proposed scheme aids in reducing the RMS error contribution due too multipath signals. This conclusion agrees with the results reported in [120].

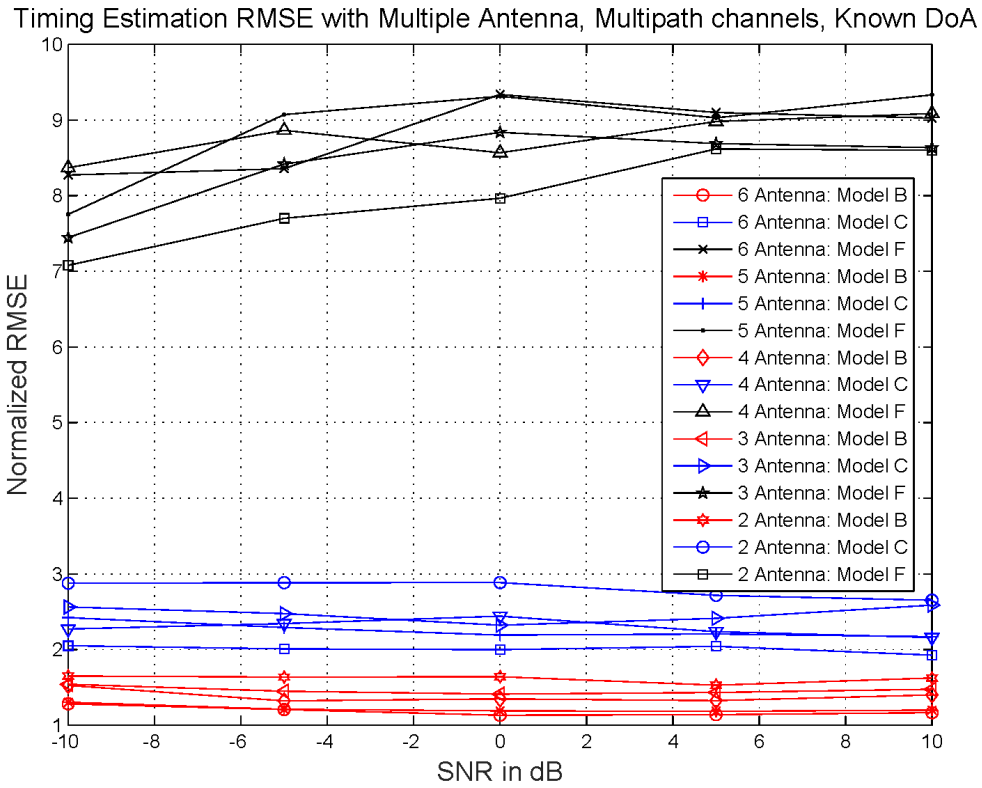


Figure 5.10: RMSE in timing estimate with multiple antennas and known DoA.

5.3.5 Summary

A novel ToA estimation approach in the time domain for OFDM-WLAN has been presented. It is based on cross-correlation of short preamble sequence and use of multiple antennas to improve the RMS error in the fractional time delay estimate. The proposed approach has been simulated for different IEEE 802.11 channel models and compared against the approach where the total length of the preamble is cross-correlated. Computer simulations show that the normalized RMS error remains largely constant for SNRs

above -5dB for all the channel models. However, when the number of receiver antennas are increased the RMS error decreases. It is seen that the availability of DoA information provides no additional improvement in estimation of the ToA. The proposed ToA estimation method is based on synchronization technique presented in Chapter 4 which could be implemented as an hardware efficient module.

.

Page Intentionally
Left Blank

6 Conclusions & Future Work

6.1 Conclusions

In this thesis, low complexity algorithms and architectures for realizing spectrum sensing, time synchronization, fast acquisition of FH signals and fractional time delay ToA estimation of burst signals have been presented. The proposed schemes aid in implementing hardware efficient burst communication waveforms in cognitive radios. All the proposed schemes are based on segmentation, decimation and cross-correlation of received signal samples which are processed at digital IF or down converted to baseband for processing. The segmented and decimated samples are processed further and cross-correlated to perform spectrum sensing or time synchronization.

In Chapter 3, a low complexity spectrum sensing algorithm and architecture has been presented. The proposed method is based on cross-correlation approach. It is shown to be effective for detection of continuous broadcast signals such as ATSC DTV, PAL-B Analog TV signal and QPSK modulated burst signal. MATLAB simulation results showed that the detection performance for ATSC digital TV, PAL-B analog TV and burst QPSK modulated signals is ~100% for all three signals types above 0dB SNR. For continuous signals it is even better with 100% detection for -10dB SNR. The probability of false detection is also consistently below 5%. It has been shown that the performance

of detection improves with the number of ISDV vectors used for averaging and cross-correlation for continuous signal detection. In the case of burst signals, an increase in the number of vectors increases the sensing duration. The sensing duration needs to be maintained to be shorter than the burst duration for achieving better performance in terms of probability of false alarm (P_{fa}) and probability of detection (P_d). The selection of number of vectors in the proposed scheme allows to dynamically trade-off the computational complexity for performance. A reconfigurable hardware architecture has been presented detailing the FPGA implementation of the proposed low complexity spectrum sensing technique.

For successfully demodulating and decoding the burst signals, the receiver needs to time synchronize with the incoming burst, within guard interval of the signal structure. Making use of the multiple preamble symbols present in the burst signal, a hardware efficient time synchronization technique has been presented in Chapter 4. The proposed technique is based on cross-correlation of the stored preamble sequence with a single preamble sequence which is derived from the received preamble samples. The proposed technique's robustness against multipath environments and carrier frequency offsets has been demonstrated through MATLAB simulations. Simulation results also showed that the proposed synchronization scheme performs better, in terms of the synchronization rate achieved, than the similar techniques proposed in the literature. The segmented sample structure of the derived preamble sequence of the proposed scheme, in which samples from different preamble repetitions are picked to form the derived preamble, enables hardware efficient implementation of the cross-correlator. An architecture for FPGA implementation of the synchronization scheme has also been proposed and it has been shown that the proposed pipelined cross-correlator architecture results in reduction of the minimum usable clock period thereby increasing the maximum possible speed as well as

reduction in the power consumption and usage of FPGA hardware resources such as DSP blocks, logic and signals over the conventional cross-correlation implementations. The proposed synchronization approach is suitable for efficient hardware implementation with improved synchronization performance.

Further in Chapter 5, the proposed time synchronization technique has been adapted to realize fast acquisition of Frequency Hopping burst signals and for estimating fractional time delay ToA of OFDM-WLAN signals. Given the hardware efficient implementation of the proposed time synchronization technique, it has been adapted for use at digital IF and with signum cross-correlators. The multiple parallel realizations of the time synchronization at digital IF allow for searching of the FH burst at multiple frequency bins simultaneously, thereby reducing the overall acquisition time of the FH burst signal. With the use of signum cross-correlators, when ZC preamble sequences are used, the overall hardware complexity is further reduced with minimal impact on the time synchronization performance with the proposed technique. MATLAB simulations showed the robustness of the scheme over different LTE propagation channel environments. The proposed hardware architecture has been implemented and it has been shown that overall hardware complexity is low.

Apart from time synchronization for demodulation of burst signals, the proposed time synchronization scheme has been adapted for fractional time delay ToA estimation as presented in Chapter 5. By increasing the digital oversampling ratio of the received burst signal and with the use of multiple antennas, the normalized RMS error is reduced for the estimated ToA. This has been applied to OFDM-WLAN signal's short preambles and MATLAB simulations show that the normalized RMS error at 10dB SNR is less than 3 digital samples for Model-B and Model-C propagations channels. For the Model-F

propagation channel, it is seen that the multiple number of antennas give no improvement in ToA estimate error.

The proposed low complexity algorithms and architectures for spectrum sensing, time synchronization, fast acquisition of FH signals and fractional time delay ToA estimation of burst signals are based on segmented processing of the oversampled received signal. It has been shown that this results in hardware efficient implementation of the proposed functional modules in a transceiver. With this work, it has become a possibility to reduce the hardware resources required for realizing certain important functions in Cognitive Radios, trade-off hardware complexity for better spectrum sensing detection performance and achieve better time synchronization performance with reduced hardware implementation complexity.

6.2 Future Work

Analysis, simulations and hardware realizations have shown that the proposed spectrum sensing, time synchronization and its adaptions are robust over different propagation channel scenarios and are hardware resource efficient. These techniques should prove useful in reducing the overall implementation complexity of a cognitive radio. However, with cognitive radios evolving into cognitive radio networks (CRNs), the co-operative spectrum sensing and time synchronization across multiple CR nodes in the CRN is important. As such the next steps would be

- To develop the proposed spectrum sensing scheme in a CRN with collation of sensing data from multiple nodes and improving the overall performance of the spectrum sensing over the complete CRN.

- To develop the proposed time synchronization algorithm to synchronize multiple nodes in a CRN through derivation of a preamble sequence from samples of multiple node's preamble samples.
- To improve the ToA estimation across all the propagation channels with known / unknown direction of arrival. And also to further develop the proposed technique to jointly estimate the direction of arrival and ToA.

Page Intentionally
Left Blank

.

Author's Publications

[1] Syed Naveen Altaf Ahmed, P. K. Meher and A. P. Vinod, "Efficient Cross-Correlation Algorithm and Architecture for Robust Synchronization in Frame-based Communication Systems", in *Circuits, Systems, and Signal Processing,* Springer Publications, ISSN: 1531-5878, doi: 10.1007/s00034-017-0678-3, Nov. 2017.

[2] Syed Naveen Altaf Ahmed, A. P Vinod and P. K. Meher, "Fractional Time Delay ToA Estimation Using Short Preambles For OFDM-WLAN", Submitted to *Wireless Communications and Mobile Computing*, Wiley Journal.

[3] S. N. Altaf Ahmed, P. K. Meher and A. P. Vinod, "A low-complexity spectrum sensing technique for cognitive radios based on correlation of intra-segment decimated vectors," *2012 IEEE International Conference on Communication Systems (ICCS)*, Singapore, 2012, pp. 443-447, Jun. 2012.

[4] Syed Naveen Altaf Ahmed, P. K. Meher and A. P. Vinod, "Fast acquisition and time synchronization of frequency hopping burst signals," in *IEEE International Conference on Signals and Systems (ICSIGSYS 2017)*, Bali, Indonesia, May 2017.

[5] Syed Naveen Altaf Ahmed, P. K. Meher and A. P. Vinod, "An Efficient Spectrum Sensing Technique for Burst Signals Detection Based on Correlation and FFT", *2017 IEEE TENCON, IEEE Region Ten Conference*, Penang, 2017, pp. 684 – 688, Nov. 2017.

Page Intentionally
Left Blank

References

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," in *IEEE Personal Communications*, vol. 6, no. 4, pp. 13-18, Aug 1999.

[2] A. S. Margulies and J. Mitola, "Software defined radios: a technical challenge and a migration strategy," *1998 IEEE 5th International Symposium on Spread Spectrum Techniques and Applications - Proceedings. Spread Technology to Africa (Cat. No.98TH8333)*, Sun City, 1998, pp. 551-556 vol.2, 1998.

[3] Q. Zhao and B. M. Sadler, "A Survey of Dynamic Spectrum Access," in *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 79-89, May 2007.

[4] J. Mitola, "Cognitive radio for flexible mobile multimedia communications," *Mobile Multimedia Communications, 1999. (MoMuC '99) 1999 IEEE International Workshop on*, San Diego, CA, 1999, pp. 3-10, Nov. 1999.

[5] "DARPA: The Next Generation (XG) Program." http://www.darpa.mil/ato/programs/xg/index.htm.

[6] "TV White Space Regulatory Framework", Infocomm Development Authority of Singapore, *Getting Connected: TV White Space Workshop*, Boracay, Philippines, May 2015.

[7] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt Generation / Dynamic Spectrum Access / Cognitive Radio Wireless Networks: A Survey," *Computer Networks Journal (Elsevier)*, Vol. 50, no. 13, pp. 2127–2159, Sep. 2006.

[8] R. N. Prashob, A. P. Vinod and A. K. Krishna, "An adaptive threshold based energy detector for spectrum sensing in cognitive radios at low SNR," *2010 IEEE International Conference on Communication Systems*, Singapore, 2010, pp. 574-578, Nov. 2010.

[9] B. Shent, L. Huang, C. Zhao, Z. Zhou and K. Kwak, "Energy Detection Based Spectrum Sensing for Cognitive Radios in Noise of Uncertain Power," *2008 International Symposium on Communications and Information Technologies*, Lao, 2008, pp. 628-633, Oct. 2008.

[10] Y. Zeng, Y. C. Liang and R. Zhang, "Blindly Combined Energy Detection for Spectrum Sensing in Cognitive Radio," in *IEEE Signal Processing Letters*, vol. 15, no. , pp. 649-652, Oct. 2008.

[11] B. Farhang-Boroujeny, "Filter Bank Spectrum Sensing for Cognitive Radios," in *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1801-1811, May 2008.

[12] F. Sheikh and B. Bing, "Cognitive Spectrum Sensing and Detection Using Polyphase DFT Filter Banks," *2008 5th IEEE Consumer Communications and Networking Conference*, Las Vegas, NV, 2008, pp. 973-977, Jan. 2008.

[13] K. G. Smitha and A. P. Vinod, "A Multi-Resolution Fast Filter Bank for Spectrum Sensing in Military Radio Receivers," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 7, pp. 1323-1327, July 2012.

[14] M. Narendar, A. P. Vinod, A. S. Madhukumar and A. K. Krishna, "A tree-structured DFT filter bank based spectrum sensor for estimation of radio channel edge frequencies in military wideband receivers," *10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)*, Kuala Lumpur, 2010, pp. 534-537, May 2010.

[15] A. Sahai and D. Cabric, “Spectrum sensing: Fundamental limits and practical challenges,” in *Proc. IEEE Int. Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Tutorial, Baltimore, MD, USA, Nov. 2005.

[16] H. S. Chen, W. Gao and D. G. Daut, "Signature Based Spectrum Sensing Algorithms for IEEE 802.22 WRAN," *2007 IEEE International Conference on Communications*, Glasgow, 2007, pp. 6487-6492, Jun. 2007.

[17] Z. Tian and G. B. Giannakis, "A Wavelet Approach to Wideband Spectrum Sensing for Cognitive Radios," *2006 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, Mykonos Island, 2006, pp. 1-5, June 2006.

[18] Y. L. Xu, H. S. Zhang and Z. H. Han, "The Performance Analysis of Spectrum Sensing Algorithms Based on Wavelet Edge Detection," *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, Beijing, 2009, pp. 1-4, Oct. 2009.

[19] Shiann-Shiun Jeng, Jia-Ming Chen, Hong-Zong Lin and Chen-Wan Tsung, “Wavelet-Based Spectrum Sensing for Cognitive Radios using Hilbert

Transform", in *World Academy of Science, Engineering and Technology*, Issue: 75, pp. 596 – 600, Mar. 2011.

[20] A.V. Dandawate, and G. B. Giannakis, "Statistical tests for presence of cyclostationarity," in *IEEE Trans. Signal Processing*, vol. 42, no. 9, pp. 2355 – 2368, Sept. 1994.

[21] O. A. Y. Ojeda and J. Grajal, "Sensitivity Analysis of Cyclostationarity-Based and Radiometric Detectors for Single-Sensor Receivers," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 1, pp. 27-43, Jan. 2012.

[22] J. Lunden, V. Koivunen, A. Huttunen and H. V. Poor, "Collaborative Cyclostationary Spectrum Sensing for Cognitive Radio Systems," in *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4182-4195, Nov. 2009.

[23] H. Sadeghi, P. Azmi and H. Arezumand, "Optimal multi-cycle cyclostationarity-based spectrum sensing for cognitive radio networks," in *2011 19th Iranian Conference on Electrical Engineering*, Tehran, 2011, pp. 1-6, May 2011.

[24] Al-Habashna, A., Dobre, O.A., Venkatesan, R., Popescu, D.C., "Second-Order Cyclostationarity of Mobile WiMAX and LTE OFDM Signals and Application to Spectrum Awareness in Cognitive Radio Systems," in *IEEE Journal of Selected Topics in Signal Processing*, Vol: 6, Issue: 1, pp. 26 – 42, Feb. 2012.

[25] A. Al-Habashna, O. A. Dobre, R. Venkatesan and D. C. Popescu, "WiMAX Signal Detection Algorithm Based on Preamble-Induced Second-Order Cyclostationarity," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Miami, FL, 2010, pp. 1-5, Dec. 2010.

[26] A. Al-Habashna, O. A. Dobre, R. Venkatesan and D. C. Popescu, "Cyclostationarity-Based Detection of LTE OFDM Signals for Cognitive Radio Systems," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Miami, FL, 2010, pp. 1-6, Dec. 2010.

[27] A. Al-Habashna, O. A. Dobre, R. Venkatesan and D. C. Popescu, "Second-Order Cyclostationarity of Mobile WiMAX and LTE OFDM Signals and Application to Spectrum Awareness in Cognitive Radio Systems," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 1, pp. 26-42, Feb. 2012.

[28] T. Bourgeois and Y. Sanada, "Cyclostationarity-based detector for OFDM signals with correlated pilot subcarriers," in *2011 The 14th International Symposium on*

Wireless Personal Multimedia Communications (WPMC), Brest, 2011, pp. 1-5, Oct. 2011.

[29] IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Redline," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007) - Redline* , vol., no., pp.1-5229, March 2012.

[30] Y. Zeng, and Y.-C. Liang, "Spectrum-Sensing Algorithms for Cognitive Radio Based on Statistical Covariances," in *IEEE Trans. Vehicular Technology*, vol. 58, no. 4, pp. 1804 - 1815, May 2009.

[31] Y. Zeng, and Y.-C. Liang, "Eigenvalue-Based Spectrum Sensing Algorithms for Cognitive Radio," in *IEEE Trans. Communications*, vol. 57, no. 6, pp. 1784 – 1793, Jun. 2009.

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

[33] D. Simunic and T. S. Dhope, "Hybrid detection method for spectrum sensing in cognitive radio," in *2012 Proceedings of the 35th International Convention MIPRO*, Opatija, 2012, pp. 765-770, May 2012.

[34] Wang Jun and Bi Guangguo, "Novel autocorrelation based spectrum sensing methods for cognitive radios," in *2010 16th Asia-Pacific Conference on Communications (APCC)*, Auckland, 2010, pp. 412-417, Oct.-Nov., 2010.

[35] Z. Zhao, G. Zhong, D. Qu and T. Jiang, "Cyclostationarity-based spectrum sensing with subspace projection," in *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, Tokyo, 2009, pp. 2300-2304, Sept. 2009.

[36] Sharma, R.K., Wallace, J.W., "Improved Spectrum Sensing by Utilizing Signal Autocorrelation," in *2009 IEEE 69th Vehicular Technology Conference*, Barcelona, Spain, 2009, pp. 1 – 5, Apr. 2009.

[37] T. Ikuma and M. Naraghi-Pour, "Autocorrelation-Based Spectrum Sensing Algorithms for Cognitive Radios," in *2008 Proceedings of 17th International*

Conference on Computer Communications and Networks, St. Thomas, US Virgin Islands, 2008, pp. 1-6, Aug. 2008.

[38] Tugnait,J.K., "On autocorrelation-based multi-antenna spectrum sensing for cognitive radios in unknown noise", in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 2944 – 2947, May 2011.

[39] Simon, M. K., Omura, J. K., Scholtz, R. A., & Levitt, B. K., "Spread spectrum communications handbook", revised ed., McGraw-Hill, 1994.

[40] R. L. Peterson, R. E. Ziemer, and D. E. Borth, "Introduction to Spread Spectrum Communications", Prentice Hall, Inc., 1995.

[41] John Litva and Titus Kwok-Yeung Lo, "Digital Beamforming in Wireless Communications", Artech House, Norwood, MA, 1996.

[42] I. Chlamtac and S. Kutten, "A spatial reuse TDMA/FDMA for mobile multi-hop radio networks," in *Proceedings of IEEE INFOCOM*, 1985.

[43] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," in *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477-486, Aug 2002.

[44] "DirecNet™ Overview", *The Open Group DirecNet Task Force*, An Industry Government Alliance Developing an Open Standard enabling Highly Mobile, Directional, High Data Rate, Ad-Hoc, Mesh Communications Networks.

[45] M. Speziali, S. Bregni, M. Magarini and A. Spalvieri, "A new approach to post-FFT synchronization for DVB-T receivers," in *2016 8th IEEE Latin-American Conference on Communications (LATINCOM)*, Medellin, 2016, pp. 1 – 6, Nov. 2016.

[46] EN 300 744 – "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television," ver. 1.6.1. ETSI, Jan. 2009.

[47] Ramjee Prasad, "OFDM for wireless communications systems", Artech House; 2004.

[48] W. D. Warner and C. Leung, "OFDM/FM frame synchronization for mobile radio data communication," in *IEEE Transactions on Vehicular Technology*, vol. 42, no. 3, pp. 302-313, Aug 1993.

[49] J. Bingham, “Method and apparatus for correcting for clock and carrier frequency offset, and phase jitter in multicarrier modems,” U.S. Patent 5206886, Apr. 27, 1993.

[50] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," in *IEEE Trans. Communications.*, vol. 45, no. 12, (1997-12), pp. 1613-1621, Dec. 1997.

[51] Pei Xiao, Colin Cowan, T.Ratnarajah and Anthony Fagan, "Time synchronization algorithms for IEEE 802.11 OFDM systems," in *IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2009)*, Shanghai, (2009-12), pp. 287-290, Dec. 2009.

[52] Gul, M.M.U., Sungeun Lee and Xiaoli Ma, "Robust synchronization for OFDM employing Zadoff-Chu sequence," in *2012 46th Annual Conference on Information Sciences and Systems (CISS)*, (2012), pp. 1-6, Mar. 2012.

[53] Gul, M.M.U., Xiaoli Ma and Sungeun Lee, "Timing and Frequency Synchronization for OFDM Downlink Transmissions Using Zadoff-Chu Sequences," in *IEEE Transactions on Wireless Communications*, Vol.: 14, Issue: 3, (2015-03), pp. 1716-1729, Mar. 2013.

[54] Byungjoon Park, Hyunsoo Cheon, Changeon Kang and Daesik Hong, "A novel timing estimation method for OFDM systems," in *IEEE Communications Letters*, vol. 7, no. 5, pp. 239-241, May, 2003.

[55] Hlaing Minn, Vijay K. Bhargava and Khaled Ben Letaif, "A Robust Training and Frequency Synchronization for OFDM Systems," in *IEEE Trans. on Wireless Communications*, vol. 2, no. 4, (2003-07), pp. 822-839, Jul. 2007.

[56] M. Ruan, M. C. Reed and Z. Shi, "Training symbol based coarse timing synchronization in OFDM systems," in *IEEE Transactions on Wireless Communications*, vol. 8, no. 5, pp. 2558-2569, May 2009.

[57] Y. Guo, G. Liu, I-Tai Lu, J. Ge and Haiyang Ding, "A new time and frequency synchronization scheme for OFDM-based cooperative systems," in *IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014*, Farmingdale, NY, 2014, pp. 1-5, May. 2014.

[58] T. H. Kim and I. C. Park, "Low-Power and High-Accurate Synchronization for IEEE 802.16d Systems," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 12, pp. 1620-1630, Dec. 2008.

[59] H.-Y. Liu and C.-Y. Lee, “A low-complexity synchronizer for OFDM based UWB system,” in *IEEE Trans. Circuits Syst. II*, Brief Papers, vol. 53, no. 11, pp. 1269–1273, Nov. 2006.

[60] C. Nanda Kishore and V. Umapathi Reddy, “A frame synchronization and frequency offset estimation algorithm for OFDM system and its analysis,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, Article ID 57018, pp. 5, Mar. 2006.

[61] Baoguo Yang, K. B. Letaief, R. S. Cheng and Zhigang Cao, "Timing recovery for OFDM transmission," in *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 11, pp. 2278-2291, Nov. 2000.

[62] J. J. van de Beek, M. Sandell and P. O. Borjesson, "ML estimation of time and frequency offset in OFDM systems," in *IEEE Transactions on Signal Processing*, vol. 45, no. 7, pp. 1800-1805, Jul 1997.

[63] J. W. Choi, J. Lee, Q. Zhao and H. L. Lou, "Joint ml estimation of frame timing and carrier frequency offset for OFDM systems employing time-domain repeated preamble," in *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 311-317, Jan. 2010.

[64] H. W. Kim, K. Kang and D. S. Ahn, "ML Estimation of Time and Frequency Offset Using Antenna Diversity in OFDM Systems," in *2008 Digest of Technical Papers - International Conference on Consumer Electronics*, Las Vegas, NV, 2008, pp. 1-2, Jan. 2008.

[65] A. Czylwik, "Synchronization for systems with antenna diversity," in *Gateway to 21st Century Communications Village. VTC 1999-Fall. IEEE VTS 50th Vehicular Technology Conference (Cat. No.99CH36324)*, Amsterdam, 1999, pp. 728-732 vol.2, Sep. 1999.

[66] Meng-Han Hsieh and Che-Ho Wei, "A Low-Complexity Frame Synchronization and Frequency Offset Compensation Scheme for OFDM Systems over Fading Channels," in *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1596-1609, Sep. 1999.

[67] M. Speth, F. Classen and H Meyr, "Frame synchronization of OFDM systems in frequency selective fading channels," in *1997 Proceedings of IEEE 47th Vehicular Technology Conference*, Phoenix, USA, (1997-05), pp. 1807-1811, May 1997.

[68] S.H Muller-Weinfurtner, "Burst frame and frequency synchronization with a sandwich preamble," in *2001 Proceedings of IEEE Global Telecommunications Conference*, San Antonio, USA, (2001-11), pp. 1366-1370, Nov. 2001.

[69] A. Filippi and S. Serbetli, "OFDM symbol synchronization using frequency domain pilots in time domain," in *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, pp. 3240-3248, June 2009.

[70] T. Keller, L. Piazzo, P. Mandarini and L. Hanzo, "Orthogonal frequency division multiplex synchronization techniques for frequency-selective fading channels," in *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 6, pp. 999-1008, Jun 2001.

[71] Y. Mostofi and D.C Cox, "A Robust Timing Synchronization Design in OFDM Systems - Part II: High Mobility Cases," in *IEEE Transactions on Wireless Communications*, vol. 6, no. 12, (2007-12), pp. 4340-4348, Dec. 2007.

[72] Xiaoyan Zhao and Lizhen Cui, "A new frame synchronization algorithm for OFDM WiMAX system in Simulink," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, Harbin, 2011, pp. 2321-2325, Dec. 2011.

[73] G. Ren, Y. Chang, H. Zhang, "Synchronization Method Based on a New Constant Envelop Preamble for OFDM Systems," in *IEEE Transactions on Broadcasting*, 2005, pp. 139-143, Jan 2005.

[74] D. Chu, "Polyphase codes with good periodic correlation properties," in *IEEE Transactions on Information Theory*, vol. 18, no. 4, (1972-07), pp. 531-532, Jul. 1972.

[75] "Technical specification group radio access network; Physical layer aspects for evolved universal terrestrial radio access (UTRA) (Release 7)", Cedex, France, 3GPP TR 25.814, V7.1.0, (2006-09).

[76] L. Deneire, "Avoiding a tower of Babel-Synchronization in Wireless OFDM Systems, How to make them work?", Marc Engels, Editor, Kluwer Academic Publishers, Boston, (2002), pp. 95-111, 2002

[77] W. Yuan, S. Zhiguo and C. Kangsheng, "Robust Time Synchronization Scheme Combating Frequency Offset for IEEE 802.16a OFDM Systems," in *2006 International Conference on Communications, Circuits and Systems*, Guilin, 2006, pp. 1068-1072, Jun. 2006.

[78] C. Putman, S. Rappaport and D. Schilling, "A Comparison of Schemes for Coarse Acquisition of Frequency-Hopped Spread-Spectrum Signals," in *IEEE Transactions on Communications*, vol. 31, no. 2, pp. 183-189, Feb 1983..

[79] S. Rappaport and D. Grieco, "Spread-spectrum signal acquisition: Methods and technology," in *IEEE Communications Magazine*, vol. 22, no. 6, pp. 6-21, June 1984.

[80] S. Rappaport and D. Schilling, "A Two-Level Coarse Code Acquisition Scheme for Spread Spectrum Radio," in *IEEE Transactions on Communications*, vol. 28, no. 9, pp. 1734-1742, Sep 1980.

[81] J. Min and H. Samueli, "Synchronization techniques for a frequency-hopped wireless transceiver," in *Proceedings of Vehicular Technology Conference - VTC*, Atlanta, GA, 1996, vol.1, pp. 183-187, Apr.-May 1996.

[82] Jan-Jaap van de Beek, "Synchronization and Channel Estimation in OFDM Systems", Doctoral Thesis, Lule'a University of Technology, Division of Signal Processing, Lule'a, Sweden, Sep. 1998.

[83] J. P. Ianniello, "High-resolution multipath time delay estimation for broad-band random signals," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 3, pp. 320-327, Mar 1988.

[84] Zhen-Ya He and Li-Hua Li, "High-resolution multipath time delay estimation using a neural network," in *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Ont., 1991, vol. 2, pp. 1469-1472, Apr. 1991.

[85] M. Pallas and G. Jourdain, "Active high resolution time delay estimation for large BT signals," in *IEEE Trans. Signal Processing*, vol. 39, pp. 781–788, Apr. 1991.

[86] L. Dumont, M. Fattouche, and G. Morrison, "Super-resolution of multipath channels in a spread spectrum location system," in *Electron. Lett*., vol. 30, pp. 1583–1584, Sept. 1994.

[87] H. Saarnisaari, "TLS-ESPRIT in a time delay estimation," in *1997 IEEE 47th Vehicular Technology Conference. Technology in Motion*, Phoenix, AZ, 1997, vol.3, pp. 1619-1623, May 1997.

[88] H. Yang, D. Wang and W. Chen, "An Efficient TOA Estimation Using TLS-ESPRIT Based on FBCM Matrix," in *2006 CIE International Conference on Radar*, Shanghai, 2006, pp. 1-4, Oct. 2006.

[89] Lichun Li, Chongsen Ran and Feng Wei, "High resolution time-delay estimation based on wavelet denoising," in *2004 International Conference on Communications, Circuits and Systems (IEEE Cat. No.04EX914)*, 2004, vol.2, pp. 876-879, Jun. 2004.

[90] X. Li and K. Pahlavan, "Super-resolution TOA Estimation with diversity for indoor geolocation," in *IEEE Trans. Wireless Comm.*, vol. 3, no. 1, pp. 224-234, Jan. 2004.

[91] F. Zhao, W. Yao, C. C. Logothetis and Y. Song, "Comparison of Super-Resolution Algorithms for TOA Estimation in Indoor IEEE 802.11 Wireless LANs," in *2006 International Conference on Wireless Communications, Networking and Mobile Computing*, Wuhan, 2006, pp. 1-5, Sep. 2006.

[92] Ziming He, Yi Ma and Rahim Tafazolli, "Improved High Resolution TOA Estimation for OFDM-WLAN Based Indoor Ranging," in *IEEE Wireless Communications Letters*, vol. 2, no. 2, pp. 163 – 166, Apr. 2013.

[93] Joseph Mitola, "Software Radios: Wireless Architecture for the 21st Century," John Wiley and Sons Ltd, United States, 2000.

[94] Rohde, Ulrich, Jerry Whitaker, and Andrew Bateman, "Communications Receivers: DSP, Software Radios, and Design", McGraw-Hill, 2000.

[95] A. Makki, A. Siddig, M.M. Saad, C.J. Bleakley and J.R. Cavallaro, "High-resolution time of arrival estimation for OFDM-based transceivers," in *Electronics Letters*, vol. 51, no. 3, pp. 294–296, Feb 2015.

[96] K. Nur, S. Feng, C. Ling and W. Ochieng, "Application of the Improved FOCUSS for Arrival Time Estimation (IFATE) algorithm to WLAN high accuracy positioning services," in *2012 Ubiquitous Positioning, Indoor Navigation, and Location Based Service (UPINLBS)*, Helsinki, 2012, pp. 1-8, Oct. 2012.

[97] ATSC: "ATSC Digital Television Standard", Doc. A/53 Part 1-6:2007, *Advanced Television Systems Committee*, Washington, D.C., January 2007.

[98] International Communication Union, "Recommendation ITU-R BT.470-6, Conventional Analog Television Systems", 1998.

[99] Theodore S. Rappaport. "Wireless communications: Principles and Practice", Vol. 2. New Jersey: Prentice Hall PTR, 1996.

[100] Yong Soo Cho, Jaekwon Kim, Won Young Yang and Chung G. Kang, "OFDM for Wireless Communications Systems", Wiley-IEEE Press; 1st edition, Nov. 2010.

[101] Weisstein, Eric W. "Exponential Sum Formulas." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/ExponentialSumFormulas.html.

[102] 3GPP TS 36.101, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception", V12.5.0 (2014-11), Release 12, pp. 485-486, Nov. 2014.

[103] J. Fransaer and D. Fransaer, "Fast-Cross-Correlation Algorithm with Application to Spectral Analysis," in *IEEE Transactions on Signal Processing*, vol. 39, no. 9, (1991-09), pp. 2089-2092, Sep. 1991.

[104] Dave Hale, "An Efficient method for computing local cross-correlations of multi-dimensional signals," in *Report of Consortium Project on Seismic Inverse methods for Complex Structures*, Center for Wave Phenomena, Golden, Colorado, CWP-544, (2006-05), pp. 253-260, May 2006.

[105] C. S. Wallace, "A suggestion for a fast multiplier," in *IEEE Trans. Electronic Computers*, vol. EC-13, no. 1, (1964-01), pp. 14–17, Jan 1964.

[106] A.A. Katkar and J.E. Stine, "Modified Booth Truncated Multipliers", in *2004 Proceedings of the 14th ACM Great Lakes symposium on VLSI*, pp. 444-447, Apr. 2004.

[107] Peterson, Roger L., Rodger E. Ziemer, and David E. Borth, "Introduction to spread-spectrum communications," Vol. 995, New Jersey, Prentice Hall, 1995.

[108] Jae Won Kang, Younghoon Whang, Hyo Yol Park and Kwang Soon Kim, "Generalized Cross-correlation Properties of Chu Sequences," in *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp: 438-444, Jan. 2012.

[109] F. Haddadi, M. M. Nayebi and M. R. Aref, "On The Positive Definiteness of Polarity Coincidence Correlation Coefficient Matrix," in *IEEE Signal Processing Letters*, Volume 15, 2008, pp. 73-76, Jan. 2008.

[110] Analog Devices, Wideband RF Transceiver: AD9361, https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms3-ebz, last accessed on 13th Jan. 2017.

[111] Ahmed Makki, Abubakr Siddig, Mohamed Saad, Joseph R. Cavallaro and Chris J. Bleakley, "Indoor Localization Using 802.11 Time Differences of Arrival," in

IEEE Trans. on Instrumentation And Measurement, vol. 65, no. 3, pp. 614-623, Mar. 2016.

[112] C. Hoene and J. Willmann, "Four-way TOA and software-based trilateration of IEEE 802.11 devices," in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, Cannes, 2008, pp. 1-6, Sep. 2008.

[113] Yuan Shen and Moe Z. Win, "On the Accuracy of Localization Systems Using Wideband Antenna Arrays," in *IEEE Trans. On Communications*, vol. 58, no. 1, pp. 270-280, Jan. 2010.We

[114] Roberta Cardinali, Luca De Nardis, Maria-Gabriella Di Benedetto, and Pierfrancesco Lombardo, "UWB Ranging Accuracy in High- and Low-Data-Rate Applications," in *IEEE Trans. On Microwave Theory And Techniques*, vol. 54, no. 4, pp. 1865 - 1875, Apr. 2006.

[115] Y. Shen and M. Z. Win, "Fundamental Limits of Wideband Localization— Part I: A General Framework," in *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 4956-4980, Oct. 2010.

[116] Yiyin Wang, Geert Leus and Alle-Jan van der Veen, "Cramer-Rao bound for range estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009*, pp. 3301-3304, Apr. 2009.

[117] Syed Naveen Altaf Ahmed and Zeng Yonghong, “UWB Positioning Accuracy and Enhancements,” Submitted in *IEEE TENCON 2017, IEEE Region 10 Conference*, Kuala Lumpur, Nov. 2017.

[118] Liang Zhou and M. Saito, "A new symbol timing synchronization for OFDM based WLANs," in *2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No.04TH8754)*, 2004, Vol.2, pp. 1210-1214, Sep. 2004.

[119] Vinko Erceg., “IEEE P802.11 Wireless LANs, TGn Channel Models,” *IEEE 802.11-03/940r4*, May, 2004.

[120] Y. Shen and M. Z. Win, "Fundamental Limits of Wideband Localization— Part I: A General Framework," in *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 4956-4980, Oct. 2010.