# On Detection of OFDM Signals for Cognitive Radio Applications

Douglas Allan

Digital Signal Processing Enabled
Communications Group

Submitted in partial fulfilment of the degree of
Doctor of Philosophy

October 2019

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Signed: **Douglas Allan**

Date: **October 2019**

# Abstract

As the requirement for wireless telecommunications services continues to grow, it has become increasingly important to ensure that the Radio Frequency (RF) spectrum is managed efficiently. As a result of the current spectrum allocation policy, it has been found that portions of RF spectrum belonging to licensed users are often severely underutilised, at particular times and geographical locations. Awareness of this problem has led to the development of Dynamic Spectrum Access (DSA) and Cognitive Radio (CR) as possible solutions. In one variation of the shared-use model for DSA, it is proposed that the inefficient use of licensed spectrum could be overcome by enabling unlicensed users to opportunistically access the spectrum when the licensed user is not transmitting. In order for an unlicensed device to make decisions, it must be aware of its own RF environment and, therefore, it has been proposed that DSA could be enabled using CR. One approach that has be identified to allow the CR to gain information about its operating environment is spectrum sensing.

An interesting solution that has been identified for spectrum sensing is cyclostationary detection. This property refers to the inherent periodic nature of the second order statistics of many communications signals. One of the most common modulation formats in use today is Orthogonal Frequency Division Multiplexing (OFDM), which exhibits cyclostationarity due to the addition of a Cyclic Prefix (CP). This thesis examines several statistical tests for cyclostationarity in OFDM signals that may be used for spectrum sensing in DSA and CR. In particular, focus is placed on statistical tests that rely on estimation of the Cyclic Autocorrelation Function (CAF). Based on splitting the CAF into two complex component functions, several new statistical tests are introduced and are shown to lead to an improvement in detection performance when compared to the existing algorithms. The performance of each new algorithm is assessed in Additive White Gaussian Noise (AWGN), impulsive noise and when subjected to impairments such as multipath fading and Carrier Frequency Offset (CFO). Finally, each algorithm is targeted for Field Programmable Gate Array (FPGA) implementation using a Xilinx 7 series device. In order to keep resource costs to a minimum, it is suggested that the new algorithms are implemented on the FPGA using hardware sharing, and a simple mathematical re-arrangement of certain tests statistics is proposed to circumvent a costly division operation.

# Acknowledgements

Firstly, I would like to thank my supervisors Prof. Bob Stewart and Dr. Louise Crockett for giving me the opportunity to study for a PhD in the DSP Enabled Communications (DSPeC) group and for supporting me to continue as a researcher. I would especially like to thank them for always being available for technical discussions, supporting with the successful publication and presentation of my papers and convincing me to persist in completing my PhD thesis. I would also like to thank Prof. Muhammad Imran and Prof. Stephan Weiss for their suggestions on improving the thesis.

Secondly, I would like to thank all of my friends and colleagues in the DSPeC group. I am lucky to have been able to work alongside such a good group of people. Special thanks must go to Kenny, David N and Shawn for the many times they helped me on a technical front and for their friendship throughout the last few years. I would also like to thank my parents, family and friends for their support and encouragement during this PhD. Finally, I must give special thanks to my girlfriend Hannah for her patience and constant support throughout this process, for which I am very grateful.

# Contents

# List of Figures

# List of Acronyms

| | |
|---|---|
| **3GPP** | 3rd Generation Partnership Project |
| **AWGN** | Additive White Gaussian Noise |
| **AMPS** | Advanced Mobile Phone Service |
| **ATIS** | Alliance for Telecommunications Industry Standards |
| **AM** | Amplitude Modulation |
| **ADC** | Analogue to Digital Converter |
| **ASIC** | Application Specific Integrated Circuit |
| **AGC** | Automatic Gain Control |
| **BPSK** | Binary Phase Shift Keying |
| **BSS** | Blind Source Separation |
| **BRAM** | Block RAM |
| **CFO** | Carrier Frequency Offset |
| **CLT** | Central Limit Theorem |
| **CR** | Cognitive Radio |
| **CPE** | Common Phase Error |
| **CLB** | Configurable Logic Block |
| **CFAR** | Constant False Alarm Rate |
| **CORDIC** | Coordinate Rotational Digital Computer |
| **CAF** | Cyclic Autocorrelation Function |
| **CDF** | Cumulative Distribution Function |
| **CP** | Cyclic Prefix |
| **CSD** | Cyclic Spectral Density |
| **DARPA** | Defence Advanced Projects Agency |
| **DAC** | Digital to Analogue Converter |
| **DSP** | Digital Signal Processing |
| **DVB** | Digital Video Broadcasting |

| | |
|---|---|
| **DVB-H** | DVB - Handheld |
| **DVB-T** | DVB - Terrestrial |
| **DFT** | Discrete Fourier Transform |
| **DSB-AM** | Double Sideband AM |
| **DSA** | Dynamic Spectrum Access |
| **EDGE** | Enhanced Data Rates for GSM Evolution |
| **ETSI** | European Telecommunications Standards Institute |
| **FFT** | Fast Fourier Transform |
| **FCC** | Federal Communications Commission |
| **FPGA** | Field Programmable Gate Array |
| **FIR** | Finite Impulse Response |
| **FSM** | Finite State Machine |
| **FIFO** | First Input First Output |
| **FF** | Flip Flop |
| **FS** | Fourier Series |
| **FDD** | Frequency Division Duplexing |
| **FDMA** | Frequency Division Multiple Access |
| **FM** | Frequency Modulation |
| **FSK** | Frequency Shift Keying |
| **GLRT** | Generalised Likelihood Ratio Test |
| **GPRS** | General Packet Radio Service |
| **GSNR** | Generalised Signal to Noise Ratio |
| **GPS** | Global Positioning System |
| **GSM** | Global System for Communications |
| **GUI** | Graphical User Interface |
| **HDL** | Hardware Description Language |
| **HLS** | High Level Synthesis |
| **HOS** | Higher Order Statistics |
| **HOS** | Higher Order Statistics |
| **HSDPA** | High Speed Downlink Packet Access |
| **HSPA** | High Speed Packet Access |
| **HSUPA** | High Speed Uplink Packet Access |
| **HSUPA** | High Speed Uplink Packet Access |

| | |
|---|---|
| **ISM** | Industrial Scientific and Medical Band |
| **IOB** | Input Output Block |
| **IEEE** | Institute for Electrical and Electronics Engineers |
| **IDE** | Integrated Development Environment |
| **ICI** | Inter Carrier Interference |
| **ISI** | Inter Symbol Interference |
| **IF** | Intermediate Frequency |
| **IDFT** | Inverse Discrete Fourier Transform |
| **JTRS** | Joint Tactical Radio Service |
| **LRT** | Likelihood Ratio Test |
| **LOS** | Line of Sight |
| **LTE** | Long Term Evolution |
| **LTS** | Long Training Symbol |
| **LUT** | Look Up Table |
| **LUT** | Look Up Table |
| **MAC** | Multiply Accumulate |
| **MCM** | Multi Carrier Modulation |
| **MIMO** | Multiple Input Multiple Output |
| **NCO** | Numerically Controlled Oscillator |
| **Ofcom** | Office of Communications |
| **OFDM** | Orthogonal Frequency Division Multiplexing |
| **OSI** | Open Systems Interconnection |
| **PR** | Partial Reconfiguration |
| **PAPR** | Peak to Average Power Ratio |
| **PPDU** | Physical Layer Convergence Protocol Data Unit |
| **PSDU** | Physical Layer Convergence Protocol Service Data Unit |
| **PDF** | Probability Density Function |
| **Pd** | Probability of Detection |
| **Pfa** | Probability of False Alarm |
| **PS** | Processing System |
| **PL** | Programmable Logic |
| **QAM** | Quadrature Amplitude Modulation |
| **QPSK** | Quadrature Phase Shift Keying |

| | |
|---|---|
| **RF** | Radio Frequency |
| **RAM** | Random Access Memory |
| **RAM** | Random Access Memory |
| **RMS** | Root Mean Square |
| **RRC** | Root Raised Cosine |
| **STS** | Short Training Symbol |
| **SFG** | Signal Flow Graph |
| **SNR** | Signal to Noise Ratio |
| **SCFDM** | Single Carrier Frequency Division Multiple Access |
| **SDR** | Software Defined Radio |
| **S$\alpha$S** | Symmetric Alpha Stable |
| **SoC** | System on Chip |
| **TDD** | Time Division Duplexing |
| **TDMA** | Time Division Multiple Access |
| **TACS** | Total Access System |
| **TVWS** | TV White Space |
| **UHF** | Ultra High Frequency |
| **UMTS** | Universal Mobile Communications System |
| **VHF** | Very High Frequency |
| **VHDL** | Very High Speed Integrated Circuit HDL |
| **WCDMA** | Wideband Code Division Multiple Access |
| **WLAN** | Wireless Local Area Network |
| **XDC** | Xilinx Design Constraints |

# Chapter 1

# Introduction

## 1.1 Research Background

One of the most important challenges for the design of future wireless communication technologies will be ensuring that the Radio Frequency (RF) spectrum is used as effectively as possible. In most countries, the management of RF spectrum is the responsibility of government regulatory bodies such as the Office of Communications (Ofcom) in the UK and the Federal Communications Commission (FCC) in the US. The spectrum allocation policy adhered to by these organisations divides the spectrum into licensed and unlicensed bands. Licensed bands are assigned statically to an individual or organisation for their exclusive use over a period of time and across a certain geographical location. Conversely, bands such as the Industrial Scientific and Medical (ISM) band and the TV White Space (TVWS) band can be used by unlicensed devices.

Since the need for the RF spectrum continues to increase, the question of how best to manage and share the spectrum has become a major interest within the research community. It has been demonstrated that spectrum belonging to licensed users is very often underutilised, leading to the existence of so called "spectrum holes". A method that aims to exploit the existence of "spectrum holes" and enable spectrum sharing is the shared-use model for Dynamic Spectrum Access (DSA). In the spectrum overlay version of this model, the band of interest belongs to the primary user and the secondary user aims to opportunistically access the band when it is determined that the primary user is not transmitting, achieving this without causing interference to the primary user. It is hypothesised that this approach could be used to enable sharing of licensed spectrum with unlicensed users, thereby leading to more efficient use of the spectrum. Moreover, this model could equally be applied in unlicensed bands, to ensure that they are shared more efficiently between users.

The key enabler for DSA will be Cognitive Radio (CR); an intelligent radio system that can modify its transmissions based on decisions made about its RF operating environment. The CR will itself be enabled by Software Defined Radio (SDR), which is a radio where the physical layer is implemented purely in software, or using a hardware platform whose functionality can be controlled using software.

One of the most important components of a CR system will be the mechanism that it uses to gain knowledge about its operating environment. A major approach that has been proposed for this is spectrum sensing. Spectrum sensing is the process of determining whether or not a particular frequency band is occupied by a communications signal. Several methods exist to perform spectrum sensing including energy detection, matched filter detection and cyclostationary detection. In recent years, cyclostationary detection has become a popular research area for spectrum sensing in CR due to the fact that a large number of communications signals are cyclostationary in nature. These signals can be categorised as wide-sense cyclostationary, if their second order statistics repeat with an identifiable fundamental period. This is a powerful property to exploit for detection as it is intimately related to the structure of each particular signal, and it can be used to easily distinguish the signal from noise, which is statistically stationary.

A digital modulation method that has seen wide-scale adoption is Orthogonal Frequency Division Multiplexing (OFDM). This modulation scheme has several advantages including spectral efficiency, robust performance in multipath channels and efficient implementation using the Fast Fourier Transform (FFT) algorithm. OFDM is implemented in the physical layer of communications standards such as the IEEE 802.11 family of standards (WiFi), the Digital Video Broadcasting (DVB) standards and 3rd Generation Partnership Project (3GPP) Long Term Evolution (LTE). OFDM is also a candidate to be the basis of the new 5G mobile standard. A fundamental component of the OFDM signal is the Cyclic Prefix (CP), which is added to combat Inter Symbol Interference (ISI) between OFDM symbols and leads to a simplification of the equalisation process in the receiver. The addition of the CP causes the autocorrelation function to repeat with a fundamental period equal to the overall OFDM symbol period, and thus makes it wide-sense cyclostationary. Since it is popular and widely used, developing approaches for cyclostationary detection of OFDM waveforms within the context of DSA and CR is an important and interesting problem.

## 1.2 Research Aims

This thesis aims to present a contribution to the problem stated above by developing new algorithms for the detection of OFDM signals that exploit their inherent cyclostationarity and that can be used for spectrum sensing in a DSA and CR context. More specifically, the algorithms belong to a class of detector that uses a statistical test based on estimation of the Cyclic Autocorrelation Function (CAF) of the received OFDM signal. The proposed detectors are based on a novel technique of splitting the CAF into two complex component functions and are called the Split-CAF Generalised Likelihood Ratio Test (GLRT), Split-CAF Low Complexity, Split-CAF Spatial Sign and Split-CAF Quantised detectors.

The first objective is to derive these detectors from first principles and to assess their performances when compared to existing approaches found in the literature. The algorithms should outperform existing approaches when compared using an appropriate quantitative performance metric. The second objective is to ensure that the algorithms can offer robust detection of OFDM signals in different scenarios apart from the commonly used Additive White Gaussian Noise (AWGN) model, such as in the presence of impulsive noise, and when the transmitted signal is exposed to impairments such as multipath fading and Carrier Frequency Offset (CFO). A third major objective is to guarantee that the algorithms are computationally efficient and as such can be implemented at a reasonable cost on an FPGA. This objective includes ensuring that they are mapped to the FPGA in a manner that avoids consuming hardware resources unnecessarily.

## 1.3 Original Contributions

With the research aims outlined in the previous section in mind, the original contributions to knowledge provided by this thesis can be summarised as follows:

- In Chapter 4, a unifying description and derivation of existing CAF based detectors is provided. This includes verification of their probability distributions under the null hypothesis and an assessment of their performance for detection of IEEE 802.11a/g and DVB-Terrestrial (DVB-T) OFDM signals in AWGN and impulsive noise environments.

- In Chapter 5, a novel technique is suggested where instead of basing the statistical test on estimation of the CAF, it is based on estimation of two complex component functions of the CAF, which are formed using the real and imaginary parts of the autocorrelation lag product respectively. Subsequently, three new detectors are derived: the Split-CAF GLRT, Split-CAF Low Complexity and Split-CAF Spatial Sign detectors. Of these, the Split-CAF Low Complexity detector was published in [1]. The distributions of each new test statistic under the null hypothesis are derived and confirmed in simulation. Each of these detectors are re-derivations of the detectors introduced in Chapter 4. It is shown that the detection performance of each of the proposed algorithms surpasses the existing algorithms in AWGN and impulsive noise environments.

- A new statistical test is proposed which combines a hard quantisation with estimation of the two component functions of the CAF. This detector is called the Split-CAF Quantised detector and it is shown to perform robustly in AWGN and impulsive noise, and to be very efficient in terms of low level resources when implemented on an FPGA.

- A theoretical justification is provided for the performance improvement exhibited by the Split-CAF detectors over their traditional CAF counterparts. In addition, the detection performance of each of the proposed algorithms is assessed for time varying multipath channels and in the presence of CFO.

- In targeting the algorithms for FPGA implementation in Chapter 6, an assessment of their relative costs in terms of low level resources is provided. In order to ensure that hardware resources are not used unnecessarily, it is proposed that hardware sharing should be used when implementing the Split-CAF detectors. Furthermore, a simple mathematical re-arrangement is suggested for test statistics that are expressed as a fraction, in order to circumvent the division operation. This idea was first published in [2].

## 1.4 Thesis Organisation

In Chapter 2, a short review of the background of wireless communications is provided. This includes information on several standards for wireless communications. Following this, the concepts of DSA, CR and SDR are introduced before leading into the mathematical background required to express cyclostationarity. Finally, a short review of FPGA technology is provided.

In Chapter 3, an introduction to Multi-Carrier Modulation (MCM) and OFDM is provided. This includes a review and discussion of multipath propagation effects. Following this, the use of OFDM in the IEEE 802.11a/g and DVB-T standards is discussed.

In Chapter 4, the background to hypothesis testing for signal detection is provided and the cyclostationarity of OFDM due to the CP is established. In particular, the cyclic frequencies of IEEE 802.11a/g and DVB-T OFDM signals are verified. Following this, three cyclostationary detection algorithms found in the literature are derived and their distributions under the null hypothesis are confirmed in simulation. The algorithms are then tested for detection of IEEE 802.11a/g and DVB-T waveforms in AWGN and impulsive noise.

In Chapter 5, the proposed Split-CAF GLRT, Split-CAF Low Complexity, Split-CAF Spatial sign and Split-CAF quantised detectors are derived and their distributions under the null hypothesis are confirmed theoretically and in simulation. These are then tested in detection of IEEE 802.11a/g and DVB-T waveforms in AWGN and impulsive noise. Finally, their performance under radio impairments is assessed.

In Chapter 6, FPGA targeting of the algorithms is considered. The chapter begins by reviewing HDL Coder and Vivado software tools. This is followed by a review of FPGA hardware sharing techniques. The details of the implementation of each of the algorithms in the previous two chapters in HDL Coder is provided. This is followed by an assessment of their relative costs in terms of low level resources on the FPGA. Finally, the performance of each of the HDL Coder implementations are verified.

In Chapter 7, conclusions are drawn and future directions for this research are suggested.

## Chapter 2

# Dynamic Spectrum Access and Cognitive Radio

## 2.1 Introduction

In this chapter, an overview of the physical layer of a wireless digital communications system is provided alongside a brief review of wireless communications standards. Following this, the concepts of SDR, DSA and CR are introduced and the need for spectrum sensing is discussed. Having established the motivation and context for the research problem, the relevant mathematical background for cyclostationary signal processing is described. Finally, a background discussion of FPGAs is given and their potential role in the realisation of DSA and CR is highlighted.

## 2.2 Wireless Communications Background

Wireless communications encompasses any transfer of information between communicating entities that are not physically connected to one another. Wireless telecommunications is achieved through the use of the RF spectrum, which is a section of the electromagnetic spectrum ranging from 3kHz to 300GHz. Early uses of wireless communication via the RF spectrum include broadcast of text using Morse Code and transfer of audio using Amplitude Modulation (AM) and Frequency Modulation (FM). AM and FM are still widely used by commercial radio stations today. Over the years, RF communication has been developed further to incorporate countless other applications such as mobile/cellular communications, satellite communications, broadcasting of television signals, wireless networking between devices such as laptops, computers and tablets and wireless internet access. Another important use of the RF spectrum is radar, which has both commercial and military applications, and the Global Positioning System (GPS) for navigation applications. RF is also used for amateur radio and radio astronomy.

### 2.2.1 Wireless Digital Communications Transceiver

A useful conceptual model for the components in a modern telecommunications system is the Open System Interconnection (OSI) model [3]. This model divides the system into seven distinct layers, as illustrated in Figure 2.1.

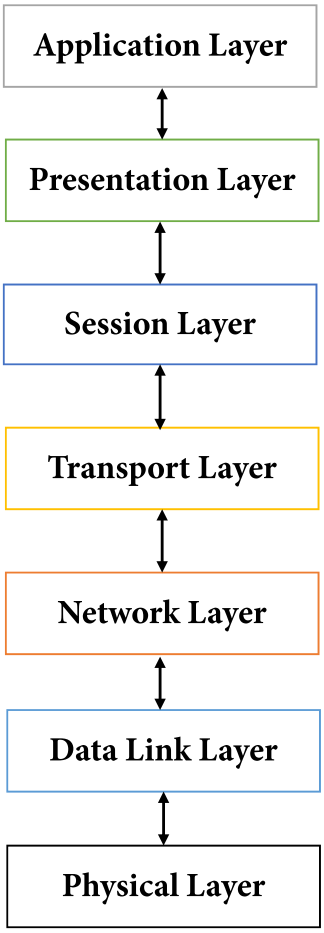


Figure 2.1: Illustration of OSI Stack

In this thesis, focus is placed on the physical layer, which encapsulates all of the functionality required to physically transmit and receive data using an appropriate transmission medium. In this case, digital data is transmitted and received wirelessly using signals in the RF spectrum. Figure 2.2 illustrates the major components of the physical layer of a single-carrier wireless digital communications system, for an architecture known as a baseband digital radio [4]. In this system, all of the digital signal processing is performed at baseband.

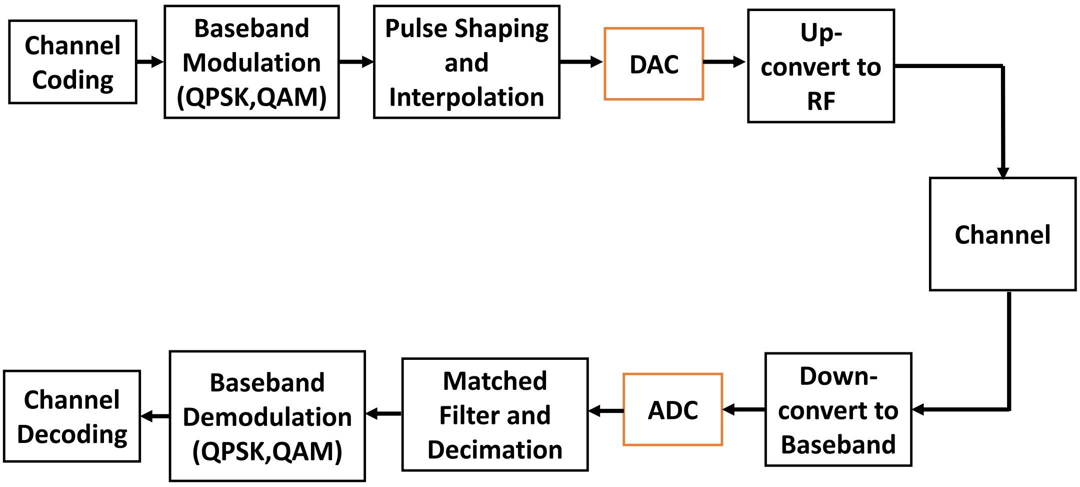

Channel Coding
Baseband Modulation (QPSK,QAM)
Pulse Shaping and Interpolation
DAC
Up-convert to RF
Channel
Down-convert to Baseband
ADC
Matched Filter and Decimation
Baseband Demodulation (QPSK,QAM)
Channel Decoding

Figure 2.2: High level illustration of Digital Communications system

In traditional Double-Sideband AM (DSB-AM), the bandwidth of the signal after modulating the carrier is twice the bandwidth of the baseband signal, leading to an inefficient use of the spectrum. In modern digital communications systems, both the In-phase (I) and Quadrature (Q) components of the carrier frequency are utilised to enable transmission of two independent digital data streams inside the same bandwidth occupied by the DSB-AM signal, thus creating a more spectrally efficient transmission. This architecture is known as digital Quadrature Amplitude Modulation (QAM). In digital communications theory, the notation of complex numbers is employed in order to simplify the mathematics used to describe the QAM architecture [5] [6].

In Figure 2.2, the physical layer processing in the transmitter begins with binary data received from a higher layer in the communications system, that is to be transmitted across the channel. The first stage is to apply channel coding, that is used in the receiver to detect and correct bit errors introduced due to imperfections in the channel [7]. Typical error-correcting codes for channel coding include convolutional codes, Reed-Solomon codes and Turbo codes. After applying the channel coding, the binary data is grouped into blocks of $m$ bits, and each block is mapped to a specific symbol. Using complex notation, the symbol $s[n]$, is represented as,

$$s[n] = s_I[n] + js_Q[n], \tag{2.1}$$

where $s_I[n]$ is the discrete-time in-phase component and $s_Q[n]$ is the discrete-time quadrature component [6]. Since each symbol represents a group of $m$ bits, the number of possible symbols,

$M$, is given by,

$$M = 2^m. \tag{2.2}$$

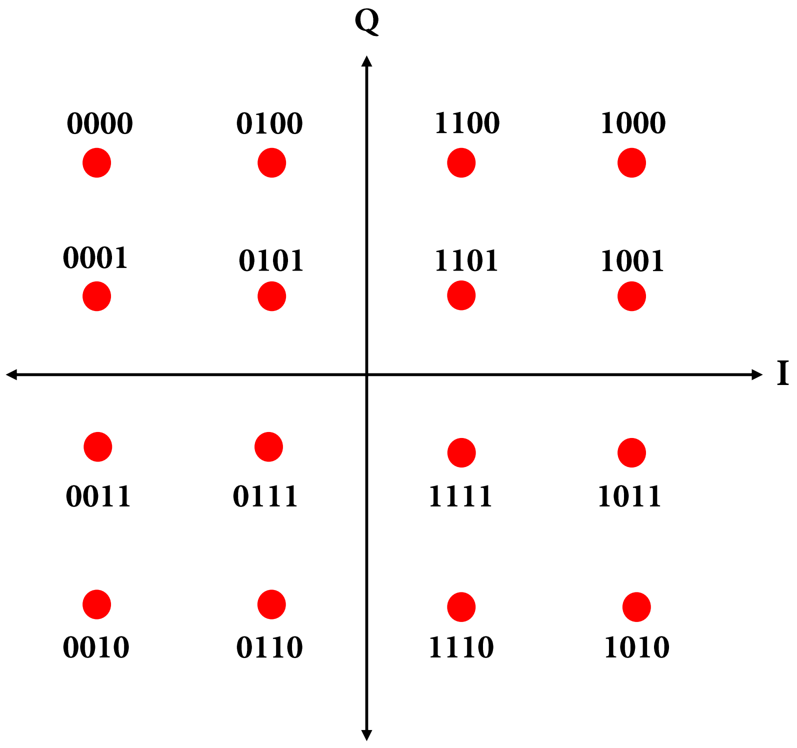


Figure 2.3: Constellation diagram for 16-QAM

There are several common symbol mapping schemes including Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), 16-QAM and 64-QAM. An example of the constellation diagram for 16-QAM is shown in Figure 2.3. In this scheme, each symbol represents a block of $m = 4$ bits, which means there is a total of $M = 16$ possible symbols. Using higher level constellations improves the spectral efficiency of the transmission, by enabling a larger amount of data to be transferred across a fixed bandwidth. The bit-rate is related to the symbol rate as

$$R_b = mR_s, \tag{2.3}$$

where $R_s$ is the symbol rate and $R_b$ is the bit-rate. The symbols are interpolated to a rate that matches the Digital to Analogue Converter (DAC) and passed through a pulse shaping filter to control the signal bandwidth, as shown in Figure 2.2. A popular pulse-shaping filter is the Root Raised Cosine (RRC) filter, whose response is determined by the Roll-off-factor (ROF), $\beta$. The signal bandwidth, $B$, at the output of the pulse shaping filter is given by,

$$B = R_s(\beta + 1). \tag{2.4}$$

Note that $\beta$ is limited in the range 0 to 1. After interpolation and filtering, the digital samples are converted to an analogue signal using the DAC. Finally, the signal is up-converted to RF, either directly, or through an Intermediate Frequency (IF) stage.

In the receiver, the modulation process in the transmitter is reversed as shown in Figure 2.2. The signal is down-converted from RF to baseband and converted to digital format using an Analogue to Digital Converter (ADC). The samples are then passed through a matched filter whose response matches the original pulse shaping filter and decimated to the symbol rate. The symbols are translated into bits through demodulation, and the resulting bits are decoded to recover the original binary data stream. It should be noted that the receiver processing is significantly more complex than illustrated in Figure 2.2, as synchronisation stages are required in order to correct for impairments introduced through the wireless channel.

In general, more complicated digital transmission techniques exist, for which the modulation process is not adequately described by the diagram in Figure 2.2. For example, as will be seen, OFDM involves organising symbols into larger blocks, where each individual symbol in the block modulates a harmonically related sub-carrier. However, the diagram does serve as a high level representation of the physical layer processing for a generic digital communications system. In subsequent chapters, all processing and detection of OFDM signals will be carried out at complex baseband.

### 2.2.2 Wireless Communications Standards

Over the past few decades, a large number of communications standards and protocols have been developed in step with the demands of consumers of wireless technologies. For example, the capabilities offered by mobile and cellular communications standards have increased significantly over the years, in order to cope with demands for increased capacity and data rate.

The first generation of mobile standards or 1G was released in the 1980s, and was called Advanced Mobile Phone Service (AMPS) in the US and the Total Access Communications System (TACs) in the UK and Europe [5] [8]. These systems were the first to employ the cellular concept, wherein a large geographical region is divided into cells, each of which is served by at least one base station. The TACs system was deployed in the band 862-890 MHz and used a Frequency Division Multiple Access (FDMA) scheme with 25kHz wide channels. TACs used FM to provide a basic voice service to users and employed Frequency Shift Keying (FSK) for control channels, operating at a data rate of 8kbps [5] [8].

For second generation (2G) systems onwards, the maintenance and development of mobile communications standards used in Europe is managed largely by the 3rd Generation Partnership Project (3GPP) [9]. 3GPP is a consortium of communications organisations which includes the European Telecommunications Standards Institute (ETSI) and Alliance for Telecommunications Industry Solutions (ATIS). The evolution of the 3GPP standards is illustrated in Figure 2.4.

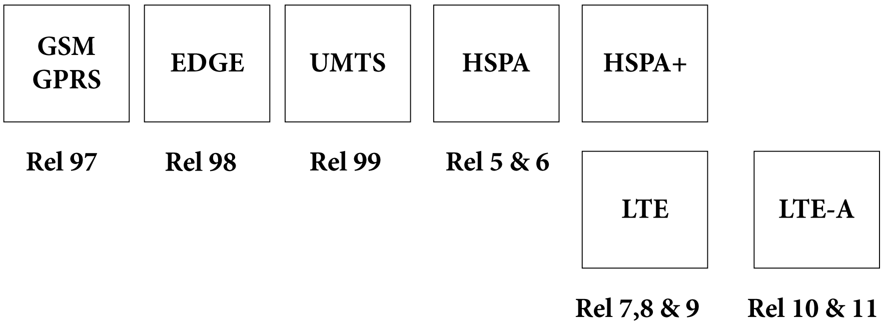


Figure 2.4: Evolution of 3GPP Mobile Standards

2G technologies use digital transmission and operate based on the Global System for Mobile Communications (GSM) standard in the UK. This standard was originally developed by ETSI but is currently maintained as part of 3GPP. The original GSM standard provided the voice service using Time Division Multiple Access (TDMA) and FSK for modulation in the physical layer [5]. Data communications services such as text and multimedia messages were then provided through the packet switched General Packet Radio Service (GPRS) in GSM Release 97, which is managed by the 3GPP [5] [10]. A further improvement on GPRS came with the Enhanced Data Rates for GSM Evolution (EDGE) service [5]. GSM systems are still widely used around the world today.

Third generation systems or 3G again represented a further enhancement of the services offered by 2G. The 3GPP standard is called the Universal Mobile Telecommunications System (UMTS) and it employs Wideband Code Division Multiple Access (WCDMA) at the physical layer. Multiple users transmit on the same frequency simultaneously but are separated by a unique spreading code. UMTS was designed to provide the services offered by 2G systems but also expanded to include services such as video streaming and broadband internet access. UMTS was first specified in 3GPP Release 99 and was further enhanced by the addition of High

Speed Downlink Packet Access (HSDPA) in Release 5 and High Speed Uplink Packet Access (HSUPA) in Release 6. These additions are collectively referred to as High Speed Packet Access (HSPA), as illustrated in Figure 2.4 [9]. A further development came with Evolved High Speed Packet Access (HSPA+) in 3GPP Release 7. Increasing the achievable data rate was realised by employing higher order modulations, Multiple Input Multiple Output (MIMO) technology and carrier aggregation. The amendments were designed to be backwards compatible with legacy UMTS systems (Rel 99) and to use the same frequency bands [9].

The next step in the development of mobile standards came with LTE, which was first specified in 3GPP Release 8 [11]. As with previous generations, LTE was designed to enhance the achievable data rate and capacity offered by existing mobile networks. This was mainly achieved by transferring from an air interface based on WCDMA to one based on OFDM for the downlink and Single Carrier Frequency Division Multiplexing (SC-FDM) for the uplink, and combining these technologies with MIMO techniques. The LTE standard supports both Frequency Division Duplexing (FDD) and Time Division Duplexing (TDD) to separate uplink and downlink channels. LTE, as specified in Release 8, is generally referred to as 3.9G as it does not quite meet the requirements of a 4G technology. However, LTE-Advanced specified in 3GPP Release 10, is more accurately described as a 4G system [9].

Another major developer of communications standards has been the Institute of Electrical and Electronic Engineers (IEEE). The most relevant of these standards is the IEEE802.11 standard for Wireless Local Area Networks (WLAN). This standard is developed and managed by the IEEE Local Area Network/Metropolitan Area Network Standards Committee or IEEE 802. Deployments of WLANs that are based on the IEEE 802.11 standard are more commonly known as WiFi networks. Since it is common for WLAN systems to be based on the IEEE 802.11 standard, the term WiFi is often used to refer to any 802.11 WLAN. WiFi systems have seen deployment on a global scale in settings such as schools, universities, office buildings and people's homes. The OFDM physical layer was first implemented as part of the IEEE 802.11a standard introduced in 1999 for operation in the 5GHz band. This was extended for operation in the 2.4GHz ISM band in the IEEE 802.11g standard. In the"roll-up" document IEEE 802.11-2012 [12], the IEEE 802.11a/g OFDM physical layer is described with enhancements to provide increased throughput under the umbrella of IEEE 802.11n. In IEEE 802.11n, higher throughputs are achieved using several methods such as by exploiting MIMO technology and optionally reducing the length of the CP relative to the length of the data symbol.

This brief review of wireless communications standards has served to highlight the diversity and proliferation of wireless technologies that are in use today. Developments in these technologies have been required to accommodate the need to support larger numbers of users and to provide higher data rates for applications such as video streaming. The coupling of rapid advancement in the capabilities of hardware platforms such as DSPs and FPGAs, in step with Moore's law, and the development of advanced modulation and signal processing techniques such as OFDM and MIMO, have made meeting these demands achievable.

## 2.3 DSA and CR

As illustrated in the previous section, the demand for wireless communications services has led to the development of a wide range of technologies and standards. However, with this growth, comes the problem of how to ensure that the spectrum is managed and shared in an optimal manner. In the following sections, the concepts of DSA, CR and SDR will be introduced, which may combine to form a solution to this problem.

### 2.3.1 Software Defined Radio

In DSA and CR systems, a major requirement will be that the radio is flexible and reconfigurable, such that it can adjust its operation in response to changes in the RF environment. Consequently, SDR has been proposed as the basis for the implementation of CR [13].

SDR was succintly defined by the IEE and SDR forum as "Radio in which some or all of the physical layer functions are software defined." [14] [15]. This definition can be expanded to a radio whose core physical layer components including coding, modulation, filtering and fundamental operating parameters such as frequency band and transmitter power, are either implemented in software or can be controlled through software [13] [16]. This paves the way for a universal system that can support a diverse range of communications standards and technologies from a single piece of radio hardware [17]. Some candidate platforms for the implementation of SDR include DSPs and FPGAs, whose functionality can be reconfigured after the initial manufacturing stage, unlike Application Specific Integrated Circuits (ASICs).

The idea of an all-purpose SDR platform is not new and first emerged frm the military domain. An early example was the SPEAKeasy system, developed by the Defense Advanced Research Projects Agency (DARPA) [18]. The SPEAKEasy system combined a software defined physical layer with a flexible RF front end to support multi-band operation. The main impetus behind the development of the SPEAKEasy system was that several branches of the military used different and, thus, incompatible communications standards. Therefore, it was difficult for the different sectors to communicate easily in the field. The purpose of the SPEAKEasy system was to design a radio capable of supporting ten distinct standards, on a single programmable device. Another important requirement of the SPEAKEasy system was to allow future updates to support newer standards and features, as and when necessary. A Texas Instruments processor and FPGAs were also used for the radio hardware. Another military SDR system that followed SPEAKEasy was the Joint Tactical Radio System (JTRS).

### 2.3.2 Dynamic Spectrum Access and Cognitive Radio

Currently, the spectrum is controlled by government organisations, such as Ofcom and the FCC. It has been found that the spectrum assigned to licensed users is often underutilised both temporally and geographically. A solution that has been offered to address this problem is DSA.

DSA is a blanket term that encompasses a number of alternative approaches to allow spectrum to be shared efficiently. There are three categories of DSA: common-use, exclusive-use and shared-use [16] [19]. The common-use model refers to an unlicensed band that is shared equally among various users. This type of DSA can be found in the ulicensed ISM band [16]. In the exclusive-use and shared-use models, spectrum users are divided into two categories: primary users and secondary users. The primary user is the licensed user, and thus they hold exclusive rights to the spectrum assigned to them. The secondary user is an unlicensed entity that aims to make use of the spectrum belonging to the primary user. The mechanism by which the secondary users make use of the primary user spectrum differs between the exclusive-use and shared-use DSA models. In the exclusive-use model, the secondary user is given exclusive use of the spectrum for a given amount of time, with the explicit knowledge and cooperation of the primary user. Conversely, in the shared-use model, the secondary user accesses the spectrum without the knowledge of the primary user, provided that they do not appreciably interfere with the operation of the primary user. [16] This model is the most relevant to the work carried out in this thesis, as sensing the primary user band is a core requirement.

There are two major approaches for secondary user transmitters in the shared-use model: spectrum underlay and spectrum overlay. In spectrum underlay, the secondary user transmits at a sufficiently low power such that the generated interference has a negligible impact on the primary user. This method implicitly assumes that the primary user never stops transmitting and thus does not make use of the so-called “spectrum holes” [16]. The alternative approach is spectrum overlay, where the secondary user opportunistically accesses the primary user spectrum when it detects a “spectrum hole” [16]. Inherent in this model is the requirement for the radio to gather information about its operating environment and then to use this information to decide whether or not it is suitable to begin transmitting. A hypothesised enabling technology for DSA is CR [19].

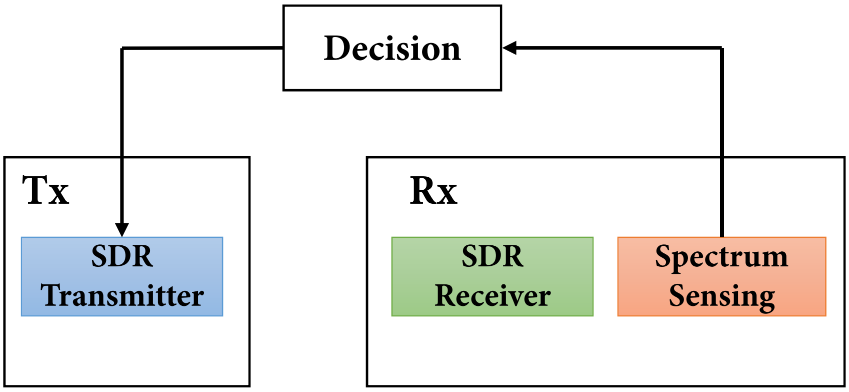


Figure 2.5: High level illustration of components of CR

A CR is a radio system that has awareness of its RF environment. It uses the information it derives from this awareness in order to modify its fundamental operating parameters. These parameters include carrier frequency, transmit power and physical layer components [13] [16]. For example, if a CR device determined that a particular band was in use by the primary user at a given time, it could use this information to modify its carrier frequency, thus allowing it to assess the occupancy of a different band. Since the operation of the CR is influenced by decisions it makes about its environment, it must be reconfigurable by nature. Therefore, CR will itself be based on SDR, since this will be the technology that enables it to self-reconfigure [13]. In this research, the mechanism by which the CR is able to make sense of its environment is of particular interest. The core components of a CR transceiver and their interconnections are illustrated in Figure 2.5. The first component is a multi-standard SDR receiver that is able to facilitate reception of different secondary user signals and the second component is the spectrum sensing module, which allows the CR to make an informed decision regarding its operating environment. This information can then be used to determine the correct time to begin transmitting.

Spectrum sensing in the CR receiver is achieved by sampling a band of interest and determining whether or not it contains a primary user signal. This can be achieved locally by a single CR device or through cooperation between different CR devices, using either a centralised or distributed approach [16]. Some of the most common techniques to achieve spectrum sensing include matched filtering, energy detection and cyclostationary feature detection. Each of these methods will be described in greater detail in Chapter 4, with cyclostationary feature detection being the focus of the research described in this thesis.

Due to the nature of various communications signals, it is often the case that their statistical properties vary periodically. This is known as the cyclostationary property of the signal. In digital communications, the periodicity is usually related to the symbol rate or carrier frequency. This property can be used very effectively for spectrum sensing in a CR device since different signals of interest have unique cyclic features, thus allowing them to be easily identified. It is also the case that noise is statistically stationary, meaning that cyclostationarity is a very useful tool for distinguishing a signal embedded in noise. It is often claimed that using cyclostationary feature detection is disadvantageous due to its computational complexity. However, a major contribution of this thesis will be to show that it can be performed effectively at a very low cost, and thus that it is a viable option for use in the spectrum sensing module of the CR receiver. The necessary statistical background required to understand cyclostationarity will be introduced in the following section.

Another key consideration for the implementation of CR is the chosen hardware platform. Due to their inherent flexibility and re-programmability, FPGAs are a natural choice. Since the CR will likely be based on SDR, it is necessary for the physical layer hardware to be re-configured during operation. This can be achieved using technologies such as Partial Reconfiguration (PR) which enables a portion of the FPGA to be re-programmed dynamically while the remainder continues operating as normal [15]. Since FPGAs are a prime candidate for the implementation of CR, it makes sense to investigate their use for the implementation of spectrum sensing algorithms. Therefore, in Chapter 6, various existing and proposed solutions for cyclostationary feature detection are targeted for implementation using a Xilinx FPGA device and their relative hardware costs are compared. These algorithms are also of interest for other applications aside from spectrum sensing for CR, so prototyping them using FPGAs is a useful step before considering implementation using ASICs. A review of the fundamentals of FPGAs will be given in the final section of this chapter.

## 2.4 Mathematical Background

As mentioned in the previous section, one of the key components of a CR device is the spectrum sensing module, since it will rely on this to gain knowledge about its operating environment. This thesis is concerned with the development and implementation of effective and efficient cyclostationary feature detection algorithms for spectrum sensing. Therefore, in this section, we will introduce the necessary mathematical concepts required in order to gain a deeper understanding of cyclostationary signal processing.

### 2.4.1 CDF and PDF

A random variable is defined as a function that maps the outcome of an experiment to a real or complex number. The abstract space that contains every possible outcome of the experiment is called the sample space and random variables can either be continuous or discrete. A continuous random variable can take on an infinite number of possible values, whereas a discrete random variable can only take on a finite number of distinct values. In this discussion, we will focus on continuous random variables in order to introduce fundamental concepts. Since a random variable is by definition random, it must be characterised statistically. The first metric available to describe a random variable is the Cumulative Distribution Function (CDF), which is defined as follows,

$$F_X(x) = Pr[X \leq x], \tag{2.5}$$

where $Pr[.]$ denotes probability, $X$ corresponds to the random variable and $x$ is the argument passed to the CDF. Intuitively, the CDF gives the probability that a random variable takes on a value less than or equal to the argument passed to it. The Probability Density Function (PDF) is defined as the first derivative of the CDF and is expressed mathematically as,

$$f_X(x) = \frac{d}{dx} F_X(x), \tag{2.6}$$

where $\frac{d}{dx}$ denotes the first derivative. Substituting the value $x$ into (2.6), determines the probability that the random variable will take on a value between $x$ and $x + dx$. The CDF and PDF fully characterise the random variable in a statistical sense and it is sufficient to have either, since it is possible to derive one from the other. For a complex random variable, the PDF and CDF cannot be computed directly, so the behaviour can be characterised by the joint CDFs and PDFs of the real and imaginary parts [20].

### 2.4.2 Moments

The first moment of a random variable is the mean, $\mu$, which is given by,

$$\mu = E[X] = \int_{-\infty}^{\infty} x f_X(x) dx, \tag{2.7}$$

where $E[.]$ is the expected value. In practice, the PDF is not required to compute the expected value since it is well approximated by computing the average. The second moment is the mean square value,

$$E[X^2] = \int_{-\infty}^{\infty} x^2 f_X(x)dx. \tag{2.8}$$

The second central moment or the variance is,

$$\sigma^2 = E[(X - \mu)^2] = \int_{-\infty}^{\infty} (x - \mu)^2 f_X(x)dx. \tag{2.9}$$

This measures the extent to which a random variable varies around its mean value, and is always positive [20]. A larger variance indicates a larger spread of values around the mean. The square root of the variance is referred to as the standard deviation and is denoted as $\sigma$. Note that the second moment and second central moment are equivalent when $\mu = 0$. Higher order moments include skewness and kurtosis. However, these are not relevant to the work carried out in this thesis.

Joint moments can be defined in order to characterise the statistical relationships between sets of random variables. Consider the case where there are two random variables $X$ and $Y$. The second joint moment is called the correlation [20] and is defined as,

$$r_{XY} = E[XY^*]. \tag{2.10}$$

Note the conjugate operator $*$ is included in the definition for the case where the random variables are complex. The covariance between $X$ and $Y$ is defined as,

$$c_{XY} = E[(X - \mu_X)(Y - \mu_Y)^*] = E[XY^*] - \mu_X \mu_Y. \tag{2.11}$$

The covariance is equivalent to the correlation if $\mu_X = \mu_Y = 0$ [20]. Having defined the key tools for statistically characterising a random variable, we can now move on to define a random process.

### 2.4.3 Random Processes

The concept of a random process follows directly from the concept of a random variable, in that it assigns a random signal to the outcome of an experiment. Random processes are fundamental to signal processing and communications theory, as many signals of interest are characterised as random processes. In order to illustrate the concept of a random process, consider an experiment where the outputs of several signal generators are observed simultaneously. In this case, the sample space consists of an ensemble (two shown) of random time-varying signals as in Figure 2.6.

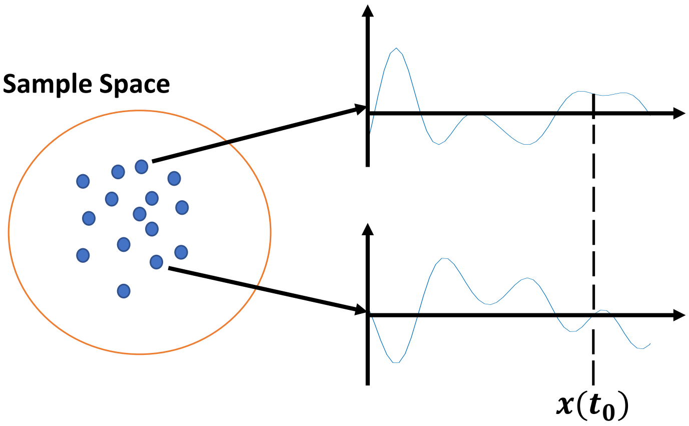


Figure 2.6: Illustration of Random Process

If the output of each signal generator is captured at time $t_0$, there is a value for $x(t_0)$ for each signal in the sample space, as illustrated in Figure 2.6. Therefore, $x(t_0)$ can be viewed as a random variable and the CDF, PDF and moments can be used to describe it statistically. From this definition, it is clear that a random process is simply a collection of random variables, each of which is defined at a particular time instant. The relationships between different random variables of the process (e.g $x(t_0)$ and $x(t_1)$), can be ascertained by calculating their joint PDFs, CDFs and moments. Of particular interest is the autocorrelation function which is defined as,

$$R_{xx}(t,\tau) = E[x(t)x^*(t-\tau)], \tag{2.12}$$

where $\tau$ is called the lag parameter and $x(t)x^*(t-\tau)$ is the autocorrelation lag product. It is clear that the autocorrelation function measures the significance of the relationship between two random variables in the process, separated by a fixed time step $\tau$. A related function is the autocovariance function,

$$C_{xx}(t,\tau) = E[(x(t) - E[x(t)])(x(t-\tau) - E[x(t-\tau)])^*]. \tag{2.13}$$

The autocovariance function equals the autocorrelation function when $E[x(t)] = E[x(t-\tau)] = 0$.

Having defined the concept of a random process, some relevant classifications of random processes can be defined.

### 2.4.4 Stationary Processes

A random process is called stationary if its statistical parameters are time-invariant. There are two forms of stationary process: strict-sense stationary and wide-sense stationary. A process is strict-sense stationary if its $n^{th}$ order joint CDF is independent of time [20],

$$F_X(x(t_0), ..., x(t_n)) = F_X(x(t_0 + \tau), ..., x(t_n + \tau)). \tag{2.14}$$

In other words, the joint CDF of any $n$ random variables drawn from the random process is the same, regardless of the point in time that it is measured. It follows that all joint moments and moments of random variables in the process are also independent of time. A random process is said to be wide-sense stationary if its mean (first moment) and its autocorrelation function (second joint moment) are time-invariant i.e.,

$$\mu(t) = \mu, \tag{2.15}$$

$$R_{xx}(t, \tau) = R_{xx}(\tau). \tag{2.16}$$

As can be observed from (2.16), the autocorrelation function is only dependent on the lag parameter $\tau$ for a stationary process. A prominent example of a stationary process is AWGN. In this case, the process can be described fully using only its first and second order moments.

### 2.4.5 Cyclostationary Processes

As mentioned earlier, it is often the case that communications signals are cyclostationary in nature. This allows signals to be distinguished from background noise, since it is stationary. As with stationary processes, there are two kinds of cyclostationary process: strict-sense cyclostationary and wide-sense cyclostationary. As many communications signals fall into the latter category, it is sufficient to define a wide-sense cyclostationary process for the purposes of this thesis. A process is called wide-sense cyclostationary if its mean and autocorrelation function are periodic with fundamental period $T_0$ as defined below,

$$\mu(t) = \mu(t + T_0), \tag{2.17}$$

$$R_{xx}(t,\tau) = R_{xx}(t+T_0,\tau). \tag{2.18}$$

In this case, the autocorrelation function is dependent on both $\tau$ and $t$, and repeats after a specific interval of $T_0$. In this thesis, we are interested in the periodic nature of the autocorrelation function as this will be exploited for detection purposes in the following chapters. The fundamental cyclic frequency of a cyclostationary process is defined as,

$$\alpha_0 = \frac{1}{T_0}. \tag{2.19}$$

The process will also have harmonically related cyclic frequencies at integer multiples of $\alpha_0$. Since the signal detection algorithms relevant to this problem do not rely on Higher Order Statistics (HOS), the assumption that our signals of interest can be designated wide sense cyclostationary is sufficient.

As the autocorrelation function is periodic, it can be decomposed into a Fourier Series (FS),

$$R_{xx}(t,\tau) = \sum_{n=-\infty}^{\infty} R_{xx}^{n/T_0}(\tau) e^{j2\pi\frac{n}{T_0}t}, \tag{2.20}$$

where the subscript $n$ defines the range of cyclic frequencies, and $R_{xx}^{n/T_0}(\tau)$ is the CAF. The CAF is expressed as,

$$R_{xx}^{n/T_0}(\tau) = \frac{1}{T_0}\int_0^{T_0} R_{xx}(t,\tau) e^{-j2\pi\frac{n}{T_0}t} dt. \tag{2.21}$$

### 2.4.6 Ergodic Processes

Up until this stage, the focus has been placed on random processes, which have been defined as an ensemble of random time varying functions or signals. In order to characterise a random process, statistical parameters are measured across the entire ensemble of signals in the process. However, it is impossible to observe and record the entire random process. In fact, only a single realisation can be captured in a practical situation.

An ergodic process is defined as a process whose ensemble statistics can be reliably estimated from observation of a single realisation of the process. As the observation interval increases, so too does the reliability of the estimates of the statistical parameters. Consider estimating the autocorrelation function of a wide-sense stationary process using an average in time as,

$$\hat{R}_{xx}(t,\tau) = \frac{1}{T}\int_0^{T} x(t)x^*(t-\tau) dt, \tag{2.22}$$

where $\hat{R}_{xx}(t,\tau)$ denotes an estimate of the true autocorrelation function $R_{xx}(t,\tau)$. If $\hat{R}_{xx}(t,\tau)$ converges asymptotically to $R_{xx}(t,\tau)$ in a mean-square sense, then the random process is said to be autocorrelation ergodic [20], and thus a reliable estimate of the autocorrelation function can be made from observation of a single realisation of the process. For a wide-sense cyclostationary process, an estimate of the true CAF can be computed as,

$$\hat{R}_{xx}^{n/T_0}(\tau) = \frac{1}{T_0}\int_0^{T_0} x(t)x^*(t-\tau)e^{-j2\pi\frac{n}{T_0}t}dt. \tag{2.23}$$

Again, if $\hat{R}_{xx}^{n/T_0}(\tau)$ converges asymptotically to $R_{xx}^{n/T_0}(\tau)$ in a mean-square sense, then the process can be called cyclic autocorrelation ergodic. It is a key assumption for all of the algorithms described in this thesis, that the signals we wish to detect belong to the family of wide-sense cyclostationary and cyclic autocorrelation ergodic random processes, and thus that we can identify them by computing (2.23).

## 2.5 Overview of FPGA Technology

In this final section, the fundamentals of FPGA technology will be reviewed, since they are a prime candidate for the implementation of SDR and CR systems. Also, each of the cyclostationary detection algorithms introduced in Chapters 4 and 5 are targeted for FPGA implementation in Chapter 6.

An FPGA is a special form of integrated circuit that consists of an array of low level logic blocks and interconnections that can be re-programmed after manufacture. This stands in opposition to an ASIC, which has a fixed function and cannot be re-programmed. FPGAs are popular because of their re-programmability, capacity for parallel processing and low power consumption. For these reasons, FPGAs are ideally suited for embedded Digital Signal Processing (DSP) applications. They are also very useful for prototyping of specialist hardware, that will subsequently be implemented in an ASIC. There are two major manufacturers of FPGAs, Xilinx and Intel and, for the purposes of this thesis, focus will be placed on devices manufactured by Xilinx.

### 2.5.1 FPGA Programming

An FPGA design is defined using a Hardware Description Language (HDL) such as Very High Speed Integrated Cicruit HDL (VHDL) or Verilog. Aside from using VHDL or Verilog directly, it is also possible to generate an FPGA design using a tool operating at a higher level of abstraction, from which the VHDL or Verilog code is generated. Common high level tools include High Level Synthesis (HLS) [21], Xilinx System Generator [22] and MathWorks HDL Coder [23]. In HLS, the design is implemented in C and C++ and the VHDL code is generated from this. Xilinx System Generator is a tool which comprises a library of blocks that are used inside the Simulink environment in MATLAB. Simulink is a block based tool which is ideally suited for modelling and simulation of signal processing systems. HDL Coder is a similar tool to Xilinx System Generator, except that it is more vendor agnostic and can be used to target both Xilinx and Intel devices. HDL Coder will be described in more detail in Chapter 6, where it is used to implement the cyclostationary detection algorithms.

### 2.5.2 FPGA Architecture and Design

The state of the art Xilinx devices are the UltraScale+ and UltraScale devices, and each have Virtex and Kintex variants [24] [25]. These devices were preceded by the 7 Series devices [26] which consist of four families: Spartan-7, Artix-7, Kintex-7 and the Virtex-7. In Chapter 6, an Artix-7 FPGA will be used for targeting the hardware designs. Xilinx also manufactures System-On-Chip (SoC) technologies in the form of Zynq devices which include the Zynq UltraScale+ MPSoC [27] and the Zynq-7000 [28]. The Zynq-7000 devices incorporate a tightly coupled ARM processor and an FPGA from either the Artix-7 or Kintex-7 families [15]. These architectures allow designs to be partitioned effectively into hardware and software components and the devices incorporate highly effective data transfer between the ARM processor or the Processing System (PS) and the FPGA or Programmable Logic (PL).

The main building block found on a Xilinx 7 series FPGA device is called a Configurable Logic Block (CLB). The vast majority of the FPGA consists of an array of CLBs that can be connected together via special programmable interconnections. The array of CLBs is known collectively as the FPGA fabric. A CLB is further comprised of two logic slices as illustrated in Figure 2.7.

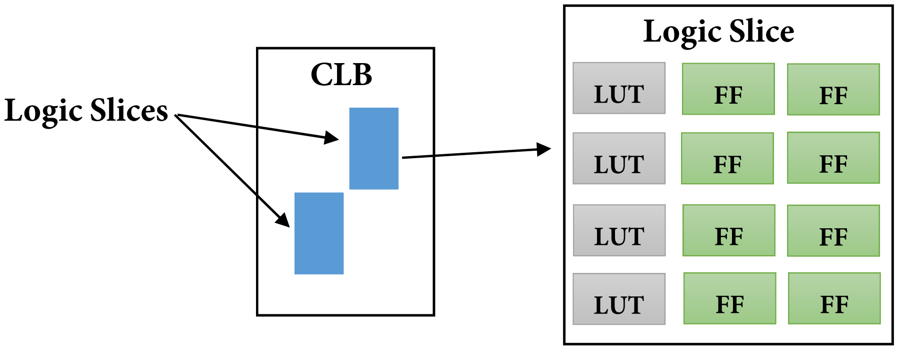


Figure 2.7: Illustration of Xilinx CLB

The slice itself contains Look up Tables (LUTs) and Flip-Flops (FFs) as in Figure 2.7, which are the fundamental logic blocks that enable the synthesis of combinatorial and sequential circuits. Each LUT can be configured to implement a particular combinatorial boolean function by storing the output for each possible combination of its inputs. In this way, the LUT provides a very efficient method of implementing potentially very complex boolean functions. The LUTs are combined to implement larger and more complicated functions. LUTs can also be configured to implement memories such as Read Only Memory (ROM), and Random Access Memory (RAM) which permits reading and writing operations. For the 7 series FPGAs, each logic slice contains four six-input LUTs as shown in Figure 2.7. Sequential logic is defined as logic with state (i.e. memory) and the fundamental building block of sequential logic is the FF. FFs are used for fundamental tasks in FPGA design such as implementing delays in digital filters, pipeline registers and Finite State Machines (FSMs). In 7 series devices, each slice contains eight FFs. Other logic blocks include multiplexers and carry-logic. A switching matrix lies adjacent to each CLB, and controls routing within the CLB and facilitates its connection to other resources on the FPGA. Finally, Input/Output Blocks (IOBs) serve as the external interface and are the conduit through which data passes into and out of the FPGA.

Aside from the logic fabric, there are two other major resources of interest: the DSP48E1 slice and the Block RAMs (BRAMs). The DSP48E1 is a special purpose processing element optimised for high speed arithmetic in FPGAs. It is especially useful for DSP applications and a simplified illustration of its architecture is shown in Figure 2.8.

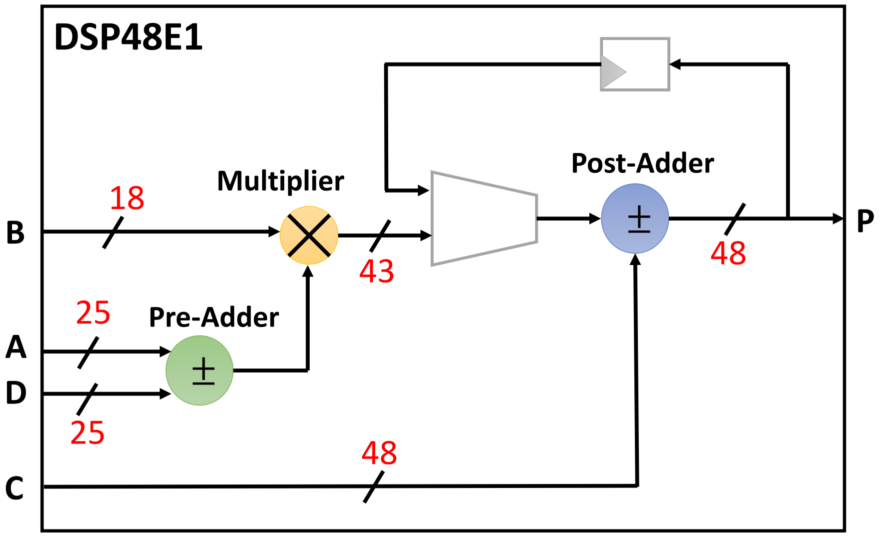


Figure 2.8: Illustration of Xilinx DSP48E1

The three major components are a pre-adder, highlighted in green in Figure 2.8, a multiplier highlighted in yellow and a post-adder highlighted in blue. The DSP48E1 get its name from the fact that the output fixed point wordlength is 48 bits. A fundamental application of the DSP48E1 is implementation of the Multiple Accumulate (MAC) operation needed for Finite Impulse Response (FIR) filters and other DSP algorithms. Each DSP48E1 can be used to implement a single coefficient of the FIR filter (or two if coefficient symmetry is exploited), and they can be cascaded together to implement the entire filter. Note that the pre-adder is included precisely to exploit the coefficient symmetry that exists for linear phase FIR filters. However, DSP48E1s are a scarce resource on low cost FPGAs, and so it often of interest to optimise a design so as to minimise their use.

BRAMs are specialist memory blocks that can be used to implement ROM, RAM and First Input First Output (FIFO) buffers. On 7 series devices, each BRAM can store 36Kb of data where $K$ is 1024 bits. The default word length of the data stored in a BRAM is 18 bits, meaning that the BRAM comprises 2048 memory elements in this configuration. However, the number of memory elements in the BRAM is dynamic and can be altered depending on the wordlength of the data that is to be stored. If a significant amount of memory is required in the implementation of a design, then BRAMs should be targeted, since the alternative is to use Distributed RAM which is implemented using LUTs. For high memory applications, using Distributed RAM can lead to a high resource cost in terms of the FPGA fabric and is likely to lead to a slower design.

When designing for FPGAs, it is normally of interest to optimise the design based on one of three metrics: area, speed or power consumption. Area optimisation refers to optimising such that the design consumes as few resources as possible. Speed optimisation refers to ensuring that the design can be clocked at as high a rate as possible, and this dictates the maximum rate at which data can be processed. Finally, optimising for power consumption means ensuring that the design consumes as little power as possible. This could be achieved by making sure particular elements of the design process data at a low rate. This can sometimes be achieved by applying clever DSP such as using the polyphase decomposition, when designing interpolation and decimation filters.

## 2.6 Chapter Summary

This chapter started with a review of a generic wireless digital communications architecture as a precursor to a discussion of some common wireless standards. Due to the wide deployment of wireless technologies and standards, it is important to make sure that the RF spectrum is used as efficiently as possible. Therefore, DSA, CR and SDR, were presented as approaches which could be used to enable better sharing of spectrum among users. Following this, cyclostationary detection algorithms were introduced as a promising approach for spectrum sensing in DSA and CR systems. In particular, the fundamental mathematical background of cyclostationary random processes was reviewed as a basis for understanding these algorithms. Finally, FPGAs were identified as a candidate platform for the implementation of future CR systems. Subsequently, the fundamentals of Xilinx FPGAs and design flows were described as necessary background for the work described in Chapter 6.

## Chapter 3

# Orthogonal Frequency Division Multiplexing

### 3.1 Introduction

This chapter introduces some of the fundamental principles and applications of OFDM modulation. The OFDM modulation format has become ubiquitous in modern wireless standards, and so it is likely that primary users of the spectrum in the considered DSA scenario will employ OFDM in the physical layer. Therefore, it is of interest to develop efficient spectrum sensing methods for OFDM signals.

In the first portion of this chapter, multipath channels are described as a motivation for the development of OFDM modulation. Following this, an overview and mathematical treatment of OFDM is provided and some practical issues such as timing and frequency synchronisation and channel estimation/equalisation are highlighted. The next section briefly describes how OFDM has been applied as part of wireless communications standards such as IEEE 802.11a/g and DVB-T, as these signals are used as a basis for testing the detection algorithms introduced in the coming chapters.

### 3.2 Multipath Propagation

OFDM has been designed to offer robust performance in multipath channels and, as such, the fundamentals of these channels will be reviewed briefly. In a typical wireless channel, the transmitted signal will be reflected, refracted and scattered by objects before it reaches the receiver [6]. The signal observed at the receiver in a wireless channel consists of a combination of delayed and attenuated copies of the transmitted signal which arrive through multiple paths. Each distinct signal path in the wireless channel is called a multipath component and is associated with either a single or multiple scatterers [5] [6]. If the transmitter and/or receiver is moving, then the attenuation and delay associated with each multipath component varies with time. The constructive and destructive interference of the multipath components leads to fading of the transmitted signal [5] [6]. In modelling multipath channels, we are concerned with the multipath components that can be resolved at the receiver. As will be seen, the resolvability of multipath components at the receiver is intimately related to the characteristics of the transmitted signal. In practice, each resolvable component will consist of several unresolvable components [6].

From a time domain point of view, the wireless channel is characterised by the channel delay spread, $d_s$. This can be defined loosely as the difference in time between the arrival of the first multipath component and the last multipath component. From a frequency domain point of view, the channel is characterised by the coherence bandwidth, $B_c$. This is defined as the bandwidth over which the channel effects can be said to be effectively correlated. $d_s$ and $B_c$ are approximately inversely proportional,

$$B_c \approx \frac{1}{d_s}. \tag{3.1}$$

There are two forms of small-scale fading: flat fading and frequency-selective fading. Flat fading occurs when the symbol duration, $T_{sym}$, is significantly larger than the delay spread of the channel i.e $T_{sym} > d_s$ or equivalently when the signal bandwidth is significantly smaller than the coherence bandwidth i.e. $R_s < B_c$. Note that $T_{sym}$ and $R_s$ are inversely proportional [5] [6]. In a flat fading channel, the signal is not spread significantly in time, meaning that all copies generally arrive within the duration of a single symbol. As a result, the various multipath components are not resolvable and the receiver perceives them as a single path. The unresolvable components add constructively and destructively and cause amplitude and phase distortion in the received signal. Due to limited spreading in time, there should be very little interference between the current data symbol and any previously transmitted symbols. In the frequency domain, the fading effect is approximately correlated across all frequencies of the signal, since the signal bandwidth is small compared to $B_c$. In a time varying channel, the amplitude and phase distortion associated with the combination of the unresolvable components will vary randomly [6].

Frequency-selective fading occurs when the symbol duration is small compared to the delay spread of the channel i.e. $T_{sym} < d_s$ or when the signal bandwidth is large compared to the coherence bandwidth i.e. $R_s > B_c$ [5] [6]. In the time domain, the signal has been spread in time by a duration longer than the symbol period. In this scenario, it is possible to resolve at least some of the various multipath components at the receiver and more significant interference will occur between the current data symbol and previous data symbols due to the time spreading. This phenomenon is called Inter Symbol Interference (ISI) and it can have a very detrimental effect on the performance of the wireless communications system. In the frequency domain, component frequencies of the signal that are separated by a value of $B_c$ or greater will experience uncorrelated fading effects.

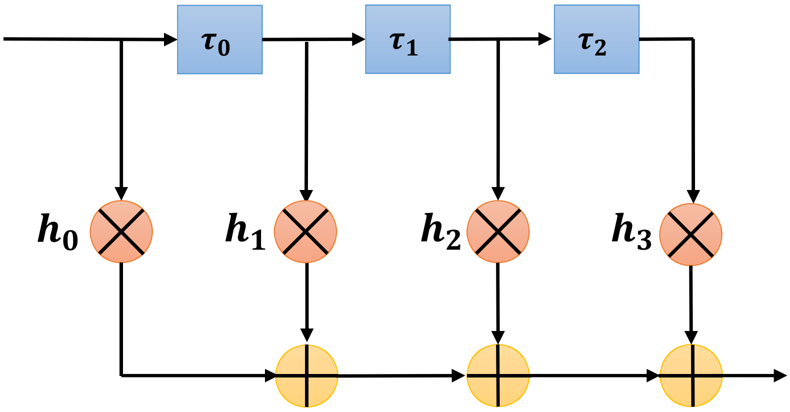


Figure 3.1: Illustration of FIR Multipath Channel

Multipath propagation effects are modelled in software as an FIR filter, as shown in Figure 3.1. Each coefficient $h$ represents a discrete multipath component and $\tau$ represents the delays of the various paths. In a flat fading scenario, the channel consists of a number of unresolvable multipath components and as such this can be modelled by an FIR filter with a single coefficient. For a frequency selective channel, the FIR filter consists of multiple coefficients. However, a filter with multiple coefficients can still be a flat fading channel if the bandwidth of the signal is small in comparison to $B_c$. The coefficients of the FIR filter are expressed as complex numbers since each multipath component introduces a scaling and phase shift. If there is no significant Line of Sight (LOS) component, the coefficients are drawn from a complex normal distribution with a zero mean. Consequently, the magnitudes of the multipath filter coefficients follow a Rayleigh distribution and the phases are uniformly distributed. Multipath channels with no LOS component are referred to as Rayleigh fading channels. Conversely, if a LOS component exists, the coefficients are drawn from a complex normal distribution with a non-zero mean. This means that the magnitudes of the coefficients are Rician distributed and the phases are uniformly distributed. This type of multipath channel is called a Rician fading channel.

The time varying nature of the channel is caused by relative motion between the transmitter and the receiver and by movement of objects in the signal path. There are two consequences of the time varying nature of the channel; the multipath channel coefficients change with time and frequency spreading occurs. The Doppler shift is the change in the frequency of a signal as a result of relative movement. The maximum Doppler shift for a multipath channel is calculated as follows,

$$f_d = \frac{v f_c}{c}, \tag{3.2}$$

where $f_d$ is the maximum Doppler shift, $v$ is the velocity of the transmitter/receiver, $f_c$ is the carrier frequency and $c$ is the velocity of the propagating electromagnetic wave. There are two classifications for the time-varying multipath channel: slow fading and fast fading. These are defined based on the channel coherence time, $T_c$, and the Doppler spread, $D_s$, which is measured in Hertz. $T_c$ is the length of time over which the multipath channel coefficients can be considered invariant. Each of the multipath components will experience different Doppler shifts depending on angle of arrival and velocity, so Doppler spread characterises the range of Doppler shifts for the multipath channel. The Doppler spread is related to $f_d$ as,

$$D_s = 2f_d. \tag{3.3}$$

The channel coherence time and Doppler spread are approximately inversely proportional [5],

$$T_c \approx \frac{1}{D_s}. \tag{3.4}$$

A multipath channel is described as fast fading if the coherence time is small compared to the symbol duration i.e. $T_c < T_{sym}$ or equivalently if the signal bandwidth is small compared to the Doppler spread i.e. $D_s > R_s$. Conversely, a channel is said to be slow fading if the channel coherence time is large compared to the symbol duration i.e. $T_c > T_{sym}$ or the signal bandwidth is large compared to the Doppler spread, $R_s > D_s$. When modelling multipath channels in software, two of the most important parameters to specify are the desired delay spread and the maximum Doppler shift.

## 3.3 OFDM Modulation

OFDM is a special case of MCM that is noted for its resilience to ISI, its relatively simple requirements in terms of equalisation and its ease of implementation using DSP. It has become a dominant technology in modern communications systems, such as in the IEEE802.11 WLAN family of standards, the DVB standards and the 3GPP LTE standard for mobile communications. It is also the basis of the 5G New Radio (NR) standard.

## 3.4 Overview of MCM

OFDM is an example of MCM that involves splitting the available bandwidth into a number of parallel sub-channels. As with any digital communications system, the data bits are linearly mapped into symbols according to a modulation scheme such as BPSK or QAM. However, instead of transmitting using a single carrier, the data symbols are allocated to and transmitted across a number of parallel sub-carriers. The number of sub-carriers, $N_s$, is chosen such that the bandwidth of each sub-channel is small compared to $B_c$. This is equivalent to ensuring the the symbol duration on each sub-channel, is large compared to $d_s$. This process ensures that the individual signals on each sub-channel experience flat fading and limited ISI [5].

## 3.5 Overview of OFDM

As noted previously, OFDM is a special case of MCM that is spectrally efficient and is implemented using an established DSP algorithm. The OFDM system employs a set of sub-carriers that are spaced in frequency at intervals of $1/T_u$, where $T_u$ is referred to as the useful OFDM symbol duration. In discrete-time, the useful symbol duration is normalised by the sampling frequency of the OFDM system and is denoted as $N_u$. The frequency spacing is the smallest spacing that maintains orthogonality between the sub-carriers. In the frequency domain, the sub-channels overlap, leading to a very efficient usage of the available spectrum [5] [29].

The first stage in generating an OFDM signal involves dividing the input symbol stream into blocks of $N_s$ symbols. Each of the $N_s$ symbols modulates one of a set of harmonically related sub-carriers, and the individual signals are multiplexed together to form the useful OFDM symbol. This process can be described mathematically in discrete time,

$$x[n] = \sum_{k=0}^{N_s-1} X[k] e^{j\frac{2\pi nk}{N}}, \tag{3.5}$$

where $k$ is the sub-carrier index, $X[k]$ represents the underlying BPSK/QAM symbols, and $n = 0, 1, 2....N_s - 1$ is the sample index. Note that $N_s$ in (3.5) is equivalent to $N_u$. The OFDM modulation process is mathematically equivalent to the Inverse Discrete Fourier Transform (IDFT),

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi nk}{N}}, \tag{3.6}$$

where $N$ is the total number of points in the IDFT, $k$ is the IDFT bin index and $X[k]$ are the frequency domain samples to be transformed to the time domain. This shows that OFDM modulation and demodulation can be performed efficiently using an IDFT/DFT pair. Furthermore, if $N$ is made equal to a power of two, then modulation and demodulation can be performed using the FFT. The FFT is a computationally efficient algorithm for computing the Discrete Fourier Transform (DFT), and is very suitable for hardware implementation in FPGAs and other devices.

Having generated the useful OFDM symbol, the next stage in the modulation process is the addition of a CP. The CP consists of a portion of the useful OFDM symbol that is added to the front of the symbol. The primary purpose of the CP is to mitigate ISI between OFDM symbols, due to the time spreading introduced by the multipath channel. The length of the CP in relation to the useful OFDM symbol is chosen based on the expected delay spread of the channel. However, due to the fact that the CP is redundant and carries no information, its length must be minimised to ensure that the transmission is as efficient as possible. Once the CP is added to the useful OFDM symbol, the full OFDM symbol is created as shown in Figure 3.2.

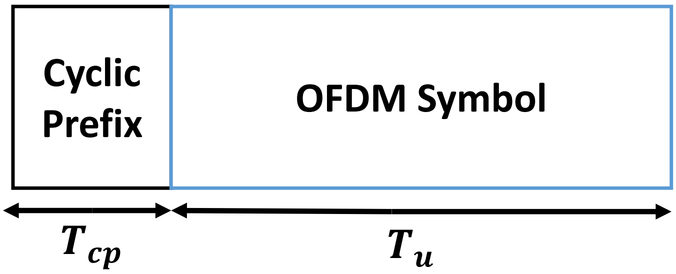


Figure 3.2: Illustration of OFDM Symbol

In the above diagram, $T_{cp}$ refers to the CP duration. The total symbol duration is $T_{ofdm} = T_{cp} + T_u$. In discrete-time, $T_{ofdm}$ and $T_{cp}$ are $N_{ofdm}$ and $N_{cp}$ respectively.

A key benefit arising from the addition of a CP, is that it simplifies equalisation in the receiver [5]. Consider an OFDM symbol that is constructed using $N_s$ sub-carriers in vector form,

$$x = [x[0], x[1], ......x[N_s - 1]]', \tag{3.7}$$

where $'$ denotes transposition. The symbol is passed through a multipath channel and the channel output, $y[n]$, is obtained through linear convolution with the channel impulse response $h[n]$,

$$y[n] = h[n] * x[n] = \sum_k h[k]x[n-k], \tag{3.8}$$

where $*$ denotes convolution. If a CP is appended to the original OFDM symbol, the linear convolution becomes a circular convolution [5],

$$y[n] = \sum_k h[k]x[n-k]_{N_s}. \tag{3.9}$$

Due to the addition of the CP, the OFDM symbol is periodic with $N_s$ or $N_u$. Circular convolution in the time domain is equivalent to multiplication in the frequency domain [30]. Therefore, the received sub-carriers $Y[k]$ are given by the following,

$$Y[k] = X[k]H[k], \tag{3.10}$$

where $H[k]$ are the frequency domain channel coefficients. Therefore, equalisation in the receiver is reduced to estimating $H[k]$ and multiplying by $1/H[k]$ to recover the original sub-carriers.

In summary, the design of OFDM can be understood through the lens of multipath propagation. The wideband channel is divided into a set of sub-channels, whose bandwidth is small compared to $B_c$. This ensures that each sub-channel experiences flat fading and limited ISI. Due to the fact that modulation and demodulation are implemented digitally using an IFFT/FFT pair, OFDM can be efficiently implemented in hardware. It is spectrally efficient since the sub-carriers overlap in the frequency domain and it is highly resistant to ISI introduced by the channel due to the use of a CP. Furthermore, the use of a CP simplifies the receiver processing because it reduces the equalisation process to a single multiplication per sub-carrier. Due to the many benefits associated with OFDM, it has become the backbone of a number of commercial communications standards and technologies.

## 3.6 Synchronisation in OFDM

Although the benefits of using OFDM are clear, it does present some issues, that need to be addressed. Firstly, using a CP adds redundancy to the system and contributes to an overall reduction in the achievable data rate. OFDM also suffers from a large Peak to Average Power ratio

(PAPR) which causes problems in the radio power amplifier. However, of particular importance is synchronisation in the receiver. The orthogonality of the sub-carriers must be maintained in order for the OFDM system to operate properly. Therefore, the next section will provided a brief overview of synchronisation in OFDM.

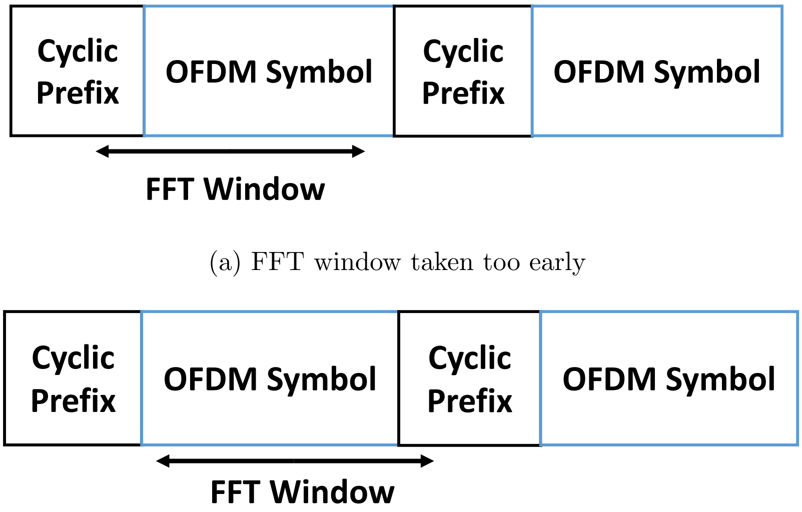
### 3.6.1 Timing offset

Timing offsets are defined as either fractional timing offsets or integer timing offsets as measured in relation to the OFDM system sampling rate [31]. Fractional timing offsets are caused by a mismatch in the sampling phase between the transmitter and receiver sampling clocks. The effect of a fractional timing offset on the OFDM transmission can be understood by recalling the following DFT property [30] [31],

$$x[n-\tau] \xrightarrow{\mathscr{F}} X[k]e^{-j\frac{2\pi k\tau}{N}}, \tag{3.11}$$

where $\tau$ is the normalised timing offset (to the sampling period) and $\xrightarrow{\mathscr{F}}$ denotes Fourier transform. Note that the rotation is dependent on $k$, which is the sub-carrier index in OFDM. Therefore, the phase rotation will be more severe for the outer sub-carriers. However, the timing offset is constant and therefore is unchanged across OFDM symbols. In practice, the rotations introduced by fractional timing offsets are coupled together with multipath channel induced rotations and can be removed through equalisation.

Integer timing offsets cause a misalignment of the FFT window in the receiver [29] [31]. The estimated start of a symbol can occur before or after the ideal starting point. If the estimated starting point is earlier than the actual beginning, then it will generally fall within the CP. If the estimate occurs after the ideal point, then it will fall somewhere inside the OFDM symbol. Both of these scenarios are illustrated in Figures 3.3 (a) and (b).



(a) FFT window taken too early

(b) FFT window taken too late

Figure 3.3: Illustration of integer timing offset

Referring to Figure 3.3 (a), if the symbol start is estimated to begin within the CP, then the result will be a sub-carrier rotation in the frequency domain, as is the case with a fractional offset [29]. However, this only holds true if the CP is longer in duration than the delay spread of the channel plus the timing offset. If this is not the case then IBI will occur, since samples will be taken from a position affected by previous OFDM symbols due to the time spreading of the channel. However, the CP is usually made significantly longer than the delay spread of the channel, to prevent this from occurring [29] [31].

Referring to Figure 3.3 (b), where the symbol start is estimated to begin after the actual starting point, it is seen that the FFT window will include samples from the CP of the following OFDM symbol. This must be avoided since it will lead to IBI and ICI due to violation of the circular comvolution property [29]. However, due to the freedom that is provided through the addition of the CP, a receiver can be designed to ensure that the timing estimate falls into the CP region where no interference from other symbols exists. This will introduce a phase rotation in the frequency domain that can be handled as part of the equalisation process.

### 3.6.2 Carrier Frequency Offset

Due to the fact that an OFDM signal consists of a set of overlapping sub-carriers, it is very susceptible to frequency offsets. Frequency offsets are introduced by a mismatch between the transmitter and receiver local oscillators, and Doppler shift introduced by the time varying multipath channel. The frequency offset means that there is no longer an integer number of cycles of each sub-carrier within the FFT window. This leads to interference between the various sub-carriers, which is known as Inter Carrier Interference (ICI). This can have a detrimental effect on receiver performance, and therefore the offset must be estimated and corrected as accurately as possible. The effect of a frequency offset in the time domain can, again, easily be conceptualised by recalling the following DFT property [30],

$$x[n]e^{-j\frac{2\pi n\epsilon}{N}} \xrightarrow{\mathscr{F}} X[k-\epsilon], \tag{3.12}$$

where $n$ is the time domain sample index and $\epsilon$ is the frequency offset normalised by the sampling frequency $f_s$. The offset, $e^{-j\frac{2\pi n\epsilon}{N}}$, must be estimated and corrected prior to the FFT, in order to mitigate the ICI that will occur.

As is the case with timing offsets, frequency offsets can also be split in to two distinct categories: integer and fractional. In this case, they are measured in relation to the OFDM sub-carrier spacing, $\Delta f$. The time domain correction algorithms that are typically employed to correct for the frequency offset, are constrained by the system parameters, meaning that only a finite range of frequency offsets can successfully be estimated and corrected for. However, constraints are often placed on the amount of oscillator drift that can be tolerated in a system in order to ensure that this limitation does not adversely affect performance. A popular algorithm that can be used for symbol timing and frequency synchronisation in OFDM is the Schmidl and Cox algorithm [32].

### 3.6.3 Phase Noise and Residual Frequency Offset

Phase Noise is a phenomenon caused by the fact that the local oscillator is not a perfect sinusoid. A random phase term is added to the local oscillator that causes the spectrum of the carrier to spread out. It can also be caused by jitter in the sampling clock [29] [31].

The effect of phase noise in OFDM systems is twofold. It can cause ICI (which cannot be corrected), and it introduces a fixed phase rotation to each sub-carrier in the OFDM symbol, whose value changes on a symbol by symbol basis. The fixed rotation is known as Common Phase Error (CPE) [31]. CPE is also introduced due to residual frequency offsets that arise from frequency synchronisation performed prior to FFT demodulation, which is never perfect in a practical system. CPE may have limited effects for the first few OFDM symbols, but will grow steadily worse as time progresses leading to a degradation in system performance.

Fortunately, CPE can be estimated and tracked using the pilot sub-carriers that are embedded in the OFDM symbols. Pilots are redundant sub-carriers that are embedded within the useful OFDM symbol, whose locations are known to the receiver and can be used for phase tracking and channel estimation.

### 3.6.4 Sampling Frequency Offset

It is likely that a mismatch will exist between the transmitter and receiver sampling frequencies. This can be visualised as an offset in sampling phase that changes with time. The effect in OFDM is equivalent to a fractional timing offset, except that it changes on a symbol by symbol basis [33]. This means that the rotations must be tracked, as is the case with CPE. Normally, standardised technologies place limits on the allowable deviation from the ideal sampling frequency to minimise the degradation it causes to performance.

### 3.6.5 Channel Estimation and Equalisation

Although this it is not a synchronisation problem, estimating the fading channel and correcting it is a very important aspect of receiver processing. Recall that the advantage of OFDM is that it divides a wideband channel (i.e. greater than $B_c$) into a number of parallel narrowband (less than $B_c$) sub-channels. This ensures that each sub-channel experiences flat fading and limited ISI. However, the signal is still wideband, meaning that it will be subject to the frequency selective nature of the channel, and different sub-carriers will experience uncorrelated fading [5].

In order to equalise the channel, the complex channel coefficient for each sub-channel must first be estimated. In the majority of systems, this is achieved through the use of dedicated pilot sub-carriers or training symbols. For example, in IEEE 802.11a/g systems, a dedicated training symbol is placed at the beginning of the OFDM frame. The underlying symbol for each sub-carrier is known at the receiver. Therefore, a channel estimate can readily be obtained by dividing the received sub-carriers by the know sub-carriers in accordance with Equation 3.10. One possible approach to equalisation is then to scale each of the sub-carriers in the remaining OFDM symbols by $1/H[k]$, i.e. the inverse of the estimated channel coefficient.

## 3.7 OFDM in Wireless Standards

Due to its various advantages, OFDM has been adopted widely and forms the backbone of various wireless and wired communications systems. These include digital television standards such as DVB-T and DVB-Handheld (DVB-H) [34] and the wireless mobile standard, 3GPP LTE [11]. As described earlier, it is used in WLANs i.e. IEE 802.11a and various amendments to this standard such as IEEE 802.11g and IEEE 802.11n. Furthermore, OFDM is the enabler for a number of newer technologies such as Li-Fi [35]. However, focus will be placed on the application of OFDM in the IEEE 802.11a/g Non-HT and DVB-T standards, as these are most relevant to the work carried out in this thesis.

### 3.7.1 OFDM in IEEE 802.11a/g Non-HT

In this section, a review of the design of the OFDM signal in the IEEE 802.11a/g Non-HT standard specified in the 2012 roll-up document [12] will be reviewed. The OFDM physical layer defines several operating modes that support data communications rates of 6,9,12,18,24,36,48 and 54Mb/s respectively. The OFDM symbol consists of 64 sub-carriers of which 52 are used. The value 64 is chosen as a power of 2 to enable efficient implementation of the modulation and demodulation processes using FFT algorithms. The underlying modulation methods used for each of the sub-carriers are BPSK, QPSK, 16-QAM and 64-QAM. The standard specifies channel coding using convolutional codes with code rates of 1/2, 2/3 and 3/4 respectively. In the fully clocked mode, the baseband sampling rate is $f_s = 20MHz$.

Data is transferred over the air in frames in the IEEE 802.11a/g Non-HT standard. The data sent as part of each frame consists of a PHY header and a payload that contains data sent from the MAC sub-layer in the form of the Physical Layer Convergence Protocol Service Data Unit (PSDU), and tail and pad bits. The data frame is illustrated in Figure 3.4.

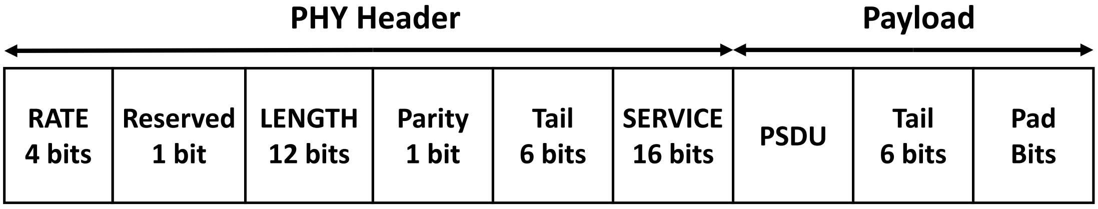


Figure 3.4: Illustration of data frame in IEEE 802.11a/g Non-HT standard

The bits contained within the data frame are transmitted across the channel in a PLCP Protocol Data Unit (PPDU). The format of the PPDU frame is illustrated in Figure 3.5.

| | | |
|---|---|---|
| PHY<br>Preamble | SIGNAL<br>1 SYMBOL | DATA<br>Variable number of<br>OFDM Symbols |

Figure 3.5: Illustration of PPDU Frame in IEEE 802.11a/g Non-HT standard

The first component of the PPDU frame is the PHY preamble. This portion of the frame contains no data and the preamble is composed of repetitions of two training symbols: the Short Training Symbol (STS) and the Long Training Symbol (LTS). According to the standard, the STS should be used for Automatic Gain Control (AGC) convergence, diversity selection, timing synchronisation and coarse frequency synchronisation [29]. The LTS is reserved for channel estimation and fine frequency synchronisation and can be used for timing synchronisation as well.

The first OFDM symbol in the PPDU frame is the SIGNAL symbol. The data transmitted in the SIGNAL field is the first 24 bits of the PHY header, i.e. from the RATE field to the Tail field as seen in Figure 3.4. The useful OFDM symbol is generated using 64 sub-carriers: 48 carry data, 4 are pilots, 11 are null sub-carriers, and one is a DC carrier. A CP of one quarter of the length of the useful symbol, i.e. 16 samples, is then appended to the front. In generating the SIGNAL symbol, the 24 bit string is first passed through through a convolutional encoder with a code rate of 1/2 that generates a 48 bit output. This data is then interleaved using an approach specified in the standard. Each of these bits is then mapped to a BPSK symbol and each symbol is assigned to one of the 48 data carrying sub-carriers. The reason that BPSK and a code rate of 1/2 are used to generate the SIGNAL symbol is that this combination performs the best at low Signal to Noise Ratios (SNRs). This is important as the SIGNAL symbol must be received successfully in order to ensure that the remainder of the PPDU frame can be demodulated and decoded. Of particular importance are the RATE and LENGTH fields which inform the receiver of the underlying modulation method (BPSK, QPSK, 16-QAM, 64-QAM), code rate and length of the frame.

The first part of the data frame to be transferred as part of the DATA portion of the PPDU frame is the SERVICE field. The field comprises 16 bits and is divided into two sections. The first 7 bits are used for the initialisation of the descrambler in the receiver, and the remaining 9 bits are reserved for future use. All of the bits in the SERVICE and Payload fields are passed through a scrambler before the data is sent to a convolutional encoder with generator polynomials g0 = 122 and g1 = 171 and a rate of 1/2. The different code rates are achieved through a method called ‘puncturing’. After the convolutional encoder, the bits are mapped to one of the four possible modulation methods: BPSK, QPSK, 16-QAM and 64-QAM. The resulting complex numbers are divided into blocks of 48 and are mapped to the data carrying sub-carriers of the OFDM symbol. As before, 4 pilot sub-carriers, null sub-carriers and a DC sub-carrier are added to the symbol. Finally, the CP, which is 1/4 the length of the useful symbol, is added at the beginning of the symbol to prevent ISI. Once all data has been mapped to OFDM symbols, the symbols are concatenated to form the DATA portion of the PPDU frame. The PHY preamble and SIGNAL symbols are appended to the front of the DATA field and the formation of the PPDU frame is complete. The final stage in the process is upconversion to RF and transmission over the air [29].

### 3.7.2 OFDM in DVB-T

In this section, the construction of the OFDM signal for the DVB-T standard [34] will be reviewed. The purpose of the DVB-T standard is to provided digital terrestrial television in the Ultra High Frequency (UHF) and Very High Frequency (VHF) bands and the standard specifies three different channel spacings: 6 MHz, 7 MHz and 8 MHz. The DVB-T standard has two configurations: “2K” mode and “8K” mode. An additional “4K” mode is defined for the DVB-H standard but this is beyond the scope of this review.

In DVB-T, several stages are performed to transfer data from the output of the MPEG-2 transport multiplexer, to a suitable format for transmission over the air. In the first stage, the binary data is passed through a scrambler [34]. This is followed by two stages of channel coding and interleaving. In the outer coding and interleaving stages, a Reed-Solomon code is implemented and this is followed by convolutional interleaving. Subsequently, in the inner coding stage, a convolutional code is applied with possible code rates of 1/2, 2/3, 3/4, 5/6 and 7/8. The different code rates are achieved through the puncturing process. The inner interleaving stage consists of bit-wise interleaving and symbol interleaving. Further details of the channel coding and interleaving processes can be found in [34].

The data are mapped to either QPSK, 16-QAM, 64-QAM, non-uniform 16-QAM or non-uniform 64-QAM symbols. The configuration of the OFDM physical layer is different for the “2K” and “8K” transmission modes. In each mode, the signal is transmitted as a frame which comprises 68 OFDM symbols and a group of four individual frames is called a super-frame. For

8MHz channels, the symbols contain 1,705 sub-carriers with an FFT size of 2048 in the "2K" mode and 6,817 used sub-carriers with an FFT size of 8192 in the "8K" mode. The duration of the useful symbol period is $896\mu s$ and $224\mu s$ at $f_s = 9.1429$ MHz and, subsequently, the sub-carrier spacings are 1.116kHz and 4.464kHz respectively. As was the case for the IEEE 802.11a/g standard, pilot sub-carriers are included to facilitate channel estimation and synchronisation at the receiver. Finally, there are four possible CP lengths for each mode: 1/4, 1/8, 1/16 and 1/64. Therefore, in "2K" mode, the CP durations are 56 $\mu s$, 28 $\mu s$, 14 $\mu s$ and 7 $\mu s$ respectively. In "8K" mode, the CP durations are $224\mu s$, $112\mu s$, 56 $\mu s$ and $28\mu s$. In Chapter 4, the structure of the OFDM test signals will comply with the IEEE 802.11a/g and DVB-T standards.

## 3.8 Chapter Summary

In this chapter, the OFDM digital modulation format has been described. OFDM is an example of MCM, which uses a set of orthogonal and overlapping sub-carriers. The main advantage of MCM techniques such as OFDM is that they divide the wideband channel into several narrowband sub-channels, whose bandwidth is less than the coherence bandwidth of the channel. This ensures that each sub-channel is flat fading and experiences limited ISI. It is also advantageous since it can be implemented efficiently using the FFT algorithm and is spectrally efficient due to the fact that the sub-carriers overlap closely.

Because of its many advantages, OFDM has been implemented in the physical layer of several standards including IEEE 802.11a/g, DVB-T and 3GPP LTE. The CP is added to provided immunity to ISI introduced through the multipath channel. Furthermore, having a CP makes the channel equivalent to a circular convolution, which means that equalisation can be achieved using a single tap equaliser after applying the FFT. As will be seen in Chapter 4, the CP can also be used as a unique identifier of the OFDM signal, due to the fact that it causes the autocorrelation function to be periodic.

# Chapter 4

# Cyclostationary Feature Detection for OFDM Signals

## 4.1 Introduction

In future, robust and accurate detection of various waveforms will be required in order to facilitate the implementation of intelligent radio systems. For example, a prime motivation in CR systems is the ability to accurately detect the presence of a primary user waveform, thus allowing the CR to make opportunistic use of the existing RF spectrum. As mentioned previously, OFDM plays a pivotal role in modern communications systems, meaning that reliable and cost effective detection of OFDM signals is an important issue. At the beginning of this chapter, an overview of signal detection as it is applied in an engineering context is provided and some common detection strategies are discussed. It is established that due to the inclusion of a CP, OFDM signals exhibit non-conjugate symbol rate cyclostationarity.

Following this, a review is conducted into solutions for cyclostationary detection that can be found in the literature and three prominent methods that base the detection on estimation of the CAF are introduced. These algorithms are attractive in that they can be shown to exhibit a Constant False Alarm Rate (CFAR) and only require knowledge of the OFDM signal structure and cyclic frequency. In the subsequent sections, each test statistic is derived and their probability distributions under the null hypothesis are confirmed. Having derived each of the algorithms, their performances are compared for the detection of IEEE 802.11a/g Non-HT and DVB-T signals in AWGN and impulsive noise environments.

## 4.2 Overview of Detection Methods

Three prominent methods employed for signal detection purposes are the energy detector, the matched filter and the feature detector [36] [37]. In this section, a general overview of the detection problem as it relates to these different methods will be given. For the purposes of this thesis, the detection problem is formulated as a binary hypothesis test [38]. In digital communications, a possible binary hypothesis test is the following,

$$\begin{aligned} H_0 &: x[n] = w[n], \\ H_1 &: x[n] = s[n] + w[n], \end{aligned} \tag{4.1}$$

where $H_0$ is the null hypothesis, $H_1$ is the alternative hypothesis, $x[n]$ is the received signal, $s[n]$ is the signal of interest and $w[n]$ is background noise. In modern day communications systems, these signals are represented using complex numbers due to the use of QAM receivers. In this situation, $H_0$ states that only noise is received, whereas $H_1$ states that a signal of interest plus noise is received. It is the task of the receiver to determine which of the two hypotheses is true. This is done by computing a test statistic, $\hat{T}$, using the observed data and comparing it to a pre-defined threshold, $\eta$. If $\hat{T} > \eta$, it is decided that $H_1$ is true. Conversely, if $\hat{T} \leq \eta$ then it is decided that $H_0$ is true.

The binary hypothesis test is characterised by two important metrics: the Probability of False Alarm ($P_{fa}$) and the Probability of Detection ($P_d$). These terms are encountered when detection problems are discussed in an engineering context, and originated when the first radar systems were being designed [38]. $P_{fa}$ is a conditional probability and is expressed as follows,

$$P_{fa} = Pr[\hat{T} > \eta | H_0]. \tag{4.2}$$

This quantifies the probability that the test statistic exceeds the threshold given that $H_0$ is true. This represents the case where it is incorrectly decided that $H_1$ is true, when in fact $H_0$ is true. As a rule, it is desirable to minimise $P_{fa}$ as much as possible. Similarly, $P_d$ is also a conditional probability and is expressed as,

$$P_d = Pr[\hat{T} > \eta | H_1]. \tag{4.3}$$

This is the probability that the test statistic exceeds the threshold given that $H_1$ is true. Therefore, this represents the case where it is correctly decided that $H_1$ is true. Unlike $P_{fa}$, it is desirable to maximise $P_d$ as much as possible.

For each of the detection algorithms discussed in this thesis, the probability distribution of the test statistic under $H_0$ is completely known. This knowledge is exploited in order to set a threshold for the detector based on a maximum allowable $P_{fa}$. A formula for computing $\eta$ is derived by re-writing (4.2) as follows,

$$P_{fa} = 1 - Pr[\hat{T} \leq \eta | H_0]. \tag{4.4}$$

Recall that the CDF of a random variable $X$ is defined as $F_X(x) = Pr[X \leq x]$. Taking this into account, (4.4) can be written as,

$$P_{fa} = 1 - [F_{\hat{T}}(\eta) | H_0]. \tag{4.5}$$

If (4.5) is re-arranged and $F_{\hat{T}}^{-1}$ is applied to both sides, a final formula for $\eta$ is [39],

$$\eta = [F_{\hat{T}}^{-1}(1 - P_{fa}) | H_0]. \tag{4.6}$$

For each of the algorithms described in this thesis, $P_d$ is evaluated through simulation, based on the value of $\eta$ that is determined using (4.6). Therefore, the derivation of the probability distributions of the test statistics under $H_1$ is not carried out. The performance of the detectors can be ascertained by evaluating $P_d$ as a function of Signal to Noise Ratio (SNR), as will be seen later in this chapter.

The energy detector is the simplest detector and has the advantage that it requires no explicit knowledge of the signal of interest. The test statistic $\hat{T}$ is computed as,

$$\hat{T} = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2. \tag{4.7}$$

If the value of $\hat{T}$ exceeds $\eta$, then it is decided that a signal is present in the observed data. Although its simplicity is attractive, the energy detector does suffer from some drawbacks. Most importantly, setting $\eta$ is dependent on the noise variance [36] [40] [41]. This has implications for the reliability of the detection since estimation of the noise variance must be performed in the receiver to enable accurate detection. Clearly, estimating the noise variance will be imperfect in a practical system, and so this requirement can have a negative impact on detector performance, i.e. it could lead to an increase in the number of false alarms and/or missed detections. Another problem of this detection scheme is that it cannot actually determine what type of waveform it is detecting, i.e. it has no discriminatory capability. Energy detection schemes have been studied extensively and some example papers are [41], [42], [43], [44].

The matched filter is used to maximise the SNR of the signal at its output [40]. It effectively performs correlation between the received signal and a locally stored copy of the known signal. This correlation is performed using a matched filter by setting the coefficients to a time-reversed and conjugated version of the known signal. It is clear that the matched filter is in opposition to the energy detector, in that it requires intimate knowledge of the transmitted signal. In any practical system, the transmitted signal will not be known at the receiver. Therefore, matched filtering must be performed using pre-defined training sequences and pilots. The matched filter is only optimal for operation in an AWGN channel and can suffer from degradation in more realistic channels, due to multipath effects and lack of synchronisation. This implies that a matched filter may require a prior synchronisation stage to be performed in order to enable accurate detection, which would be undesirable in terms of additional complexity. Some papers that consider matched filter detectors for spectrum sensing include [45], [46], [47].

Feature detectors aim to detect a signal by exploiting the presence of particular features in the transmitted signal. As will be seen, a common approach for the detection of OFDM signals is to exploit the fact that the CP causes the autocorrelation function to repeat with an identifiable fundamental period. Feature detectors can be placed in between the energy detector and the matched filter, in terms of the knowledge that they require about the parameters of the transmitted signal. The remainder of this chapter will consider statistical tests that can be used to reliably detect the presence of cyclostationarity in the received signal.

## 4.3 Detectable Features of OFDM Signals

In an ideal communications system, a signal would consist entirely of data in order to maximise information transfer. However, it is often the case that redundancy must be deliberately added in order to ensure that successful communication can be achieved. In OFDM systems, training symbols and pilots (which carry no data) are added to the signal for the purposes of synchronisation and channel estimation/equalisation in the receiver. Equally, the CP is used to prevent ISI introduced by the dispersive nature of the wireless channel. However, these forms of redundancy can also be exploited as unique identifiers to detect and classify different OFDM systems. Research has been conducted into using different redundancies in OFDM for detection and classification. For example, the authors in [48], [49] and [50], exploit the cyclostationarity that is introduced by the inclusion of pilot sub-carriers in each OFDM symbol. Various researchers, including those in, [39] and [51], have utilised CP-induced cyclostationarity in order to detect and classify OFDM systems. In this thesis, focus is placed exclusively on CP based cyclostationary detection since this is a consistent feature of practical OFDM systems. For ease of reference, Figure 4.1 again illustrates the composition of a typical OFDM symbol.

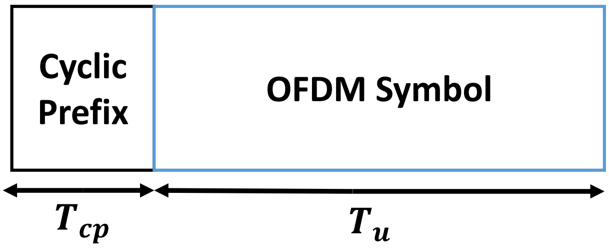


Figure 4.1: Illustration of typical OFDM symbol

As discussed in Chapter 3, the symbol period $T_{ofdm}$ can be broken into two parts: $T_u$ which is the useful OFDM symbol period, and $T_{cp}$ which is the CP duration, i.e the section of the symbol that carries data. The discrete-time equivalents of these are $N_{ofdm}$, $N_u$ and $N_{cp}$, respectively. As described previously, the CP is formed by copying a section of data from the end of the OFDM symbol and appending it to the front. The immediate consequence is that data samples separated by a duration of $N_u$ samples are identical. Based on the assumption of ergodicity, an estimate of the autocorrelation function can be computed in discrete-time from a single realisation of the OFDM process as follows,

$$\hat{R}_{xx}[n,\nu] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]x^*[n-\nu], \tag{4.8}$$

where $n$ is the sample index, $N$ is the period of observation, $\nu$ is the discrete autocorrelation lag and $x[n]x^*[n-\nu]$ is the discrete autocorrelation lag product. When computing the autocorrelation function for an OFDM signal, it is expected that a significant peak should appear at $\nu = N_u$, due to the fact that samples separated by this interval are identical. Figure 4.2 shows the magnitude of $\hat{R}_{xx}[n,\nu]$ at various lags for an IEEE 802.11a/g Non-HT OFDM signal at $f_s$ = 20MHz with $N_{ofdm}$ = 80, $N_u$ = 64 and $N_{cp}$ = 16, as computed in MATLAB. A total of 80 different lags are shown.

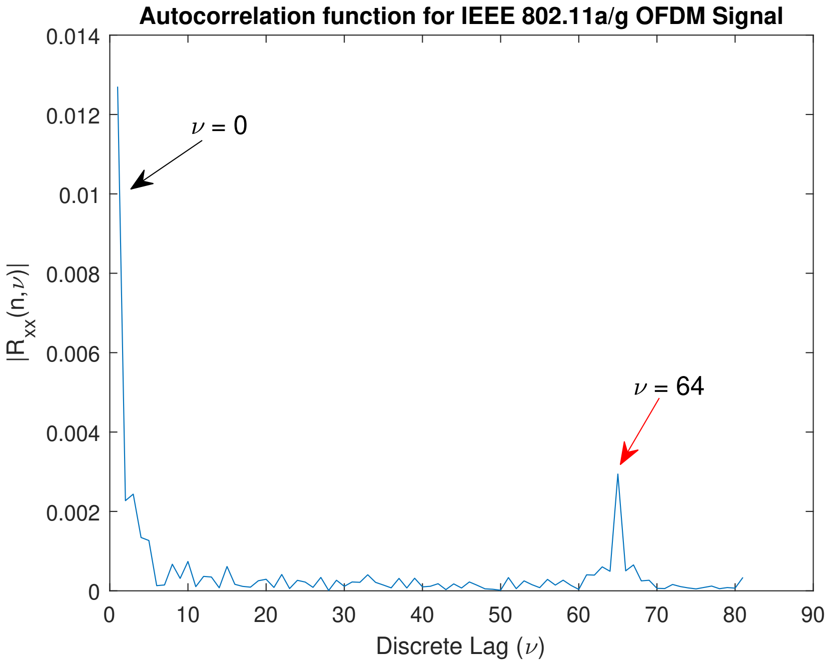


Figure 4.2: Autocorrelation function for IEEE 802.11a/g Non-HT OFDM signal

As indicated by the red arrow in Figure 4.2, a peak can clearly be seen protruding at a lag equal to $\nu = N_u = 64$. Also, notice that there is a peak at $\nu = 0$ marked by the black arrow. Computing (4.8) at $\nu = 0$ gives us an estimate of the power of the OFDM signal as expressed below,

$$\hat{R}_{xx}[n,0] = \frac{1}{N}\sum_{n=0}^{N-1}\left|x[n]\right|^2 . \tag{4.9}$$

Notice that basing a spectrum sensing algorithm on an autocorrelation detector at $\nu = 0$ is equivalent to an energy detector. As a typical transmission consists of multiple OFDM symbols, the correlation between the cyclic prefix and the end of the OFDM symbol repeats regularly. For an OFDM process, the autocorrelation function resembles a periodic pulse train with a fundamental period of $T_0$ samples, where $T_0 = T_{ofdm}$ or equivalently $N_0 = N_{ofdm}$ [39]. Assuming cyclo-ergodicity, an estimate of the discrete-time CAF can be obtained from a single realisation of the process as follows,

$$\hat{R}^{\alpha}_{xx}[\nu] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]x^*[n-\nu]e^{-j2\pi\alpha n}, \tag{4.10}$$

where $\alpha$ is now the digital cyclic frequency, i.e. it has been normalised to the input sampling frequency. Estimating the CAF across a range of discrete cyclic frequencies can be achieved by computing the Discrete Fourier Transform (DFT) of the discrete autocorrelation lag product and scaling the result by a factor of 1/N. This is defined as,

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]x^*[n-\nu]e^{-j2\pi kn/N}. \tag{4.11}$$

It is clear that each DFT bin, $k$, corresponds to an estimate of the CAF at a particular discrete cyclic frequency. In practice, the DFT is computed using the FFT algorithm. Figure 4.3 shows $\left|X[k]\right|$ for the IEEE 802.11a/g signal, where the FFT size is $N = 4096$. Note only the first $N/2$ bins are shown since this corresponds to cyclic frequencies in the range 0 to $f_s/2$. It is clear through visual inspection that several peaks exist. The peaks marked by the red arrows correspond to $\alpha_0 = 1/N_0$ and its two most significant integer multiples or harmonics. In the case of IEEE 802.11a/g at $f_s = 20$MHz, $\alpha_0 = 1/80$. Since $N_0 = N_{ofdm}$, OFDM is said to exhibit non-conjugate symbol rate induced cyclostationarity. It is called non-conjugate cyclostationarity because the autocorrelation lag product includes a complex conjugate in its standard formulation. If the quantity $E[x(t)x(t-\tau)]$ was periodic, then the signal would be said to exhibit conjugate symbol rate induced cyclostationarity. The peak at $\alpha = 0$ is simply (4.8) i.e. an estimate of the autocorrelation function.

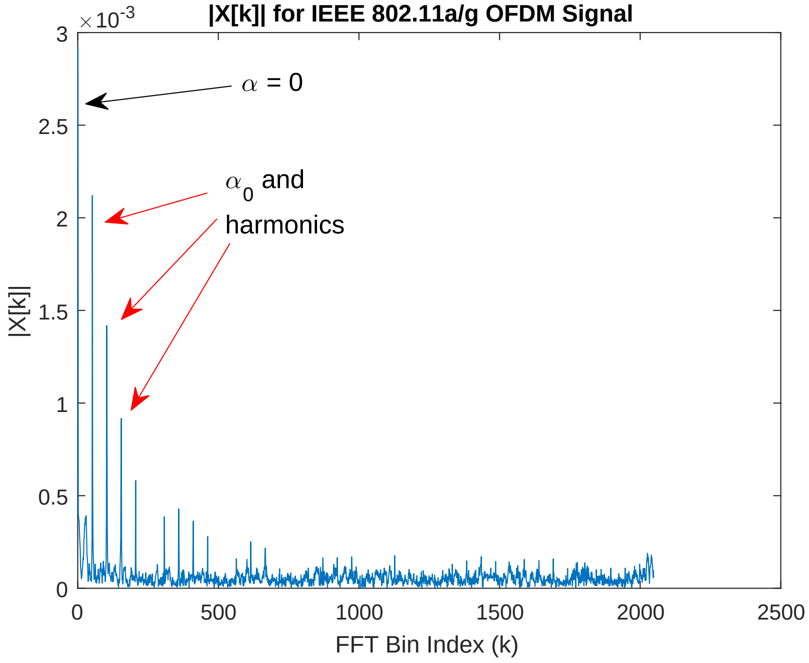


Figure 4.3: $\left|X[k]\right|$ for IEEE 802.11a/g OFDM signal

The standard autocorrelation function can be thought of as a special case of the CAF evaluated at $\alpha = 0$. Strictly speaking, $N$ should be made to equal to an integer multiple of $N_0$ since an FS is being approximated. However, for the FFT algorithm used to generate Figure 4.3, the FFT is computed with a duration that is equal to a power of two. If $N_0$ is a non-integer multiple of the FFT length, then the cyclic frequency will not lie in an exact FFT bin and some spectral leakage will occur. This is the case for the IEEE 802.11a/g signal used to generate Figure 4.3. However, it is still possible to uncover the cyclostationarity of the OFDM signal using an observation interval that is a non-integer multiple of $N_0$. The cyclostationary features introduced through the use of a CP are unique to each separate OFDM system, making them very powerful tools for effective detection and classification. In the next section, the cyclostationarity of DVB-T signals will be discussed as this standard will be used as a test signal later in this chapter.

### 4.3.1 Cyclostationarity of DVB-T signals

As described in Chapter 3, the two operational modes for DVB-T are 2K transmission mode and 8K transmission mode [34]. In 2K mode, $N_u = 2048$, which is equivalent to $T_u = 224\mu s$ at $f_s = 9.1429$MHz. Conversely, in 8K mode, $N_u = 8192$ which corresponds to $T_u = 896\mu s$ at $f_s = 9.1429MHz$. Therefore, in order to detect DVB-T signals based on estimation of the CAF, two parallel detection branches would be required. For the 2K mode branch, the CAF would be computed at $\nu = 2048$ and, for the 8K mode branch, it would be computed at $\nu = 8192$. Furthermore, due to the fact that there are four possible CP lengths in each of the modes, this means that there are four possible cyclic frequencies for each mode. Each cyclic frequency would have to be tested for individually in order to cover all possible configurations of the DVB-T signal. In 2K mode, the four cyclic frequencies are $\alpha_0 = 1/2560$, $\alpha_0 = 1/2304$, $\alpha_0 = 1/2176$ and $\alpha_0 = 1/2112$ respectively. Conversely, in 8K mode, the cyclic frequencies are $\alpha_0 = 1/10240$, $\alpha_0 = 1/9216$, $\alpha_0 = 1/8704$ and $\alpha_0 = 1/8448$. In Chapter 5, the existing and proposed algorithms will be tested for each possible OFDM signal configuration in 2K mode.

## 4.4 Literature Review

There is a vast body of literature covering the theory and application of cyclostationary signal processing for problems as diverse as signal detection, Blind Source Separation (BSS), beamforming and angle of arrival estimation [52] [53]. An excellent summary and tutorial on the background theory of cyclostationarity and its use in the fields of signal processing and communications can be found in the book edited and compiled by Gardner [54]. This book introduces the theoretical foundations of the topic, such as the Cylic Wiener-Khintchine theorem and the concept of spectral correlation. The Cyclic Wiener-Khintchine theorem states that the Fourier transform of the CAF is the Cyclic Spectral Density (CSD) function. Spectral correlation is a useful interpretation of the CAF and the CSD, which expresses them as the cross-correlation and cross-spectral density of frequency-shifted versions of the signal respectively [54] [55]. It also discusses applying cyclostationarity for blind channel estimation and synchronisation in communications receivers. Also, in [56], Gardner establishes the cyclostationarity of several common digital modulation schemes, including PSK and FSK.

In terms of applying cyclostationarity for spectrum sensing in CR and DSA systems, many of the papers are based on either algorithms that were developed in the past for different applications or new algorithms developed specifically for the problem at hand. The vast majority of detection schemes are based on estimation of either the CSD or the CAF. In [57], the optimum multi-cycle CSD detector is derived for operation in AWGN environments. However, this has a major drawback in that it cannot be implemented without phase synchronisation [57] [58]. Similarly, in [59], multi-cycle and single-cycle detectors are derived to operate in non-Gaussian noise environments for surveillance and reconnaissance applications. However, the test statistic requires knowledge of the noise PDF to be implemented practically [58] [59]. These detectors have subsequently been considered for CR applications in, e.g. [58]. In an earlier paper on applying cyclostationary detection in CR, the authors exploit a normalised version of the CSD [60]. Several other papers also base their detection algorithms for CR applications on estimation of the CSD including [61], [62], [63], [64], [65], [66], [67] and [68].

Narrowing down the literature search further, it is of particular interest to focus on detection schemes that are based entirely on estimation of the CAF. This is done with a view to introducing new algorithms in Chapter 5, that are based on splitting the traditional CAF into two component functions, which utilise the real and imaginary parts of the autocorrelation lag product respectively. The constraint associated with using this approach in the detection of OFDM signals is that reasonable knowledge of $\nu$ and $\alpha$ are required. However, all of this information is available if the OFDM symbol structure is known and, for standard waveforms, this knowledge is freely available. In a highly influential paper, the authors introduce GLRTs to detect the presence of cyclostationarity [69]. Note that the GLRT test statistics are derived for both the CAF (time domain) and CSD (frequency domain). However, only the time domain test is of relevance to this work. The time domain GLRT test statistic has been applied in several works including [70], [71], [72], [73] and [74]. Modified versions of these algorithms that differ due to the estimation of the covariance matrix can be found in [75], [76], [2], [39] and [77]. Another interesting detection algorithm that can be used based on an estimation of the CAF is the Low Complexity detector in [78]. Note that the detector is not referred to as the "Low Complexity" detector in the literature. Similar test statistics can be found in [79], [80], [81] and [82]. A final algorithm that has a prominent place in the literature is the Spatial Sign cyclostationary detector, first published for CR applications in [83]. It has been subsequently discussed in [84] and [39]. In [85], the effects of this function on the cyclostationary features of certain signals is studied from a mathematical perspective. Furthermore, in [86], several novel variations of the Spatial Sign cyclostationary detector are suggested, which exhibit a reduced cost in terms of low level resources when targeted for implementation on an FPGA.

In the following sections, due to the fact that each of them feature prominently in the literature, the GLRT, Low Complexity and Spatial Sign detectors will be derived and analysed, and their performances assessed, in detection of IEEE 802.11a/g Non-HT and DVB-T waveforms in both AWGN and impulsive noise environments.

## 4.5 GLRT Detector

The GLRT cyclostationary feature detector was first described in [69]. This test statistic was modified by the authors in [39], [75], [76] and [77] for use in cognitive radio systems. The difference between the original and modified detectors is the method used for calculating the elements of the covariance matrix used in the GLRT test statistic. The analysis in this thesis will be based on the modified algorithms and will only consider cyclostationarity in the second order statistics of the signal, since the feature of interest is the periodicty of the autocorrelation function.

It is clear from the estimated CAFs shown in Figure 4.3, that the bins corresponding to non-cyclic frequencies are non-zero. However, the true CAFs at these frequencies are equal to zero since cyclostationarity is not present. Since a finite observation interval has been used to estimate the CAF, an error component has been introduced with respect to the true value of the CAF, due to the fact that the estimation is imperfect [69]. This is quantified mathematically as,

$$\hat{R}^{\alpha}_{xx}[\nu] = R^{\alpha}_{xx}[\nu] + e^{\alpha}_{xx}[\nu], \tag{4.12}$$

where $e^{\alpha}_{xx}[\nu]$ is the estimation error term and $R^{\alpha}_{xx}[\nu]$ is the true CAF. This error term will tend asymptotically to zero with the observation interval. However, the observation must be restricted to a finite interval for practical reasons. Therefore, in order to determine whether or not a cyclic frequency is present, a statistical test is performed, since simply checking for a non-zero value is insufficient due to the presence of the error term [69]. In order to derive such a test, the authors in [69] begin by arranging the real and imaginary parts of $\hat{R}_{xx}^{\alpha}[\nu]$ in vector form as follows,

$$\hat{\mathbf{r}}_{\mathbf{xx}} = [\Re(\hat{R}_{xx}^{\alpha}[\nu]) \quad \Im(\hat{R}_{xx}^{\alpha}[\nu])]. \tag{4.13}$$

The real and imaginary parts of $R_{xx}^{\alpha}[\nu]$ can also be expressed in vector form,

$$\mathbf{r}_{\mathbf{xx}} = [\Re(R_{xx}^{\alpha}[\nu]) \quad \Im(R_{xx}^{\alpha}[\nu])]. \tag{4.14}$$

The relationship between these two vectors is,

$$\hat{\mathbf{r}}_{\mathbf{xx}} = \mathbf{r}_{\mathbf{xx}} + \mathbf{e}_{\mathbf{xx}}, \tag{4.15}$$

where $\mathbf{e}_{\mathbf{xx}}$ is the estimation error vector, whose elements are the real and imaginary parts of the estimation error term,

$$\mathbf{e}_{\mathbf{xx}} = [\Re(e_{xx}^{\alpha}[\nu]) \quad \Im(e_{xx}^{\alpha}[\nu])]. \tag{4.16}$$

The following binary hypothesis test is formulated,

$$\begin{aligned} H_0 &: \hat{\mathbf{r}}_{\mathbf{xx}} = \mathbf{e}_{\mathbf{xx}}, \\ H_1 &: \hat{\mathbf{r}}_{\mathbf{xx}} = \mathbf{r}_{\mathbf{xx}} + \mathbf{e}_{\mathbf{xx}}. \end{aligned} \tag{4.17}$$

The null hypothesis, $H_0$, states that the estimated CAF consists solely of the error term, covering the case where no cyclostationarity is present, i.e. when $\mathbf{r}_{\mathbf{xx}} = [0\ 0]$. Conversely, the alternative hypothesis,$H_1$, states that the estimated CAF comprises the true CAF plus the error term. In order to formulate a test to decide between each of the hypotheses, the probability distribution of $\hat{\mathbf{r}}_{\mathbf{xx}}$ under each hypothesis is determined. It is proved in [69], that $\hat{\mathbf{r}}_{\mathbf{xx}}$ follows a multivariate normal distribution with two dimensions and that the only difference in this distribution under each hypothesis is the mean. Under $H_0$, the mean is [0 0] and under $H_1$ the mean is $\mathbf{r}_{\mathbf{xx}}$, i.e. the true CAF.

In the case where the probability distributions and all of their parameters are completely known for both hypotheses, the hypothesis test is called ‘simple’. According to the Neyman-Pearson lemma, the optimal strategy to decide between $H_0$ and $H_1$ in the case of a simple hypothesis test, is the Likelihood Ratio Test (LRT) [38]. This can be formulated for the hypothesis test in (4.17) as,

$$\hat{T}_{LRT} = \frac{f(\hat{\mathbf{r}}_{\mathbf{xx}}|H_1)}{f(\hat{\mathbf{r}}_{\mathbf{xx}}|H_0)}, \tag{4.18}$$

where $f$ denotes the PDF. Since $\hat{\mathbf{r}}_{\mathbf{xx}}$ follows a multi-variate normal distribution under each hypothesis, the LRT becomes,

$$\hat{T}_{LRT} = \frac{e^{-1/2(\hat{\mathbf{r}}_{\mathbf{xx}}-\mathbf{r}_{\mathbf{xx}})\mathbf{\Sigma}_{\mathbf{xx}}^{-1}(\hat{\mathbf{r}}_{\mathbf{xx}}-\mathbf{r}_{\mathbf{xx}})'}}{e^{-1/2(\hat{\mathbf{r}}_{\mathbf{xx}})\mathbf{\Sigma}_{\mathbf{xx}}^{-1}(\hat{\mathbf{r}}_{\mathbf{xx}})'}}, \tag{4.19}$$

where $e$ is the base of the natural logarithm, $\mathbf{\Sigma}_{\mathbf{xx}}^{-1}$ is the inverse covariance matrix of $\hat{\mathbf{r}}_{\mathbf{xx}}$ and $'$ denotes the transpose. The hypothesis test for this particular problem is not simple since $\mathbf{r}_{\mathbf{xx}}$ and $\mathbf{\Sigma}_{\mathbf{xx}}$ are unknown. In the case where certain parameters are unknown, the hypothesis test is called 'composite' and the statistical test is the GLRT. This has the same formulation as the LRT except that estimates of the parameters of the distribution are substituted in place of the true parameters. Therefore, to formulate the GLRT for this problem, $\mathbf{r}_{\mathbf{xx}}$ is replaced by $\hat{\mathbf{r}}_{\mathbf{xx}}$ and $\mathbf{\Sigma}_{\mathbf{xx}}^{-1}$ is replaced by $\hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1}$. Substituting into (4.19) the GLRT becomes,

$$\hat{T}_{GLRT} = e^{1/2(\hat{\mathbf{r}}_{\mathbf{xx}})\hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1}(\hat{\mathbf{r}}_{\mathbf{xx}})'}. \tag{4.20}$$

The test statistic is obtained by taking the natural logarithm of (4.20) and scaling by a factor of 2 [69] [72],

$$\hat{T}_{GLRT} = \hat{\mathbf{r}}_{\mathbf{xx}}\hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1}\hat{\mathbf{r}}_{\mathbf{xx}}'. \tag{4.21}$$

The value $\hat{T}$ will always be a positive scalar because $\hat{\mathbf{\Sigma}}_{\mathbf{xx}}^{-1}$ is positive definite. In using the test statistic in (4.21), it is necessary to compute $\hat{\mathbf{r}}_{\mathbf{xx}}$ for a particular cyclic frequency of interest and to compute $\hat{\mathbf{\Sigma}}_{\mathbf{xx}}$. For two zero mean random variables, the entries of $\hat{\mathbf{\Sigma}}_{\mathbf{xx}}$ can be computed as follows,

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx}} = \begin{bmatrix} \hat{E}[\Re(\hat{R}_{xx}^{\alpha}[\nu])^2] & \hat{E}[\Re(\hat{R}_{xx}^{\alpha}[\nu])\Im(\hat{R}_{xx}^{\alpha}[\nu])] \\ \hat{E}[\Im(\hat{R}_{xx}^{\alpha}[\nu])\Re(\hat{R}_{xx}^{\alpha}[\nu])] & \hat{E}[\Im(\hat{R}_{xx}^{\alpha}[\nu])^2] \end{bmatrix}. \tag{4.22}$$

It is clear that several instances of $\hat{\mathbf{r}}_{\mathbf{xx}}$ are required in order to compute the entries of (4.22). This can be achieved by taking the FFT of the autocorrelation lag product and scaling by a factor of $1/N$, which is the method employed in [76] [2] [77] [39]. However, having to compute an FFT adds unwanted complexity to the detection process and means that evaluation of the cyclic frequency is limited by the FFT resolution. This problem can be overcome, but requires a modification of the test statistic in (4.21). The elements of $\hat{\mathbf{r}}_{\mathbf{xx}}$ are given by,

$$\Re(\hat{R}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1}\Re(x[n]x^*[n-\nu]e^{-j2\pi\alpha n}), \tag{4.23}$$

$$\Im(\hat{R}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1}\Im(x[n]x^*[n-\nu]e^{-j2\pi\alpha n}). \tag{4.24}$$

The quantities $\Re(x[n]x^*[n-\nu]e^{-j2\pi\alpha n})$ and $\Im(x[n]x^*[n-\nu]e^{-j2\pi\alpha n})$ can be denoted as random variables, $X$ and $Y$. Using this notation, the quantities $\Re(\hat{R}^{\alpha}_{xx}[\nu])$ and $\Im(\hat{R}^{\alpha}_{xx}[\nu])$ are equivalent to the sample mean of $X$ and $Y$, which will be denoted as $\bar{X}$ and $\bar{Y}$. Therefore, (4.23) and (4.24) can be written as,

$$\bar{X} = \frac{1}{N}\sum_{n=0}^{N-1} X_n, \tag{4.25}$$

$$\bar{Y} = \frac{1}{N}\sum_{n=0}^{N-1} Y_n. \tag{4.26}$$

Thus, $\hat{\mathbf{\Sigma}}_{\mathbf{xx}}$ (4.22) can be re-expressed as,

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx}} = \begin{bmatrix} \hat{E}[\bar{X}^2] & \hat{E}[\bar{X}\bar{Y}] \\ \hat{E}[\bar{Y}\bar{X}] & \hat{E}[\bar{Y}^2] \end{bmatrix}. \tag{4.27}$$

It has been established previously that $\bar{X}$ and $\bar{Y}$ are normally distributed random variables under both hypotheses. Therefore, according to the Central Limit Theorem (CLT), $\hat{\mathbf{\Sigma}}_{\mathbf{xx}}$ can also be written as,

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx}} = \begin{bmatrix} \hat{E}[X^2]/N & \hat{E}[XY]/N \\ \hat{E}[YX]/N & \hat{E}[Y^2]/N \end{bmatrix}. \tag{4.28}$$

This is useful because it means that the covariance matrix can be estimated directly from the data in $X$ and $Y$ and, as such, repeated measurements of $\bar{X}$ and $\bar{Y}$ are not required. The scaling by a factor of $1/N$ in (4.28) can be removed by simply scaling each element by $N$. However, this requires that the test statistic in (4.21) is scaled by $N$ to give,

$$\hat{T}_{GLRT} = N\hat{\mathbf{r}}_{\mathbf{xx}}\hat{\mathbf{\Sigma}}^{-\mathbf{1}}_{\mathbf{xx}}\hat{\mathbf{r}}'_{\mathbf{xx}}. \tag{4.29}$$

Finally, $\hat{\mathbf{\Sigma}}_{\mathbf{xx}}$ can be computed as,

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx}} = \begin{bmatrix} \hat{E}[X^2] & \hat{E}[XY] \\ \hat{E}[XY] & \hat{E}[Y^2] \end{bmatrix}. \tag{4.30}$$

Each of the terms in (4.30) is calculated as [39],

$$\hat{E}[X^2] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(x[n]x^*[n-\nu]e^{-j2\pi\alpha n})^2, \tag{4.31}$$

$$\hat{E}[XY] = \hat{E}[YX] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(x[n]x^*[n-\nu]e^{-j2\pi\alpha n})\Im(x[n]x^*[n-\nu]e^{-j2\pi\alpha n}), \tag{4.32}$$

$$\hat{E}[Y^2] = \frac{1}{N}\sum_{n=0}^{N-1} \Im(x[n]x^*[n-\nu]e^{-j2\pi\alpha n})^2. \tag{4.33}$$

Therefore, substituting into (4.29), the final test statistic is

$$\hat{T}_{GLRT} = N\frac{(\Re(\hat{R}^{\alpha}_{xx}[\nu]))^2\hat{E}[Y^2] + (\Im(\hat{R}^{\alpha}_{xx}[\nu]))^2\hat{E}[X^2] - 2(\Re(\hat{R}^{\alpha}_{xx}[\nu]))(\Im(\hat{R}^{\alpha}_{xx}[\nu]))\hat{E}[XY]}{\hat{E}[X^2]\hat{E}[Y^2] - (\hat{E}[XY])^2}. \tag{4.34}$$

The distribution of (4.34) under $H_0$, is a $\chi^2_{2M}$ distribution where $M$ is the number of autocorrelation lags used in the detector and $2M$ is the degrees of freedom of the $\chi^2$ distribution [69] [72] [39]. Since only a single lag is considered, $M = 1$, and as such the test statistic is $\chi^2_2$ distributed under $H_0$. Given that the distribution of the test statistic under $H_0$ is known, a threshold can be set that limits $P_{fa}$ to a desired value. For example, suppose that is desired to limit $P_{fa}$ to 0.1, then this is substituted into (4.6) as follows,

$$\eta = F^{-1}_{\chi^2_2}(0.90). \tag{4.35}$$

This leads to a threshold for the detector of $\eta = 4.6052$ since it is $\chi^2_2$ distributed. In order to verify that the test statistic is indeed $\chi^2_2$ under $H_0$, it is assumed the input to the detector under $H_0$ is a complex normal white noise signal with independent and identically distributed (IID) real and imaginary parts as follows,

$$\mathcal{N}(0, \sigma_n^2/2), \tag{4.36}$$

where $\sigma_n^2$ is the variance of the complex normal white noise signal and $\mathcal{N}$ denotes the normal distribution. The test statistic will be $\chi^2_2$ regardless of the value of $\sigma_n^2$, since $\mathbf{r_{xx}} = [0\ 0]$ in all cases. The complex normal white noise signal can be generated in MATLAB using the *randn* function for both the real and imaginary parts. Figure 4.4. plots the simulated CDF $(F_{\hat{T}}(\eta)|H_0)$ for $\sigma_n^2 = 5$ against the theoretical $\chi_2^2$ CDF generated using the *chi2cdf* function in MATLAB. The range of the threshold $\eta$ is 0 to 40, giving a total of 41 tested thresholds. The detector is set up to detect $\alpha_0$ for an IEEE 802.11a/g OFDM signal. It is clear that there is strong agreement between the simulated CDFs and the theoretical distribution, thus confirming that the test statistic is $\chi_2^2$ distributed under $H_0$. Figure 4.5 compares the $P_d$ vs. SNR performance of the test statistic (4.34) with the FFT approach described in [76] [2] [77] [39], using the test statistic in (4.21). The test signal is an IEEE 802.11a/g OFDM signal in an AWGN channel with SNR in the range -30dB to 0dB. The observation interval is $N = 16384$ samples (FFT size) and $P_{fa} = 0.1$ for both detectors. It can be observed that there is no performance difference between the time domain and FFT domain methods.

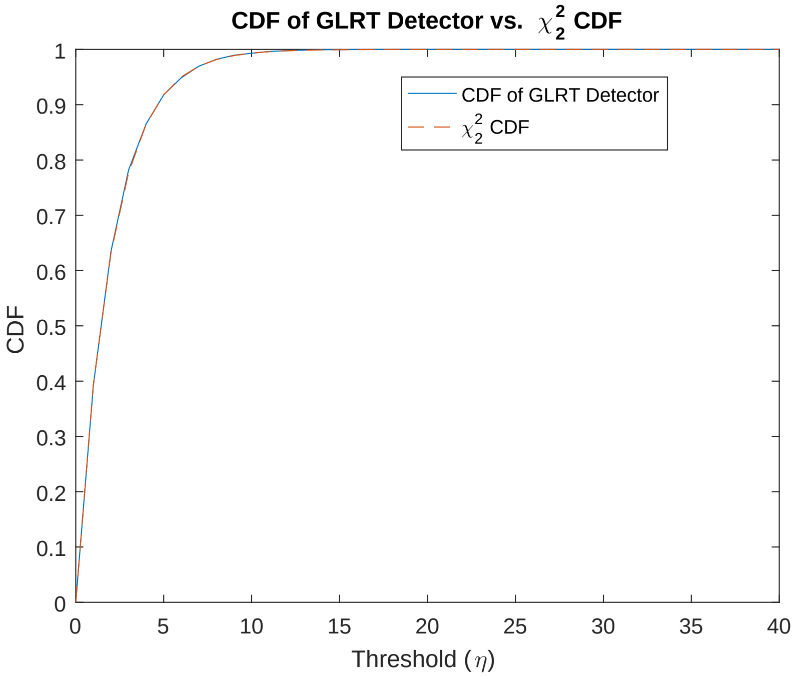


Figure 4.4: CDF of GLRT detector vs. $\chi_2^2$ CDF

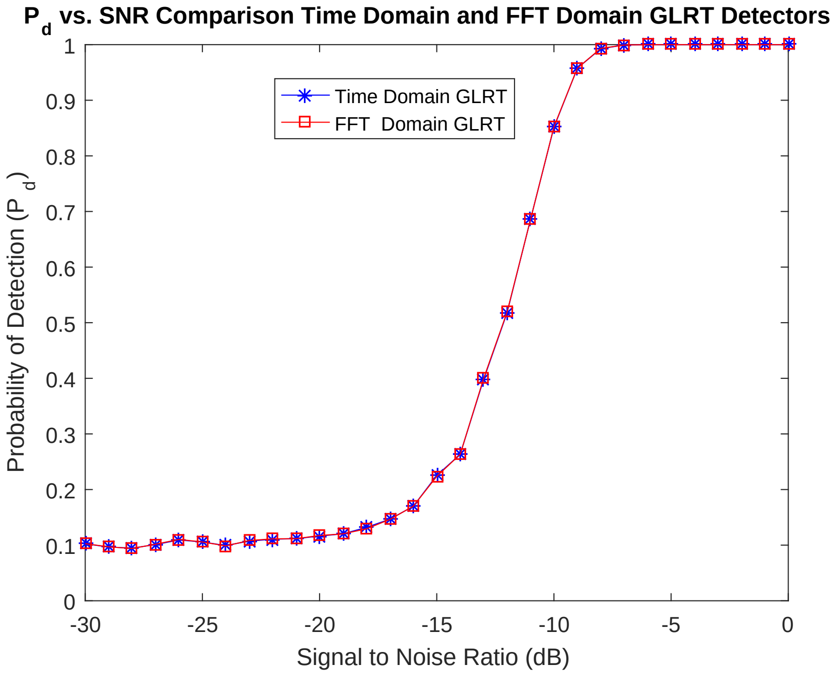


Figure 4.5: $P_d$ vs. SNR comparison Time Domain and FFT Domain GLRT detectors

## 4.6 Low Complexity Detector

In this section, a test statistic is introduced that can be derived by further analysing the elements of $\mathbf{\Sigma_{xx}}$, assuming that under $H_0$, the input signal is a complex normal white noise signal with real and imaginary parts distributed according to (4.36). Under this condition, the quantities $X$ and $Y$ are statistically independent, meaning that $E[XY]$ in (4.30) is separable and can be written as $E[X]E[Y]$. Furthermore, there is no cyclostationarity present in the input signal, so $E[X] = E[Y] = 0$. This reduces $\mathbf{\Sigma_{xx}}$ to,

$$\mathbf{\Sigma_{xx}} = \begin{bmatrix} E[X^2] & 0 \\ 0 & E[Y^2] \end{bmatrix}. \tag{4.37}$$

It is clear that only the variance terms remain in $\mathbf{\Sigma_{xx}}$. In order to determine $E[X^2]$ and $E[Y^2]$, it is instructive to expand the product $x[n]x^*[n-\nu]e^{-j2\pi\alpha n}$. Firstly, let us denote, $x[n] = x_r[n] + jx_i[n]$, $x^*[n-\nu] = x_r[n-\nu] + jx_i[n-\nu]$ and $e^{-j2\pi\alpha n} = cos(2\pi\alpha n) - jsin(2\pi\alpha n)$. After substitution and expansion X is given by the following,

$$\begin{aligned} X = & \, x_r[n]x_r[n-\nu]cos(2\pi\alpha n) \\ & - x_i[n]x_i[n-\nu]cos(2\pi\alpha n) \\ & + x_r[n]x_i[n-\nu]sin(2\pi\alpha n) \\ & + x_i[n]x_r[n-\nu]sin(2\pi\alpha n). \end{aligned} \tag{4.38}$$

The mean value of sine and cosine waveforms with peak amplitudes of 1 is zero, and their Root Mean Square (RMS) value is $1/\sqrt{2}$. Therefore, they have a variance of 1/2. [87]. The variance of the product of n independent zero mean random variables is equal to the product of their individual variances. Therefore, the variance of each of the four terms in (4.38) can be calculated as,

$$\frac{\sigma_n^2}{2}.\frac{\sigma_n^2}{2}.\frac{1}{2} = \frac{\sigma_n^4}{8}. \tag{4.39}$$

The four product terms in (4.38) are also independent random variables meaning that the total variance is given by the sum of their individual variances leading to,

$$E[X^2] = \frac{\sigma_n^4}{2}. \tag{4.40}$$

A similar approach can be used to determine $E[Y^2]$ . Following from this, $\mathbf{\Sigma_{xx}}$ reduces to,

$$\mathbf{\Sigma_{xx}} = \begin{bmatrix} \frac{\sigma_n^4}{2} & 0 \\ 0 & \frac{\sigma_n^4}{2} \end{bmatrix}. \tag{4.41}$$

Now considering $\hat{\mathbf{\Sigma}}_{\mathbf{xx}}$ is an estimate of $\mathbf{\Sigma_{xx}}$, it can be assumed that $\hat{E}[X^2] \approx \hat{E}[Y^2] \approx \hat{\sigma}_n^4/2$ and $\hat{E}[XY] \approx 0$. Substituting this into (4.29) reduces the test statistic to,

$$\hat{T}_{LC} = \frac{2N\left|\hat{R}_{xx}^{\alpha}[\nu]\right|^2}{\hat{\sigma}_n^4}. \tag{4.42}$$

This test statistic still requires an estimate of $\sigma_n^4$ but is computationally more efficient than (4.34), as $\hat{E}[X^2]$ and $\hat{E}[Y^2]$ do not need to be estimated independently and $\hat{E}[XY]$ does not need to be calculated. As a result, we refer to this detector as the “Low Complexity” detector.

$\hat{\sigma}_n^4$ can be estimated efficiently as,

$$\hat{\sigma}_n^4 = \frac{1}{N}\sum_{n=0}^{N-1}\left|x[n]x^*[n-\nu]\right|^2. \tag{4.43}$$

Therefore, the test statistic becomes,

$$\hat{T}_{LC} = \frac{2N\left|\hat{R}_{xx}^{\alpha}[\nu]\right|^2}{\frac{1}{N}\sum_{n=0}^{N-1}\left|x[n]x^*[n-\nu]\right|^2}. \tag{4.44}$$

Since (4.34) is $\chi_2^2$ distributed under $H_0$, it follows that (4.44) is also $\chi_2^2$ under $H_0$ as confirmed in Figure 4.6.

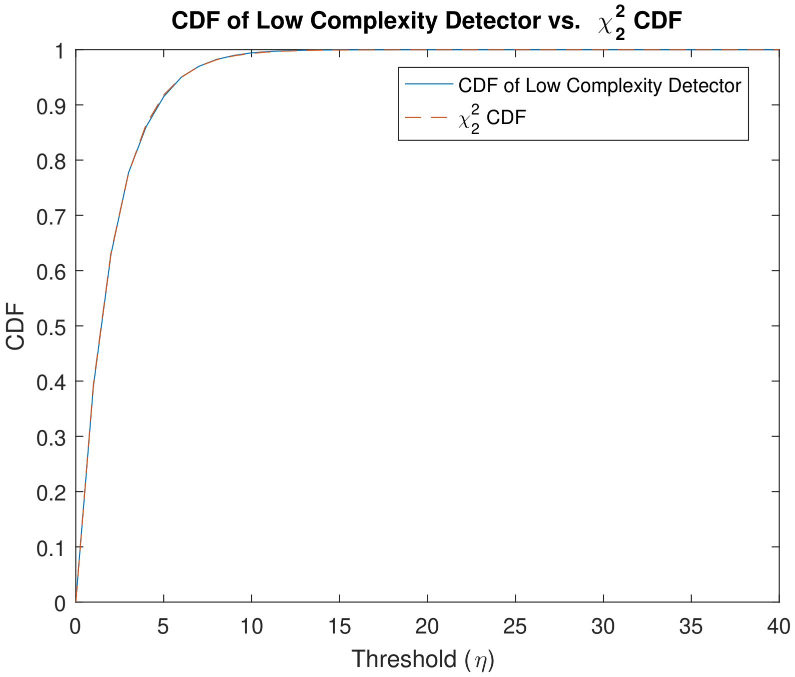


Figure 4.6: CDF of Low Complexity Detector vs. $\chi_2^2$ CDF

Scaling by a factor of 2 on the numerator of (4.44) is unnecessary if it is realised that the $\chi^2$ distribution is a special case of the $\Gamma(\theta, \lambda)$ distribution, where $\theta$ is the shape parameter and $\lambda$ is the scale parameter. The relationship between the distributions is,

$$\chi_v^2 = \Gamma(v/2, 2), \tag{4.45}$$

where $v$ denotes the degrees of freedom of the $\chi^2$ distribution. Therefore, a $\chi_2^2$ distributed random variable is equivalent to a $\Gamma(1,2)$ random variable. If a $\Gamma(1,2)$ random variable is divided by a factor of 2, the result is $\Gamma(1,1)$ distributed. Thus, the final test statistic can be written as,

$$\hat{T}_{LC} = \frac{N\left|\hat{R}^{\alpha}_{xx}[\nu]\right|^2}{\frac{1}{N}\sum_{n=0}^{N-1}\left|x[n]x^*[n-\nu]\right|^2}. \tag{4.46}$$

Figure 4.7 compares the CDF of (4.46) and the theoretical $\Gamma(1,1)$ CDF. These match very closely, thus justifying the preceding analysis.

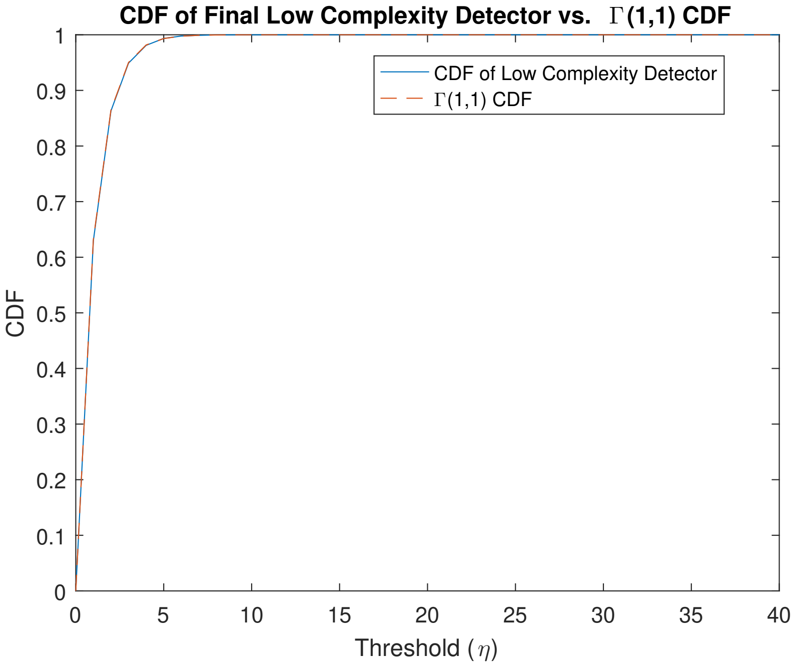


Figure 4.7: CDF of Final Low Complexity Detector vs. $\Gamma(1,1)$ CDF

## 4.7 Spatial Sign Detector

In this section, a detector developed by the authors in [83] is introduced. The detector is based on applying the spatial sign function to the complex input signal, which reduces the elements of the covariance matrix in the GLRT test statistic (4.22) to constant values, regardless of the signal variance. As a result, elements of the covariance matrix do not need to be estimated, as was the case with the GLRT and Low Complexity detectors. The normalisation applied at the input of the detector has been shown to provide robust detection performance in impulsive noise environments when compared to other solutions [58], and this will also be demonstrated in Section 4.8. The spatial sign function is defined as,

$$s[n] = \begin{cases} \dfrac{x[n]}{|x[n]|}, & x[n] \neq 0 \\ 0, & x[n] = 0 \end{cases}. \tag{4.47}$$

In practice, it is assumed that the chance of receiving a zero valued sample, i.e. $x[n] = 0$ is negligible. The CAF after applying the spatial sign function is,

$$\hat{R}^{\alpha}_{ss}[\nu] = \frac{1}{N} \sum_{n=0}^{N-1} s[n]s^{*}[n-\nu]e^{-j2\pi\alpha n}, \tag{4.48}$$

where $\hat{R}_{xx}[\nu]$ has been replaced with $\hat{R}_{ss}[\nu]$ signifying that the complex input data has been passed through the spatial sign function. The CAF in (4.48) can be estimated across a range of cyclic frequencies by taking the FFT of $s[n]s^{*}[n-\nu]$ and scaling by a factor of $1/N$ as follows,

$$S[k] = \frac{1}{N} \sum_{n=0}^{N-1} s[n]s^{*}[n-\nu]e^{-j2\pi nk/N}, \tag{4.49}$$

where $S[k]$ replaces $X[k]$ defined previously. Figure 4.8 shows the magnitude of (4.49) for an FFT size of $N = 16384$, with an IEEE 802.11a/g OFDM signal. It can be observed that the cyclostationary features are still retained after transformation using the spatial sign function. In [85], a mathematical justification for this is provided. Therefore, it is possible to detect the presence of the signal of interest using cyclostationarity after applying the spatial sign function. In order to compute the spatial sign function directly, an absolute value calculation is required, which consists of two real multiply operations, an addition and a square root. It also requires a division to normalise the complex input samples by their absolute values. Both square root and division are difficult and costly to implement in FPGA hardware. Since it is desirable to produce architectures that are as amenable to FPGA implementation as possible, it would be beneficial to avoid the square root and division operations.

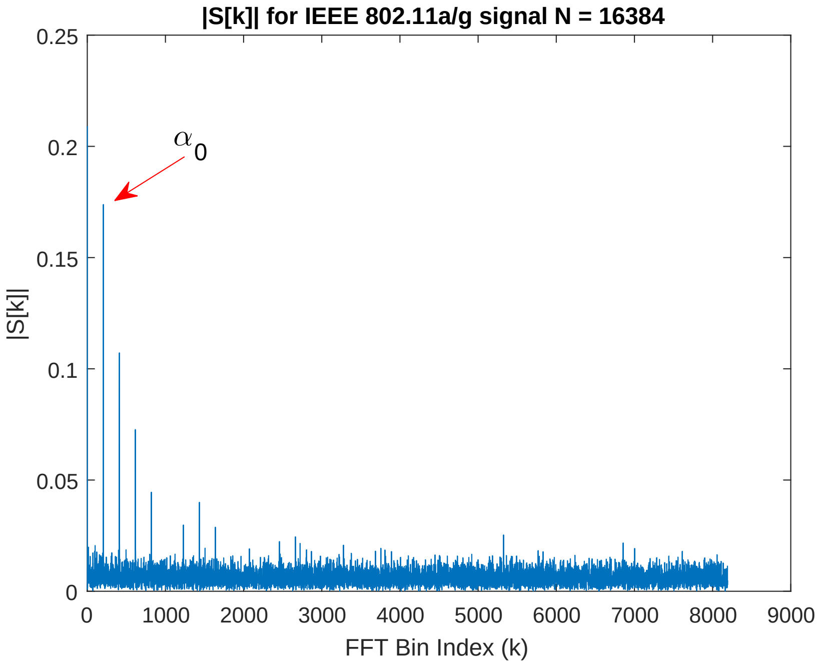


Figure 4.8: $\left|S[k]\right|$ for IEEE 802.11a/g signal N = 16384

This is possible if it is realised that the spatial sign function is equivalent to the following,

$$s[n] = \frac{x[n]}{\left|x[n]\right|} = cos(\phi[n]) + jsin(\phi[n]), \tag{4.50}$$

where $\phi[n]$ is the time dependent phase of $x[n]$. The implementation is immediately simplified as the calculation can be performed using two Co-ordinate Rotational Digital Computer (CORDIC) [88] stages. Having applied the spatial sign function, the random variable $X$ as defined in Section 4.5 can be written as,

$$\begin{aligned} X = {} & cos(\phi[n])cos(\phi[n-\nu])cos(2\pi\alpha n) \\ & - sin(\phi[n])sin(\phi[n-\nu])cos(2\pi\alpha n) \\ & + cos(\phi[n])sin(\phi[n-\nu])sin(2\pi\alpha n) \\ & + sin(\phi[n])cos(\phi[n-\nu])sin(2\pi\alpha n). \end{aligned} \tag{4.51}$$

The variance of each of the four terms in (4.51) is 1/8 and therefore the total variance is,

$$E[X^2] = \frac{1}{2}. \tag{4.52}$$

It is also true under $H_0$ that $E[XY] = E[X]E[Y] = 0$, leading to the final covariance matrix,

$$\mathbf{\Sigma_{ss}} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}. \tag{4.53}$$

Substituting the inverse of (4.53) into (4.29) with $\hat{\mathbf{r}}_{\mathbf{xx}}$ replaced by its spatial sign equivalent ,$\hat{\mathbf{r}}_{\mathbf{ss}}$, leads to the test statistic,

$$\hat{T}_{SS} = 2N\left|\hat{R}^{\alpha}_{ss}[\nu]\right|^2, \tag{4.54}$$

which is $\chi^2_2$ distributed under $H_0$. Again, scaling by a factor of 2 is unnecessary and can be removed, giving the final test statistic [58] [83],

$$\hat{T}_{SS} = N\left|\hat{R}^{\alpha}_{ss}[\nu]\right|^2. \tag{4.55}$$

The distribution of (4.55) is $\Gamma(1,1)$ as confirmed in Figure 4.9.

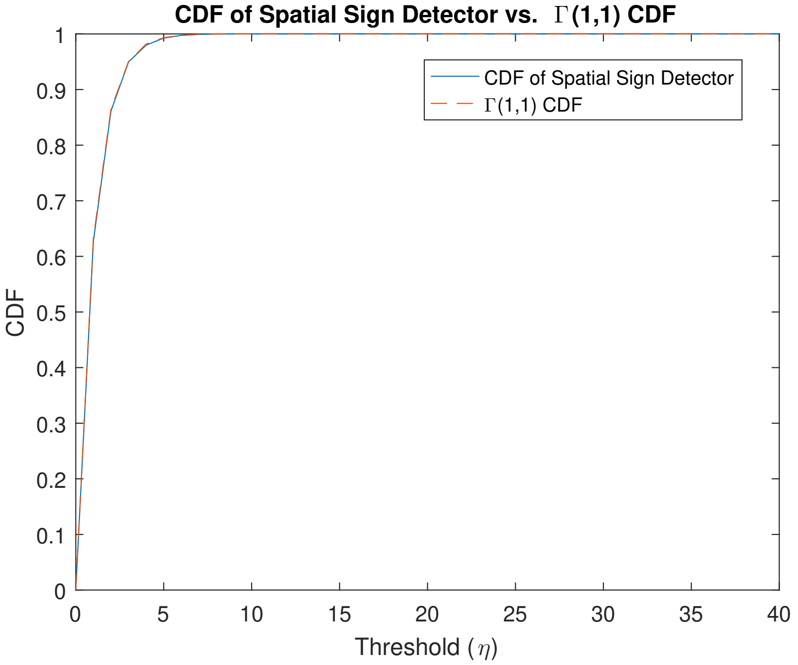


Figure 4.9: CDF of Spatial Sign Detector vs. $\Gamma(1,1)$ CDF

## 4.8 Detection Performance Comparison

Having derived each of the detectors in the previous sections, it is now necessary to compare their relative performance in terms of $P_d$ for both AWGN and impulsive noise environments.

The performance will be evaluated for both IEEE 802.11a/g and DVB-T signals. Figure 4.10 compares $P_d$ vs. SNR curves for the GLRT, Low Complexity and Spatial Sign detectors in an AWGN channel. Assuming a zero mean signal, the SNR in dB is defined as,

$$SNR = 10log_{10}(\frac{\sigma_s^2}{\sigma_n^2}), \tag{4.56}$$

where $\sigma_s^2$ is the signal variance and $\sigma_n^2$ is the noise variance. The input test signal is IEEE 802.11a/g Non-HT, the thresholds for the detectors have been chosen to guarantee a $P_{fa} = 0.1$ and the observation interval is equal to $N = 1600$ samples, which is equivalent to 20 OFDM symbols. $P_d$ is calculated by computing 5000 test statistics at each SNR level. At each iteration of the simulation, new realisations of the IEEE 802.11a/g signal and the noise are generated.

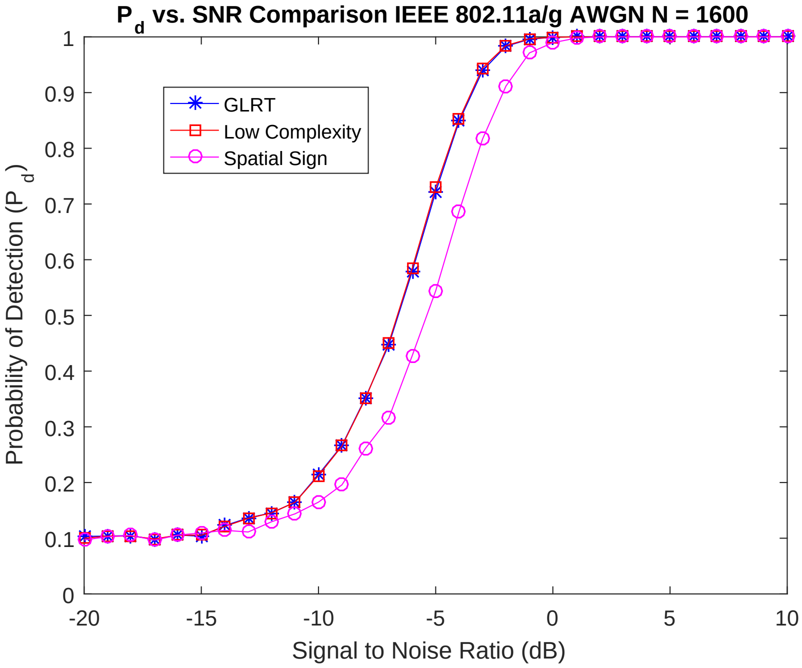


Figure 4.10: $P_d$ vs. SNR Detector Performances for IEEE802.11a/g in AWGN

It can be seen that the GLRT detector and the Low Complexity detector give the best overall performance in an AWGN channel, with almost identical performances for each SNR level. It is clear that for the Spatial Sign detector, there is a performance drop when compared to the GLRT and Low Complexity detectors. This result is observed in [39] and [58]. The performance drop can be explained by the fact that the spatial sign function discards the amplitude information of the input signal [85]. This is disadvantageous since the reduction in detector complexity comes at the cost of a reduced performance in an AWGN channel. Therefore, it can be concluded that the best detection performance can be achieved by choosing between the GLRT and Low Complexity detectors, assuming operation in an AWGN channel .

The relationship between the performances of each of the detectors can also be observed for a DVB-T signal. In this case, $N_u = 2048$ and $N_{cp} = 512$, leading to $N_{ofdm} = 2560$. Therefore, the fundamental cyclic frequency is $\alpha_0 = 1/2560$. Figure 4.11 shows $P_d$ vs. SNR curves for each of the detection algorithms, with an observation interval of $N = 51200$ samples which is equivalent to 20 OFDM symbols and $P_{fa} = 0.1$. It can be observed that the best detection performance is achieved by using either the GLRT or the Low Complexity detector and it is seen that the Spatial Sign detector exhibits a reduced detection performance in the AWGN channel as expected.

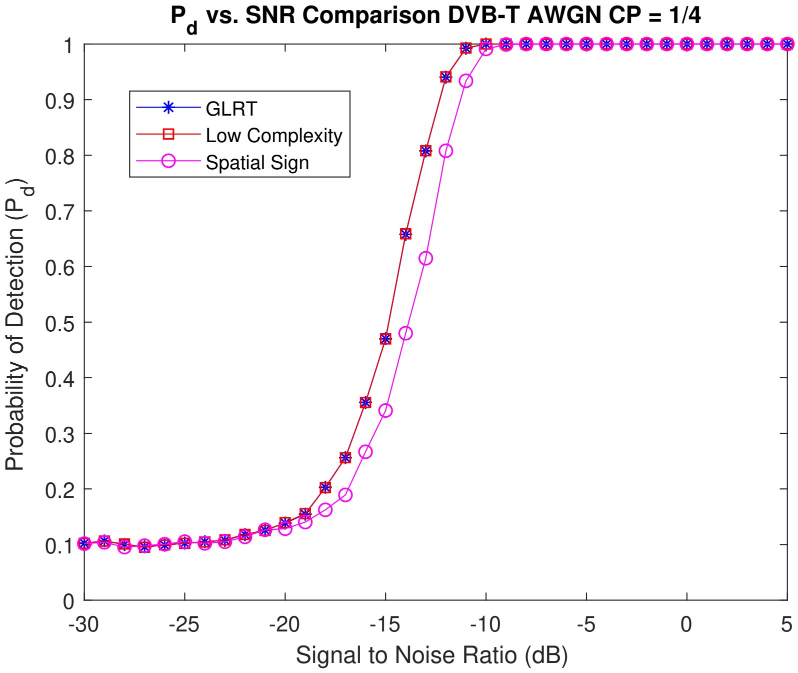


Figure 4.11: $P_d$ vs. SNR Detector Performances for DVB-T in AWGN

Having assessed the performances of each of the detectors in an AWGN channel for IEEE802.11a/g Non-HT and DVB-T OFDM signals, it is now instructive to assess their performances in impulsive noise environments. This follows the analysis carried out by the authors in [58]. Typical sources of natural impulsive noise include atmospheric and receiver noise and man-made sources include microwave ovens, car ignitions systems and radiation from power lines [58] [89]. As described in [58], impulsive noise can be adequately modelled using contaminated Gaussian noise distributions and Symmetric Alpha Stable (S$\alpha$S) distributions. It has been found that contaminated Gaussian noise models give a good approximation to the Middleton Class A impulsive noise model. Similarly, S$\alpha$S distributions give a good approximation to the Middleton Class B model [58]. Therefore, we will use both of these to assess the performance of each detector in impulsive noise.

In the simulations for Chapter 4 and 5, the following complex contaminated Gaussian noise model will be employed as described in [58],

$$0.95\mathcal{N}_C(0,\sigma^2) + 0.05\mathcal{N}_C(0,100\sigma^2), \tag{4.57}$$

where $\mathcal{N}_C$ denotes the complex Gaussian distribution. This model is used in [58] and 5% of the time the noise is drawn from a complex Gaussian distribution with a variance of $100\sigma^2$. In the case of the contaminated Gaussian noise model, the SNR is still defined as in (4.56) where $\sigma_n^2$ is equal to $\sigma^2$.

Figures 4.12 and 4.13 compare the performances of the detectors in the presence of the contaminated Gaussian impulsive noise. Again, $N_{cp} = 512$ for the DVB-T signal.

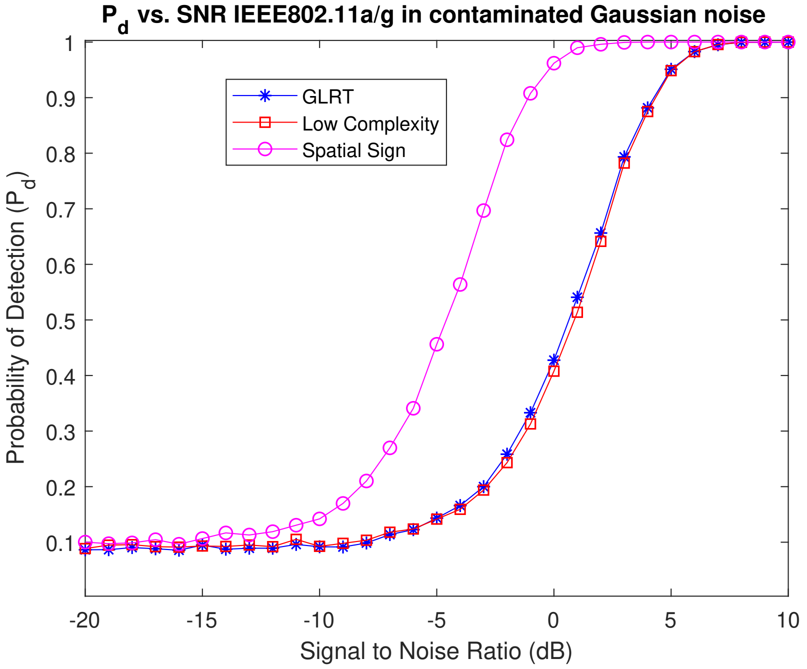


Figure 4.12: $P_d$ vs. SNR detector comparisons for IEEE802.11a/g in contaminated Gaussian Impulsive noise

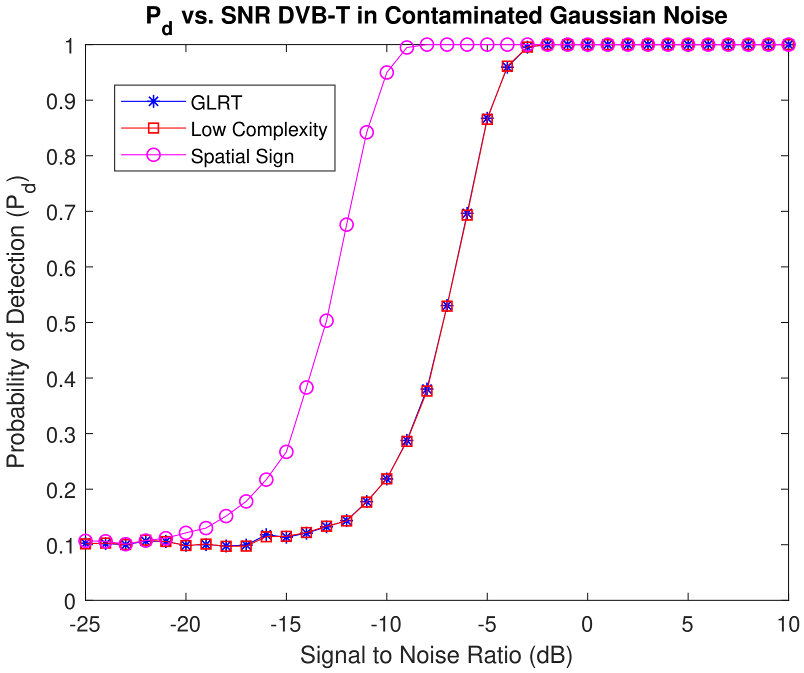


Figure 4.13: $P_d$ vs. SNR detector comparisons for DVB-T in contaminated Gaussian Impulsive noise

It is clear that the performance of the GLRT and Low Complexity detectors degrades significantly. Conversely, the Spatial Sign detector outperforms both the GLRT and Low Complexity detectors in the contaminated Gaussian noise. This insensitivity to impulsive noise is achieved because of the normalisation applied by the spatial sign function. This result was also observed in [58].

As described earlier, impulsive noise can be adequately modelled using an S$\alpha$S distribution. Stable distributions are a family of probability distributions that can be used for modelling heavy tails and skewness [90]. The stable distribution is characterised by four parameters: the stability parameter ($\alpha$), the skewness parameter ($\beta$), the scale parameter ($s$) and the location parameter ($\mu$). The S$\alpha$S distribution corresponds to the case where $\beta = 0$ and the Cauchy distribution is a special case of the symmetric stable distribution with $\alpha = 1$. In [58], noise samples from the Cauchy distribution are added to the signal of interest, in order to demonstrate the robustness of the Spatial Sign detector in impulsive noise environments. Therefore, this approach will also be employed in this thesis. In order to assess the performance of each detector in Cauchy impulsive noise, a new metric is defined called the Generalised Signal to Noise Ratio (GSNR). This is given by,

$$GSNR = 10log_{10}(\frac{\sigma_s^2}{s}), \tag{4.58}$$

where $s$ corresponds to the scale parameter of the Cauchy noise. Figures 4.14 and 4.15 compare the performances of each detector in Cauchy impulsive noise as a function of GSNR for both IEEE802.11a/g and DVB-T signals.

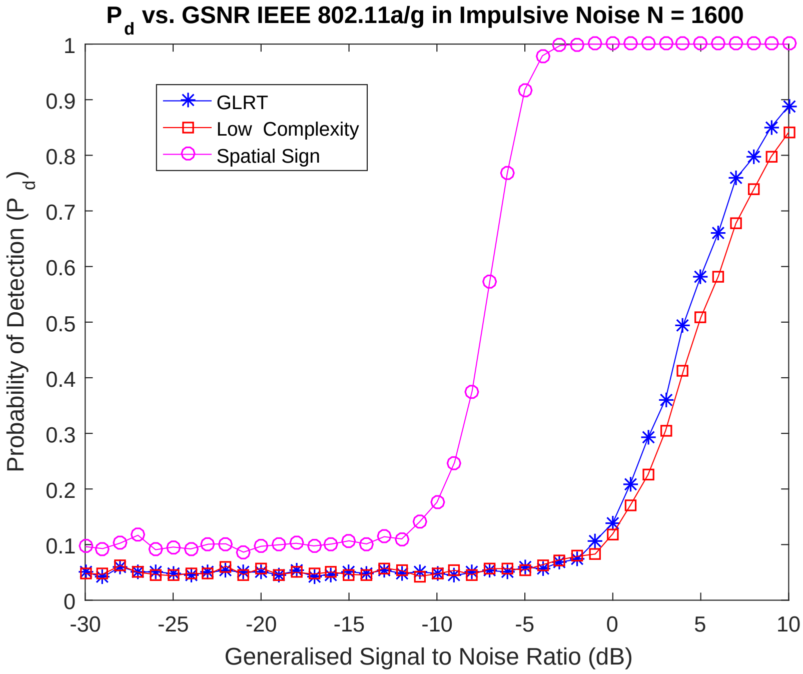


Figure 4.14: $P_d$ vs. GSNR detector comparisons for IEEE802.11a/g in Cauchy Impulsive noise

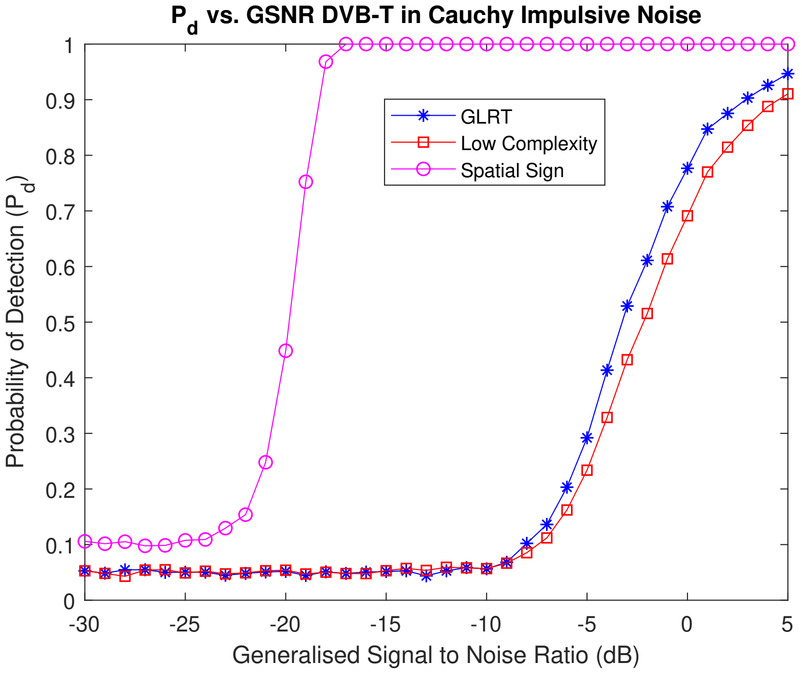


Figure 4.15: $P_d$ vs. GSNR detector comparisons for DVB-T in Cauchy Impulsive noise

As was the case for the contaminated Gaussian noise, it is clear that the Spatial Sign detector performs robustly in the presence of the Cauchy distributed noise. However, the performance of the GLRT and Low Complexity detectors is severely degraded. It can also be observed that as the GSNR decreases, $P_d$ does not converge to the theoretical $P_{fa}$ of 0.1 for the GLRT and Low Complexity detectors. The theoretical false alarm rate is not achieved because the Cauchy noise violates the asymptotic normality of the CAF. The Cauchy noise has a non-finite variance and thus the Generalised CLT applies rather than the CLT. Using the Spatial Sign function ensures that the asymptotic normality of the CAF is retained, because the variance of the output signal is finite and constant and thus the CLT is satisfied. Therefore, the theoretical false alarm can be achieved, even when the input signal is corrupted by Cauchy noise. Overall, the Spatial Sign detector has the most robust performance in both AWGN and impulsive noise.

## 4.9 Chapter Summary

In conclusion, this chapter has introduced the concept of cyclostationary feature detection for OFDM signals. This problem is an especially important part of the research into CR, which has been offered as a solution to the problem of spectrum underutilisation. Several cyclostationary feature detection algorithms that exist in the literature have been introduced and derived and their ability to detect both IEEE 802.11a/g Non-HT and DVB-T OFDM signals has been established.

Each of these detectors exhibits a CFAR and only requires knowledge of the OFDM signal structure and cyclic frequencies. Knowledge of these details is realistic, since they are publicly available for the various commercial OFDM waveforms in use today. It has been shown that the GLRT and the Low Complexity detectors provide the most robust performance, assuming operation in an AWGN channel. The Spatial Sign detector exhibits a slight reduction in performance in the AWGN channel due to the loss of amplitude information. However, the Spatial Sign detector provides robust performance in impulsive noise environments, whereas the performances of the GLRT and Low Complexity detectors degrade significantly.

# Chapter 5

# Improved Cyclostationary Feature Detection Algorithms

## 5.1 Introduction

Having derived several cyclostationary feature detection algorithms and compared their performances in the detection of both IEEE 802.11a/g Non-HT and DVB-T OFDM signals, four new detection algorithms will be introduced. Each of these is derived based on a novel method where the CAF is split into two component functions, formed using the real and imaginary parts of the autocorrelation lag product respectively. The first three algorithms can be considered as reformulations of the GLRT, Low Complexity and Spatial Sign detectors introduced in the previous chapter. The fourth algorithm combines quantisation of the complex OFDM signal at the detector input, with the proposed method of basing the detection on estimation of the two component functions of the traditional CAF. The four proposed cyclostationary detectors are called the Split-CAF GLRT, Split-CAF Low Complexity, Split-CAF Spatial Sign and Split-CAF Quantised detectors respectively.

The probability distributions of the test statistics under the null hypothesis are confirmed theoretically and in simulation. The performances of the detectors are established for the detection of both IEEE802.11a/g Non-HT and DVB-T OFDM signals in AWGN and impulsive noise environments. Furthermore, the performance of the algorithms under radio impairments such as multipath fading and CFO are investigated.

## 5.2 Proposed Detection Algorithms

### 5.2.1 Component Functions

In the previous chapter, each cyclostationary detection algorithm was based on estimation of the CAF, as re-stated below,

$$\hat{R}_{xx}^{\alpha}[\nu] = \frac{1}{N} \sum_{n=0}^{N-1} x[n]x^*[n-\nu]e^{-j2\pi\alpha n}. \tag{5.1}$$

Due to the fact that the received OFDM signal is corrupted by noise, the autocorrelation lag product $x[n]x^*[n-\nu]$ will be a complex valued signal. This leads us to define a modified CAF that only considers the real part of $x[n]x^*[n-\nu]$ as follows,

$$\hat{I}_{xx}^{\alpha}[\nu] = \frac{1}{N} \sum_{n=0}^{N-1} \Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}. \tag{5.2}$$

Similarly, another modified CAF can be formulated using the imaginary part of $x[n]x^*[n-\nu]$ as,

$$\hat{Q}_{xx}^{\alpha}[\nu] = \frac{1}{N} \sum_{n=0}^{N-1} \Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}. \tag{5.3}$$

Note that the letters $I$ and $Q$ have been used in keeping with the concept of I/Q data. The functions $\hat{I}_{xx}^{\alpha}[\nu]$ and $\hat{Q}_{xx}^{\alpha}[\nu]$ are related to $\hat{R}_{xx}^{\alpha}[\nu]$ as,

$$\hat{R}_{xx}^{\alpha}[\nu] = \hat{I}_{xx}^{\alpha}[\nu] + j\hat{Q}_{xx}^{\alpha}[\nu]. \tag{5.4}$$

Therefore, they can be thought of as two component functions that combine to form the traditional CAF. In the following sections, several detection algorithms are introduced that compute separate test statistics using $\hat{I}_{xx}^{\alpha}[\nu]$ and $\hat{Q}_{xx}^{\alpha}[\nu]$, and sum the results to form a final test statistic. These are called the Split-CAF GLRT, Split-CAF Low Complexity, Split-CAF Spatial Sign and Split-CAF Quantised detectors.

### 5.2.2 Split-CAF GLRT Detector

In this section, a new test statistic is introduced that is similar in formulation to the GLRT detector in Section 4.5 of Chapter 4. The derivation of the Split-CAF GLRT begins by arranging the real and imaginary parts of $\hat{I}^{\alpha}_{xx}[\nu]$ in vector form as follows,

$$\mathbf{\hat{I}_{xx}} = [\Re(\hat{I}^{\alpha}_{xx}[\nu]) \quad \Im(\hat{I}^{\alpha}_{xx}[\nu])]. \tag{5.5}$$

Similarly, the real and imaginary parts of $\hat{Q}^{\alpha}_{xx}[\nu]$ can also be arranged in vector form,

$$\mathbf{\hat{Q}_{xx}} = [\Re(\hat{Q}^{\alpha}_{xx}[\nu]) \quad \Im(\hat{Q}^{\alpha}_{xx}[\nu])]. \tag{5.6}$$

The elements of $\mathbf{\hat{I}_{xx}}$ and $\mathbf{\hat{Q}_{xx}}$ are calculated as follows,

$$\Re(\hat{I}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1} \Re(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}), \tag{5.7}$$

$$\Im(\hat{I}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1} \Im(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}), \tag{5.8}$$

$$\Re(\hat{Q}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1} \Re(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}), \tag{5.9}$$

$$\Im(\hat{Q}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1} \Im(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}). \tag{5.10}$$

The quantities $\Re(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})$, $\Im(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})$, $\Re(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})$ and $\Im(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})$ are random variables and will be denoted by the symbols $X_I$, $Y_I$, $X_Q$ and $Y_Q$ respectively. Similarly, $\Re(\hat{I}^{\alpha}_{xx}[\nu])$, $\Im(\hat{I}^{\alpha}_{xx}[\nu])$, $\Re(\hat{Q}^{\alpha}_{xx}[\nu])$ and $\Im(\hat{Q}^{\alpha}_{xx}[\nu])$ are also random variables and will be denoted as $\bar{X}_I$, $\bar{Y}_I$, $\bar{X}_Q$ and $\bar{Y}_Q$ respectively. If we assume that $X_I$, $Y_I$, $X_Q$ and $Y_Q$ are drawn from a probability distribution with finite mean and variance, it follows from the CLT that $\bar{X}_I$, $\bar{Y}_I$, $\bar{X}_Q$ and $\bar{Y}_Q$ are normally distributed random variables under both hypotheses for a sufficiently large $N$. Under $H_0$, the means of $\mathbf{\hat{I}_{xx}}$ and $\mathbf{\hat{Q}_{xx}}$ are zero since no cyclostationarity is present, and these quantities only have values due to estimation errors. Under $H_1$, the mean of $\mathbf{\hat{I}_{xx}}$ is equal to $\mathbf{I_{xx}}$, i.e. the true value which is unknown. Therefore, a GLRT test statistic can be formed for $\hat{I}^{\alpha}_{xx}[\nu]$ as,

$$\hat{T}_I = \mathbf{\hat{I}_{xx}}\mathbf{\hat{\Sigma}^{-1}_{xx_I}}\mathbf{\hat{I}'_{xx}}. \tag{5.11}$$

Similarly, a test statistic can be formed for $\hat{Q}^{\alpha}_{xx}[\nu]$ as,

$$\hat{T}_Q = \mathbf{\hat{Q}_{xx}}\mathbf{\hat{\Sigma}^{-1}_{xx_Q}}\mathbf{\hat{Q}'_{xx}}. \tag{5.12}$$

As before, the main task in formulation of these detectors is estimation of the covariance matrices. The estimated covariance matrix for (5.11) is given by,

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx_I}} = \begin{bmatrix} \hat{E}[\bar{X}_I^2] & \hat{E}[\bar{X}_I\bar{Y}_I] \\ \hat{E}[\bar{Y}_I\bar{X}_I] & \hat{E}[\bar{Y}_I^2] \end{bmatrix}. \tag{5.13}$$

Similarly, the estimated covariance matrix for (5.12) is computed as,

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx_Q}} = \begin{bmatrix} \hat{E}[\bar{X}_Q^2] & \hat{E}[\bar{X}_Q\bar{Y}_Q] \\ \hat{E}[\bar{Y}_Q\bar{X}_Q] & \hat{E}[\bar{Y}_Q^2] \end{bmatrix}. \tag{5.14}$$

Using the CLT, it is possible to formulate (5.13) as,

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx_I}} = \begin{bmatrix} \hat{E}[X_I^2]/N & \hat{E}[X_IY_I]/N \\ \hat{E}[Y_IX_I]/N & \hat{E}[Y_I^2]/N \end{bmatrix}, \tag{5.15}$$

and similarly (5.14) can be computed as,

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx_Q}} = \begin{bmatrix} \hat{E}[X_Q^2]/N & \hat{E}[X_QY_Q]/N \\ \hat{E}[Y_QX_Q]/N & \hat{E}[Y_Q^2]/N \end{bmatrix}. \tag{5.16}$$

The scaling factor of $1/N$ in (5.15) and (5.16) can be removed by multiplying through by $N$ giving the final estimated covariance matrix for (5.11),

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx_I}} = \begin{bmatrix} \hat{E}[X_I^2] & \hat{E}[X_IY_I] \\ \hat{E}[Y_IX_I] & \hat{E}[Y_I^2] \end{bmatrix}, \tag{5.17}$$

and similarly for (5.12) as,

$$\hat{\mathbf{\Sigma}}_{\mathbf{xx_Q}} = \begin{bmatrix} \hat{E}[X_Q^2] & \hat{E}[X_QY_Q] \\ \hat{E}[Y_QX_Q] & \hat{E}[Y_Q^2] \end{bmatrix}. \tag{5.18}$$

However, the scaling by $N$ must be applied to the test statistics in (5.11) and (5.12). This leads to a final test statistic for $\hat{I}_{xx}^{\alpha}[\nu]$,

$$\hat{T}_I = N\hat{\mathbf{I}}_{\mathbf{xx}}\hat{\mathbf{\Sigma}}_{\mathbf{xx_I}}^{\mathbf{-1}}\hat{\mathbf{I}}_{\mathbf{xx}}', \tag{5.19}$$

and for $\hat{Q}^{\alpha}_{xx}[\nu]$,

$$\hat{T}_Q = N\hat{\mathbf{Q}}_{\mathbf{xx}}\hat{\mathbf{\Sigma}}^{-\mathbf{1}}_{\mathbf{xx_Q}}\hat{\mathbf{Q}}^{\prime}_{\mathbf{xx}}. \tag{5.20}$$

The elements of (5.17) and (5.18) are calculated as follows,

$$\hat{E}[X_I^2] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})^2, \tag{5.21}$$

$$\hat{E}[X_I Y_I] = \hat{E}[Y_I X_I] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})\Im(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}), \tag{5.22}$$

$$\hat{E}[Y_I^2] = \frac{1}{N}\sum_{n=0}^{N-1} \Im(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})^2, \tag{5.23}$$

$$\hat{E}[X_Q^2] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})^2, \tag{5.24}$$

$$\hat{E}[X_Q Y_Q] = \hat{E}[Y_Q X_Q] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})\Im(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}), \tag{5.25}$$

$$\hat{E}[Y_Q^2] = \frac{1}{N}\sum_{n=0}^{N-1} \Im(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})^2. \tag{5.26}$$

Using (5.19), the final test statistic for $\hat{I}^{\alpha}_{xx}[\nu]$ can be written as,

$$\hat{T}_I = N\frac{(\Re(\hat{I}^{\alpha}_{xx}[\nu]))^2\hat{E}[Y_I^2] + (\Im(\hat{I}^{\alpha}_{xx}[\nu]))^2\hat{E}[X_I^2] - 2(\Re(\hat{I}^{\alpha}_{xx}[\nu]))(\Im(\hat{I}^{\alpha}_{xx}[\nu]))\hat{E}[X_I Y_I]}{\hat{E}[X_I^2]\hat{E}[Y_I^2] - (\hat{E}[X_I Y_I])^2}, \tag{5.27}$$

and using (5.20), the final test statistic for $\hat{Q}^{\alpha}_{xx}[\nu]$ is,

$$\hat{T}_Q = N\frac{(\Re(\hat{Q}^{\alpha}_{xx}[\nu]))^2\hat{E}[Y_Q^2] + (\Im(\hat{I}^{\alpha}_{xx}[\nu]))^2\hat{E}[X_Q^2] - 2(\Re(\hat{Q}^{\alpha}_{xx}[\nu]))(\Im(\hat{Q}^{\alpha}_{xx}[\nu]))\hat{E}[X_Q Y_Q]}{\hat{E}[X_Q^2]\hat{E}[Y_Q^2]}. \tag{5.28}$$

Under $H_0$, both $\hat{T}_I$ and $\hat{T}_Q$ are $\chi^2_2$ distributed random variables. It is not desirable to use either $\hat{T}_I$ or $\hat{T}_Q$ as test statistics on their own, for reasons that are explained in Section 5.4.2. Because of this, it is proposed that $\hat{T}_I$ and $\hat{T}_Q$ be summed together to form a new test statistic,

$$\hat{T}_{SC-GLRT} = \hat{T}_I + \hat{T}_Q. \tag{5.29}$$

Note that the justification for this approach will be provided in Section 5.4.2. If the numerator of (5.27) is denoted as $N_I$, the denominator of (5.27) is denoted as $D_I$, the numerator of (5.28) is denoted as $N_Q$ and the denominator of (5.28) as $D_Q$, then $\hat{T}_{SC-GLRT}$ is calculated as,

$$\hat{T}_{SC-GLRT} = N\frac{N_I D_Q + N_Q D_I}{D_I D_Q}. \tag{5.30}$$

Since both $\hat{T}_I$ and $\hat{T}_Q$ are independent and $\chi_2^2$ distributed, $\hat{T}_{SC-GLRT}$ is $\chi_4^2$ distributed. This is because the sum of $n$ independent $\chi_2^2$ random variables is $\chi_{2n}^2$ distributed, as confirmed in Figure 5.1.

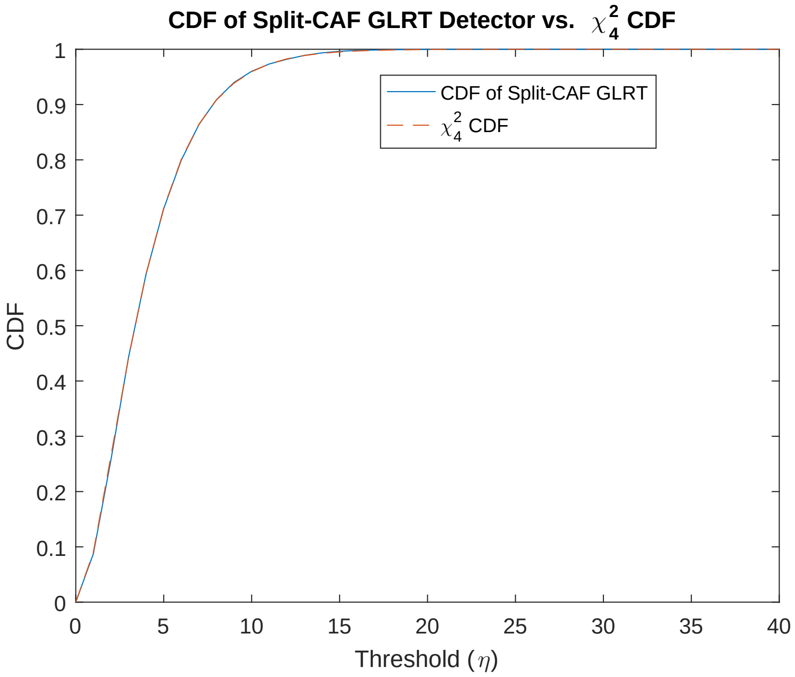


Figure 5.1: CDF of Split-CAF GLRT Detector vs. $\chi_4^2$ CDF

### 5.2.3 Split-CAF Low Complexity Detector

A Split-CAF Low Complexity detector can be derived by further analysing the Split-CAF GLRT, assuming that the input signal under $H_0$ is a complex normal white noise signal with real and imaginary parts distributed according to (4.36). This detector is described in the paper [1], written by the author of this thesis.

The first step is to analyse the covariance matrix in (5.17). Note that this analysis will equally apply to the covariance matrix in (5.18). Due to the fact that $X_I$ and $Y_I$ are statistically independent, $E[X_I Y_I] = E[X_I]E[Y_I]$. Furthermore, under $H_0$, the input signal is not cyclostationary so $E[X_I] = E[Y_I] = 0$. This reduces (5.17) to,

$$\mathbf{\Sigma_{xx_I}} = \begin{bmatrix} E[X_I^2] & 0 \\ 0 & E[Y_I^2] \end{bmatrix}. \tag{5.31}$$

In order to determine the variance of $X_I$, it is first necessary to expand the product. Let us denote $x[n] = x_r[n] + jx_i[n]$, $x^*[n-\nu] = x_r[n-\nu] + jx_i[n-\nu]$ and $e^{-j2\pi\alpha n} = cos(2\pi\alpha n) - jsin(2\pi\alpha n)$ as before. After some mathematical manipulation, $X_I$ can be written as,

$$\begin{aligned} X_I = {} & x_r[n]x_r[n-\nu]cos(2\pi\alpha n) \\ & + x_i[n]x_i[n-\nu]cos(2\pi\alpha n). \end{aligned} \tag{5.32}$$

Using the same process as for the Low Complexity detector, it can be shown that,

$$E[X_I^2] = \sigma_n^4/4. \tag{5.33}$$

Similarly, for $Y_I$ it can be shown that,

$$E[Y_I^2] = \sigma_n^4/4. \tag{5.34}$$

This reduces the covariance matrix in (5.31) to,

$$\mathbf{\Sigma_{xx_I}} = \begin{bmatrix} \sigma_n^4/4 & 0 \\ 0 & \sigma_n^4/4 \end{bmatrix}. \tag{5.35}$$

Therefore, it can be assumed that $\hat{E}[X_I^2] \approx \hat{E}[Y_I^2] \approx \hat{\sigma}_n^4/4$ and $\hat{E}[X_I Y_I] \approx 0$. Substituting into (5.19) reduces the test statistic to the following,

$$\hat{T}_I = \frac{4N}{\hat{\sigma}_n^4}\left|\hat{I}_{xx}^{\alpha}[\nu]\right|^2, \tag{5.36}$$

which will be $\chi_2^2$ distributed under $H_0$. Multiplying the numerator of (5.36) by a factor of 4 is unnecessary and so the test statistic becomes,

$$\hat{T}_I = \frac{N}{\hat{\sigma}_n^4}\left|\hat{I}_{xx}^{\alpha}[\nu]\right|^2. \tag{5.37}$$

Since (5.36) is $\chi_2^2$ distributed, it follows that (5.37) is $\Gamma(1, 0.5)$ distributed. As mentioned, a similar approach can be followed for the test statistic in (5.20), leading to,

$$\hat{T}_Q = \frac{N}{\sigma_n^4}\left|\hat{Q}_{xx}^{\alpha}[\nu]\right|^2. \tag{5.38}$$

Having defined the Low Complexity test statistics for both $\hat{I}_{xx}^{\alpha}[\nu]$ and $\hat{Q}_{xx}^{\alpha}[\nu]$, they can now be combined as was the case for the Split-CAF GLRT,

$$\hat{T}_{SC-LC} = \frac{N\left(\left|\hat{I}_{xx}^{\alpha}[\nu]\right|^2 + \left|\hat{Q}_{xx}^{\alpha}[\nu]\right|^2\right)}{\hat{\sigma}_n^4}. \tag{5.39}$$

Finally, the test statistic can be computed as,

$$\hat{T}_{SC-LC} = \frac{N\left(\left|\hat{I}_{xx}^{\alpha}[\nu]\right|^2 + \left|\hat{Q}_{xx}^{\alpha}[\nu]\right|^2\right)}{\frac{1}{N}\sum_{n=0}^{N-1}\left|x[n]x^*[n-\nu]\right|^2}. \tag{5.40}$$

Since (5.37) and (5.38) are $\Gamma(1, 0.5)$ distributed, it follows that (5.40) is $\Gamma(2, 0.5)$ distributed, as confirmed in Figure 5.2.

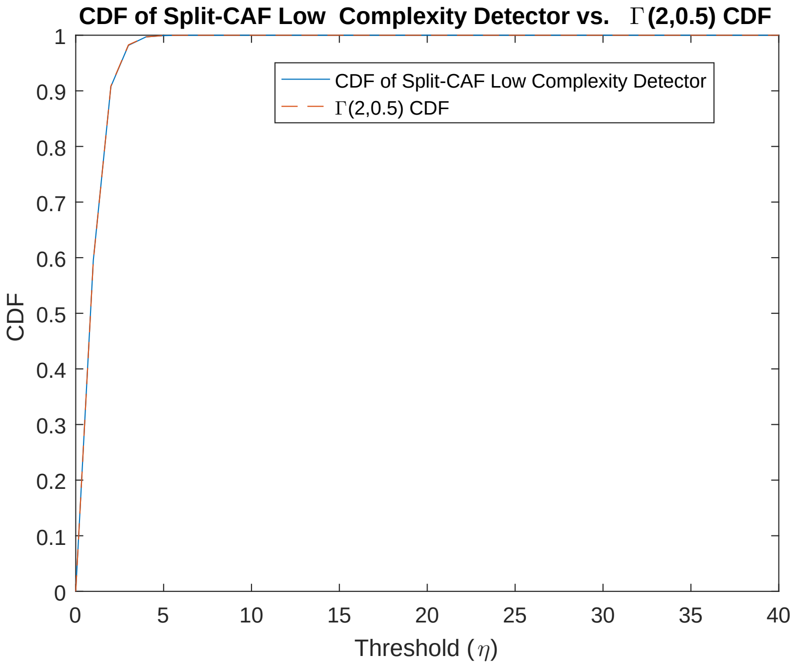


Figure 5.2: CDF of Split-CAF Low Complexity Detector vs. $\Gamma(2, 0.5)$ CDF

### 5.2.4 Split-CAF Spatial Sign Detector

As demonstrated in Chapter 4, the Spatial Sign detector has benefits related to the fact that it employs a test statistic with a reduced complexity and it has been shown to perform well in impulsive noise environments. Recall that after applying the spatial sign function, the CAF is given by the following,

$$\hat{R}_{ss}^{\alpha}[\nu] = \frac{1}{N}\sum_{n=0}^{N-1} s[n]s^{*}[n-\nu]e^{-j2\pi n\alpha}. \tag{5.41}$$

The component function, $\hat{I}_{xx}^{\alpha}[\nu]$, after applying the spatial sign function is,

$$\hat{I}_{ss}^{\alpha}[\nu] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(s[n]s^{*}[n-\nu])e^{-j2\pi\alpha n}. \tag{5.42}$$

Similarly, the component function, $\hat{Q}_{xx}^{\alpha}[\nu]$, after applying the spatial sign function is,

$$\hat{Q}_{ss}^{\alpha}[\nu] = \frac{1}{N}\sum_{n=0}^{N-1} \Im(s[n]s^{*}[n-\nu])e^{-j2\pi\alpha n}. \tag{5.43}$$

As with the original Spatial Sign detector, the elements of the covariance matrix reduce to constants. The covariance matrix in (5.35) becomes,

$$\mathbf{\Sigma_{ss_I}} = \begin{bmatrix} 1/4 & 0 \\ 0 & 1/4 \end{bmatrix}. \tag{5.44}$$

Therefore, it follows that a test statistic for $\hat{I}_{ss}^{\alpha}[\nu]$ is,

$$\hat{T}_I = 4N\left|\hat{I}_{ss}^{\alpha}[\nu]\right|^2, \tag{5.45}$$

which is $\chi_2^2$. Dividing by a factor of 4 leads to,

$$\hat{T}_I = N\left|\hat{I}_{ss}^{\alpha}[\nu]\right|^2, \tag{5.46}$$

which is $\Gamma(1, 0.5)$ distributed. A similar test statistic can be defined for $\hat{Q}_{ss}^{\alpha}[\nu]$,

$$\hat{T}_Q = N\left|\hat{Q}_{ss}^{\alpha}[\nu]\right|^2. \tag{5.47}$$

Combining (5.46) and (5.47) as for the previous two detectors, leads to a final test statistic for the Split-CAF Spatial Sign detector,

$$\hat{T}_{SC-SS} = N(\left|\hat{I}_{ss}^{\alpha}[\nu]\right|^2 + \left|\hat{Q}_{ss}^{\alpha}[\nu]\right|^2). \tag{5.48}$$

Figure 5.3 confirms the distribution of the test statistic under $H_0$.

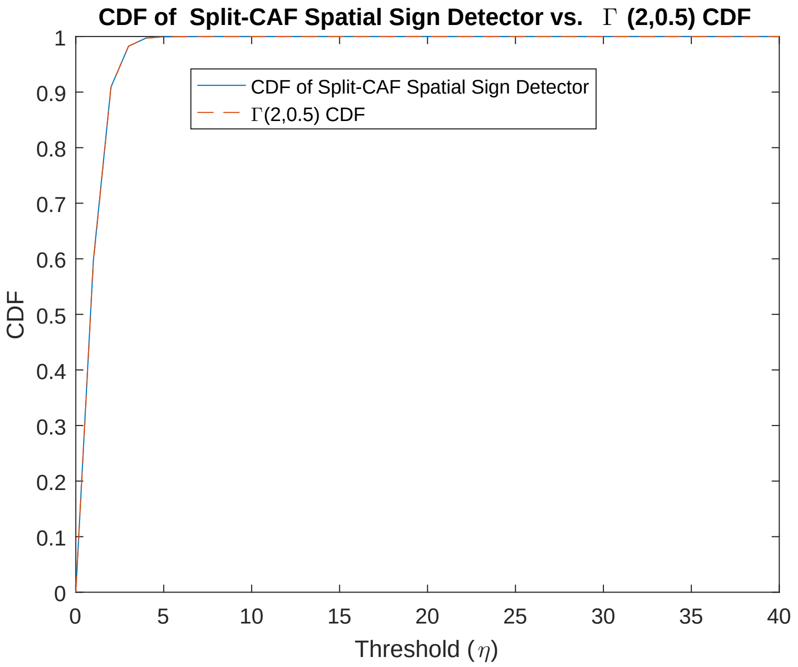


Figure 5.3: CDF of Split-CAF Spatial Sign Detector vs. $\Gamma(2, 0.5)$ CDF

### 5.2.5 Split-CAF Quantised Detector

In this section, a detector is proposed that attempts to exploit the cyclostationarity of the complex OFDM signal after its real and imaginary parts have been quantised to one of two levels, based on the sign of the samples. This approach was inspired by the simple formulation of the Spatial Sign detector, and its robustness to the effects of impulsive noise.

Taking an example, if the value of the real part of the input signal was -0.25, then the sample would be quantised to -1. Conversely, if the value was 0.25, then it would be quantised to 1. Firstly, let us express $x[n]$ as follows,

$$x[n] = x_r[n] + jx_i[n]. \tag{5.49}$$

The first step in deriving this detector is to confirm that the cyclostationarity of the signal is maintained after it has been quantised. The quantisation of $x_r[n]$ is equivalent to applying the

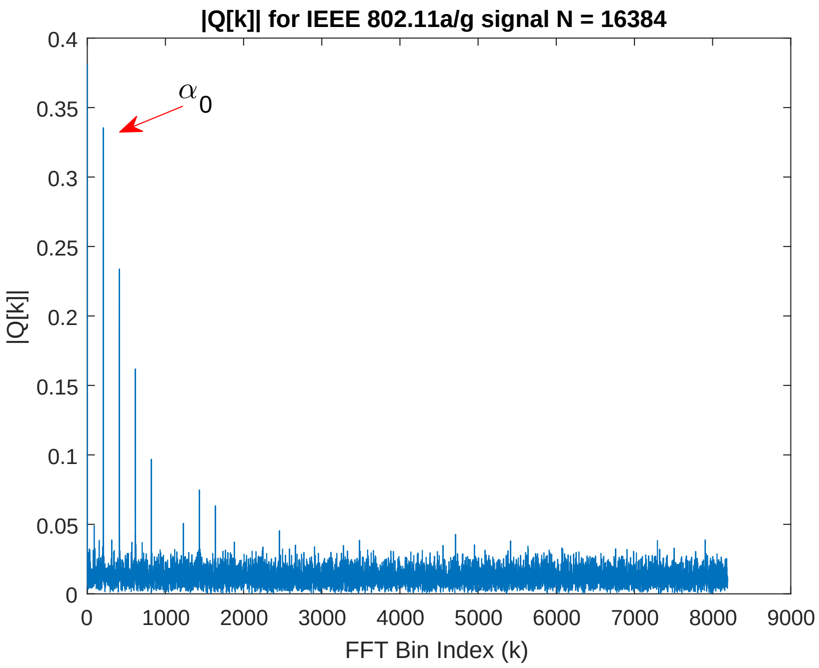


Figure 5.4: $\left|Q[k]\right|$ for IEEE 802.11a/g signal N = 16384

sign function which is expressed as follows,

$$q_r[n] = sgn(x_r[n]) = \begin{cases} -1, & \Re(x[n]) < 0, \\ 0, & \Re(x[n]) = 0, \\ 1, & \Re(x[n]) > 0, \end{cases} \tag{5.50}$$

where $q_r[n]$ will be used to denote the quantised $x_r[n]$. This can equally be defined for $x_i[n]$,

$$q_i[n] = sgn(x_i[n]) = \begin{cases} -1, & \Im(x[n]) < 0, \\ 0, & \Im(x[n]) = 0, \\ 1, & \Im(x[n]) > 0, \end{cases} \tag{5.51}$$

where $q_i[n]$ will be used to denote the quantised $x_i[n]$. As was the case in the derivation of the Spatial Sign detector, it is assumed that the chance of receiving a zero sample is small. The quantised complex data can be expressed as,

$$q[n] = q_r[n] + jq_i[n]. \tag{5.52}$$

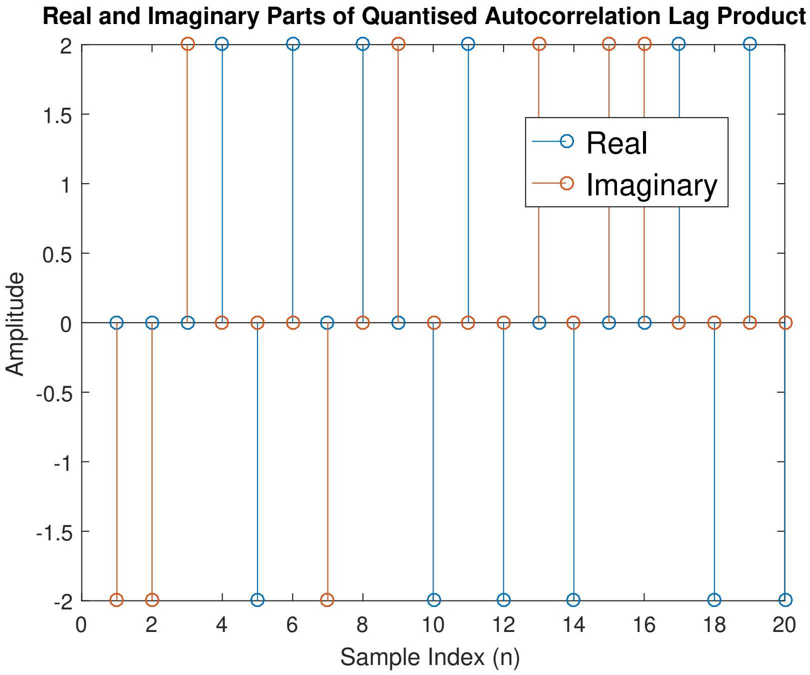


Figure 5.5: Real and Imaginary parts of $q[n]q^*[n-\nu]$

An estimate of the quantised CAF is defined as follows,

$$\hat{R}^{\alpha}_{qq}[\nu] = \frac{1}{N}\sum_{n=0}^{N-1} q[n]q^*[n-\nu]e^{-j2\pi n\alpha}. \tag{5.53}$$

This can be evaluated across a range of cyclic frequencies by taking the FFT of the quantised autocorrelation lag product $q[n]q^*[n-\nu]$ and scaling by factor of $1/N$. Figure 5.4 shows this for an IEEE 802.11a/g OFDM signal with $N = 16384$. The scaled FFT is denoted as $Q[k]$. It can be observed that the cyclostationarity has been retained after applying the quantisation as the spikes corresponding to $\alpha_0$ and its harmonics can clearly be seen protruding above the noise floor. Therefore, this would appear to confirm that it is possible to use the cyclostationarity of the quantised signal for detection of OFDM.

In order to formulate this detector using the same method as employed for the previous three detectors, it is necessary to look more closely at the quantised autocorrelation lag product $q[n]q^*[n-\nu]$. Due to the fact that the quantisation step has been applied, $q[n]q^*[n-\nu]$ can only have the values 2, -2, $2j$ or -$2j$. Therefore, $q[n]q^*[n-\nu]$ is not complex valued as was assumed in the formulation of the previous detectors, but is in fact only either real or imaginary. This means that when $\Re(q[n]q^*[n-\nu])$ has a value of 2, $\Im(q[n]q^*[n-\nu])$ is equal to 0 and vice-versa. Figure 5.5 shows $\Re(q[n]q^*[n-\nu])$ and $\Im(q[n]q^*[n-\nu])$ plotted together for a period of $n = 20$ samples.

The input to the quantiser is a complex normal white noise signal and $\nu = 64$. This confirms that $q[n]q^*[n-\nu]$ can only take the stated values. However, it is clear that $\Re(q[n]q^*[n-\nu])$ and $\Im(q[n]q^*[n-\nu])$ are separate signals and each carry information. Therefore, it is still possible to use the Split-CAF technique applied for the previous detectors. The first component CAF can be defined as,

$$\hat{I}^{\alpha}_{qq}[\nu] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(q[n]q^*[n-\nu])e^{-j2\pi\alpha n}. \tag{5.54}$$

The second component CAF is,

$$\hat{Q}^{\alpha}_{qq}[\nu] = \frac{1}{N}\sum_{n=0}^{N-1} \Im(q[n]q^*[n-\nu])e^{-j2\pi\alpha n}. \tag{5.55}$$

Assuming that no zero samples are received, both $q_r[n]$ and $q_i[n]$ can only take on the values of $-1$ or 1. Therefore,

$$\sigma^2 = \frac{1}{N}\sum_{n=0}^{N-1} \left|q[n]\right|^2 = 2. \tag{5.56}$$

Since $\sigma^2 = 2$, it follows that $\sigma^4 = 4$. This can be confirmed by visual inspection of Figure 5.5 since $\sigma^4$ can be calculated as,

$$\sigma^4 = \frac{1}{N}\sum_{n=0}^{N-1} \left|q[n]q^*[n-\nu]\right|^2. \tag{5.57}$$

This leads to the following covariance matrix for $\hat{I}^{\alpha}_{qq}[\nu]$,

$$\mathbf{\Sigma_{qq_I}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5.58}$$

Therefore, a test statistic for $\hat{I}^{\alpha}_{qq}[\nu]$ is,

$$\hat{T}_I = N\left|\hat{I}^{\alpha}_{qq}[\nu]\right|^2, \tag{5.59}$$

which will be $\chi^2_2$ distributed under $H_0$. An identical test statistic can be formulated for $\hat{Q}^{\alpha}_{qq}[\nu]$,

$$\hat{T}_Q = N\left|\hat{Q}^{\alpha}_{qq}[\nu]\right|^2. \tag{5.60}$$

As was done for the previous three detectors, (5.59) and (5.60) can be added together to produce a final test statistic,

$$\hat{T}_{SC-Q} = N\Big(\Big|(\hat{I}^{\alpha}_{qq}[\nu])\Big|^2 + \Big|(\hat{Q}^{\alpha}_{qq}[\nu])\Big|^2\Big). \tag{5.61}$$

Since both (5.59) and (5.60) are $\chi^2_2$ distributed, it follows that (5.61) is $\chi^2_4$ distributed. This is confirmed in Figure 5.6.

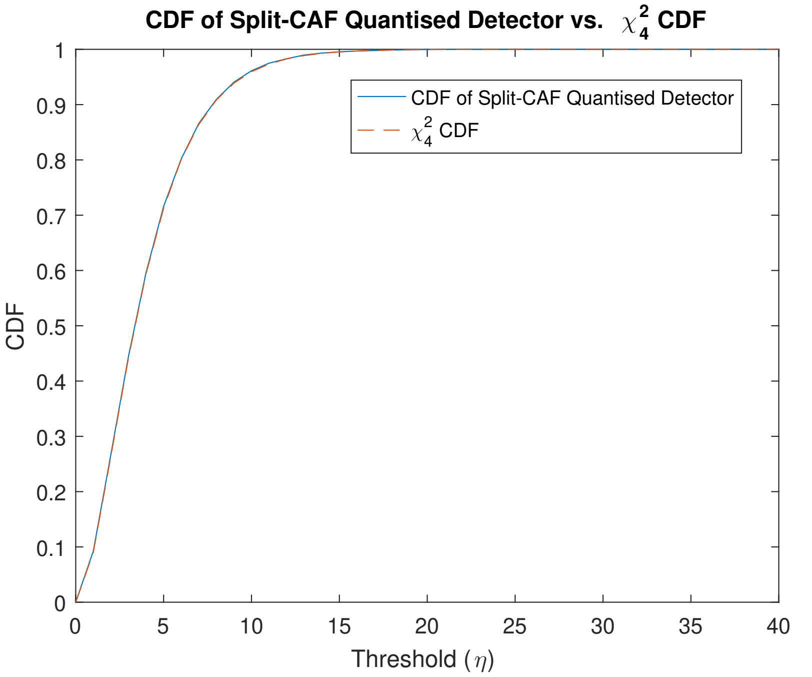


Figure 5.6: CDF of Split-CAF Quantised Detector vs. $\chi^2_4$ CDF

## 5.3 Performance Comparison with Previous Detectors

In this section, the performance of the proposed detectors will be compared with the performance of the detectors introduced in Chapter 4. Figure 5.7 compares $P_d$ vs. SNR curves for the GLRT, Low Complexity, Spatial Sign, Split-CAF GLRT, Split-CAF Low Complexity, Split-CAF Spatial Sign and Split-CAF Quantised detectors. Note that the abbreviation LC has been used to denote Low Complexity and S-C to denote Split-CAF in the figure legend. The test signal is IEEE 802.11a/g OFDM in AWGN, $N = 1600$ samples (20 OFDM symbols) and $P_{fa} = 0.1$. As was the case in Chapter 4, $P_d$ is calculated by computing 5000 test statistics at each tested SNR level.

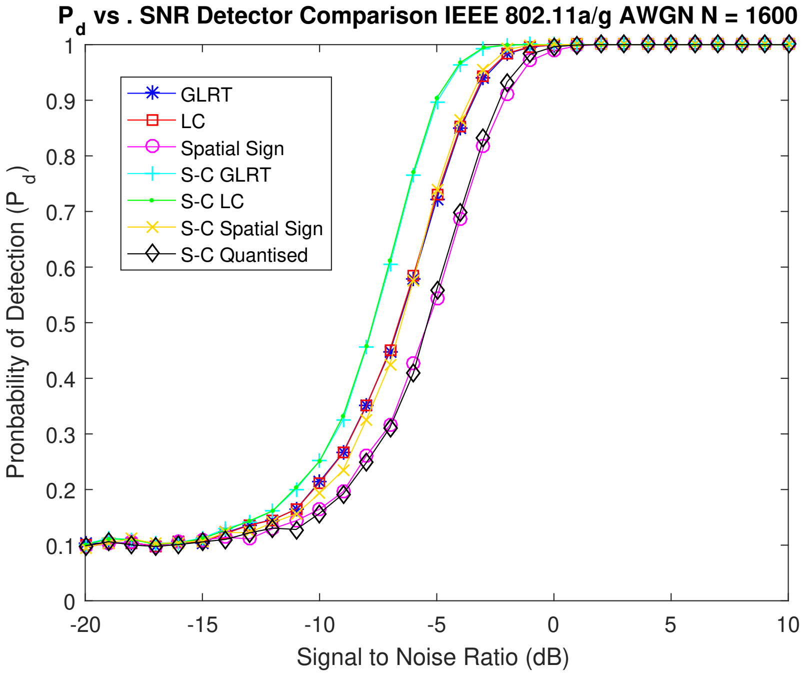


Figure 5.7: $P_d$ vs. SNR comparison of all detectors IEEE 802.11a/g in AWGN

It can be observed that the best overall performance in the AWGN channel is achieved by the Split-CAF GLRT and the Split-CAF Low Complexity detectors, since they have almost identical performance curves. The next best performance is achieved by the GLRT, Low Complexity and Split-CAF Spatial Sign detectors as their performance curves overlap closely. This result demonstrates that the Split-CAF Spatial Sign detector has been able to bridge the performance gap that exists between the GLRT, Low Complexity and Spatial Sign detectors in an AWGN channel, as was highlighted in Chapter 4. Finally, the poorest performance in an AWGN channel is exhibited by the Spatial Sign and Split-CAF Quantised detectors. These results clearly demonstrate the performance benefit that can be gained by employing the Split-CAF technique when compared to the traditional CAF. In order to highlight this further, Figure A.1 in Appendix A compares the Split-CAF GLRT and GLRT detectors, Figure A.2 compares the Split-CAF Low Complexity and Low Complexity detectors and Figure A.3 compares the Split-CAF Spatial Sign and Spatial Sign detectors. In all cases, it is clear that the proposed detectors outperform the existing algorithms, which are based on estimation of the traditional CAF. This is justified theoretically in Section 5.4.2.

Figure 5.8 compares all of the detectors for a DVB-T test signal in AWGN. In this case, $N_u = 2048$ and $N_{cp} = 512$, leading to a total $N_{ofdm} = 2560$. The CP is formed of 1/4 of the useful OFDM symbol. The observation interval is $N = 51200$ (20 OFDM symbols) and $P_{fa} =$

0.1.

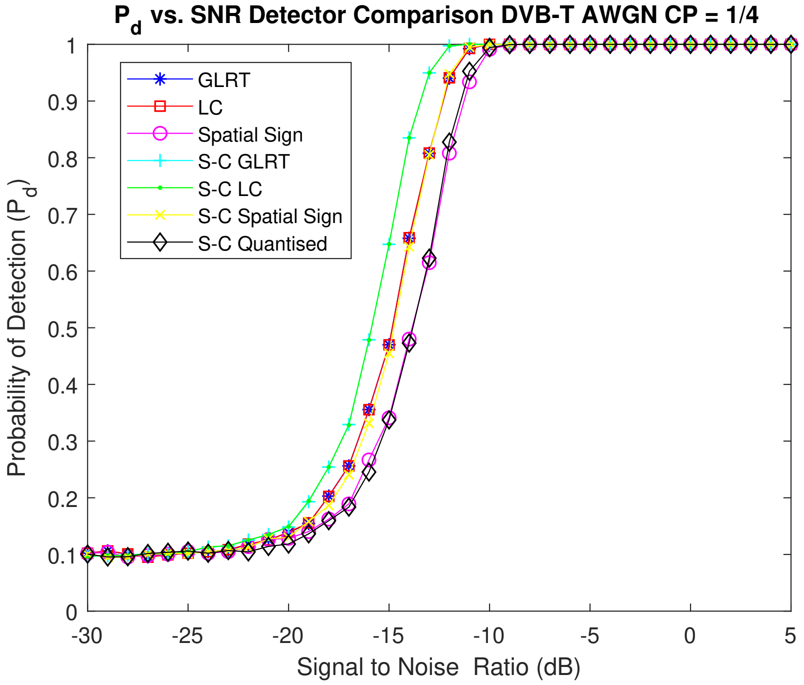


Figure 5.8: $P_d$ vs. SNR comparison of all detectors DVB-T in AWGN CP = 1/4

Again, it can be seen that the best overall performance in the AWGN channel is achieved by the Split-CAF GLRT and GLRT detectors. These are followed by the GLRT, Low Complexity and Split-CAF Spatial Sign detectors. Finally, the poorest performance is exhibited by the Spatial Sign and Split-CAF Quantised detectors. Furthermore, Figures A.4 - A.6 show the results for CP lengths of 1/8, 1/16 and 1/32 respectively.

As was the case in Chapter 4, it is instructive to assess the performances of the proposed detectors in impulsive noise environments. In the first instance, the contaminated Gaussian noise model in (4.57) will be employed. Figures 5.9 and 5.10 compare all of the detectors in contaminated Gaussian noise for the IEEE 802.11a/g and DVB-T test signals. As before, CP =1/4 for the DVB-T signal. Figures A.7 - A.9 show the results for CP = 1/8, 1/16 and 1/32 respectively.

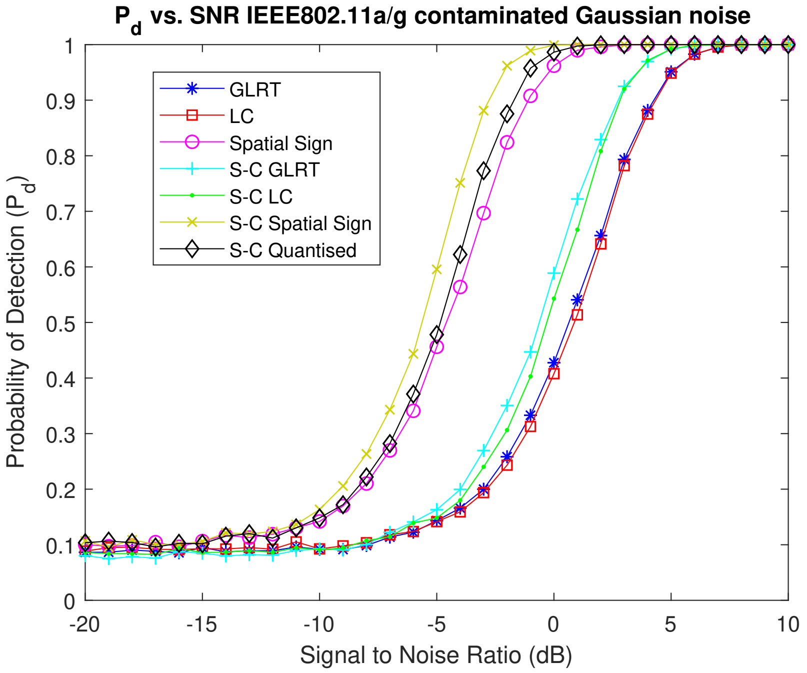


Figure 5.9: $P_d$ vs. SNR comparison all detectors for IEEE802.11a/g in contaminated Gaussian noise

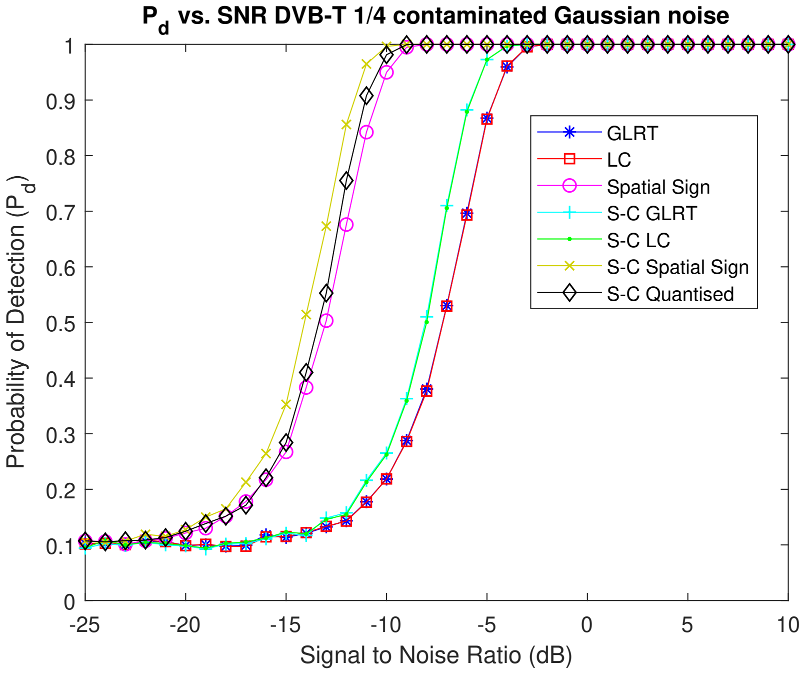


Figure 5.10: $P_d$ vs. SNR comparison all detectors for DVB-T CP = 1/4 in contaminated Gaussian noise

It can be observed that the Split-CAF Spatial Sign detector achieves the best performance in the contaminated Gaussian noise. This is closely followed by the Split-CAF Quantised and Spatial

Sign detectors. The Split-CAF Quantised detector provides a slight performance enhancement over the Spatial Sign detector, although this is very minor. As with the Spatial Sign detector, the robustness of the Split-CAF Spatial Sign and Split-CAF Quantised detectors is due to the fact that they apply a normalising operation. The next best performance is achieved by the Split-CAF GLRT and Split-CAF Low Complexity detectors albeit at a significantly reduced performance when compared to the AWGN channel. Finally, the poorest performance is exhibited by the GLRT and Low Complexity detectors. Therefore, it can be concluded that the proposed Split-CAF Spatial Sign detector achieves the best performance in the contaminated Gaussian noise, outperforming the Spatial Sign detector, which was shown to be the most robust to this type of noise in [58]. Also, both the Split-CAF GLRT and Split-CAF Low Complexity detectors perform better in this noise than their traditional CAF counterparts.

As was the case in Chapter 4, the performances of the detectors in Cauchy impulsive noise will now be assessed. Figures 5.11 and 5.12 show the results for IEEE 802.11a/g and DVB-T signals respectively. Interestingly, it can be observed that the most robust performance is achieved by the Split-CAF Quantised detector in Cauchy noise. This is very closely followed by the Split-CAF Spatial Sign detector, which performs marginally worse in all cases. This contrasts with the contaminated Gaussian noise case, where the Split-CAF Spatial Sign detector exhibited the best performance overall. However, both the Split-CAF Quantised and Split-CAF Spatial Sign detectors outperform the Spatial Sign detector.

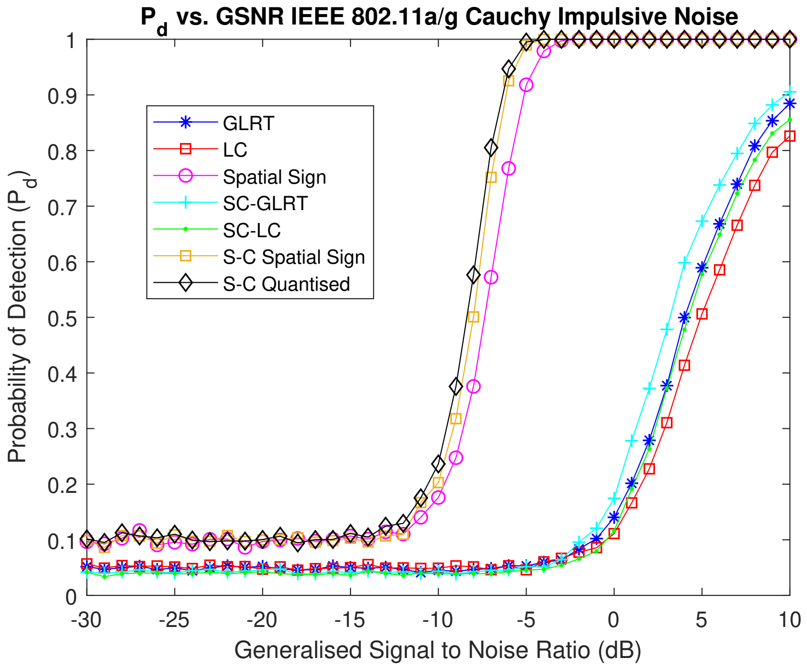


Figure 5.11: $P_d$ vs. GSNR detector comparison for IEEE802.11a/g in Cauchy Impulsive noise

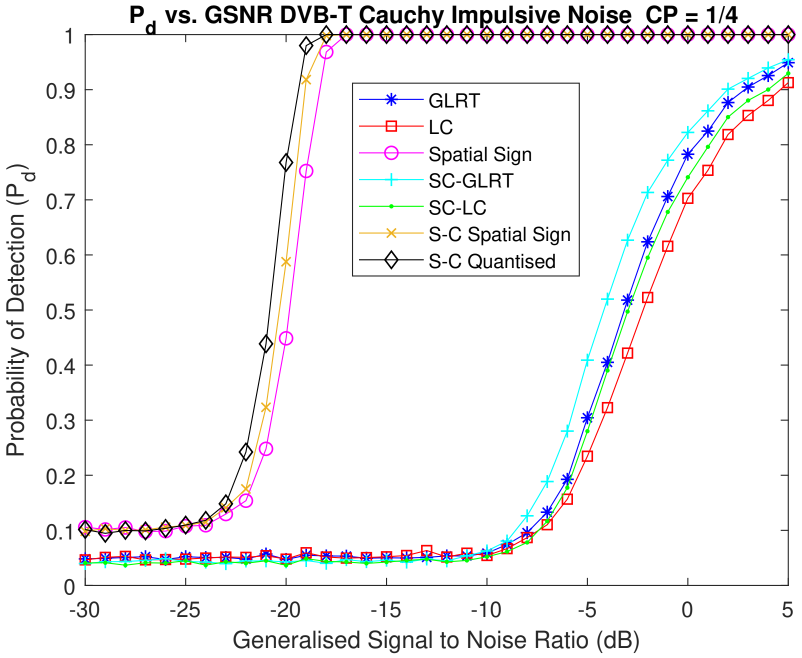


Figure 5.12: $P_d$ vs. GSNR detector comparison for DVB-T CP = 1/4 in Cauchy Impulsive noise

The Split-CAF GLRT and Split-CAF Low Complexity detectors show significantly degraded performance and do not achieve the theoretical false alarm rate. This is due to the reasons described in Section 4.8. Further results for different DVB-T CP lengths are shown in Figures A.10 - A.12.

Table 5.1 summarises the results presented in this section by ranking the performance of all of the detectors in both AWGN and impulsive noise environments, based on the tests conducted with IEEE 802.11a/g and DVB-T OFDM signals.

Table 5.1: Performance Ranking for detectors in AWGN and Impulsive Noise

| Detector | Ranking (AWGN) | Ranking (CGN) | Ranking (Cauchy) |
|---|---|---|---|
| GLRT | $2^{nd}$ | $5^{th}$ | $5^{th}$ |
| Low Complexity | $2^{nd}$ | $5^{th}$ | $7^{th}$ |
| Spatial Sign | $3^{rd}$ | $3^{rd}$ | $3^{rd}$ |
| Split-CAF GLRT | $1^{st}$ | $4^{th}$ | $4^{th}$ |
| Split-CAF Low Complexity | $1^{st}$ | $4^{th}$ | $6^{th}$ |
| Split-CAF Spatial Sign | $2^{nd}$ | $1^{st}$ | $2^{nd}$ |
| Split-CAF Quantised | $3^{rd}$ | $2^{nd}$ | $1^{st}$ |

Note that CGN denotes contaminated Gaussian noise. From Table 5.1, it is clear that if operating in an AWGN channel, the best detectors to choose are the Split-CAF GLRT and Split-CAF Low

Complexity detectors. In contaminated Gaussian noise, the best performing detector is the Split-CAF Spatial Sign detector. In contrast, the best performance in Cauchy noise is achieved by the Split-CAF Quantised detector. However, in order to determine the best performing detector, it is necessary to consider their performances in all environments jointly. With this in mind and referring to Table 5.1, it appears that the Split-CAF Spatial Sign detector achieves the highest overall ranking for all considered noise environments, achieving a ranking of 2nd place in AWGN, 1st in contaminated Gaussian noise and 2nd place in Cauchy noise.

## 5.4 Performance Under Radio Impairments

Having assessed the proposed algorithms in AWGN and impulsive noise, it is now instructive to examine their performance under common radio impairments. Specifically, the impact of impairments introduced by multipath channels and CFO will be considered.

### 5.4.1 Performance in Multipath Channels

For testing purposes, it will be assumed that an IEEE 802.11a/g signal is passed through a frequency-selective channel with a rayleigh PDF i.e. having no LOS component.

This form of channel can be modelled in MATLAB using the *stdchan* function, which generates a channel filter object that is passed to the $filter$ function along with the generated test signal. The *stdchan* function requires four parameters: the input sampling period, the maximum Doppler shift in Hz, the channel type and the RMS delay spread. In the case of an IEEE 802.11a/g signal, the input sampling period is $50ns$ and the channel type is “802.11g.”

Figure 5.13 compares $P_d$ vs. SNR for all of the detectors for a channel with $f_d = 200$Hz and a RMS delay spread of 250ns. Figures 5.14 and 5.15 compare the detectors for $f_d = 200$Hz and RMS delay spreads of $d_s = 500$ns and $d_s = 750$ns respectively. These can be considered moderate, high and very high delay spreads. The background noise is AWGN, the observation interval is $N = 1600$ samples and $P_{fa} = 0.1$.

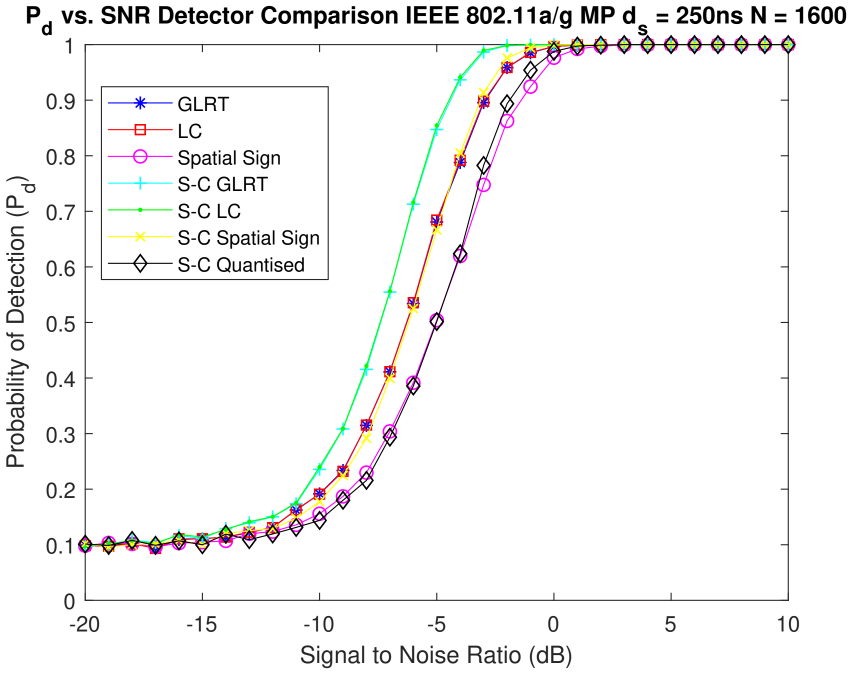


Figure 5.13: $P_d$ vs. SNR comparison IEEE 802.11a/g in Multipath Channel $d_s$ = 250ns

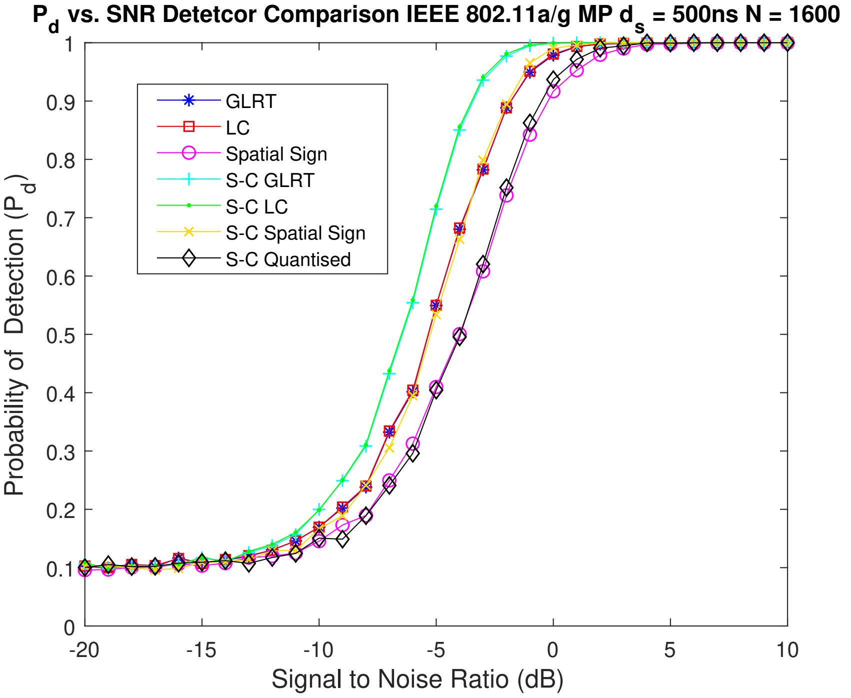


Figure 5.14: $P_d$ vs. SNR comparison IEEE 802.11a/g in Multipath Channel $d_s$ = 500ns

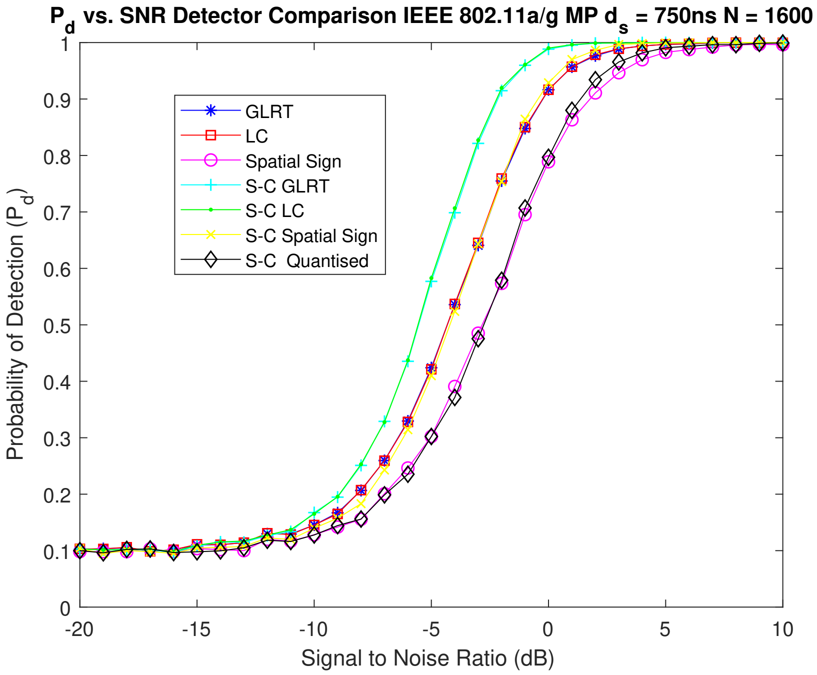


Figure 5.15: $P_d$ vs. SNR comparison IEEE 802.11a/g in Multipath Channel $d_s = 750$ns

It should be noted that, using (3.3) and (3.4), $f_d = 200$Hz is equivalent to a $T_c$ of approximately 2.5ms. This is greater than the detector observation interval, which is equal to 80 $\mu s$. Therefore, the multipath channel can be considered static, and only changing delay spread will affect the detector performance. Referring to Figures 5.13 - 5.15, it can be observed that increasing delay spread leads to decreasing detection performance for all detectors. This is caused by the fact that interference from other symbols reduces the strength of the correlation between the CP and the end of the OFDM symbol, making cyclostationary detection more difficult. As the delay spread increases in severity, the correlation becomes weaker and, therefore, the performance degrades accordingly. Figure 5.16 compares the performance of the Split-CAF GLRT for an AWGN channel only and channels with $d_s$ = 250ns, 500ns and 750ns.

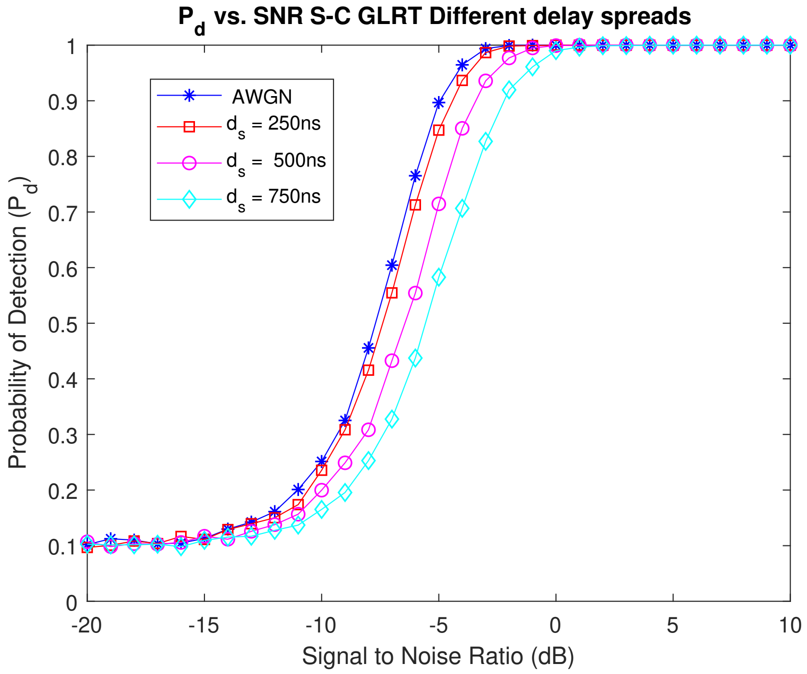


Figure 5.16: $P_d$ vs. SNR Split-CAF GLRT in AWGN with different delay spreads

As expected, it can be seen that as the delay spread increases, the performance of the detector reduces. However, the results show that detection is still viable in a multipath channel, albeit at a lower overall performance when compared to the ideal AWGN channel.

Having determined that the detectors are sensitive to the delay spread of the channel, it is now instructive to understand how they are affected by time varying channels. As reviewed in Chapter 3, the maximum Doppler spread ($D_s$) is approximately inversely proportional to the coherence time ($T_c$). For an observation interval of N = 1600 samples, the total duration is 20 OFDM symbols which is $80\mu s$ at $f_s$ = 20MHz. Figure 5.17 compares the performance of the detectors in a multipath channel with $d_s = 500ns$ and $T_c = 40\mu s$. This is equivalent to a coherence time of 10 OFDM symbols and using (3.3) and (3.4), $f_d = 12.5kHz$.

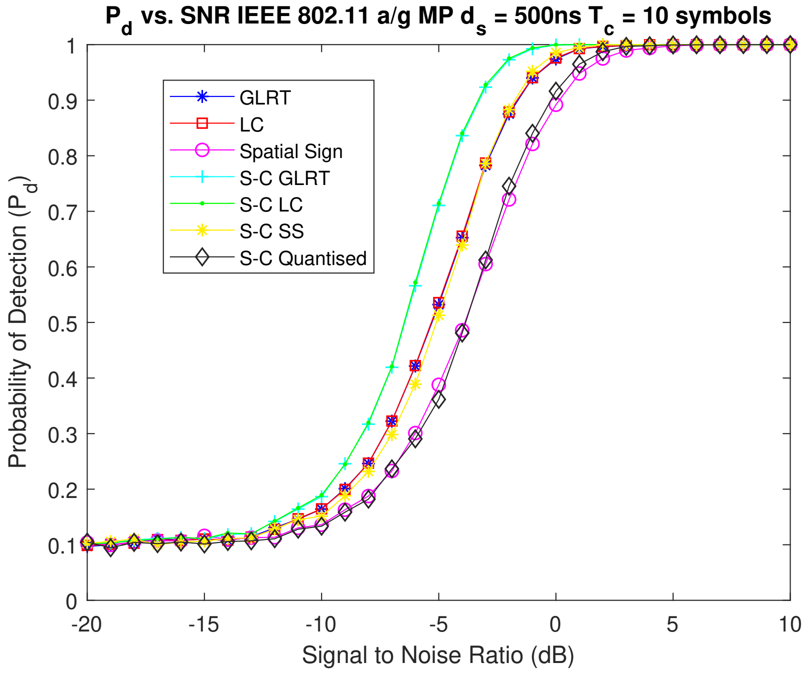


Figure 5.17: $P_d$ vs. SNR in Multipath with $d_s$ = 500ns and $T_c$ = 10 symbols

It can be observed that the performance of the detectors is unaffected for $T_c = 10$ symbols, when compared to the static channel in Figure 5.13. Figures 5.18, 5.19 and 5.20 show the results for $T_c = 5$ symbols, $T_c = 2$ symbols and $T_c = 1$ symbol. These correspond to $f_d = 25$kHz, 62.5kHz and 125kHz respectively.

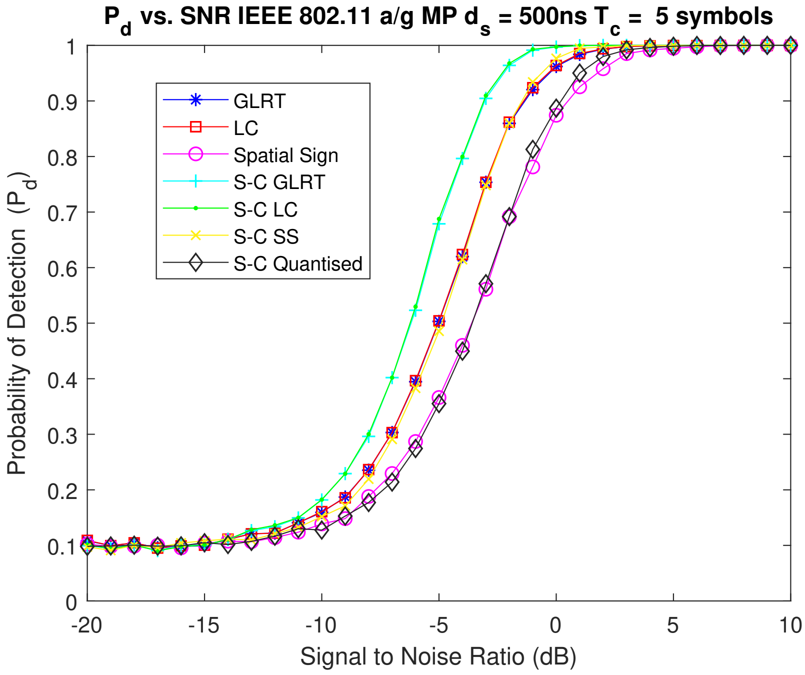


Figure 5.18: $P_d$ vs. SNR in Multipath with $d_s$ = 500ns and $T_c$ = 5 symbols

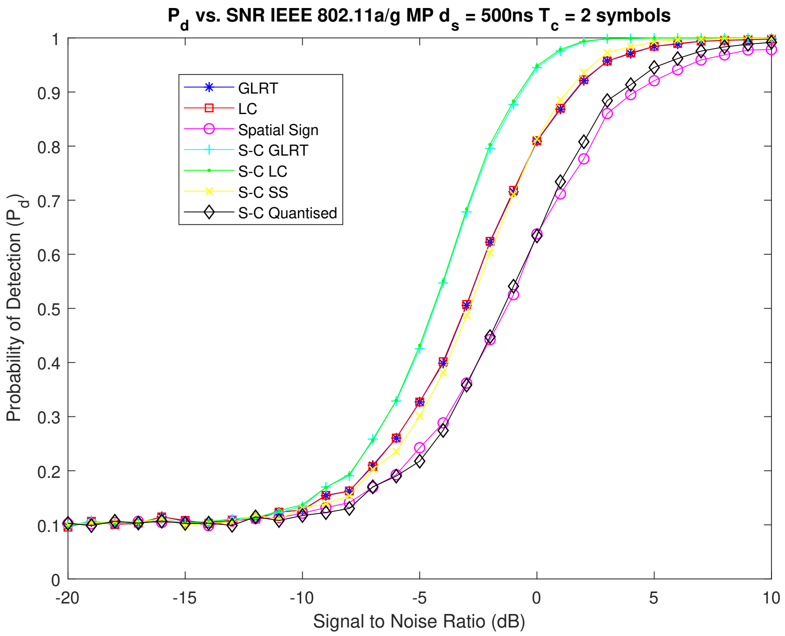


Figure 5.19: $P_d$ vs. SNR in Multipath with $d_s$ = 500ns and $T_c$ = 2 symbols

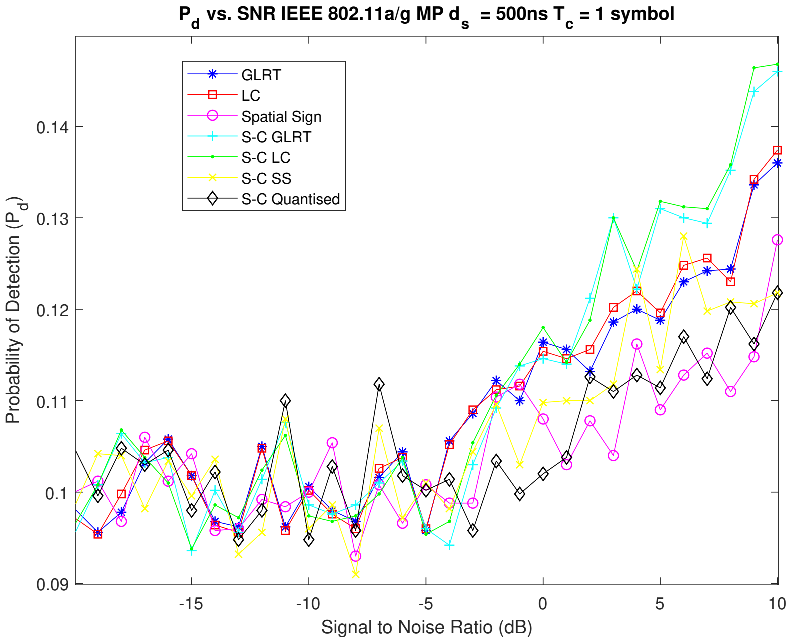


Figure 5.20: $P_d$ vs. SNR in Multipath with $d_s$ = 500ns and $T_c$ = 1 symbol

In Figure 5.18, a slight reduction in performance can be observed when compared to Figure 5.17 for $T_c$ = 10 symbols. However, in Figure 5.19 for $T_c$ = 2 symbols, the performance begins to degrade noticeably. Finally, for $T_c$ = 1 symbol in Figure 5.20, the cyclostationarity is completely corrupted and detection is no longer possible. Figure 5.21 compares the Split-CAF GLRT for a range of different coherence times.

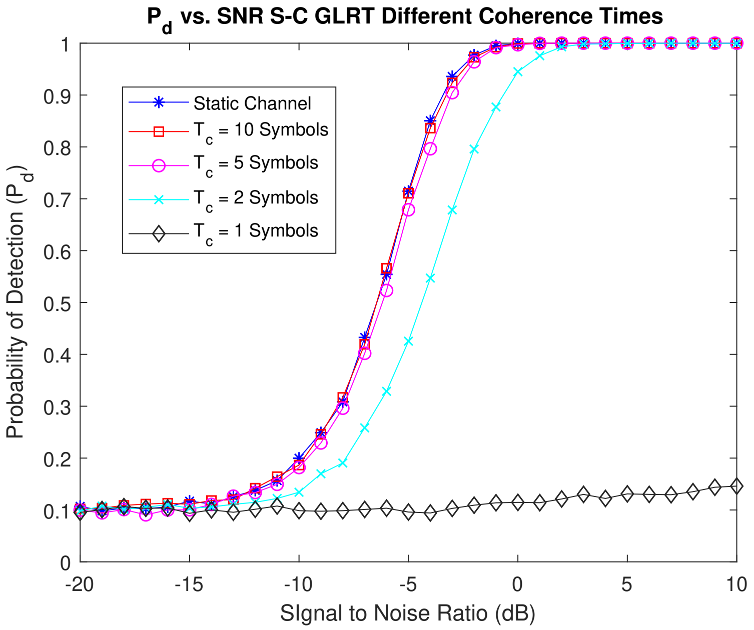


Figure 5.21: $P_d$ vs. SNR Split-CAF GLRT in AWGN with different coherence times

This shows that detection is still possible in channels where $T_c$ is less than the detector observation interval. However, as $T_c$ moves closer to the duration of an OFDM symbol, the effect of the time varying channel on the underlying cyclostationarity becomes more severe. As can be seen at a $T_c = 1$ symbol, the cyclostationarity is completely destroyed and, thus, detection is impossible.

To conclude, it can be stated that detection performance is sensitive to increasing delay spread of a multipath channel, because ISI absorbed by the CP leads to a reduction in the strength of the correlation between the CP and the end of the OFDM symbol. However, detection is still possible for moderate to high delay spreads, at a reduced performance when compared to the ideal channel. Furthermore, for fast varying channels, where $T_c$ approaches the OFDM symbol period, a marked degradation is observed when compared to the static channel. However, it has been shown that detection is still possible in time varying channels, where $T_c$ is less than the detector observation interval.

### 5.4.2 Carrier Frequency Offset

In any practical system, there will be a frequency offset between the transmitter and receiver, which will be translated to baseband. In OFDM receivers, this offset must be estimated and corrected in order to recover the data from the OFDM signal. This can be achieved using training symbols or the CP. However, it would be beneficial for the detectors to be robust to this impairment so that a frequency synchronisation stage is not necessary prior to detecting the signal. Figures 5.22 - 5.25 show the performances of the Split-CAF GLRT, Split-CAF Low Complexity, Split-CAF Spatial Sign and Split-CAF Quantised detectors for various different CFOs. In all cases, it is assumed that the value of CFO remains constant over the duration of the detector observation interval. The offsets range from 0Hz (no offset) to 600kHz in increments of 50kHz. Again, the test signal is IEEE 802.11a/g, $N = 1600$, $P_{fa} = 0.1$ and the background noise is AWGN.

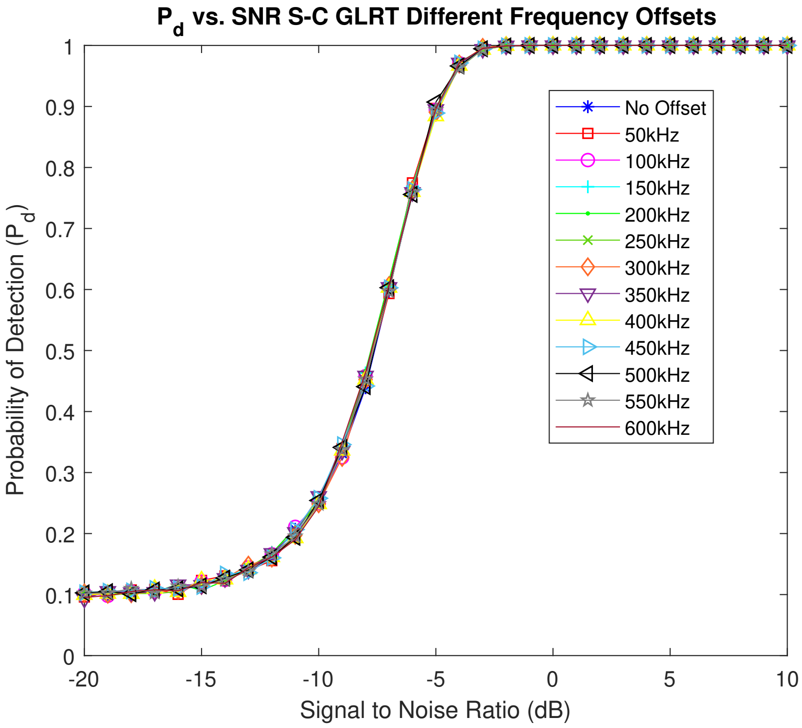


Figure 5.22: $P_d$ vs. SNR Split-CAF GLRT for Different Frequency Offsets

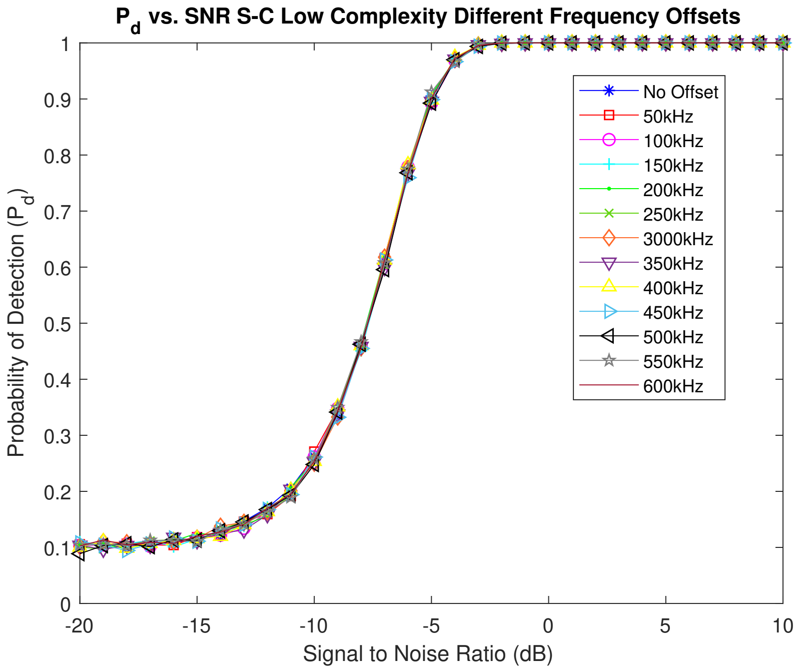


Figure 5.23: $P_d$ vs. SNR Split-CAF Low Complexity for Different Frequency Offsets

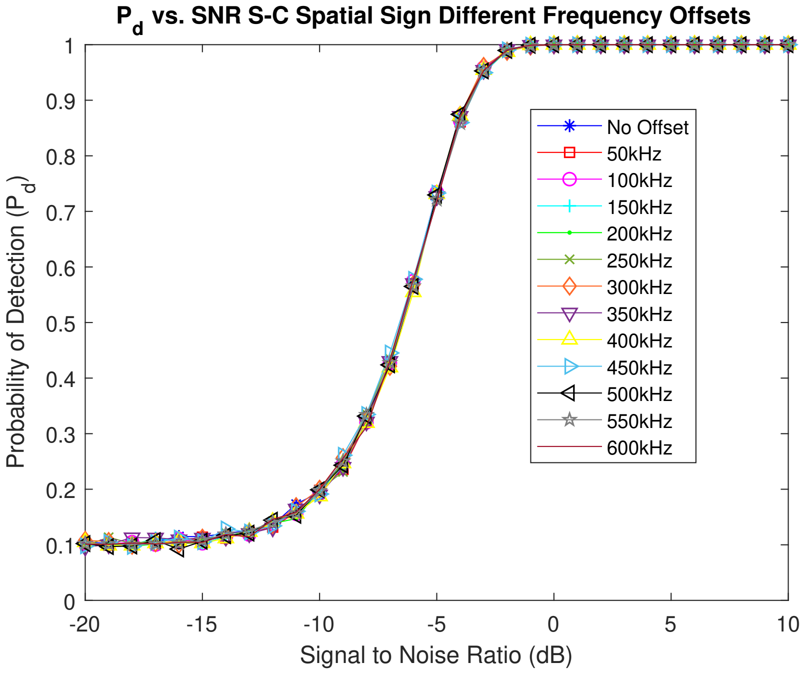


Figure 5.24: $P_d$ vs. SNR Split-CAF Spatial Sign for Different Frequency Offsets

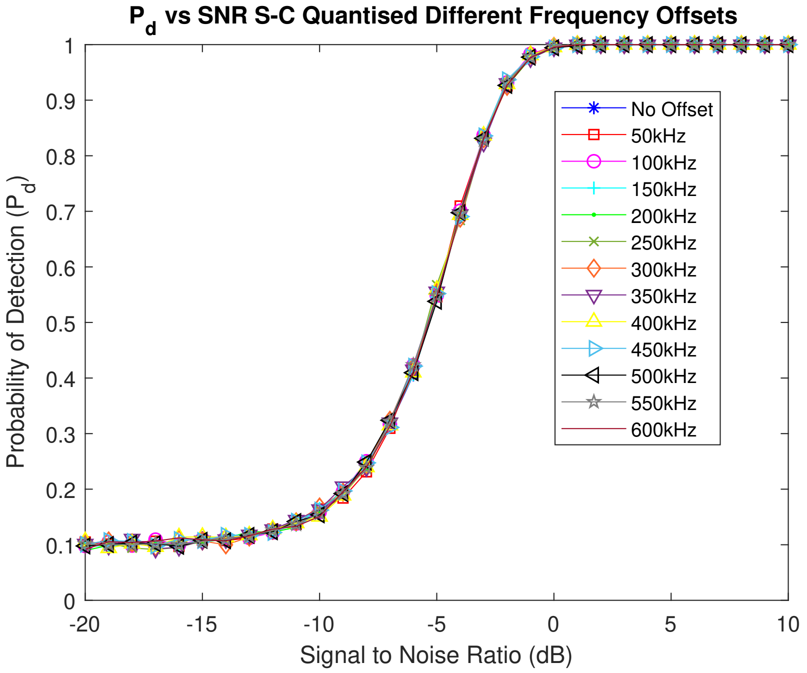


Figure 5.25: $P_d$ vs. SNR Split-CAF Quantised for Different Frequency Offsets

It is clear from the simulations, that there is no deviation in performance associated with the different frequency offsets. Therefore, no prior frequency synchronisation stage is required to use the proposed detectors. This is clearly very advantageous, as building in a frequency synchronisation stage would add unwanted complexity to the detection process.

In order to understand the reason that the proposed algorithms are resistant to the effects of CFO, it is necessary to examine the performances of $\hat{T}_I$ in (5.27) and $\hat{T}_Q$ in (5.28) for different frequency offsets. Recall that both of these test statistics were summed together to form a final test statistic for the Split-CAF GLRT detector. Figure 5.26 shows $P_d$ vs. SNR for $\hat{T}_I$ across a range of frequency offsets. Similarly, Figure 5.27 shows $\hat{T}_Q$ for the same offsets. Referring to Figure 5.27, it can be seen that in the ideal AWGN channel (i.e. with 0 Hz) offset, $P_d = P_{fa}$ for all tested SNRs. This indicates that the mean of $\hat{\mathbf{Q}}_{\mathbf{xx}}$ (5.6) does not change from zero when the signal is present, indicating that there is no detectable periodicity in the imaginary part of the autocorrelation lag product. Therefore, $\mathbf{Q}_{\mathbf{xx}}$ has a value of zero under both hypotheses and $\hat{T}_Q$ is always equal to a value drawn from the $\chi^2_2$ distribution. This proves that $\hat{T}_Q$ cannot be used as a detector. Conversely, the real part of the autocorrelation lag product is periodic when the signal is present. Therefore, the performance improvement provided by the Split-CAF GLRT over the GLRT is attributable only to $\hat{T}_I$. This improvement in performance is due to the fact that $\hat{T}_I$ actually discards the imaginary part of the autocorrelation lag product, which

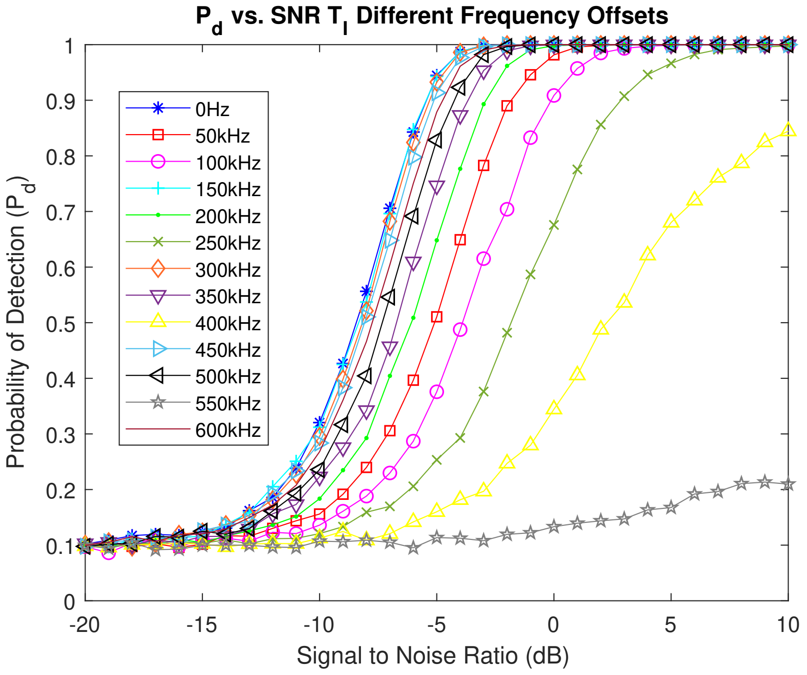


Figure 5.26: $P_d$ vs. SNR $\hat{T}_I$ for Different Frequency Offsets

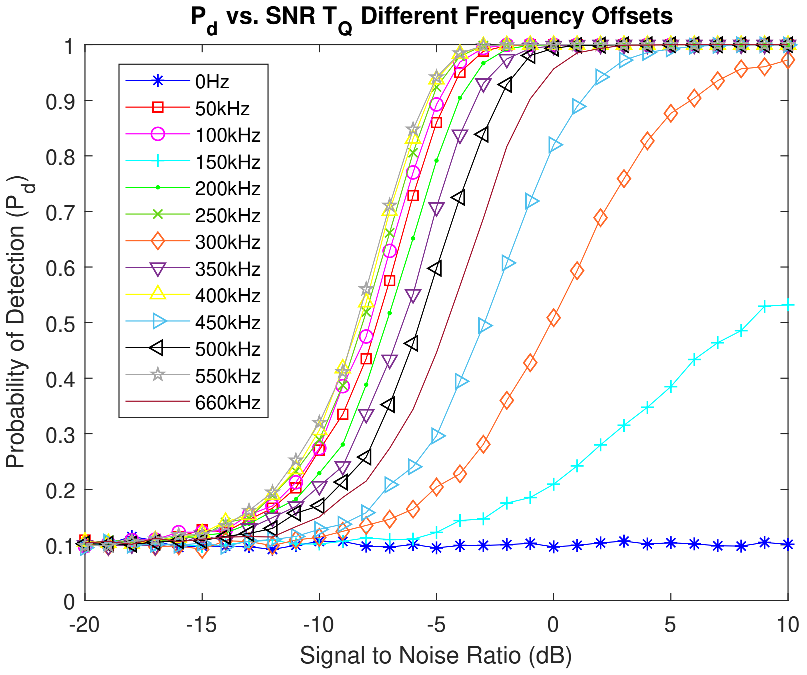


Figure 5.27: $P_d$ vs. SNR $\hat{T}_Q$ for Different Frequency Offsets

is effectively a noise term and reduces the strength of the correlation with the cyclic frequency. However, it can be observed from Figure 5.26, that the performance of $\hat{T}_I$ varies depending on the frequency offset. For example, for an offset of 550kHz, the performance of $\hat{T}_I$ is degraded severely.

At the same offset, $\hat{T}_Q$ shows excellent performance. As established previously and shown in Figure 5.27, the periodicity cannot be detected in the imaginary part of the autocorrelation lag product when no frequency offset exists. Therefore, it is clear that the frequency offset causes a distortion which leads to the unnatural appearance of periodicity in the imaginary part of the autocorrelation lag product.

In order to understand the nature of this distortion, a mathematical analysis is required. Note that this analysis will also include a carrier phase offset, which is likely to exist along with the frequency offset. Again, it will be assumed that the frequency and phase offsets remain constant for the duration of the detector observation interval. In the presence of both impairments, the received signal can be expressed as,

$$x[n]e^{-j2\pi\Delta fn}e^{-j\theta}, \tag{5.62}$$

where $x[n]$ denotes the complex data, $\Delta f$ is the normalised frequency offset, $n$ is the sample index and $\theta$ is the phase offset. It follows that the autocorrelation lag product is given by,

$$x[n]e^{-j2\pi\Delta fn}e^{-j\theta}x^*[n-\nu]e^{j2\pi\Delta f(n-\nu)}e^{j\theta}, \tag{5.63}$$

which can be rewritten as,

$$x[n]x^*[n-\nu]e^{-j2\pi\Delta fn}e^{j2\pi\Delta fn}e^{-j2\pi\Delta f\nu}e^{-j\theta}e^{j\theta}. \tag{5.64}$$

This can be simplified to give,

$$x[n]x^*[n-\nu]e^{-j2\pi\Delta f\nu}. \tag{5.65}$$

It is clear that the carrier phase offset has no effect on the autocorrelation lag product. However, the frequency offset introduces a phase distortion that is dependent on $\Delta f$ and $\nu$. With a frequency offset, the real and imaginary parts of the autocorrelation lag product can be expressed as,

$$\begin{aligned} z_r[n] &= x_r[n]cos(2\pi\Delta f\nu) + x_i[n]sin(2\pi\Delta f\nu), \\ z_i[n] &= -x_r[n]sin(2\pi\Delta f\nu) + x_i[n]cos(2\pi\Delta f\nu), \end{aligned} \tag{5.66}$$

where $z_r[n]$ and $z_i[n]$ are the real and imaginary parts of the autocorrelation lag product with a frequency offset and $x_r[n]$ and $x_i[n]$ are the real and imaginary parts with no offset. Therefore,

$z_r[n]$ and $z_i[n]$ consist of both $x_r[n]$ and $x_i[n]$ terms, and the contribution of each term is dependent on $\Delta f$.

In order to illustrate the effects of this distortion with an example, consider values of $2\pi\Delta f\nu$ that cause $cos(2\pi\Delta f\nu)$ to equal 0 and $sin(2\pi\Delta f\nu)$ to equal 1 or -1. The first of these is a phase shift of $\pi/2$ and the value of $\Delta f$ which causes this phase shift can be determined by solving,

$$\Delta f = \frac{\pi/2}{2\pi\nu}. \tag{5.67}$$

With $\nu = 64$, this phase shift is caused by a frequency offset equal to 1/4 of the OFDM sub-carrier spacing. This condition will also be true for all phase shifts that are odd multiples of $\pi/2$, which corresponds to all offsets that are odd multiples of 1/4 of a sub-carrier spacing i.e. 3/4, 5/4, 7/4 etc. In each of these cases, (5.66) reduces to,

$$\begin{aligned} z_r[n] &= \pm x_i[n], \\ z_i[n] &= \pm x_r[n]. \end{aligned} \tag{5.68}$$

Therefore, for these frequency offsets, the phase distortion causes the real and imaginary parts of the autocorrelation lag product to swap places. This example serves to illustrate the reason that periodicity can be observed in the imaginary part of the autocorrelation lag product when a frequency offset is present in the system. Note that an offset of 7/4 of the sub-carrier spacing is equivalent to $546.875kHz$. Clearly, this value is very close to $550kHz$, which explains why the performance of $\hat{T_I}$ is degraded severely for this frequency offset, as seen in Figure 5.26.

To conclude this analysis, the improved performance of the Split-CAF GLRT is attributable to $\hat{T_I}$. This improvement occurs because $\hat{T_I}$ discards the imaginary part of the autocorrelation lag product, which is effectively a noise term. However, owing to the phase distortion introduced by CFO, $\hat{T_I}$ cannot be used without a prior frequency synchronisation stage. Therefore, combining both $\hat{T_I}$ and $\hat{T_Q}$ as for the Split-CAF GLRT provides a performance enhancement over the GLRT detector, while simultaneously providing immunity to CFO. Note that the Split-CAF GLRT will exhibit a slightly reduced performance when compared to $\hat{T_I}$ with no offset, due to the fact that it is formed by adding $\hat{T_Q}$, which leads to a higher overall threshold. This analysis can equally be used to explain the improved performance of the Split-CAF Low Complexity and Split-CAF Spatial Sign detectors over their CAF counterparts.

## 5.5 Chapter Summary

In conclusion, this chapter has introduced a novel method whereby the traditional CAF is replaced by the computation of two parallel component CAFs, whose arguments are the real and imaginary parts of the autocorrelation lag product. The component CAFs are used to compute two test statistics, that are summed together to form a final test statistic. This approach has been used to derive re-formulated versions of the GLRT, Low Complexity and Spatial Sign detectors introduced in Chapter 4. It has also been used in the derivation of a new test statistic that exploits the cyclostationarity of the OFDM signal after it has been quantised. It was found that the Split-CAF GLRT, Split-CAF Low Complexity and Split-CAF Spatial Sign detectors outperformed their traditional counterparts in the detection of both IEEE802.11a/g and DVB-T OFDM signals, in AWGN and impulsive noise channels.

It was determined that the best performance in an AWGN channel was achieved using either the Split-CAF GLRT or the Split-CAF Low Complexity detectors. The best performance in contaminated Gaussian impulsive noise was achieved by the Split-CAF Spatial Sign detector. Finally, the best performance in Cauchy impulsive noise was achieved by the Split-CAF Quantised detector. When considering performance in both AWGN and impulsive noise channels, it was clear that the most robust solution is the proposed Split-CAF Spatial Sign detector. Furthermore, it was found that the detectors can perform robustly in multipath channels and in the presence of CFO. It was established that using the Split-CAF approach is the only way to guarantee improved performance over traditional CAF based methods, while simultaneously guaranteeing robust performance in the presence of CFO. However, it was found that their performance in multipath channels is sensitive to increasing delay spread and decreasing coherence times.

# Chapter 6

# FPGA Targeting of Cyclostationary Feature Detectors

In this chapter, the cyclostationary feature detection algorithms discussed in the previous chapters are implemented in HDL Coder and targeted to a Xilinx 7 Series FPGA. The chapter begins by introducing *HDL Coder* and *Vivado*, which are used for FPGA targeting. HDL Coder is a tool in MATLAB that enables conversion of high-level MATLAB/Simulink models to synthesisable HDL code (VHDL and Verilog) for programming of FPGAs. Vivado is a software suite that is used to perform the synthesis and implementation stages in the FPGA design flow, and for the analysis of important design metrics such as resource utilisation and achievable clock frequency.

The main contribution of this chapter is the design and analysis of FPGA implementations of the cyclostationary detection algorithms discussed in Chapter 4 and 5. Having reviewed the details of the implementation of each of the algorithms in HDL Coder, an analysis is provided of their relative costs in terms of low level resources when targeting a Xilinx 7 Series FPGA. From a design perspective, a major goal is ensuring that hardware cost is minimised as much as possible. With this in mind, two design choices are recommended for implementation of the detection algorithms. Firstly, it is suggested that the FPGA design technique of multi-channel hardware sharing is applied when implementing the proposed detectors in Chapter 5, in order to ensure that hardware resources are not used unnecessarily. Secondly, it is recommended that a simple mathematical re-arrangement is applied to test statistics that are expressed as a fraction, which renders the division operation redundant.

## 6.1 HDL Coder

In earlier versions of the software, MATLAB could only be used for the initial simulation process. However, with the introduction of tools such as MATLAB Coder [91] and HDL Coder [23], it is now possible to generate deployable code, including software code and HDL from high level MATLAB and Simulink models.

The HDL Coder tool allows a subset of MATLAB and Simulink functionality to be converted to VHDL and Verilog code that can then be targeted to FPGAs. Although both MATLAB code and Simulink models can be used to generate HDL, focus will be placed exclusively on Simulink in this chapter. Furthermore, only VHDL code will be generated and Xilinx FPGAs will be targeted.

The HDL Coder block library consists of blocks drawn from the Simulink library, the Communications System toolbox, the DSP System toolbox and Stateflow. More recently, functionality has been added for computer vision applications. It also includes more generic HDL blocks such as Single and Dual Port RAM blocks, and FIFOs. HDL Coder also provides several HDL optimized blocks for fundamental signal processing functionality, such as a polyphase FIR block enabling fractional sampling rate conversion, a Numerically Controlled Oscillator (NCO) block for synthesis of sine and cosine waves, and a CORDIC block operating in vectoring mode for calculation of the magnitude and angle of a complex number. The process for developing an HDL Coder model in Simulink that is employed in this Chapter is illustrated in Figure 6.1.

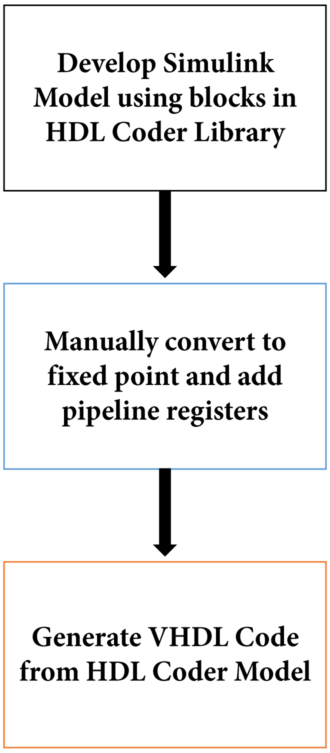

Develop Simulink Model using blocks in HDL Coder Library
Manually convert to fixed point and add pipeline registers
Generate VHDL Code from HDL Coder Model

Figure 6.1: Illustration of HDL Coder Design flow

The algorithm is first designed in Simulink with a hierarchical structure using only blocks available in the HDL Coder block library. The design hierarchy is delineated using subsystem blocks in Simulink. The top-level subsystem block corresponds to the top-level entity of the VHDL design and any subsystems within the top-level subsystem correspond to the components of the top-level design. The inputs and outputs of the subsystem correspond to ports in VHDL. The clock signal is derived from the fastest rate in the model. If multiple rates exist in the design, HDL Coder has two clocking modes available. The first method admits a single master clock to the top-level entity that is based on the fastest rate in the model, and the clocks required for the slower rates are derived from the master clock using clock enables. The second method creates a set of clock inputs to the top-level entity, where each clock corresponds to a different rate in the HDL Coder model. By default, HDL Coder employs the first method if multiple rates exist in the model.

The next stage is the manual setting of fixed point wordlengths throughout the HDL Coder model. In DSP applications, samples are expressed using a finite precision binary representation. In FPGAs, the convention is to use fixed point binary numbers due to the fact that they are less computationally intensive to handle than floating point binary numbers. In creating an FPGA design using fixed point arithmetic, a trade-off exists between achieving sufficient precision and accuracy and controlling hardware cost. A greater precision is achieved by using longer wordlengths, but this leads to a larger overall hardware cost. Therefore, it is important to bear this in mind when choosing wordlengths at the various stages of an FPGA design. When discussing fixed point wordlengths, it is instructive to use the MATLAB/Simulink notation which is as follows,

$$(sign, WL, FL). \tag{6.1}$$

The *sign* argument specifies whether the number is signed or unsigned, *WL* specifies the total wordlength and *FL* specifies the fractional wordlength. For example, (1,16,14) specifies a 16 bit signed number with 2 integer bits and 14 fractional bits.

In synchronous operation, all operations in an FPGA design are synchronised to a clock signal. Having a higher achievable clock signal frequency is desirable as it means that higher input sampling frequencies can be handled. Furthermore, ensuring that a design can be clocked at a higher frequency than the required input sampling rate can be useful for enabling time sharing of hardware, which can lead to a reduction in the resource cost of a particular implementation. The maximum achievable clock frequency of a design is dictated by the critical path. This is defined as the longest combinatorial path between clocked elements in the design, i.e FFs and is characterised by the critical path delay. The critical path delay is composed of the combinatorial delay and the routing delay and is related to the maximum clock frequency of a design as

$$f_{max} = \frac{1}{\tau_{cp}}, \tag{6.2}$$

where $f_{max}$ is the maximum clock frequency and $\tau_{cp}$ is the critical path delay. In order to minimise the critical path delay and thus increase the maximum achievable clock frequency, the technique of pipelining is employed, where registers are placed between elements in the processing chain. This process is conducted in accordance with the rules of cut set re-timing theory. HDL Coder has the facility to enable automatic insertion of pipeline registers. However, this will be done manually for all designs in this chapter in order to retain explicit control of the HDL design.

Having completed the previous two design stages in Figure 6.1, the final stage of the HDL Coder process is to generate the VHDL Code from the Simulink model. This is achieved by calling the function *makehdl* in MATLAB with the name of the top-level subsystem as an input. An alternative is to use the HDL Workflow Advisor Graphical User interface (GUI). As a matter of preference, the *makehdl* function is employed in this chapter. The generated HDL can then be exported for synthesis and implementation using an FPGA design tool such as Vivado [92]. Other competing high level tools for HDL Code generation include Xilinx System Generator (which comes as an add-on to Simulink), and Vivado HLS which enables code written in C, C++ and SystemC to be converted to VHDL and Verilog.

## 6.2 Vivado

Vivado is an Integrated Development Environment (IDE) for FPGA design that allows developers to synthesise and implement HDL code for targeting of Xilinx-7 series and Ultrascale FPGAs, as well as facilities to design systems for SoC devices such as the Zynq-7000 and Zynq Ultrascale + devices. Vivado replaced Xilinx ISE and includes an in built HDL simulator and new features such as Vivado HLS and Vivado IP integrator. The Vivado design flow as employed in this thesis is illustrated in Figure 6.2. The first stage of the process is to create a new project in Vivado. This is followed by adding the VHDL source files that were generated in HDL Coder. The third stage is to add a Xilinx Design Constraints (XDC) file that is defined using the TCL command tool language. Constraints are required in order for Vivado to produce an implementation that meets the various requirements of a given design. Of particular interest is the timing constraint, which is added to ensure that a design can meet a defined target clock frequency. Having added the constraints, the synthesis and implementation processes are carried out and a bitstream is obtained for programming of the FPGA. For all of the designs discussed in this chapter, steps 1 to 6 will be followed and the resource utilisation reports will be analysed in order to compare the relative costs of the various cyclostationary detection algorithms.

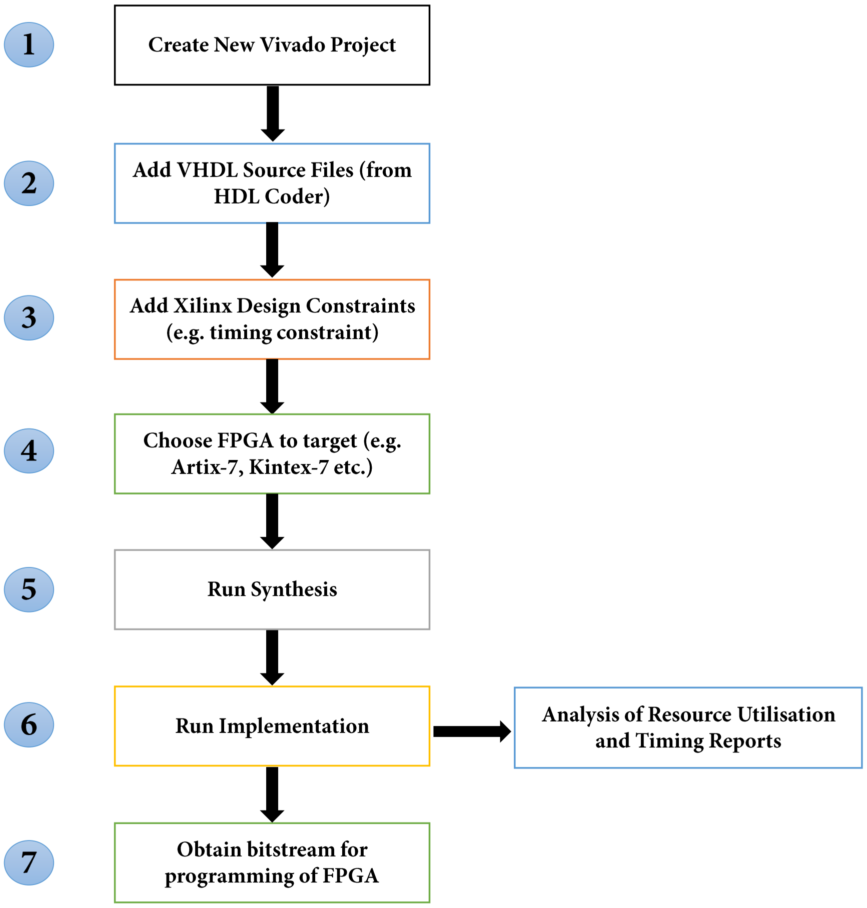

1
Create New Vivado Project
2
Add VHDL Source Files (from HDL Coder)
3
Add Xilinx Design Constraints (e.g. timing constraint)
4
Choose FPGA to target (e.g. Artix-7, Kintex-7 etc.)
5
Run Synthesis
6
Run Implementation
Analysis of Resource Utilisation and Timing Reports
7
Obtain bitstream for programming of FPGA

Figure 6.2: Illustration of Vivado Design Flow

## 6.3 Multi-Channel Hardware Sharing

In this section, a short review of of multi-channel time sharing of FPGA hardware will be provided, as this will become a central component of the implementation of the proposed Split-CAF cyclostationary detection algorithms introduced in Chapter 5. The concept of multi-channel hardware sharing is borne out of a desire to minimise the hardware cost of FPGA designs that involve applying the same signal processing hardware to multiple input data streams.

In order to illustrate this concept, consider an example where there are two parallel input data streams, each of which must be passed through the same FIR filter as shown in Figure 6.3.

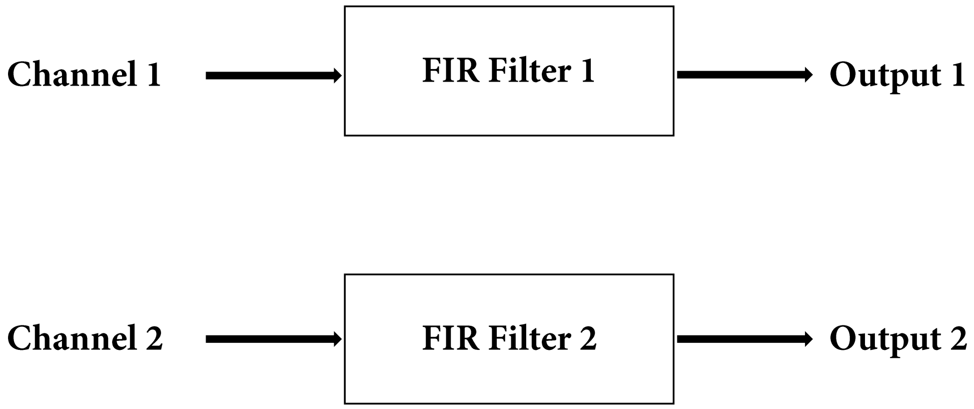


Figure 6.3: Example of multiple independent signal processing channels

The data stream on each of the independent channels is processed using the same FIR filter. Therefore, the resource cost of the total design is equivalent to two times the cost of a single FIR filter. As the number of channels increases, the cost of implementing multiple instances of the same FIR filter may become prohibitively expensive. With this in mind, it is instructive to consider how to design the hardware such that a single instance of the FIR filter can be shared between each of the input channels, making the hardware cost equivalent to one FIR filter instead of two.

In order to achieve this, a fundamental element of cut set timing theory, time scaling, can be exploited. The time scaling operation involves scaling each of the delay elements in a signal processing block by a positive integer factor $b$. In the z-domain, applying a time scaling of $b$ clock cycles to a delay of of one clock cycle results in a delay equal to

$$z^{-b}. \tag{6.3}$$

Figure 6.4 shows the Signal Flow Graph (SFG) for a parallel FIR filter with four coefficients. In the standard formulation, each of the delays in the filter is equal to $z^{-1}$.

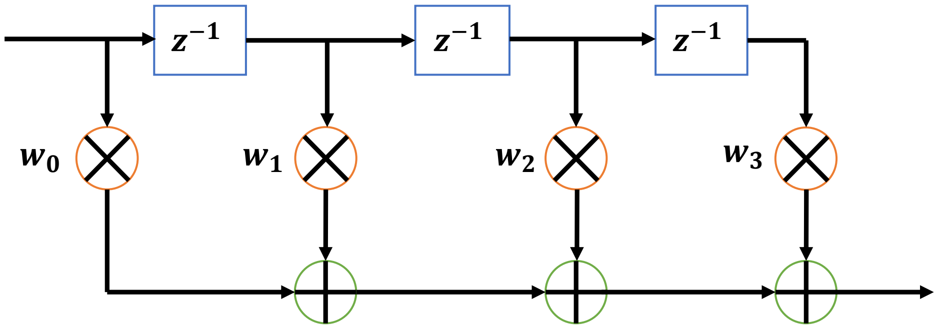


Figure 6.4: Standard Parallel FIR Filter

Now consider that each of the delay elements in a standard parallel FIR filter are scaled by a factor of $b = 2$, as illustrated in Figure 6.5.

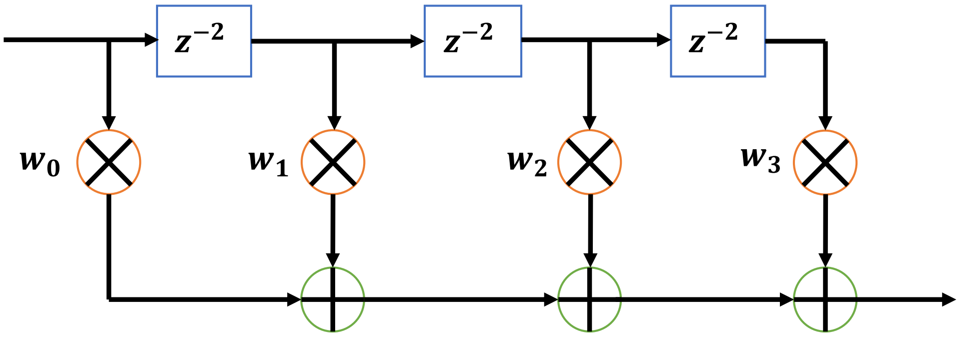


Figure 6.5: Standard FIR filter with time scaling by factor b = 2

Having applied the time scaling, the functionality of the FIR filter is changed. In order for the functionality of the time scaled filter to match the original filter, it is necessary to up sample the input to the time scaled filter by a factor of two, such that only every second input sample has a non-zero value. This means that the filter is running at two times the original sampling rate and is wastefully processing zero valued samples. However, returning to the multi-channel processing example, it can be used to 100% efficiency by replacing the zero samples with the data on a second independent channel. The channels are multiplexed into a single data stream with a rate of two times the input rate of each of the independent channels. The data is then passed into the filter with time scaling applied and the resulting data is de-multiplexed in order to recover the original independent channels. Using this configuration, it is possible to perform the filtering on each channel using a single FIR filter. The cost of achieving this is that the filter hardware must run at two times the input sample rate and all delays must be scaled by a factor of two.

In the general case, any hardware can be shared using this approach, by multiplexing the input channels into a single higher rate data stream and scaling all delays in the shared hardware by a factor equal to the number of input channels. As mentioned earlier, this approach will be a key element of the implementation of the Split-CAF cyclostationary detectors introduced in Chapter 5.

## 6.4 Implementation of Cyclostationary Detectors

Having introduced necessary background information in the previous sections, it is now time to consider the HDL Coder implementation of the cyclostationary detection algorithms described in Chapters 4 and 5. The details of the implementation of each algorithm in HDL Coder will be provided and their relative resource costs will be assessed by following the steps shown in Figure 6.2. The target device is a Xilinx Artix-7 xc7a100t csg324-1 FPGA. The test signal is an IEEE 802.11a/g OFDM signal and the observation interval for each of the detectors is $N = 1600$ with $P_{fa} = 0.1$.

### 6.4.1 GLRT Detector

In this section, details of the implementation of the GLRT detector in Section 4.5 will be provided. It is assumed that the real and imaginary parts of the data entering the HDL detector from the Analogue to Digital Converter (ADC) are 12 bit signed fixed-point numbers giving a total representable range of -2048 to 2047. This input wordlength will be assumed for all subsequent detectors. The final test statistic for the GLRT detector is repeated here for convenience:

$$\hat{T}_{GLRT} = N\frac{(\Re(\hat{R}^{\alpha}_{xx}[\nu]))^2\hat{E}[Y^2] + (\Im(\hat{R}^{\alpha}_{xx}[\nu]))^2\hat{E}[X^2] - 2(\Re(\hat{R}^{\alpha}_{xx}[\nu]))(\Im(\hat{R}^{\alpha}_{xx}[\nu]))\hat{E}[XY]}{\hat{E}[X^2]\hat{E}[Y^2] - (\hat{E}[XY])^2}. \quad (6.4)$$

Each of the individual elements of (6.4) are calculated as follows,

$$\hat{E}[X] = \Re(\hat{R}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1}\Re(x[n]x^*[n-\nu]e^{-j2\pi\alpha n}), \quad (6.5)$$

$$\hat{E}[Y] = \Im(\hat{R}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1}\Im(x[n]x^*[n-\nu]e^{-j2\pi\alpha n}), \quad (6.6)$$

$$\hat{E}[X^2] = \frac{1}{N}\sum_{n=0}^{N-1}\Re(x[n]x^*[n-\nu]e^{-j2\pi\alpha n})^2, \quad (6.7)$$

$$\hat{E}[XY] = \hat{E}[YX] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(x[n]x^*[n-\nu]e^{-j2\pi\alpha n})\Im(x[n]x^*[n-\nu]e^{-j2\pi\alpha n}), \qquad (6.8)$$

$$\hat{E}[Y^2] = \frac{1}{N}\sum_{n=0}^{N-1} \Im(x[n]x^*[n-\nu]e^{-j2\pi\alpha n})^2. \qquad (6.9)$$

The full implementation of the GLRT detector in HDL Coder is shown in Figure 6.6.

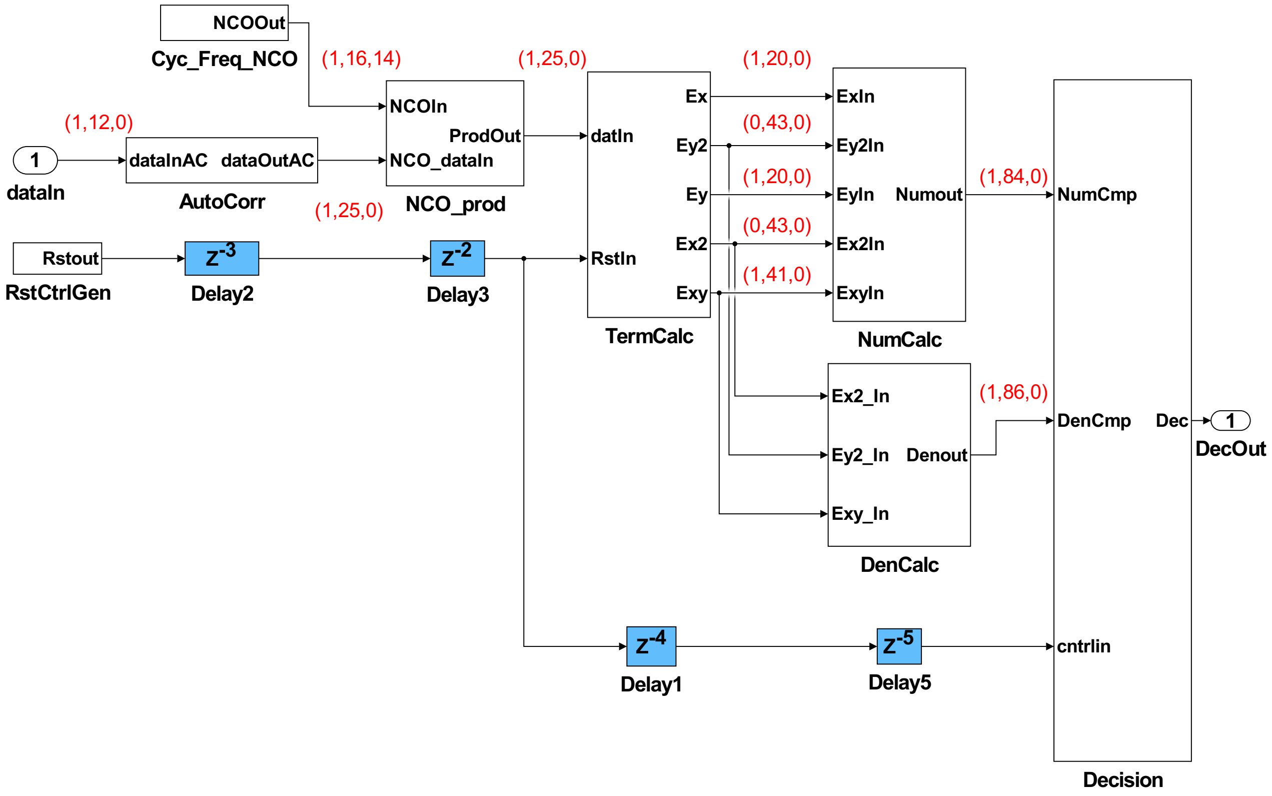


Figure 6.6: HDL Coder Implementation of GLRT Detector

The design consists of eight main entities: *AutoCorr*, *CycFreqNCO*, *NCOprod*, *RstCtrlGen*, *TermCalc*, *NumCalc*, *DenCalc* and *Decision*. The *AutoCorr* entity deals with computation of the autocorrelation lag product $x[n]x^*[n-\nu]$. This requires a delay, a complex conjugate operation and a complex multiplier. The HDL Coder implementation of this entity is shown in Figure 6.7. Note that the delay is equal to 64, as the test signal is IEEE 802.11a/g OFDM. The fixed point wordlengths at each stage of the model are highlighted in red and follow the format in (6.1). The blue delay blocks are pipeline registers that have been added to the model manually.

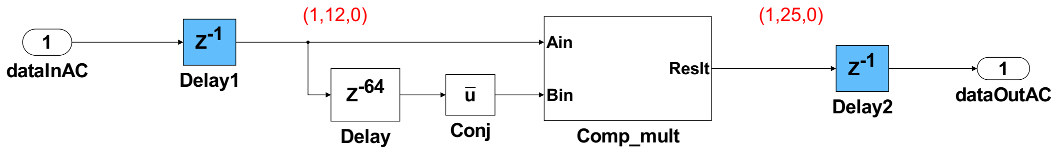


Figure 6.7: *AutoCorr* entity in HDL Coder

The multiplication of two complex numbers $a + jb$ and $c + jd$ is given by the following,

$$(a + jb) \times (c + jd) = (ac - bd) + j(bc + ad). \tag{6.10}$$

Therefore, a complex multiplication requires four real multiplies and two additions. The implementation of the complex multiplication which resides in the *NCOprod* entity is shown in Figure 6.8.

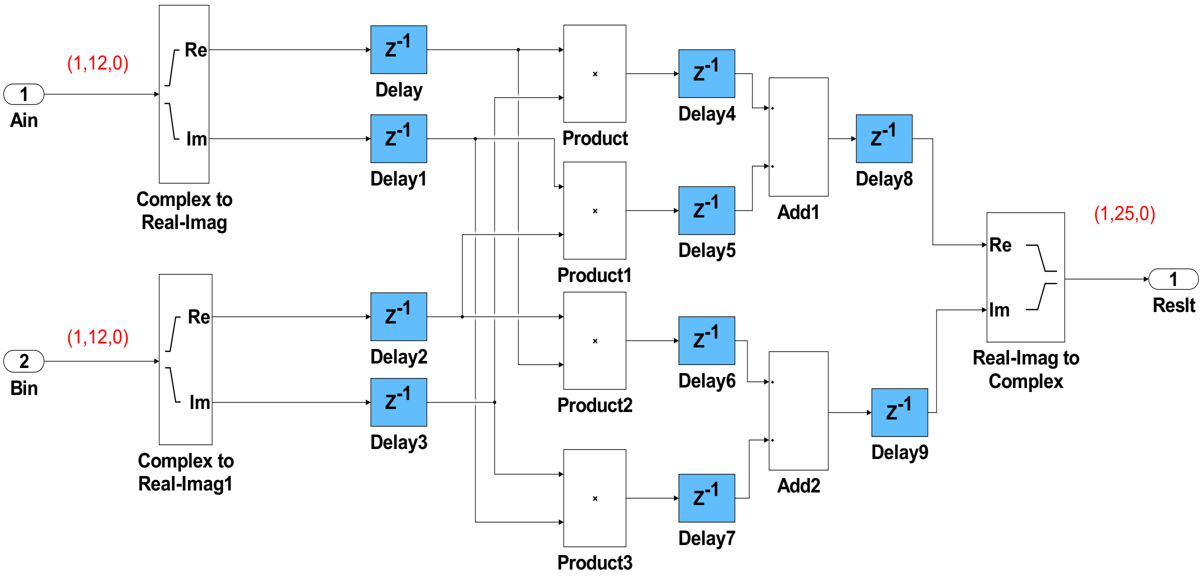


Figure 6.8: Implementation of *NCOprod* entity in HDL Coder

The *CycFreqNCO* entity generates a complex exponential at the desired cyclic frequency of $\alpha_0$ = 1/80 using the *NCO HDL Optimized* block shown in Figure 6.9.

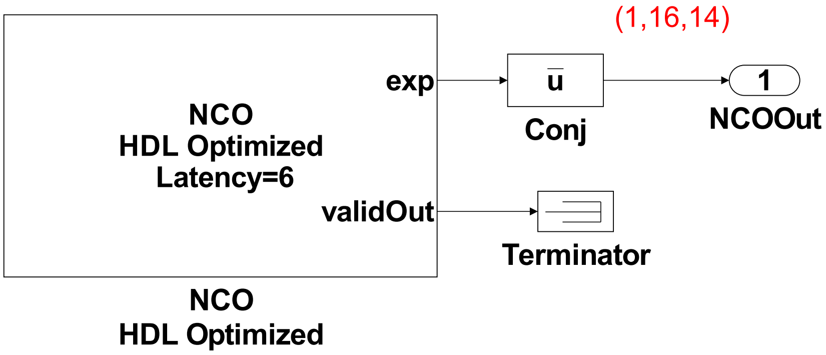


Figure 6.9: Implementation of Complex Exponential using NCO HDL Optimized Block

The output of the NCO block is complex conjugated in order to generate a complex exponential of the form,

$$e^{-j2\pi\alpha}. \tag{6.11}$$

The block requires a value to be specified for the NCO step-size, which is calculated using the following equation,

$$\mu = \frac{Mf_d}{f_s}, \tag{6.12}$$

where $\mu$ is the step-size, $M$ is the number of entries in the NCO LUT and $f_d$ is the desired frequency, i.e. the cyclic frequency of the IEEE 802.11a/g signal in Hz. The *NCOprod* entity consists of a complex multiplier implemented exactly the same as Figure 6.8, which multiplies the autocorrelation lag product and the complex exponential to produce,

$$x[n]x^*[n-\nu]e^{-j2\pi\alpha}. \tag{6.13}$$

The output wordlength of this multiplication is set to (1,25,0). Having calculated (6.13), it is possible to calculate each of the terms in (6.5) - (6.9). These are computed in the *TermCalc* entity as shown in Figure 6.10.

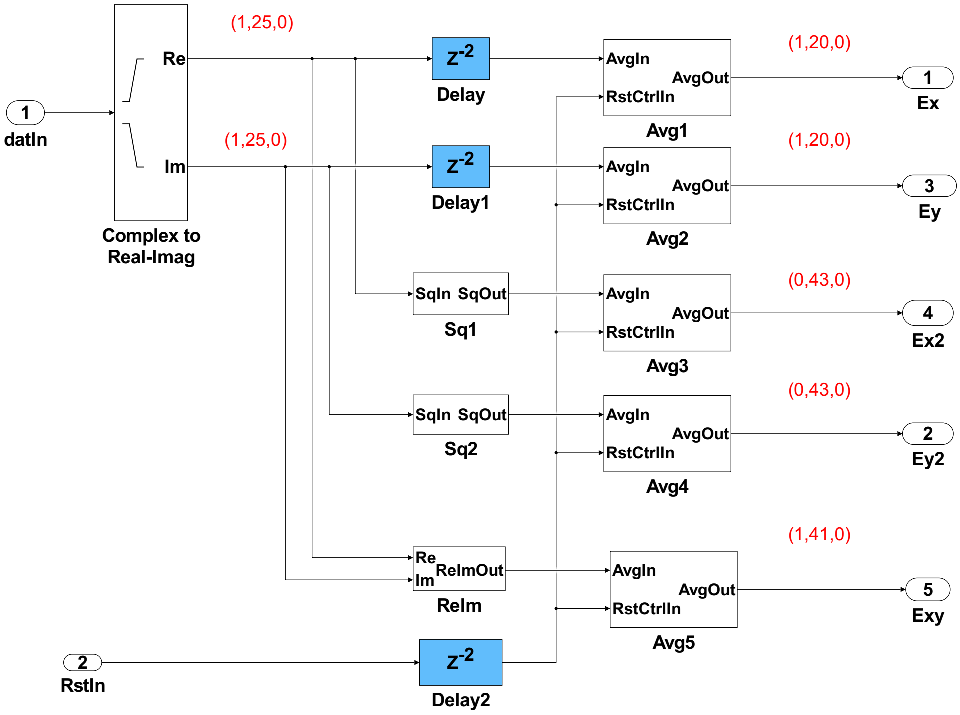


Figure 6.10: Implementation of *TermCalc* entity in HDL Coder

The *AVG* blocks are integrators which are reset at the end of every block of $N$ samples, when the test statistic is calculated. The implementation of the *AVG* blocks in HDL Coder is shown in Figure 6.11.

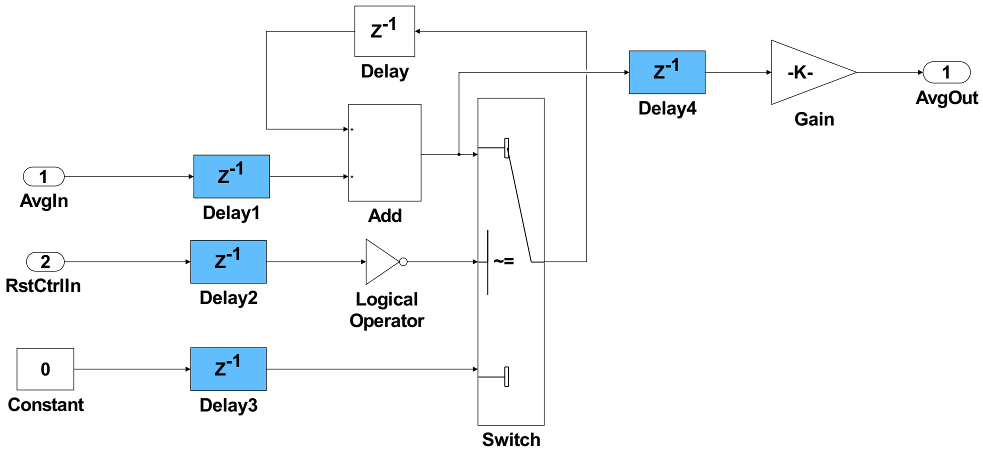


Figure 6.11: Implementation of *AVG* blocks in HDL Coder

The integrator reset is generated in the *RstCtrlGen* entity, which consists of a counter that repeatedly counts from 1 to $N$. When the counter reaches $N$, a strobe is generated which is used to reset the integrator. Having calculated the terms in (6.5) - (6.9), both the numerator and denominator terms of (6.4) can be calculated. The numerator is calculated inside *NumCalc* as shown in Figure 6.12. Similarly, the denominator of (6.4) is estimated in *DenCalc* as shown in Figure 6.13.

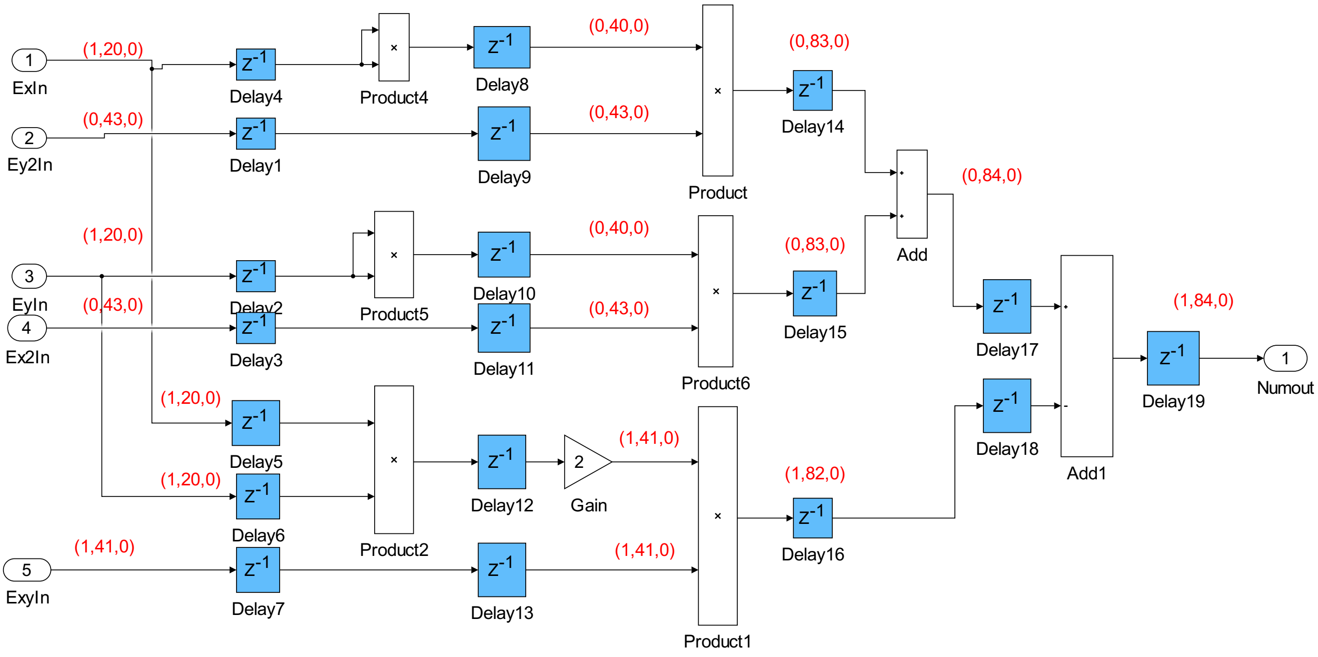


Figure 6.12: Implementation of *NumCalc* entity in HDL Coder

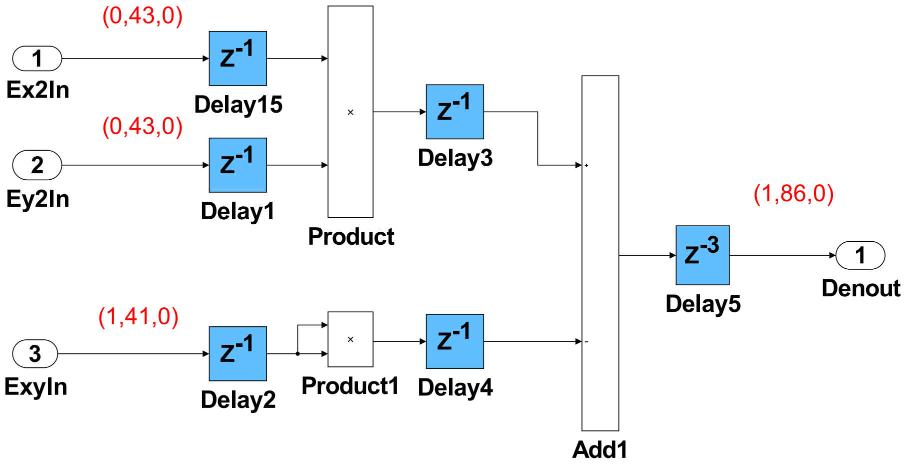


Figure 6.13: Implementation of *DenCalc* entity in HDL Coder

Having estimated the numerator and denominator of (6.4), the last step is to perform a division. However, as mentioned in the author's paper [2], the division can be avoided by applying a simple mathematical re-arrangement. If the numerator is denoted as *Num* and the denominator as *Den*, then the decision operation can be implemented as,

$$Num > Den\,\eta, \tag{6.14}$$

where $\eta$ is the threshold. The division has been replaced by a constant scaling, which is a far simpler calculation. Figure 6.14 compares $P_d$ vs. SNR curves for the original GLRT in (6.4) and with the re-arrangement (6.14) applied.

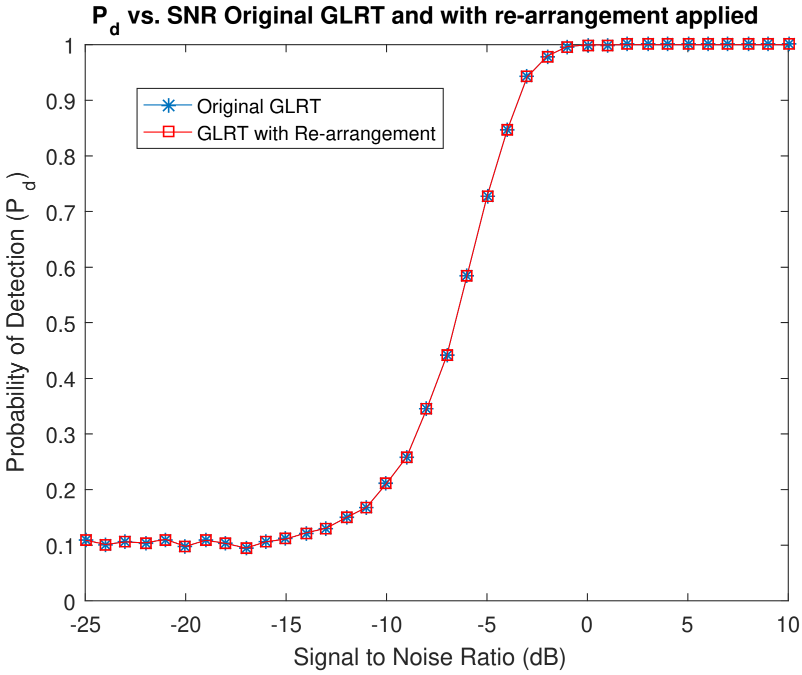


Figure 6.14: $P_d$ vs. SNR for Original GLRT and with re-arrangement applied

It can be observed that the performance matches closely, thus justifying the proposed re-arrangement of the test statistic. The implementation of the *Decision* entity is shown in Figure 6.15.

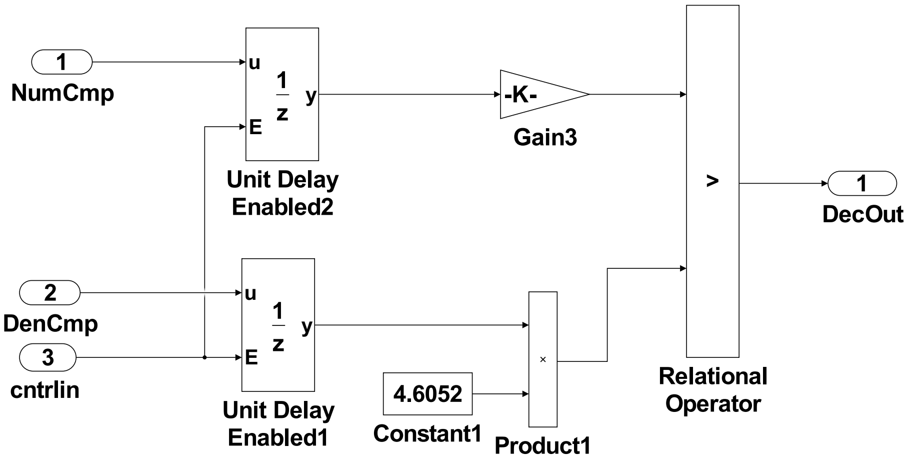


Figure 6.15: Implementation of *Decision* entity in HDL Coder

The first part of this entity consists of latches which record the numerator and denominator values at the end of each block of $N$ samples. The signal used to latch the numerator and denominator values is the same strobe used to reset the integrators in the $AVG$ blocks. In the top branch, the gain block implements the scaling by $N$ and, on the lower branch, the right hand side of (6.14) is calculated. Finally, both branches are compared using a relational operator to decide if a signal is present or not.

### 6.4.2 Low Complexity Detector

In this section, the implementation of the Low Complexity detector in Chapter 4 will be described. The final test statistic is:

$$\hat{T}_{LC} = \frac{N\left|\hat{R}^{\alpha}_{xx}[\nu]\right|^2}{\frac{1}{N}\sum_{n=0}^{N-1}\left|x[n]x^*[n-\nu]\right|^2}. \tag{6.15}$$

Figure 6.16 shows the implementation of the Low Complexity detector in HDL Coder.

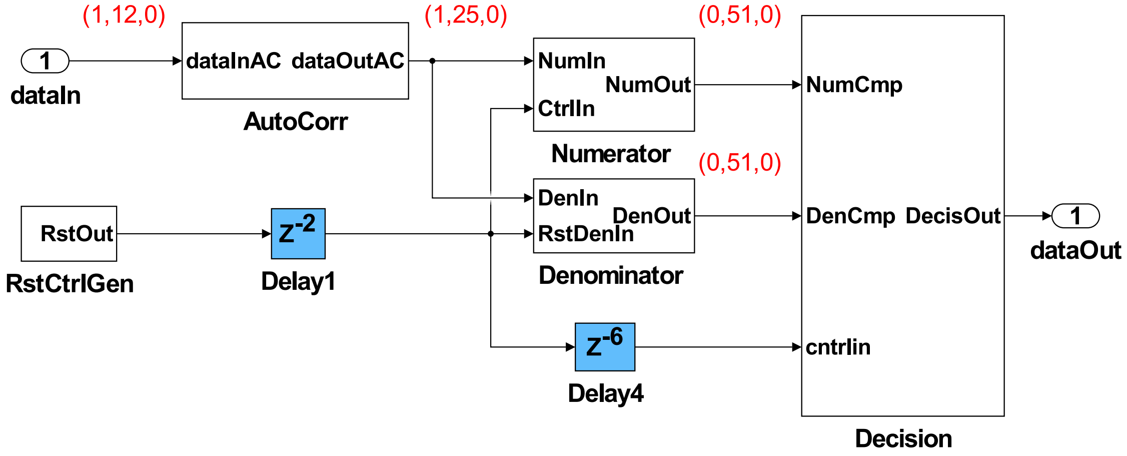


Figure 6.16: HDL Coder Implementation of Low Complexity Detector

The design consists of five main entities: *AutoCorr*, *RstCtrlGen*, *Numerator*, *Denominator* and *Decision*. The *AutoCorr* entity is implemented exactly the same as it was for the GLRT detector as shown in Figure 6.7. Similarly, the *RstCtrlGen* entity consists of a counter which counts repeatedly to $N$, and when this value is reached, a strobe is emitted and used for resetting the integrators and latching the numerator and denominator values in the *Decision* entity. The *Numerator* entity which computes the numerator of (6.15)- excluding the scaling by $N$ - is shown in Figure 6.17.

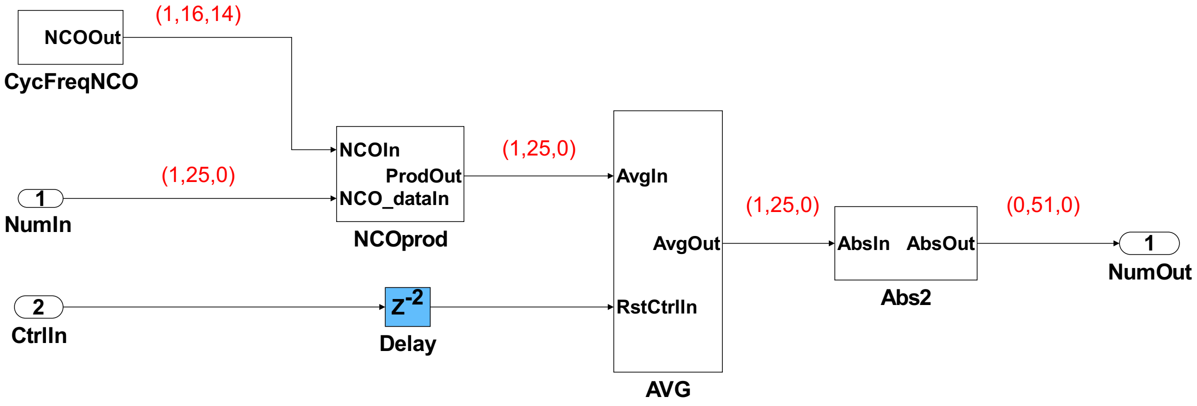


Figure 6.17: Implementation of *Numerator* entity for Low Complexity detector in HDL Coder

The *NCOprod*, *CycFreqNCO* and *AVG* blocks are implemented as shown in Figures 6.8, 6.9 and 6.11. The *Abs2* entity is implemented as shown in Figure 6.18.

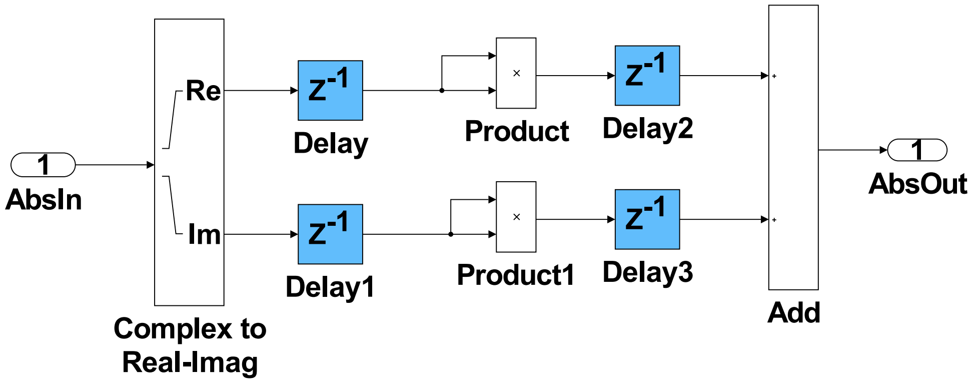


Figure 6.18: Implementation of *Abs2* entity for Low Complexity detector in HDL Coder

This requires two product blocks to square the real and imaginary parts of the complex signal and an add block to sum the results. The *Denominator* entity, which computes the denominator of (6.15), is shown in Figure 6.19.

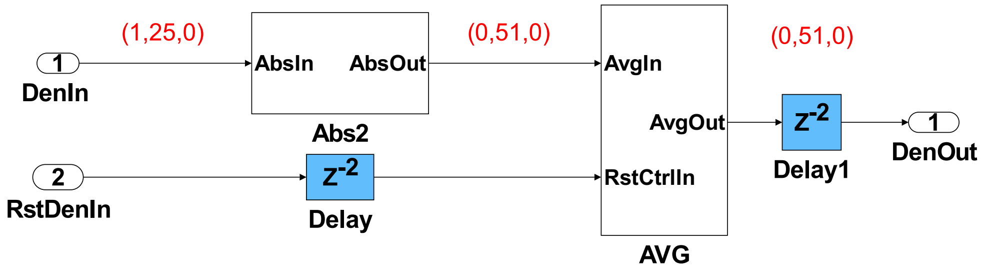


Figure 6.19: Implementation of *Denominator* entity for Low Complexity detector in HDL Coder

This consists of an *Abs2* block and an *AVG* block, which are implemented as shown in Figures 6.18 and 6.11. The final stage is the *Decision* entity. This is implemented the same as Figure 6.15 except the threshold is $\eta = 2.3026$, since the Low Complexity detector follows a $\Gamma(1,1)$ distribution under the null hypothesis.

### 6.4.3 Spatial Sign Detector

The final test statistic for the Spatial Sign detector is:

$$\hat{T}_{SS} = N\left|\hat{R}^{\alpha}_{ss}[\nu]\right|^{2}. \tag{6.16}$$

The full implementation of the Spatial Sign detector is shown in Figure 6.20.

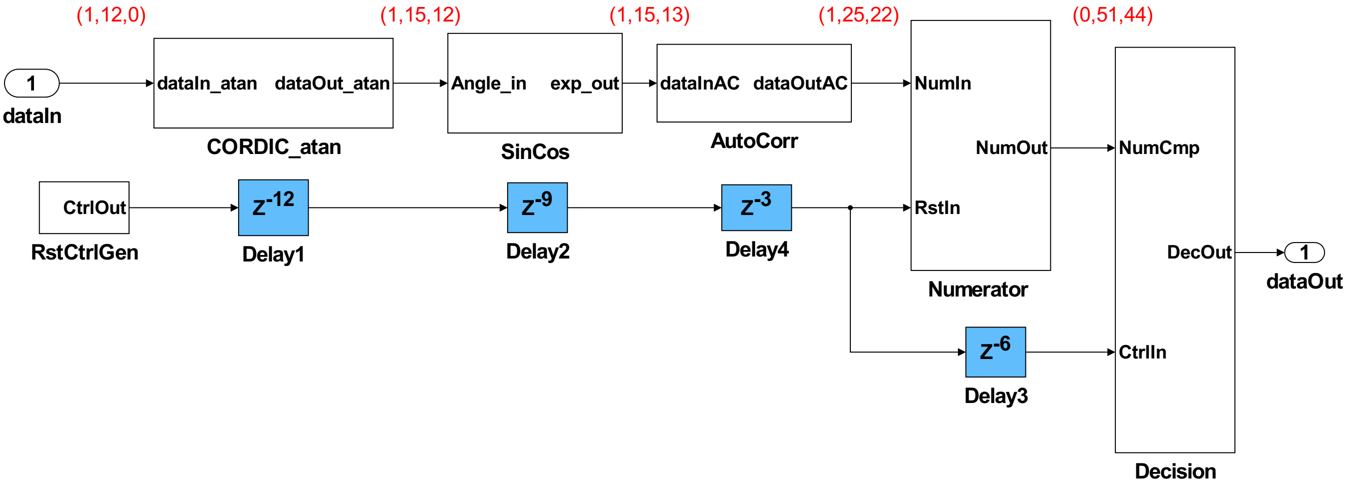


Figure 6.20: Implementation of Spatial Sign Detector in HDL Coder

In this implementation, there are six main entities: *CORDICatan*, *SinCos*, *AutoCorr*, *Numerator*, *RstCtrlGen* and *Decision*. The *AutoCorr* and *Numerator* entities are implemented as shown in Figures 6.7 and 6.17 respectively except that the wordlengths are different. The *Decision* entity is implemented as shown in Figure 6.21.

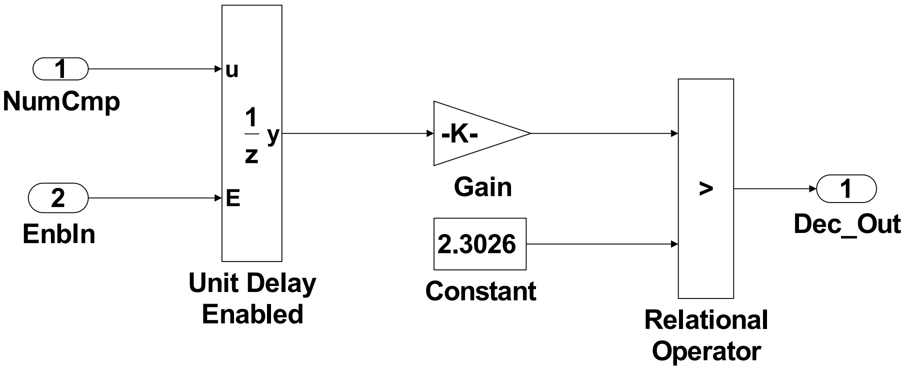


Figure 6.21: Implementation of *Decision* entity for Spatial Sign Detector

This is implemented slightly differently than the previous *Decision* modules since there is no denominator and, therefore, the test statistic is compared directly to the threshold. The Spatial Sign function is calculated by first computing the angle $\phi$ of the complex input data,

$$\phi = tan^{-1}(\frac{\Im(x[n])}{\Re(x[n])}), \tag{6.17}$$

where $x[n]$ is the complex input signal. This calculation is performed inside the *CORDICatan* entity which contains a *Complex to Magnitude-Angle HDL Optimized* block as shown in Figure

6.22.

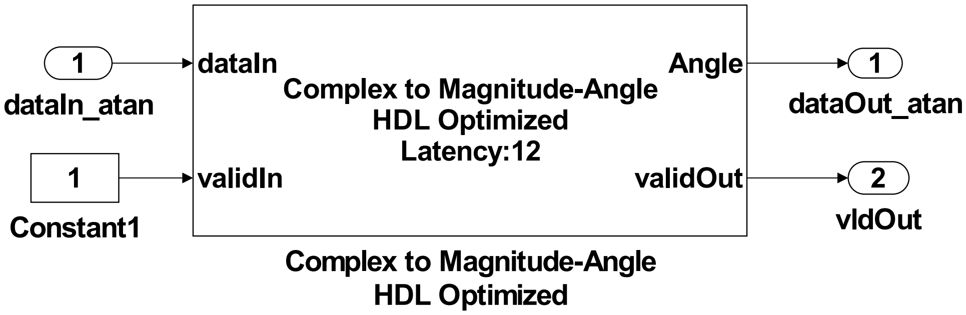


Figure 6.22: Implementation of *CORDICatan* entity in HDL Coder

This block calculates the magnitude and angle (6.17) of a complex input signal using the CORDIC algorithm [93], operating in vectoring mode. Note that only the angle output is retained, as the magnitude is not required for this algorithm. In this implementation, the number of iterations is set equal to 10 and the output angles are in the range $(-\pi, \pi)$ radians. This generates a total of 12 cycles of latency in the final implementation. The next stage in the spatial sign function is to compute the complex exponential,

$$e^{j\phi} = cos(\phi) + jsin(\phi). \tag{6.18}$$

This is achieved using CORDIC operating in rotation mode, which is implemented in the *SinCos* entity as shown in Figure 6.23. The calculation of the complex exponential has been implemented using a trigonometric function block, which can be specified to use CORDIC in rotation mode when it is targeted to an FPGA. The “UsePipelinedKernel” option in HDL Block options is selected to ensure that pipeline registers are placed in between each CORDIC cell. In this implementation, a total of 8 stages of CORDIC are employed. This generates a total of 9 cycles of latency in the final implementation.

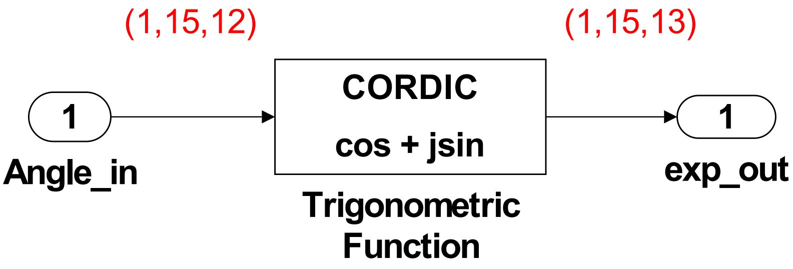


Figure 6.23: Implementation of *SinCos* entity in HDL Coder

### 6.4.4 Split-CAF GLRT Detector

In this section, the implementation of the Split-CAF GLRT detector will be described. Recall that the Split-CAF GLRT detector works by computing parallel test statistics using the two components of the traditional CAF,

$$\hat{R}^{\alpha}_{xx}[\nu] = \hat{I}^{\alpha}_{xx}[\nu] + j\hat{Q}^{\alpha}_{xx}[\nu]. \tag{6.19}$$

The test statistic computed for $\hat{I}^{\alpha}_{xx}[\nu]$ is,

$$\hat{T}_I = N\frac{(\Re(\hat{I}^{\alpha}_{xx}[\nu]))^2\hat{E}[Y_I^2] + (\Im(\hat{I}^{\alpha}_{xx}[\nu]))^2\hat{E}[X_I^2] - 2(\Re(\hat{I}^{\alpha}_{xx}[\nu]))(\Im(\hat{I}^{\alpha}_{xx}[\nu]))\hat{E}[X_IY_I]}{\hat{E}[X_I^2]\hat{E}[Y_I^2] - (\hat{E}[X_IY_I])^2}. \tag{6.20}$$

Similarly, the test statistic for $\hat{Q}^{\alpha}_{xx}[\nu]$ is,

$$\hat{T}_Q = N\frac{(\Re(\hat{Q}^{\alpha}_{xx}[\nu]))^2\hat{E}[Y_Q^2] + (\Im(\hat{I}^{\alpha}_{xx}[\nu]))^2\hat{E}[X_Q^2] - 2(\Re(\hat{Q}^{\alpha}_{xx}[\nu]))(\Im(\hat{Q}^{\alpha}_{xx}[\nu]))\hat{E}[X_QY_Q]}{\hat{E}[X_Q^2]\hat{E}[Y_Q^2] - (\hat{E}[X_QY_Q])^2}. \tag{6.21}$$

The final test statistic combines (6.20) and (6.21) as follows,

$$\hat{T}_{SC-GLRT} = N\frac{N_ID_Q + N_QD_I}{D_ID_Q}. \tag{6.22}$$

where $N_I$ and $D_I$ are the numerator and denominator of (6.20) and $N_Q$ and $D_Q$ are the numerator and denominator of (6.21). The elements of each of these test statistics are calculated as,

$$\Re(\hat{I}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1}\Re(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}), \tag{6.23}$$

$$\Im(\hat{I}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1}\Im(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}), \tag{6.24}$$

$$\Re(\hat{Q}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1}\Re(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}), \tag{6.25}$$

$$\Im(\hat{Q}^{\alpha}_{xx}[\nu]) = \frac{1}{N}\sum_{n=0}^{N-1}\Im(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}) \tag{6.26}$$

$$\hat{E}[X_I^2] = \frac{1}{N}\sum_{n=0}^{N-1}\Re(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})^2, \tag{6.27}$$

$$\hat{E}[X_I Y_I] = \hat{E}[Y_I X_I] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})\Im(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}), \quad (6.28)$$

$$\hat{E}[Y_I^2] = \frac{1}{N}\sum_{n=0}^{N-1} \Im(\Re(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})^2, \quad (6.29)$$

$$\hat{E}[X_Q^2] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})^2, \quad (6.30)$$

$$\hat{E}[X_Q Y_Q] = \hat{E}[Y_Q X_Q] = \frac{1}{N}\sum_{n=0}^{N-1} \Re(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})\Im(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n}), \quad (6.31)$$

$$\hat{E}[Y_Q^2] = \frac{1}{N}\sum_{n=0}^{N-1} \Im(\Im(x[n]x^*[n-\nu])e^{-j2\pi\alpha n})^2. \quad (6.32)$$

In order to simplify the computation, it is proposed that we assume that $D_I \approx D_Q$. This reduces the final test statistic to,

$$\hat{T}_{SC-GLRT} = N\frac{N_I + N_Q}{D_I}, \quad (6.33)$$

and removes the multiplications in (6.22). Figure 6.24 compares the $P_d$ vs. SNR curves for the test statistics in (6.22) and (6.33).

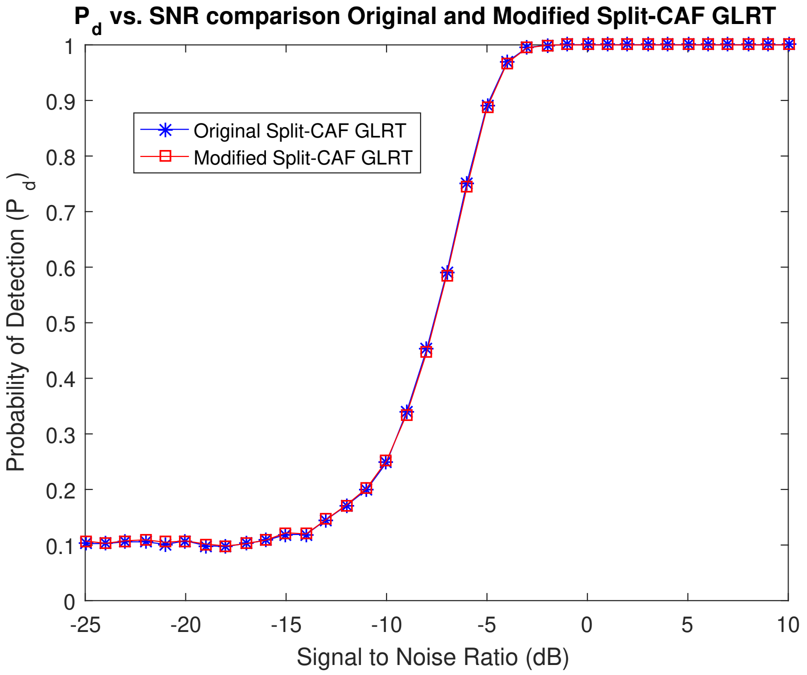


Figure 6.24: Performance comparison of original and modified Split-CAF GLRT test statistics

It can be observed that the performance curves match very closely, indicating that no significant performance penalty is incurred by adopting the simplified test statistic in (6.33). The final implementation of the Split-CAF GLRT detector in HDL Coder is shown in Figure 6.25.

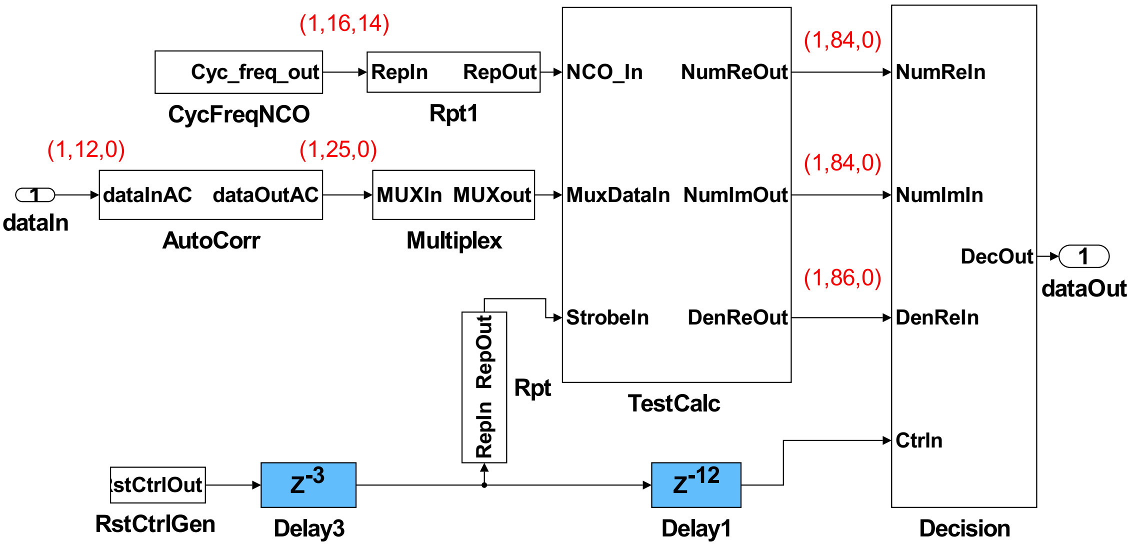


Figure 6.25: Implementation of Split-CAF GLRT Detector in HDL Coder

This architecture consists of seven main entities: *AutoCorr*, *Multiplex*, *CycFreqNCO*, *RstCtrlGen*, *Rpt*, *TestCalc* and *Decision*. The *AutoCorr*, *CycFreqNCO* and *RstCtrlGen* entities are implemented exactly the same as for the GLRT detector in Section 6.4.1. It can be observed from equations (6.23) - (6.32) that the test statistics in (6.20) and (6.21) are computed in an identical manner. The only difference is that (6.20) uses the real part of the autocorrelation lag product and (6.21) uses the imaginary part. Therefore, the hardware required to compute each test statistic can be shared using the approach described in Section 6.3. In this case, there are two independent data streams, requiring the shared hardware to run at twice the input sample rate and all delays to be scaled by a factor of two. The two independent channels are multiplexed into a single data stream in the *Multiplex* entity which is shown in Figure 6.26.

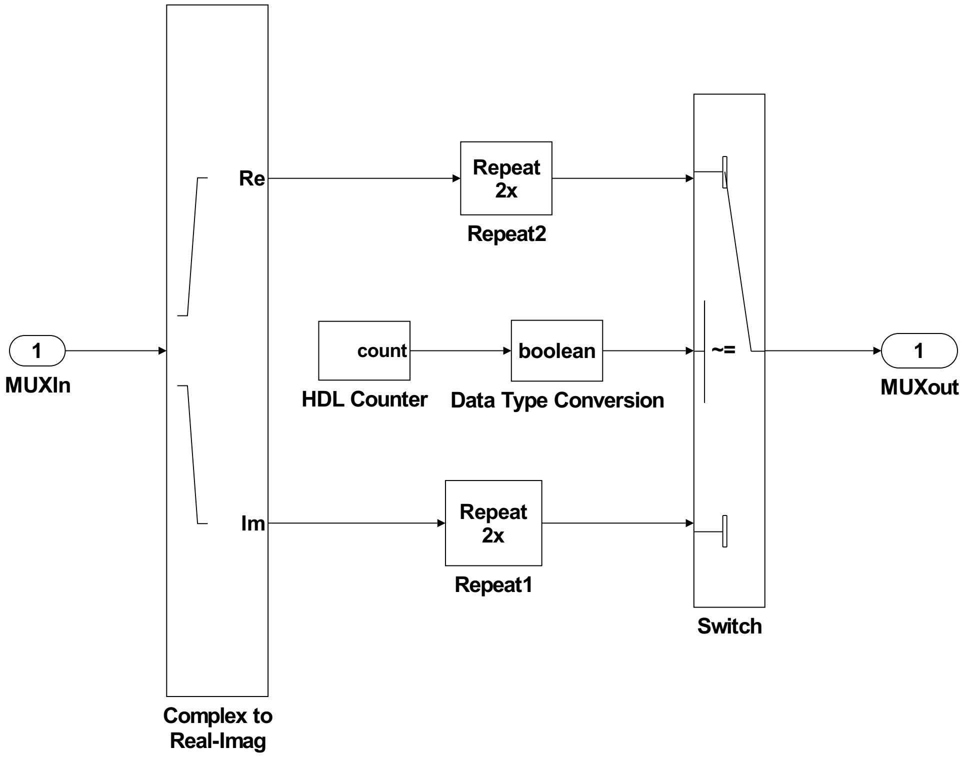


Figure 6.26: Implementation of *Multiplex* entity in HDL Coder

The autocorrelation lag product is first split into its real and imaginary parts and each signal is repeated by a factor of two. The two streams are then passed as the inputs to a multiplexer which alternately chooses between each of its inputs in step with a clock signal, operating at twice the input sampling rate. The clock signal is generated using a counter which counts from 1 to 0 repeatedly and the output of the counter is converted to a boolean signal using a data type conversion block. Therefore, the output of the circuit is a single data stream, which consists of the real and imaginary signals interleaved together. This allows a single instance of the hardware to process both the real and imaginary parts of the autocorrelation lag product and thus avoids the need to repeat the hardware twice. Note that the NCO output and the reset signal must also run at twice the input rate and, therefore, they are passed through the *Rpt* entity, which contains a repeat block. Figure 6.27 shows the shared hardware which is implemented inside the *TestCalc* entity.

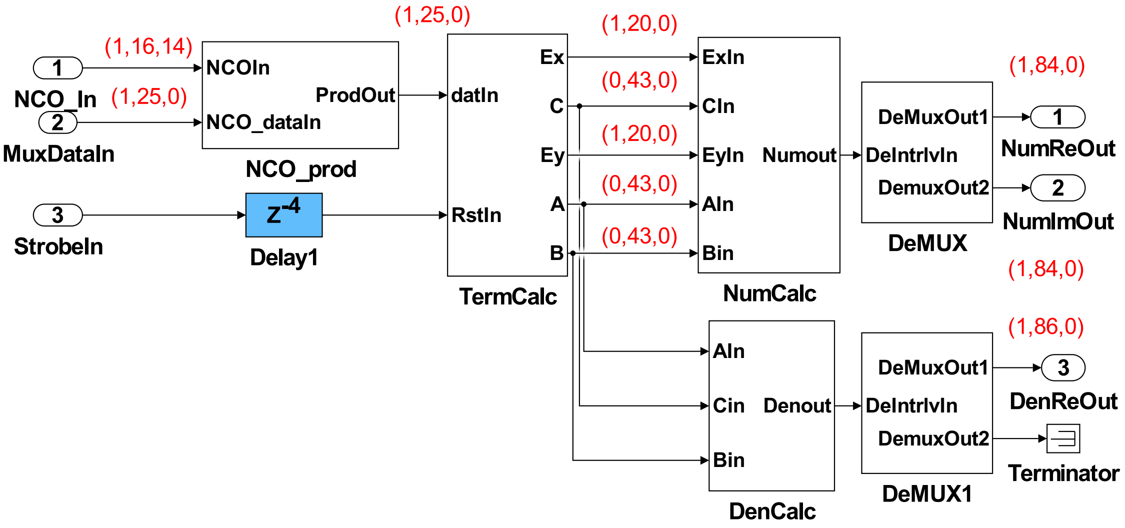


Figure 6.27: Implementation of *TestCalc* entity in HDL Coder

The *NCOprod* entity is implemented slightly differently than previously since it no longer requires a complex multiplier of the form shown in Figure 6.8. This is because the complex exponential is now scaled by a real value instead of a complex value and, thus, only two real multipliers are required instead of four. The *TermCalc*, *NumCalc* and *DenCalc* entities are implemented as shown in Figures 6.10, 6.12 and 6.13. The only difference is that all delay blocks are scaled by a factor of two in accordance with the rules for multi-channel hardware sharing. Finally, the outputs of the *NumCalc* and *DenCalc* entities are passed to a demultiplexing circuit in order to recover $N_I$, $N_Q$ and $D_I$. Notice that $D_Q$ is terminated since it is not required for the calculation in (6.33). The implementation of the demultiplexing circuit is shown in Figure 6.28.

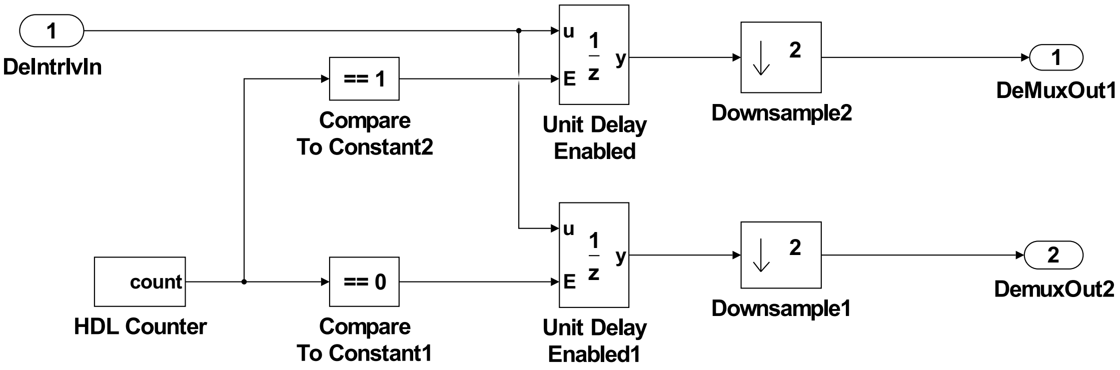


Figure 6.28: Implementation of *DeMUX* entity in HDL Coder

In the first *DeMUX* entity, the top branch latches the numerator for the test statistic employing the real part of the autocorrelation lag product and the bottom branch latches the numerator for the test statistic using the imaginary part. This de-interleaves the single data stream back into two independent streams. These are then down sampled by a factor of two in order to return them to the original sampling rate. The final stage is to compute the test statistic in (6.33), which is carried out inside the *Decision* entity as shown in Figure 6.29.

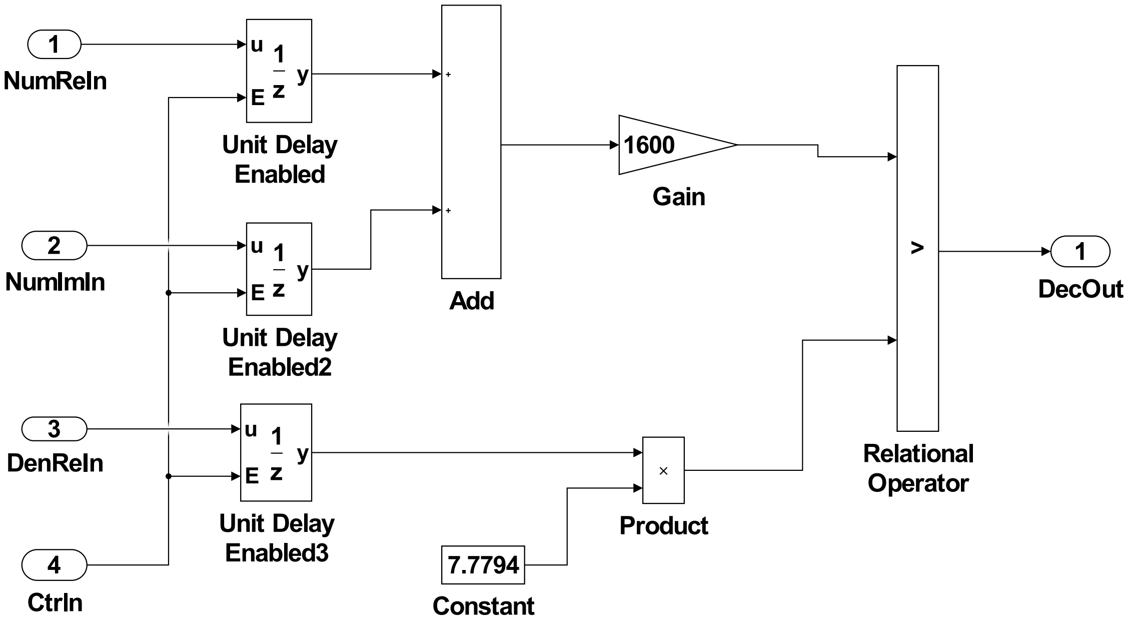


Figure 6.29: Implementation of *Decision* entity in HDL Coder

As was the case for the GLRT and Low Complexity detectors, the proposed mathematical re-arrangement in (6.14) is applied in order to avoid the division operation.

### 6.4.5 Split-CAF Low Complexity Detector

The final test statistic for this detector is:

$$\hat{T}_{SC-LC} = \frac{N\left(\left|\hat{I}^{\alpha}_{xx}[\nu]\right|^2 + \left|\hat{Q}^{\alpha}_{xx}[\nu]\right|^2\right)}{\frac{1}{N}\sum_{n=0}^{N-1}\left|x[n]x^*[n-\nu]\right|^2}. \tag{6.34}$$

The implementation of this detector in HDL coder is shown in Figure 6.30.

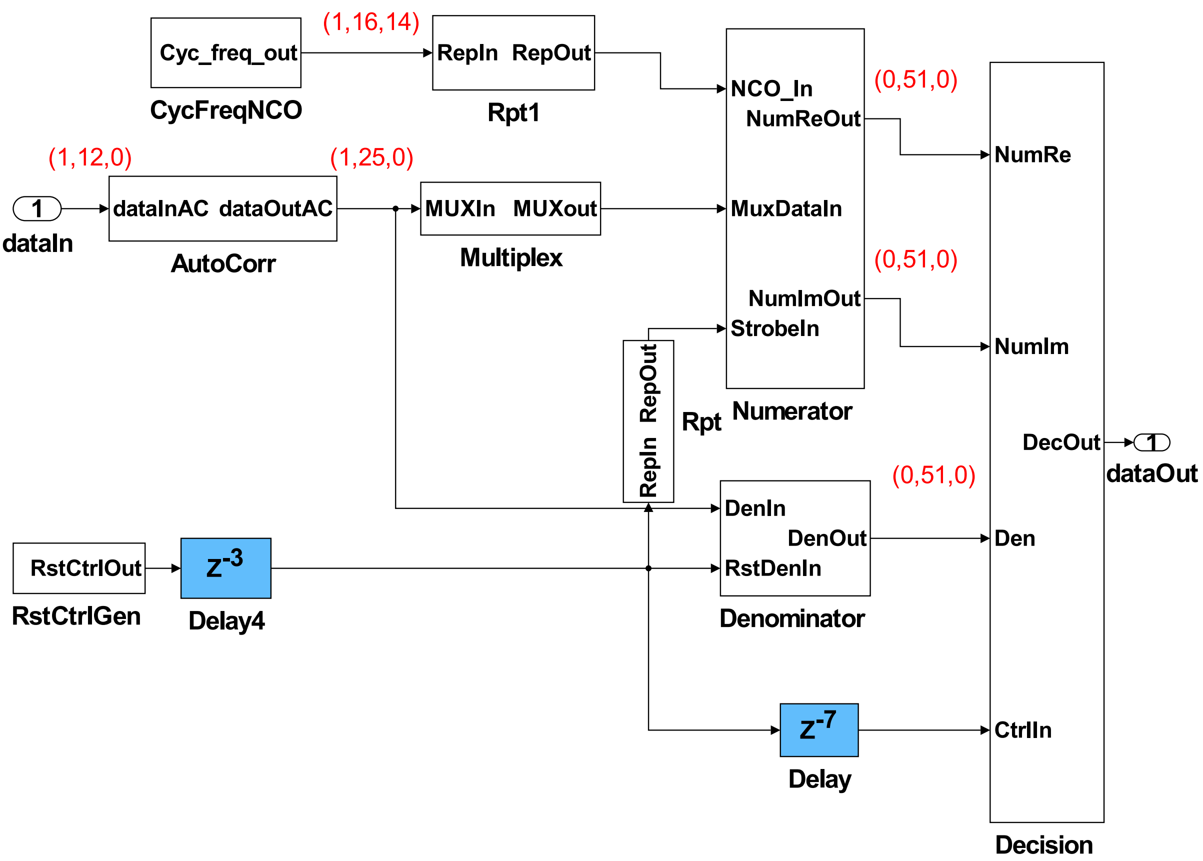


Figure 6.30: Implementation of Split-CAF Low Complexity detector in HDL Coder

This design consists of eight main entities: *AutoCorr*, *Multiplex*, *CycFreqNCO*, *Rpt*, *RstCtrlGen*, *Numerator*, *Denominator* and *Decision*. As with the Split-CAF GLRT detector, the hardware required to compute the numerator terms in (6.34) can be shared between the real and imaginary parts of the autocorrelation lag product. Therefore, implementing two instances of the *Numerator* entity is unnecessary. The *AutoCorr*, *CycFreqNCO*, *RstCtrlGen*, *Rpt* and *Multiplex* entities are implemented in the same manner as previously. The *Numerator* entity is implemented the same as in Figure 6.17. However, the *CycFreqNCO* is not included as this is not part of the shared hardware. Also, the *NCOprod* entity only requires two real multipliers instead of four and a *DeMUX* entity is included in order to recover both $\left|\hat{I}^{\alpha}_{xx}[\nu]\right|^2$ and $\left|\hat{Q}^{\alpha}_{xx}[\nu]\right|^2$. The *Denominator* entity is implemented the same as shown in Figure 6.19. Finally, the *Decision* entity is implemented as shown in Figure 6.29, with a threshold of $\eta = 1.9449$ as the test statistic is $\Gamma(2, 0.5)$ distributed under the null hypothesis.

### 6.4.6 Split-CAF Spatial Sign Detector

The final test statistic is:

$$\hat{T}_{SC-SS} = N\left(\left|\hat{I}^{\alpha}_{ss}[\nu]\right|^{2} + \left|\hat{Q}^{\alpha}_{ss}[\nu]\right|^{2}\right). \tag{6.35}$$

The full implementation of this detector in HDL Coder is shown in Figure 6.31.

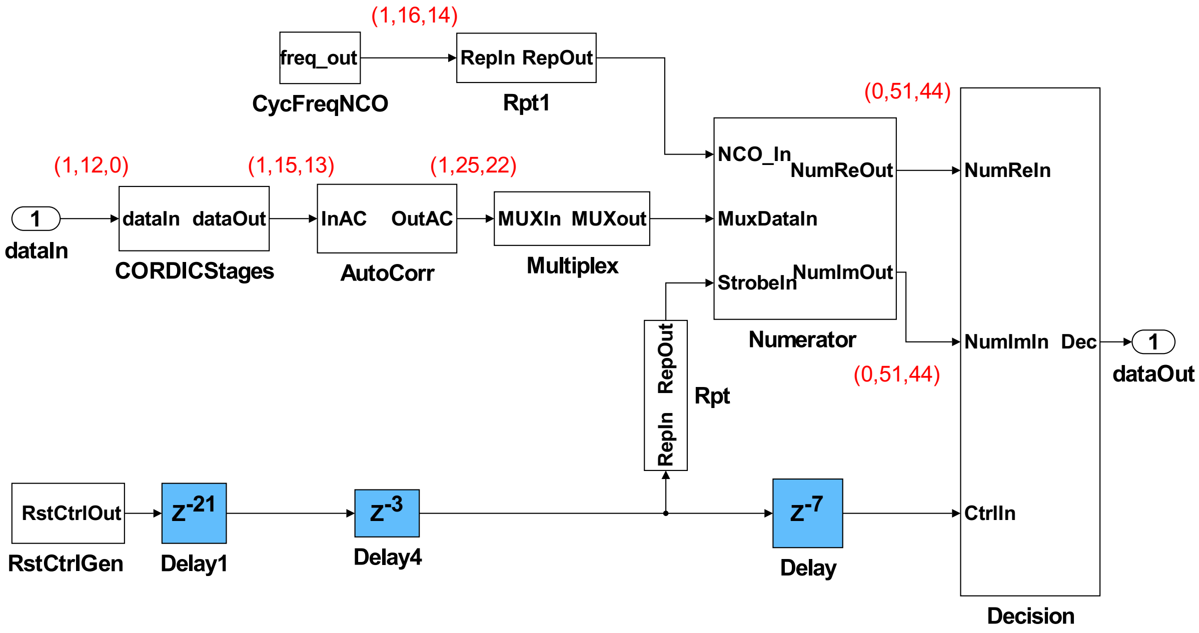


Figure 6.31: Implementation of Split-CAF Spatial Sign detector in HDL Coder

There are seven main entities: *CORDICStages*, *Multiplex*, *CycFreqNCO*, *Rpt*, *Numerator*, *RstCtrlGen* and *Decision*. The *CORDICatan* and *SinCos* entities are implemented as shown in Figures 6.22 and 6.23 and are contained within the *CORDICStages* entity. Similarly, the *RstCtrlGen*, *AutoCorr*, *CycFreqNCO*, *Multiplex* and *Rpt* entities are implemented the same as previously for the Split-CAF GLRT and Split-CAF Low Complexity detectors. The *Numerator* entity is implemented the same as for the Split-CAF Low Complexity detector with different wordlengths. Finally, the *Decision* entity is implemented as shown in Figure 6.21 except that the threshold is $\eta = 1.9449$ since the test statistic is $\Gamma(2, 0.5)$ distributed under the null hypothesis.

### 6.4.7 Split-CAF Quantised Detector

The final test statistic for this detector is:

$$\hat{T}_{SC-Q} = N\left(\left|(\hat{I}^{\alpha}_{qq}[\nu])\right|^{2} + \left|(\hat{Q}^{\alpha}_{qq}[\nu])\right|^{2}\right). \tag{6.36}$$

The implementation of the design in HDL coder is shown in Figure 6.32.

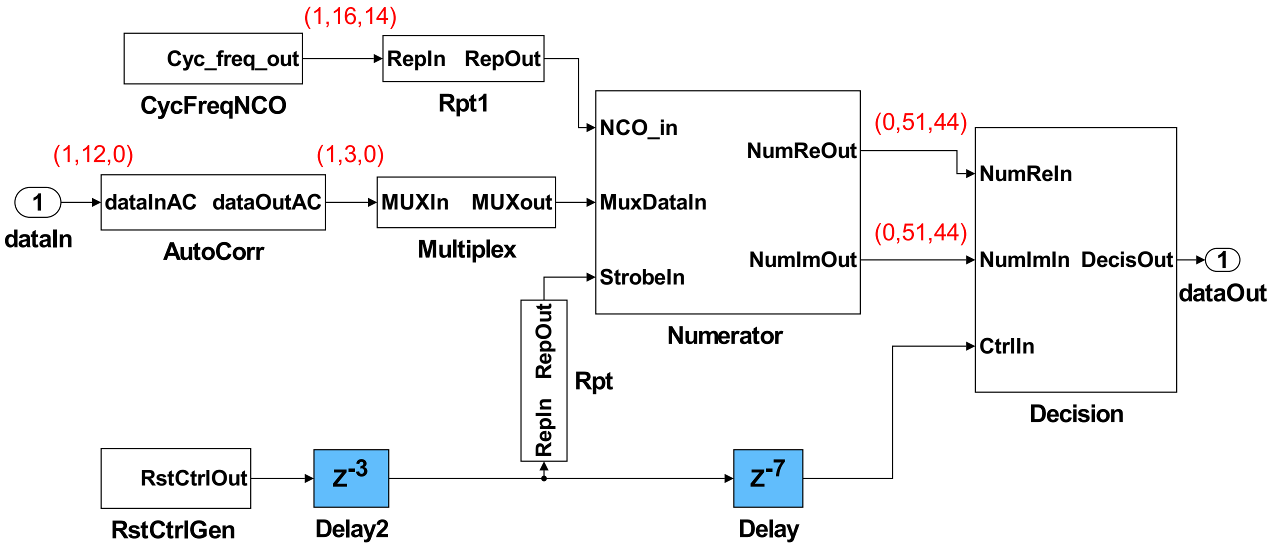


Figure 6.32: Implementation of Split-CAF Quantised detector in HDL Coder

The design consists of seven main entities: *AutoCorr*, *RstCtrlGen*, *Multiplex*, *CycFreqNCO*, *Rpt*, *Numerator* and *Decision*. All entities are implemented the same as previously for the Split-CAF Spatial Sign detector, minus the CORDIC blocks. However, the *AutoCorr* operation has been changed to incorporate the quantisation operation. This block still requires a delay, a complex conjugate and a complex multiplier which is comprised of four real multipliers as shown in Figure 6.8. However, each of the real multipliers is implemented using the circuit shown in Figure 6.33.

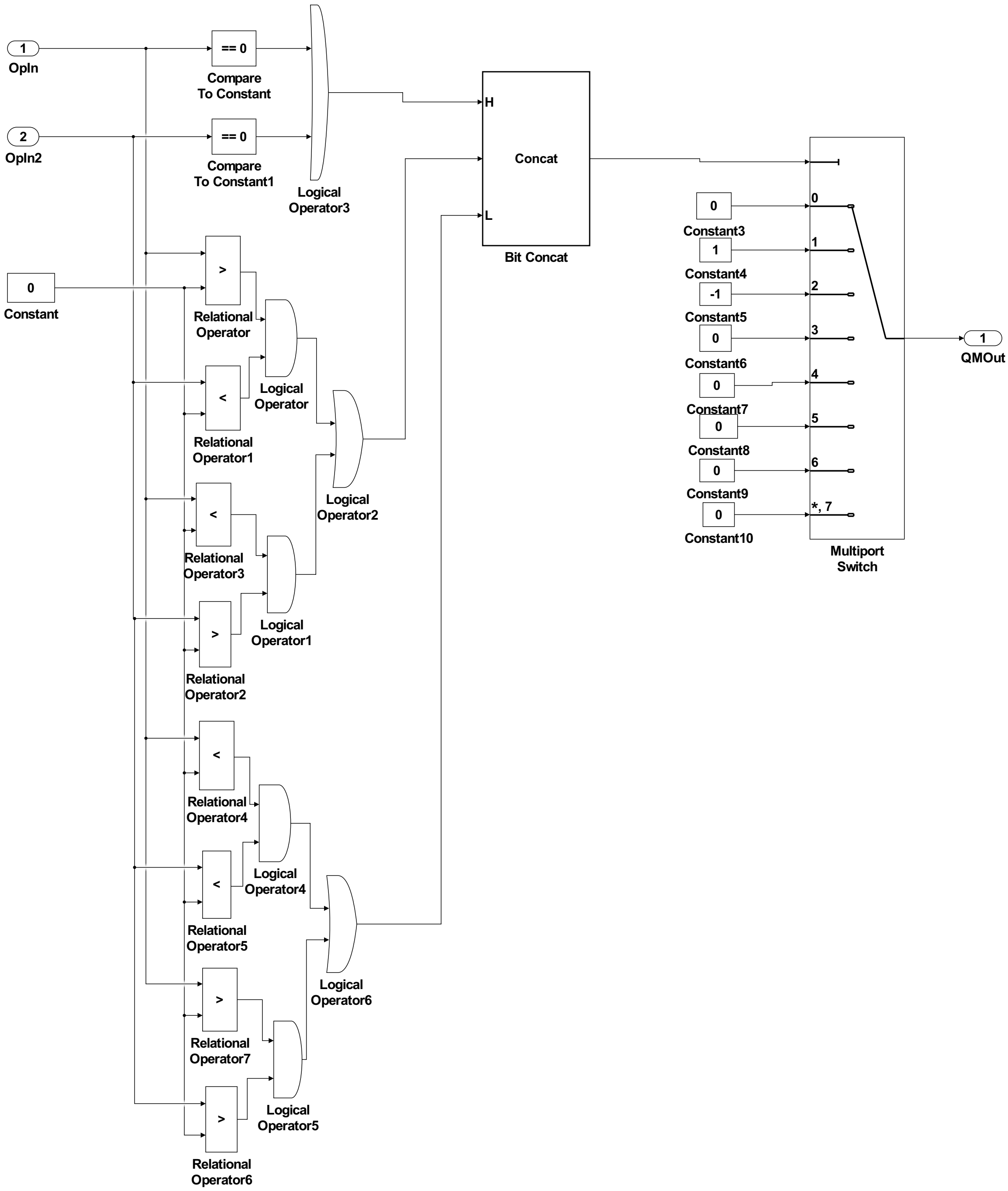


Figure 6.33: Implementation of Quantised Multiplier in HDL Coder

In the first stage of this circuit, there are three parallel branches which check the state of the input operands. Note that each input operand can be equal to zero, a positive real number or a negative real number. If either input operand is zero, then the result of the multiplication is zero. Therefore, the first branch checks for this condition and outputs a boolean 1 if this is true. Secondly, if both operands have different signs, then the result of the multiplication is negative. This condition is checked in the second parallel branch. Finally, in the third branch, the condition where both operands have the same sign is checked. In this case, the result of the multiplication is always positive. Since there are three boolean outputs, then there is theoretically eight possible states. However, only three are actually possible in practice as shown in Table 6.1.

Table 6.1: Possible States for Quantisation Circuit

| **Possible States** | **Meaning** |
|---|---|
| 001 | Same sign |
| 010 | Different signs |
| 100 | One or both equal zero |

The next stage of the circuit is a binary concatenation block which combines the three separate boolean values into a single word of the form shown in Table 6.1. Finally, this is used as the control port of a multiplexer. If the input is “001”, then the output of the multiplexer is the value 1. This is because the result is positive and thus must be quantised to 1. If the input is “010”, then the output is -1 since the result is negative and thus must be quantised to -1. Finally, if the input to the multiplexer is “100”, then the output must be zero. Note that a zero would be passed to the output of the multiplexer if any other input to the control port of the multiplexer occurred. However, as mentioned, this is not possible. Therefore, the circuit in Figure 6.33 implements a real multiplication and quantises the result. This approach has been employed instead of applying the hard quantisation before the autocorrelation function. However, both approaches would yield an identical result. The advantage of the quantisation approach is that the multiplication can be implemented efficiently using low level blocks and therefore does not require the use of embedded multipliers, which is the case for the previous six detectors.

## 6.5 Resource Cost Comparison

Having described the design of each of the considered detection algorithms in HDL Coder, it is now possible to make a comparative analysis of their resource costs when targeted for implementation on an Artix-7 FPGA device. This is achieved by carrying out steps 1-6 in Figure 6.2, after generating VHDL code from the Simulink model using the *makehdl* command in HDL Coder. In addition to the VHDL source files, a XDC file is added to the Vivado project in order to specify a timing constraint that sets the minimum acceptable clock frequency to be achieved for each design. The target device is the Artix-7 xc7a100t csg324-1 FPGA.

Tables 6.2-6.8 show the resource costs of each of the designs in terms of FFs, LUTs, BRAMs and DSP48E1s. Each design was implemented with a timing constraint of 40MHz, which is the minimum clock frequency required for the detection of IEEE 802.11a/g OFDM signals using the proposed Split-CAF detection algorithms. This is due to the fact that $f_s$ = 20MHz and the hardware is shared between two independent data streams, leading to a required clock frequency of 40MHz.

Table 6.2: Resource Utilisation of GLRT Detector

| **FPGA Resource** | **No. Used** | **No. Available** | **% Used** |
|---|---|---|---|
| Flip Flops | 2,778 | 126,800 | 2.18 |
| LUTs | 7,265 | 63,400 | 11.45 |
| Memory LUTs | 8 | 19,000 | 0.04 |
| BRAMs | 1 | 135 | 0.74 |
| DSP48E1s | 48 | 240 | 20 |

Table 6.3: Resource Utilisation of Low Complexity Detector

| **FPGA Resource** | **No. Used** | **No. Available** | **% Used** |
|---|---|---|---|
| Flip Flops | 941 | 126,800 | 0.74 |
| LUTs | 2,870 | 63,400 | 4.53 |
| Memory LUTs | 6 | 19,000 | 0.03 |
| BRAMs | 1 | 135 | 0.74 |
| DSP48E1s | 16 | 240 | 6.67 |

Table 6.4: Resource Utilisation of Spatial Sign Detector

| **FPGA Resource** | **No. Used** | **No. Available** | **% Used** |
|---|---|---|---|
| Flip Flops | 1,308 | 126,800 | 1.03 |
| LUTs | 3,068 | 63,400 | 4.84 |
| Memory LUTs | 47 | 19,000 | 0.25 |
| BRAMs | 1 | 135 | 0.74 |
| DSP48E1s | 12 | 240 | 5 |

Table 6.5: Resource Utilisation of Split-CAF GLRT Detector

| **FPGA Resource** | **No. Used** | **No. Available** | **% Used** |
|---|---|---|---|
| Flip Flops | 5,046 | 126,800 | 3.74 |
| LUTs | 7,992 | 63,400 | 12.96 |
| Memory LUTs | 161 | 19,000 | 0.85 |
| BRAMs | 1 | 135 | 0.74 |
| DSP48E1s | 46 | 240 | 19.17 |

Table 6.6: Resource Utilisation of Split-CAF Low Complexity Detector

| **FPGA Resource** | **No. Used** | **No. Available** | **% Used** |
|---|---|---|---|
| Flip Flops | 1,371 | 126,800 | 1.08 |
| LUTs | 3,201 | 63,400 | 5.05 |
| Memory LUTs | 108 | 19,000 | 0.57 |
| BRAMs | 1 | 135 | 0.74 |
| DSP48E1s | 14 | 240 | 5.83 |

Table 6.7: Resource Utilisation of Split-CAF Spatial Sign Detector

| **FPGA Resource** | **No. Used** | **No. Available** | **% Used** |
|---|---|---|---|
| Flip Flops | 1,890 | 126,800 | 1.49 |
| LUTs | 3,480 | 63,400 | 5.49 |
| Memory LUTs | 49 | 19,000 | 0.26 |
| BRAMs | 1 | 135 | 0.74 |
| DSP48E1s | 10 | 240 | 4.17 |

Table 6.8: Resource Utilisation of Split-CAF Quantised Detector

| **FPGA Resource** | **No. Used** | **No. Available** | **% Used** |
|---|---|---|---|
| Flip Flops | 986 | 126,800 | 0.78 |
| LUTs | 1,538 | 63,400 | 2.43 |
| Memory LUTs | 45 | 19,000 | 0.24 |
| BRAMs | 1 | 135 | 0.74 |
| DSP48E1s | 4 | 240 | 1.67 |

Comparing the GLRT, Low Complexity and Spatial Sign detectors, it can be seen that the most costly design in terms of LUTs and FFs is the GLRT detector. This can be attributed to the fact that the test statistic in (6.4) is appreciably more complex than the test statistics in (6.15) and (6.16) and, therefore, naturally requires more resources. The Spatial Sign detector costs more in terms of fabric than the Low Complexity detector because of the two CORDIC stages. However, the cost of the CORDIC stages could be reduced if required, e.g. by performing less iterations. Therefore, it is not necessarily the case that the Spatial Sign detector is more costly in terms of fabric than the Low Complexity detector.

In order to enable a comparison of the designs in terms of DSP48E1 slices, the multipliers required for the frequency shift by the cyclic frequency will be eliminated, as this operation could be performed using a different method e.g. using CORDIC. In total, the GLRT requires 15 real multiplications, which does not include scaling by constants such as $N$ and $\eta$, which are implemented in the FPGA fabric. However, this maps to a total of 44 DSP48E1s, due to the wordlength growth through the different stages of the detector. In contrast, the Low Complexity detector requires 8 real multiplications, but only requires 12 DSP48E1s due to the far simpler architecture. Finally, the Spatial Sign detector only requires 6 real multiplications due to the elimination of the denominator calculation, mapping to a total of 8 DSP48E1s for this implementation.

Consulting Tables 6.2-6.8, it can be seen that every design consumes less than 1% of the available BRAM and Memory LUTs, which shows that these algorithms require a very small amount of memory resources. Therefore, focus will be placed on the relative costs in terms of LUTs, FFs and DSP48E1s.

Tables 6.5-6.8 show the resource costs of the proposed Split-CAF detectors. Firstly, comparing the Split-CAF GLRT, Split-CAF Low Complexity and Split-CAF Spatial Sign detectors to their original counterparts, it can be seen that a greater amount of FFs and LUTs are required overall. The FF count is expected to increase as the delays in the shared hardware must be scaled by a factor of two. The increase in LUTs can be attributed to additional circuity which includes the multiplexing and demultiplexing operations, the repeat blocks and the adder required to compute the final test statistics. However, the increased cost in terms of LUTs is relatively modest due to the fact that hardware sharing has been employed. This increase in cost in terms of FFs and LUTs is acceptable given the performance improvement which can be achieved using the proposed algorithms and the fact that these resources are abundant for most FPGAs. However, crucially, excluding the multipliers required for the frequency shift, the Split-CAF GLRT, Split-CAF Low Complexity and Split-CAF Spatial Sign detectors require 44, 12 and 8 DSP48E1s respectively. Therefore, due to the use of the hardware sharing architecture and the proposed arrangements of the test statistics, the performance improvement associated with the Split-CAF algorithms comes at no additional cost in terms of embedded multipliers. This is significant as DSP48E1s are a less abundant resource on the FPGA and, thus, saving them is an important consideration. The proposed Split-CAF quantised detector has the lowest LUT count of all of the considered detectors and uses the second lowest amount of FFs after the Low Complexity detector, even considering the fact that time scaling is required for its implementation. Moreover, due to the elimination of the complex multiplier required for the autocorrelation calculation, the design only requires 2 real multiplications which maps to 4 DSP48E1s on the FPGA. This serves to demonstrate the computational efficiency of the proposed Split-CAF Quantised detector.

It is clear that the most hardware efficient design is the Split-CAF Quantised detector, since it consumes the least amount of LUTs and DSP48E1s and has the second lowest FF count of all of the considered approaches. Therefore, when optimising purely for resource cost this detector should be chosen. Conversely, the most costly designs are the GLRT and Split-CAF GLRT detectors. Considering that these detectors offer no performance advantage over the Low Complexity and Split-CAF Low Complexity detectors in any noise environment, the additional cost required for their implementation is not justified. Therefore, it is recommended that the Low Complexity detectors are chosen over the GLRT detectors. The Spatial Sign and Split-CAF Spatial Sign detectors are slightly more costly in terms of fabric than the Low Complexity and Split-CAF Low Complexity detectors for these implementations, due to the requirement for the CORDIC stages. However, this may not always be the case if the implementation is done differently i.e. less iterations are performed. More importantly, the Spatial Sign and Split-CAF Spatial Sign detectors require less embedded multipliers than the Low Complexity and Split-CAF Low Complexity detectors and perform better in impulsive noise environments. Furthermore, the Split-CAF Spatial Sign compares favourably to the Spatial Sign and Split-CAF Quantised detectors in terms of hardware cost, and performs most robustly in all considered noise environments.

To conclude this analysis, the most computationally efficient solution is the Split-CAF Quantised detector and this should be chosen when optimising for hardware cost. However, due to its robust performance in all considered noise environments and its modest hardware cost, it appears that the most optimal solution is the Split-CAF Spatial Sign detector. One drawback of using hardware sharing to implement the Split-CAF detectors is the need for a clock rate which is twice the input sampling period. For DVB-T and IEEE 802.11a/g signals, this translates to a required maximum of 40MHz which is easily achievable. For signals with much higher sampling rates, achieving the required clock rate may become problematic and more attention will be required to ensure that the designs are adequately pipelined. However, it is recommended that hardware sharing is employed when implementing the Split-CAF detectors due to the advantages in terms of resource cost.

## 6.6 Performance Verification

In this section, the performance of the fixed point HDL Coder models are verified. As mentioned previously, the test signal is an IEEE802.11a/g Non-HT OFDM signal observed over a duration of $N = 1600$ samples in AWGN noise. The noise variance is $\sigma_n^2 = 0.05$, $P_{fa} = 0.1$, the range of tested SNRs is -25dB to 5dB in steps of 2dB and a total of 1000 iterations are performed at each SNR level. A 2dB step and 1000 iterations were used in order to reduce the time required for the simulation. Figure 6.34 compares $P_d$ vs. SNR curves for the HDL Coder models of each detector using floating point arithmetic. The floating point models are intended as a golden reference, which can be used to verify that the fixed point designs work as expected.

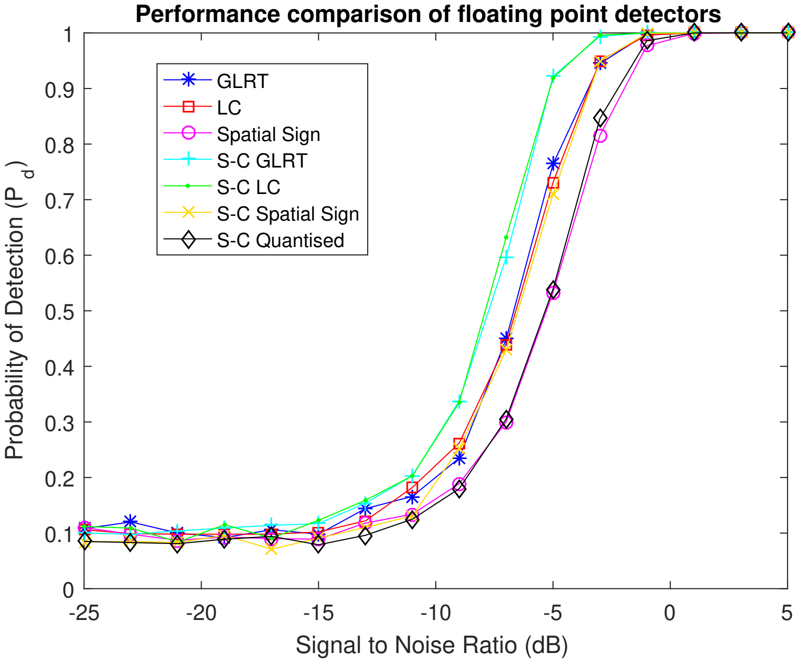


Figure 6.34: $P_d$ vs. SNR Floating Point HDL Coder Models

It can be seen that the best overall performance in the AWGN channel is obtained for the Split-CAF GLRT and Split-CAF Low Complexity detectors. This is followed by the GLRT, Low

Complexity and Split-CAF Spatial Sign detectors. Finally, the poorest performance is exhibited by the Spatial Sign and Split-CAF Quantised detectors. These results are in accordance with the findings reported in Chapters 4 and 5, thus showing that the HDL Coder models function correctly. Figures 6.35-6.41 compare the performance of the floating point and fixed point HDL Coder models. It can be observed that the curves match very closely for each of the detectors, which verifies that the designs work as expected.

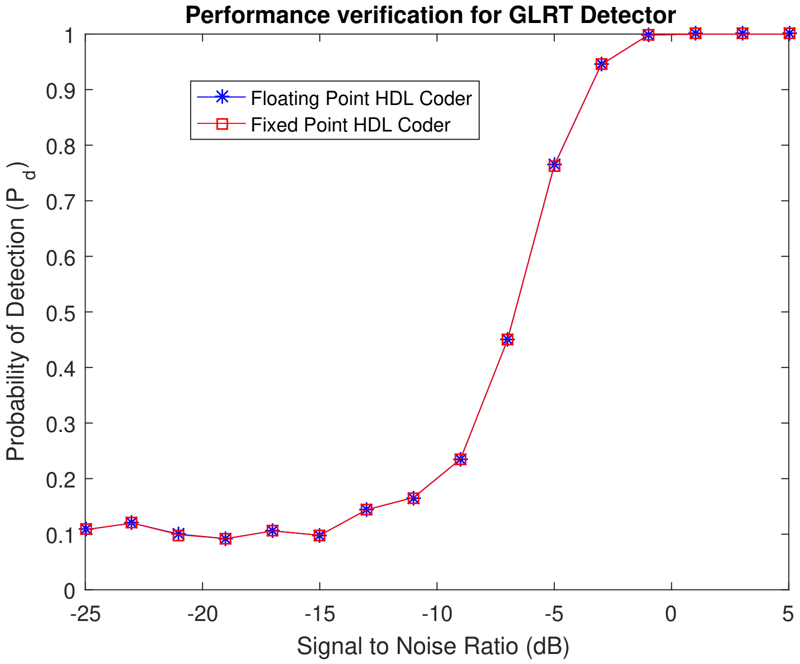


Figure 6.35: Performance verification of GLRT detector in HDL Coder

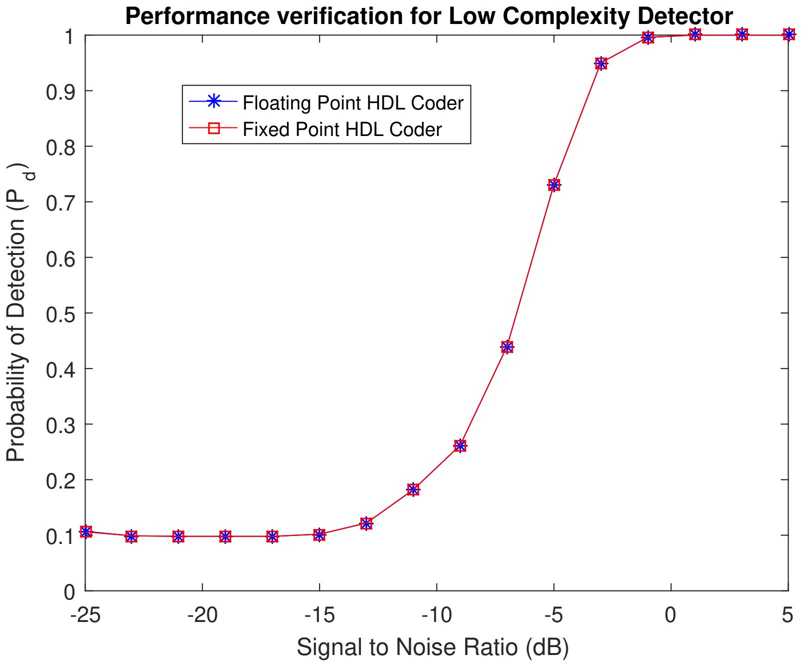


Figure 6.36: Performance verification of Low Complexity detector in HDL Coder

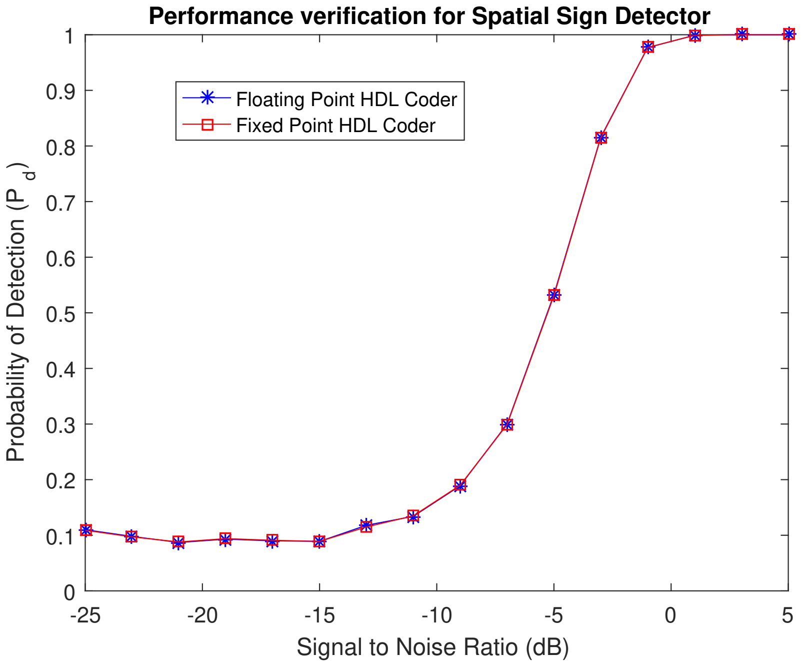


Figure 6.37: Performance verification of Spatial Sign detector in HDL Coder

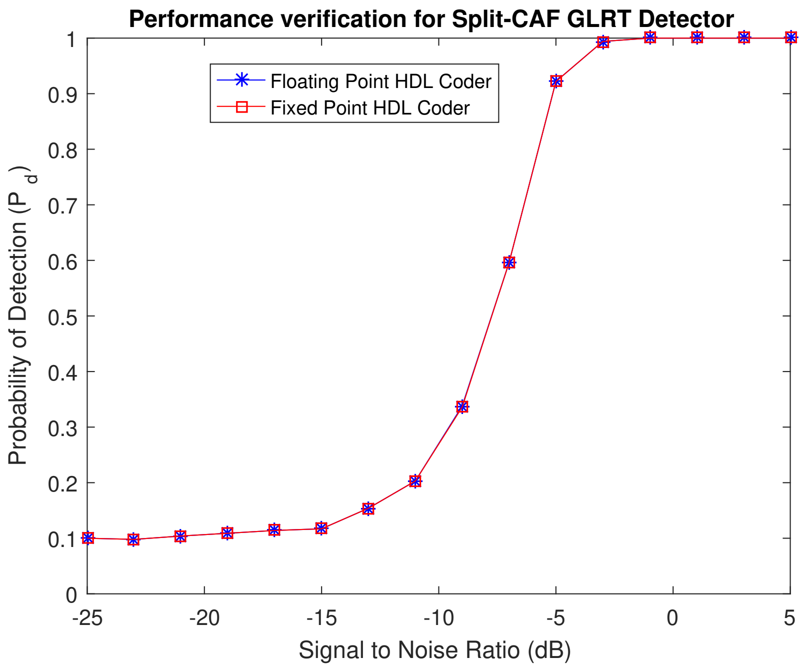


Figure 6.38: Performance verification of Split-CAF GLRT detector in HDL Coder

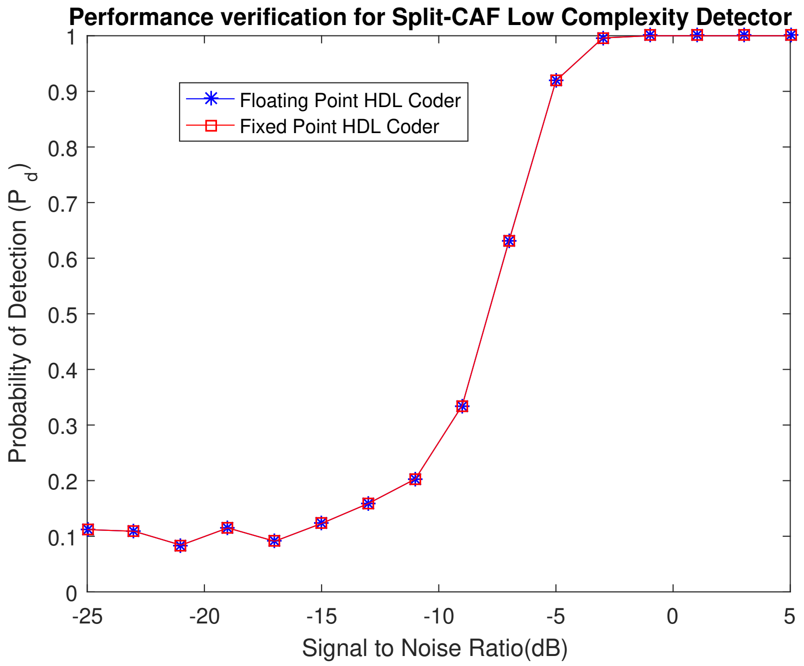


Figure 6.39: Performance verification of Split-CAF Low Complexity detector in HDL Coder

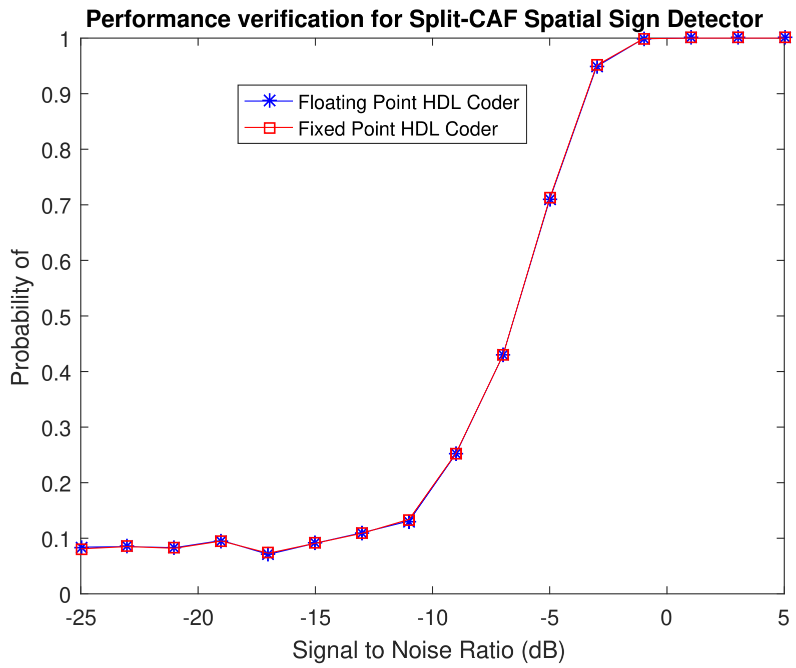


Figure 6.40: Performance verification of Split-CAF Spatial Sign detector in HDL Coder

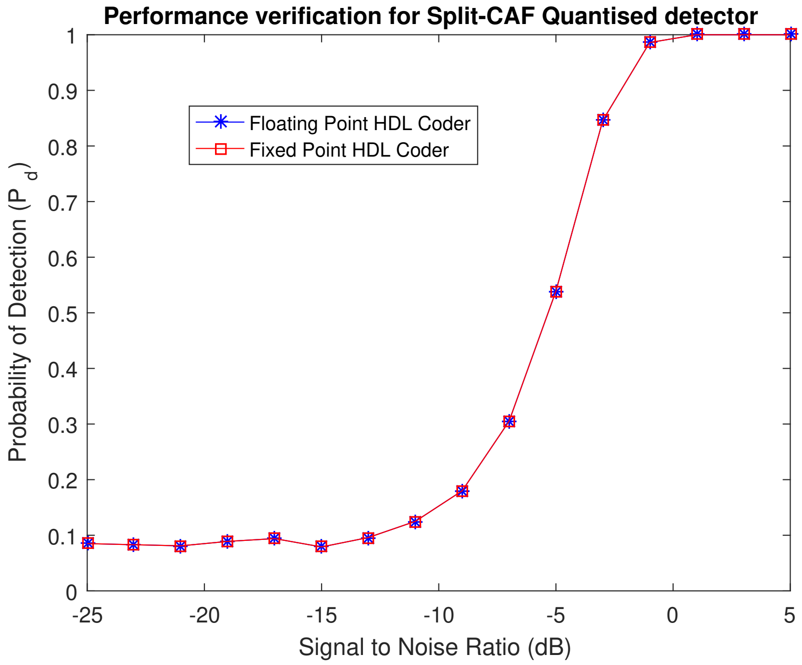


Figure 6.41: Performance verification of Split-CAF Quantised detector in HDL Coder

In addition to verifying the performance of the HDL Coder models, the designs have been tested on the Artix 7 device using the FPGA-in-the-loop feature provided by MathWorks.

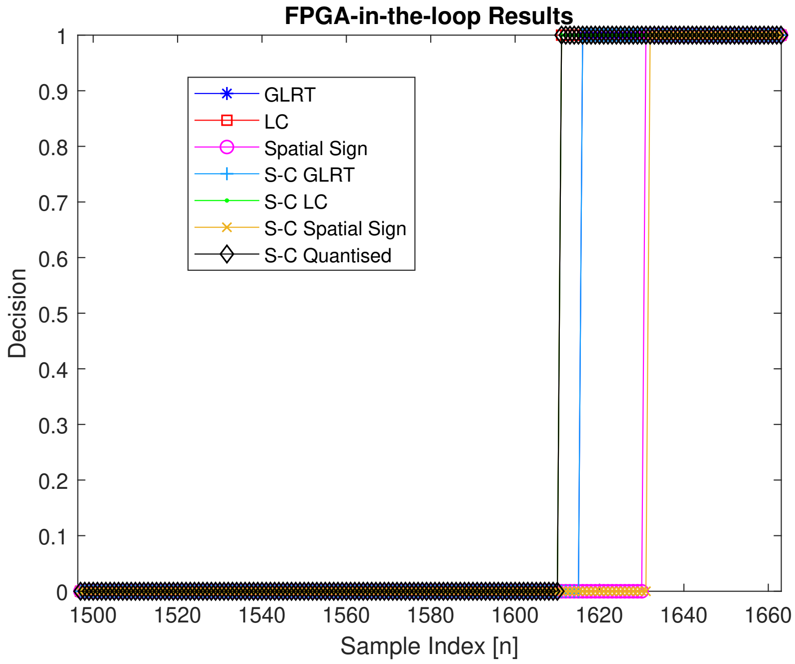


Figure 6.42: Performance verification of detectors using FPGA-in-the-loop

This feature enables simulation of hardware running on an FPGA to be conducted using the Simulink environment. Figure 6.42 shows the results of passing an IEEE 802.11a/g test signal corrupted by AWGN at an SNR of 10dB into each of the implemented detectors. Each detector returns a boolean ‘1’ after $N = 1600$ samples plus some additional latency, indicating the presence of the IEEE 802.11a/g signal and verifying that the designs work as expected on the FPGA. The latency is caused by the addition of pipeline registers in the signal path. Note the Spatial Sign and Split-CAF Spatial Sign detectors incur the largest latency due to the CORDIC stages, where registers are placed in between each cell.

## 6.7 Chapter Summary

In conclusion, this chapter has considered the FPGA implementation of the various cyclostationary feature detection algorithms introduced in this thesis. The chapter started by reviewing the software tools and features that were required for this task which included HDL Coder and Vivado. Following this, a brief review of multi-channel hardware sharing was provided.

Having covered the necessary background, a detailed description of the implementation of each detector in HDL Coder was provided. It is desirable to ensure that FPGA resources are utilised as efficiently as possible and, therefore, multi-channel hardware sharing was employed in the implementation of the proposed detectors. Furthermore, a simple mathematical re-arrangement was proposed which renders the division operation unnecessary for test statistics that are expressed as a fraction. This is useful because division can often be a costly and difficult operation to implement in FPGA hardware.

The relative resource cost of each of the algorithms was assessed for implementation on a Xilinx Artix-7 FPGA and it was shown that the most efficient design in terms of area was the proposed Split-CAF Quantised detector. Therefore, when optimising for hardware cost, this detector should be employed. In contrast, it was found that the most costly designs were the GLRT and Split-CAF GLRT detectors. Due to the fact that they offer no performance benefit in any noise environment when compared to the Low Complexity and Split-CAF Low Complexity detectors, it is recommended that these are not used. The Spatial Sign and Split-CAF Spatial Sign detectors are advantageous compared to the Low Complexity detectors because they use less embedded multipliers and perform more robustly in impulsive noise. Finally, due to its superior overall performance in AWGN and impulsive noise and its modest hardware cost, it appears that the most optimal solution is the Split-CAF Spatial Sign detector.

# Chapter 7

# Conclusions

## 7.1 Resume

At a high level, this thesis has addressed the problem of spectrum sensing of OFDM signals using cyclostationarity. In particular, several algorithms that are based on estimation of the CAF have been derived and their relative detection performance has been assessed for both AWGN and impulsive noise environments. Subsequently, the relative costs of each of the algorithms have been analysed and compared when targeting a Xilinx FPGA platform. The contents of this thesis can now be summarised in more detail.

**In Chapter 4**, an overview of the signal detection problem and the hypothesis testing framework was provided. Following this, it was established that OFDM signals are wide sense cyclostationary due to the inclusion of a CP, and that this property can be exploited for detection purposes. From the literature, three prominent algorithms were introduced that involve the computation of statistical tests based on estimation of the CAF. These were the GLRT, Low Complexity and Spatial Sign detectors. The distributions of each of these test statistics was established under the null hypothesis, allowing a threshold to be set that guarantees a desired probability of false alarm. Having derived each of the algorithms, their performance in detection of IEEE 802.11a/g and DVB-T waveforms was assessed in both AWGN and impulsive noise environments.

**In Chapter 5**, the complex CAF was split into two complex component functions, which were used as the basis for four new detection algorithms: the Split-CAF GLRT, Split-CAF Low Complexity, Split-CAF Spatial Sign and Split-CAF Quantised detectors. The Split-CAF GLRT, Split-CAF Low Complexity and Split-CAF Spatial Sign detectors can be considered re-derivations of the original algorithms introduced in Chapter 4, using the proposed CAF splitting approach. Conversely, the Split-CAF Quantised detector is a completely new algorithm that combines a hard quantisation of the input data with the CAF splitting method. As was the case for the original algorithms, the proposed test statistics were derived from first principles and their distributions under the null hypothesis were established. The performances of the proposed detectors was assessed for detection of IEEE 802.11a/g and DVB-T signals in both AWGN and impulsive noise environments and these results were compared to the algorithms in Chapter 4. Furthermore, the performance of the algorithms in multipath channels and in the presence of an uncorrected frequency offset was assessed.

**In Chapter 6**, all seven detection algorithms were implemented using HDL Coder and targeted to a Xilinx 7 Series FPGA. A thorough description of the implementation of each algorithm was provided and the relative resource costs of each algorithm were compared and analysed. In terms of FPGA design, it was suggested that a simple mathematical re-arrangement be applied to test statistics expressed as a fraction in order to circumvent a costly division operation. Also, for the Split-CAF algorithms, hardware sharing techniques were applied in order to ensure that resources were not used unnecessarily.

## 7.2 Summary of Results

In Chapter 4, it was established that the best detection performance in an AWGN channel was achieved by the GLRT and Low Complexity detectors. In contrast, it was found that the Spatial Sign detector exhibits a performance loss when compared to the GLRT and Low Complexity detectors in the AWGN channel. However, the Spatial Sign detector performs robustly in both contaminated Gaussian and Cauchy impulsive noise. In contrast, the performance of the GLRT and Low Complexity detectors degrades significantly in both types of impulsive noise. In particular, these detectors are unable to achieve the desired $P_{fa}$ when operating in Cauchy impulsive noise. These results were obtained for both IEEE 802.11a/g and DVB-T test signals.

In Chapter 5, it was found that the best performance in an AWGN channel was achieved using the Split-CAF GLRT and Split-CAF Low Complexity detectors. Further, the Split-CAF Spatial Sign detector matched the performance of the GLRT and Low Complexity detectors and the Split-CAF Quantised detector matched the performance of the Spatial Sign detector. These results are repeated in Figure 7.1 below for an IEEE 802.11a/g test signal.

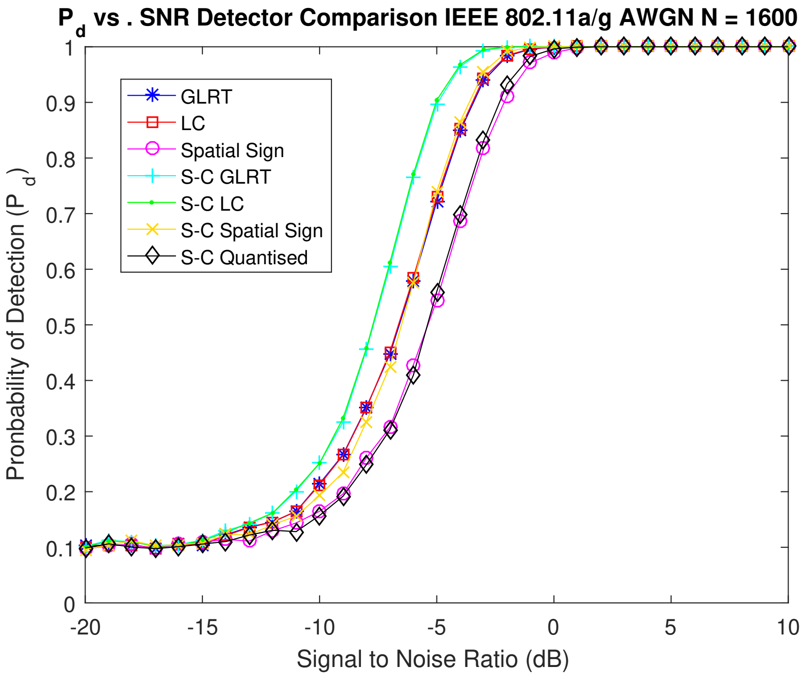


Figure 7.1: $P_d$ vs. SNR comparison of all detectors IEEE 802.11a/g in AWGN

In contaminated Gaussian noise, the best performing detector was the Split-CAF Spatial Sign detector. This was closely followed the Split-CAF Quantised detector which slightly outperformed the Spatial Sign detector. The performance of the Split-CAF GLRT and Split-CAF Low Complexity detectors degraded significantly in the contaminated Gaussian noise. However, they were still able to improve upon the GLRT and Low Complexity detectors. Conversely, in the Cauchy noise, the Split-CAF Quantised detector achieved the best overall performance. This was followed closely by the Split-CAF Spatial Sign detector which performed marginally worse. However, both detectors outperformed the Spatial Sign detector in this noise environment. As with the GLRT and Low Complexity detectors, it was found that the Split-CAF GLRT and Split-CAF Low Complexity detectors were unable to achieve the desired false alarm rate due to to the fact that the asymptotic normality of the CAF is violated by the Cauchy noise. When considering all noise environments together, it was found that the Split-CAF Spatial Sign detector achieved the best overall performance. The results for both contaminated Gaussian and Cauchy noise for an IEEE802.11a/g test signal are repeated in Figures 7.2 and 7.3.

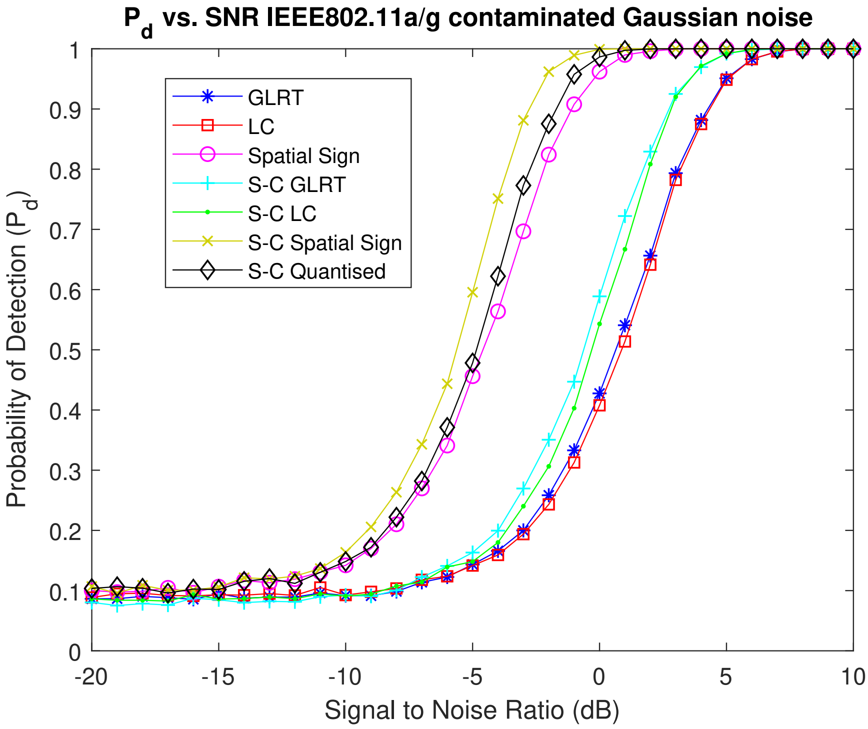


Figure 7.2: $P_d$ vs. SNR comparison of all detectors IEEE 802.11a/g in contaminated Gaussian Impulsive noise

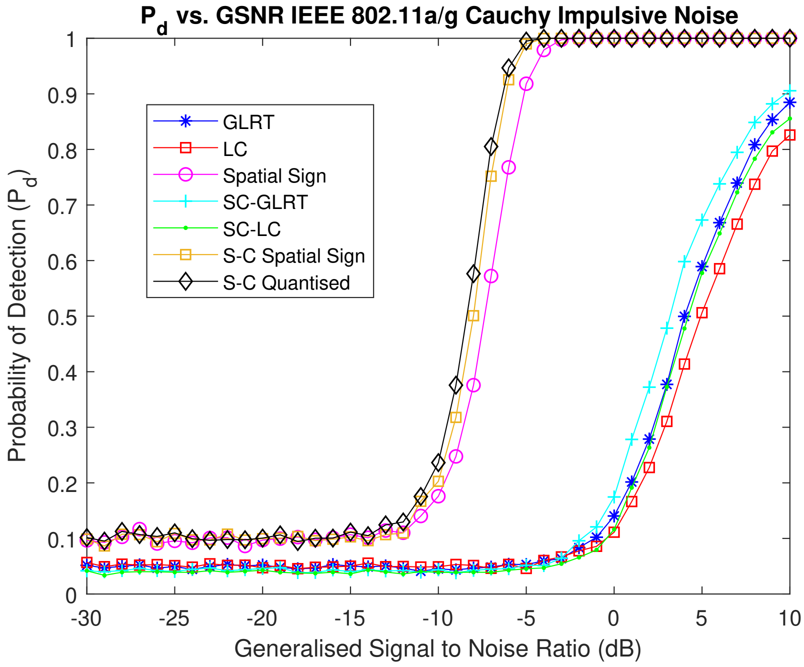


Figure 7.3: $P_d$ vs. GSNR detector comparison for IEEE802.11a/g in Cauchy Impulsive noise

This analysis has demonstrated that employing the proposed Split-CAF algorithms can offer an appreciable performance improvement over the traditional algorithms without employing any additional information e.g. autocorrelation lags and/or cyclic frequencies. Furthermore, it was shown that all of the algorithms are sensitive to increasing delay spread and decreasing coherence times in time varying multipath channels, but can perform robustly in the presence of an uncorrected frequency offset.

In Chapter 6, it was found that the most resource efficient algorithm was the Split-CAF Quantised detector and its should be chosen if optimising for hardware cost is the primary objective. It is also able to operate robustly in AWGN and outperforms the Spatial Sign detector in both types of impulsive noise. In contrast, the Split-CAF GLRT and GLRT detectors were the most computationally expensive and consumed the most resources. Due to the fact that they offer no performance advantage and are significantly more costly than the Split-CAF Low Complexity and Low Complexity detectors, it is recommended that these algorithms are not used in practice. In fact, for an AWGN channel, the most optimal solution is the Split-CAF Low Complexity detector. However, the Spatial Sign and Split-CAF Spatial Sign detectors have the advantage that they operate robustly in impulsive noise and require less embedded multipliers than the Low Complexity detectors. Furthermore, the Split-CAF Spatial Sign detector matches the performance of the Low Complexity and GLRT detectors in AWGN and outperforms the Spatial Sign detector in both impulsive noise environments. Overall, when taking into account performance in both AWGN and impulsive noise and hardware cost, the Split-CAF Spatial Sign detector is the most optimal choice of all considered detectors. A disadvantage of the approach used to implement the Split-CAF algorithms is that the designs must be capable of running at a clock of frequency equal to twice the input sampling rate, which could become problematic when detecting signals with high sampling rates. However, the savings that can be achieved through the use of hardware sharing makes it a very important feature of the implementations of the Split-CAF algorithms.

## 7.3 Future Work

In terms of future work, there are several points that would be interesting to address:

- In Chapter 5, four detection algorithms were derived based on the proposed CAF splitting technique. It would be interesting to see if it were possible to derive further test statistics using this method and to compare their performances to the proposed algorithms. In this thesis, the spatial sign and quantisation functions were applied and combined with the CAF splitting approach. However, for example, other non-linearities exist such as the Soft Limiter function [85] which could be combined with the CAF splitting to form a new detector. Furthermore, it would be interesting to see how any new algorithm might compare to the proposed algorithms when implemented on an FPGA e.g. in terms of resource cost.

- In this thesis, only a single lag and single cyclic frequency was considered in testing the performance of the proposed algorithms. Therefore, a next step would be to extend the test statistics to incorporate additional lags and/or cyclic frequencies and measure the performance improvement that can be achieved. Furthermore, only a single receive antenna was employed and, therefore, it would be interesting to see how the proposed detection algorithms could be combined with spatial diversity to achieve an improved detection performance.

- In [85], a mathematical proof is provided which justifies the observation that the cyclostationary features are preserved after applying the spatial sign function. It would be interesting to see if this method or similar could be used to justify the preservation of the cyclostationary features after applying the hard quantisation in Chapter 5.

- This thesis has focussed exclusively on applying cyclostationary detection algorithms to the detection of OFDM primary user waveforms. Therefore, a future goal for this research would be to test the algorithms proposed in Chapter 5 for different possible primary user signals and standards.

# Appendices

# Appendix A

# Additional Results

This appendix contains additional results that were omitted from the main body of the thesis. These consist mainly of results to verify the existing and proposed detectors using DVB-T signals with different CP lengths i.e. 1/8, 1/16 and 1/32 of the useful symbol duration.

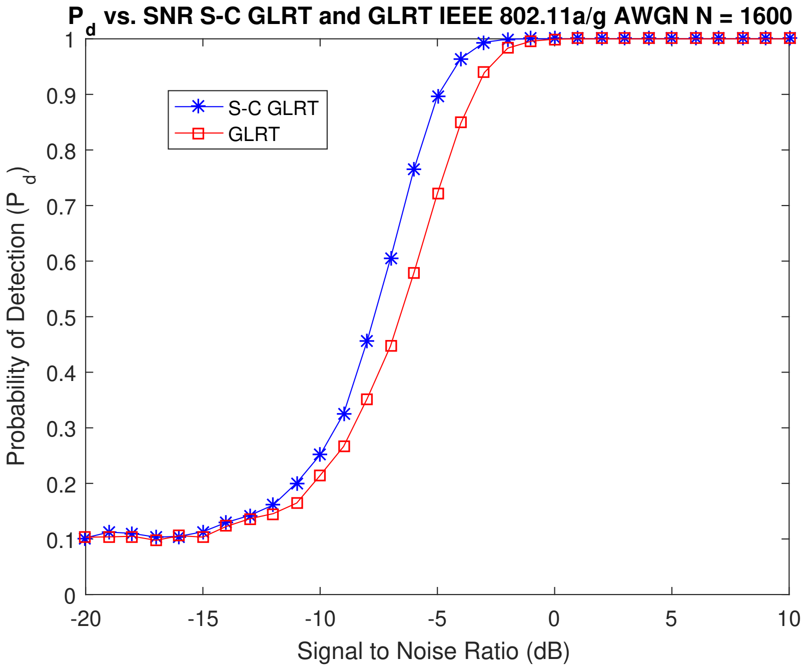


Figure A.1: $P_d$ vs. SNR comparison for Split-CAF GLRT and GLRT detectors in AWGN

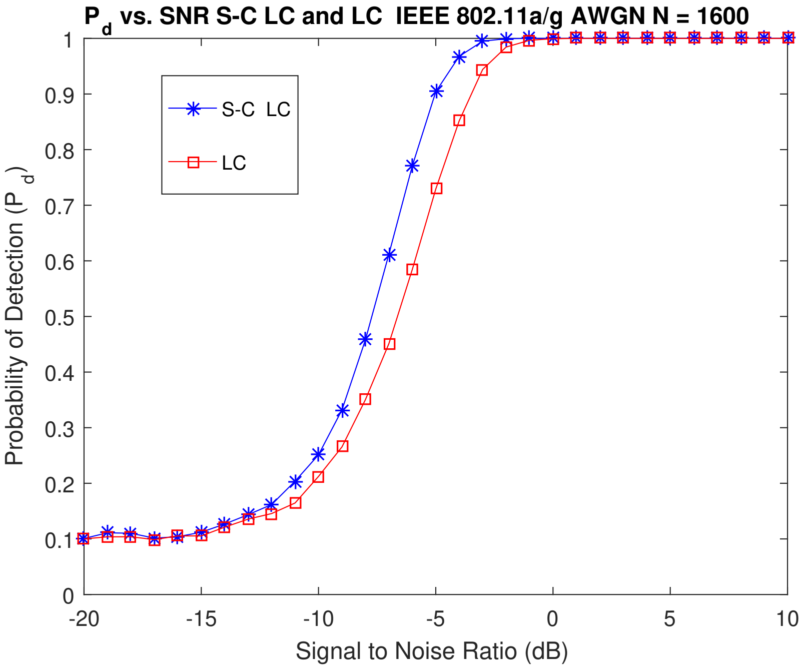


Figure A.2: $P_d$ vs. SNR comparison Split-CAF LC and LC detectors in AWGN

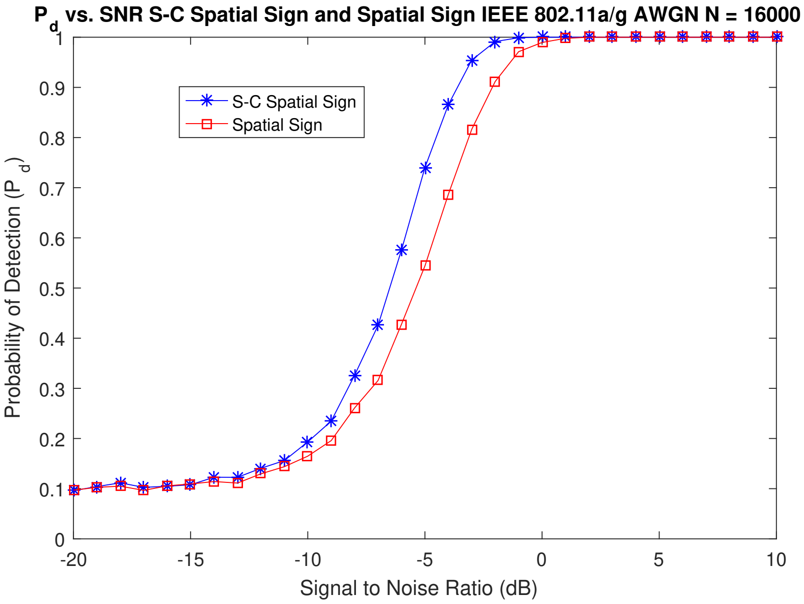


Figure A.3: $P_d$ vs. SNR comparison Split-CAF Spatial Sign and Spatial Sign Detectors in AWGN

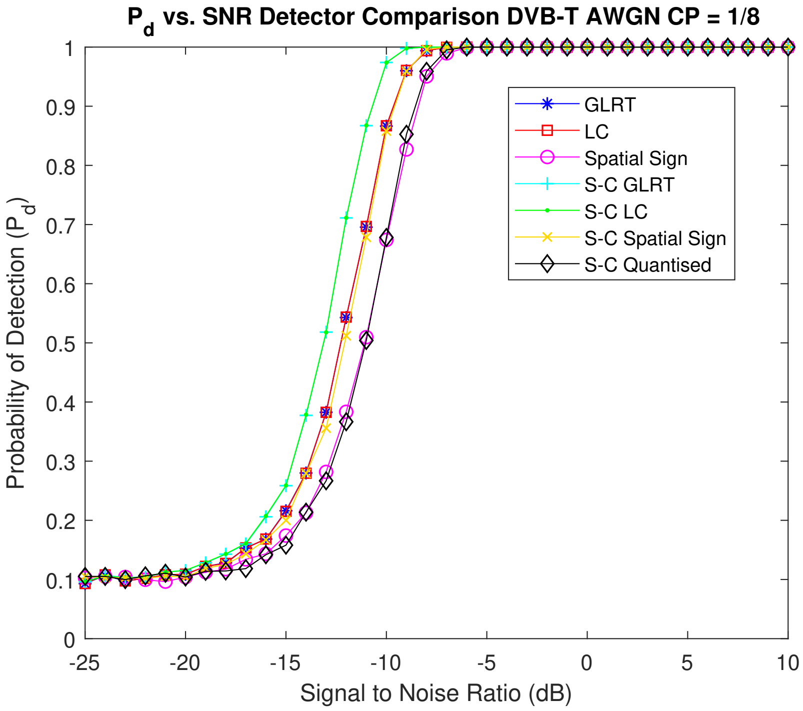


Figure A.4: $P_d$ vs. SNR comparison of all detectors DVB-T in AWGN CP = 1/8

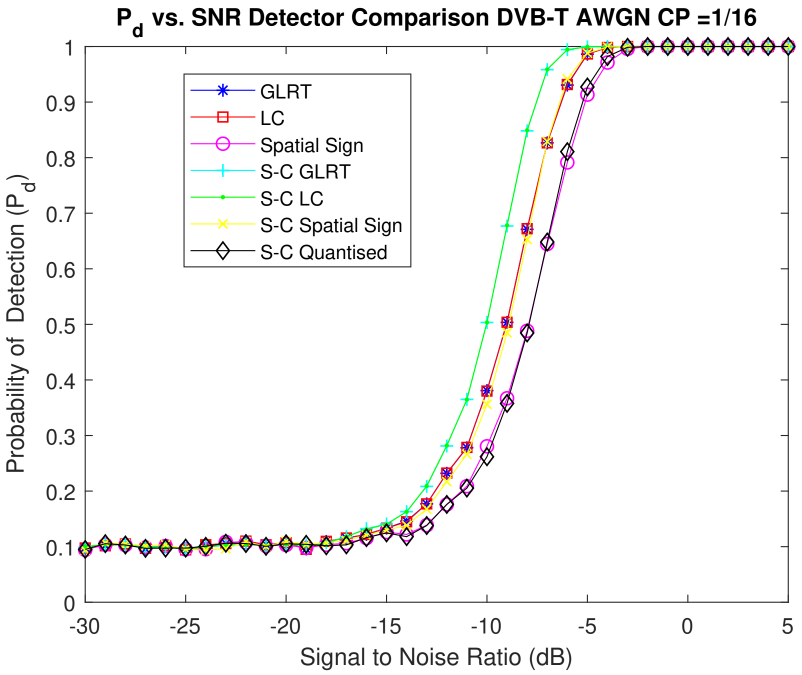


Figure A.5: $P_d$ vs. SNR comparison of all detectors DVB-T in AWGN CP = 1/16

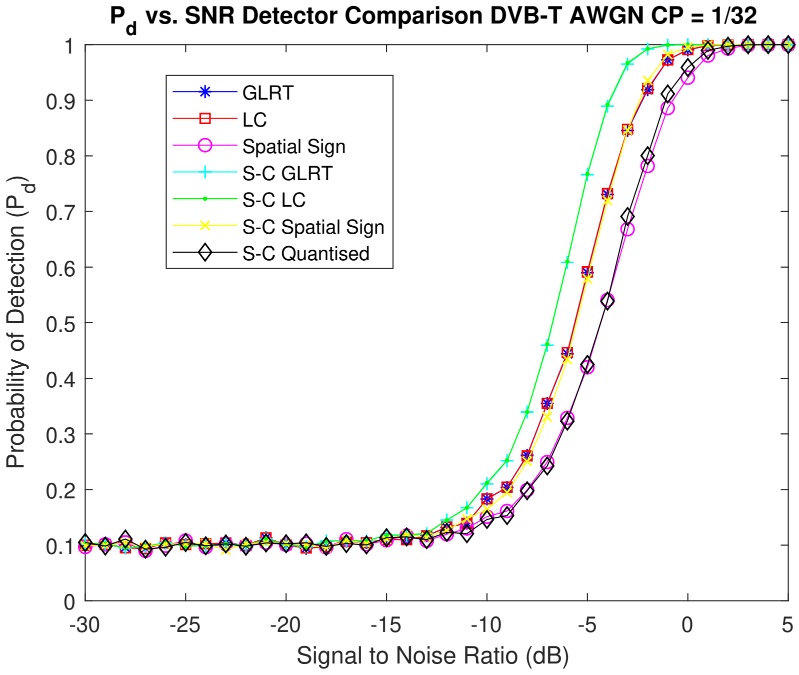


Figure A.6: $P_d$ vs. SNR comparison of all detectors DVB-T in AWGN CP = 1/32

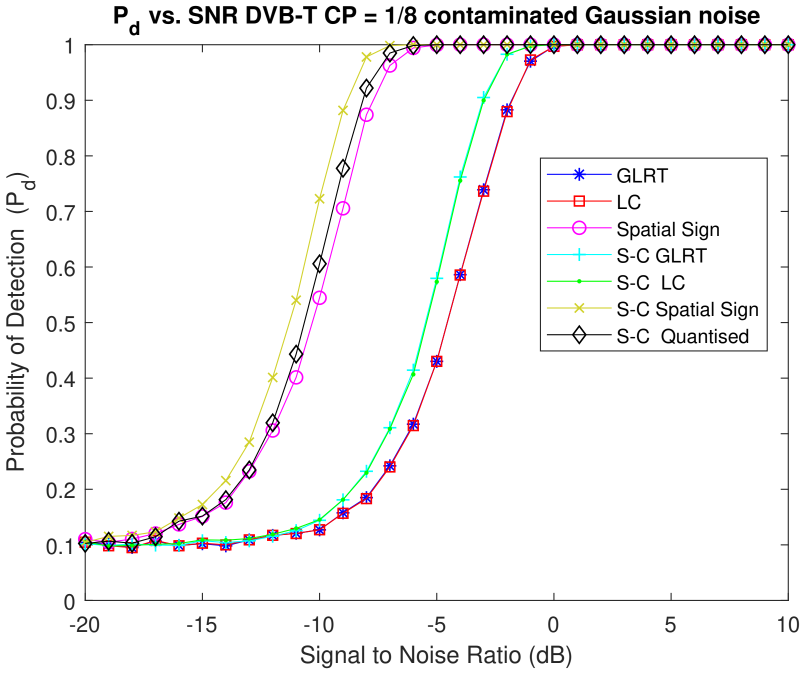


Figure A.7: $P_d$ vs. SNR comparison of all detectors DVB-T CP = 1/8 in contaminated Gaussian noise

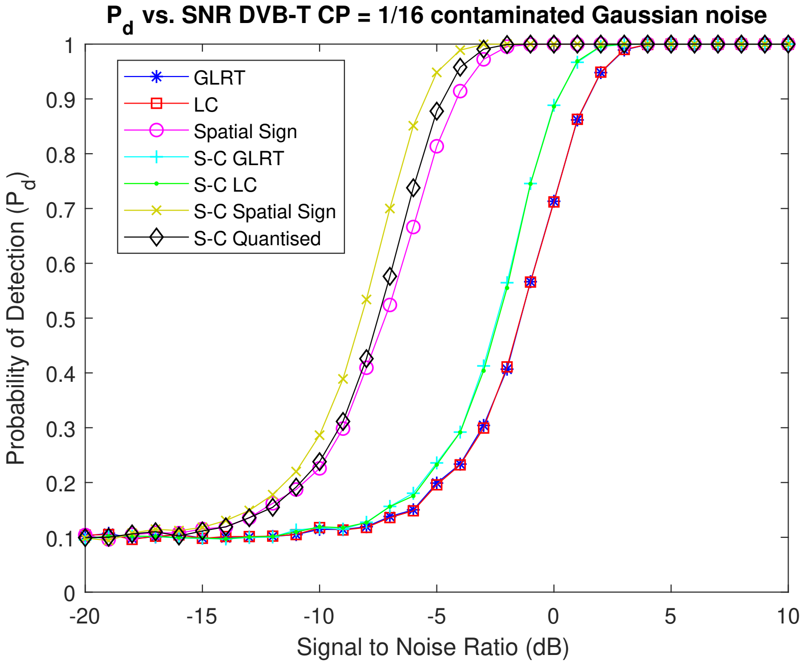


Figure A.8: $P_d$ vs. SNR comparison of all detectors DVB-T CP = 1/16 in contaminated Gaussian noise

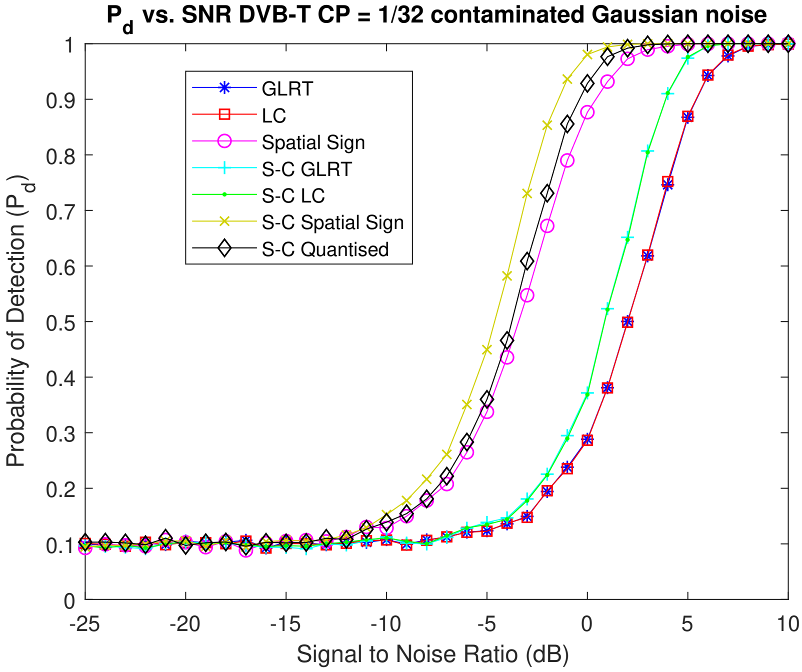


Figure A.9: $P_d$ vs. SNR comparison of all detectors DVB-T CP = 1/32 in contaminated Gaussian noise

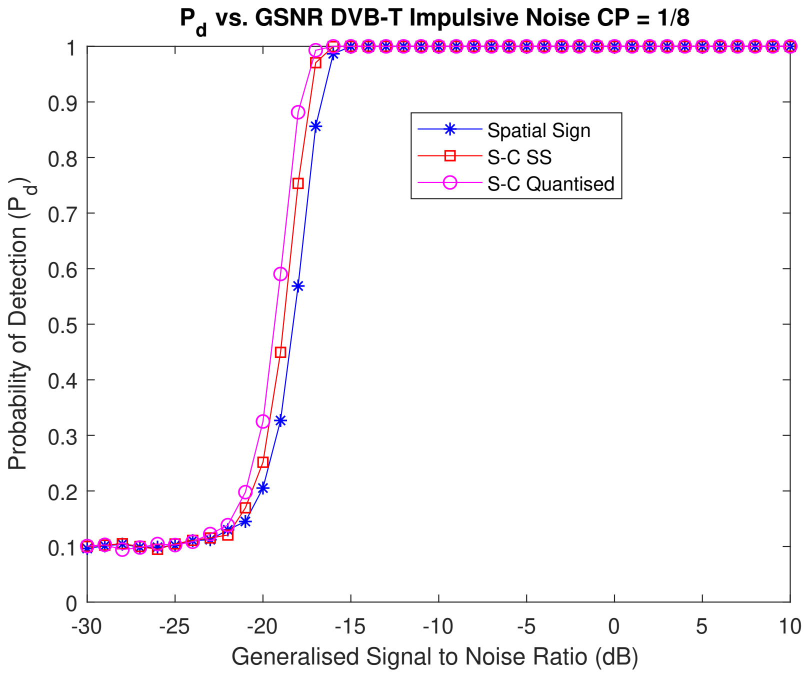


Figure A.10: $P_d$ vs. GSNR comparison DVB-T CP = 1/8 in Cauchy impulsive noise

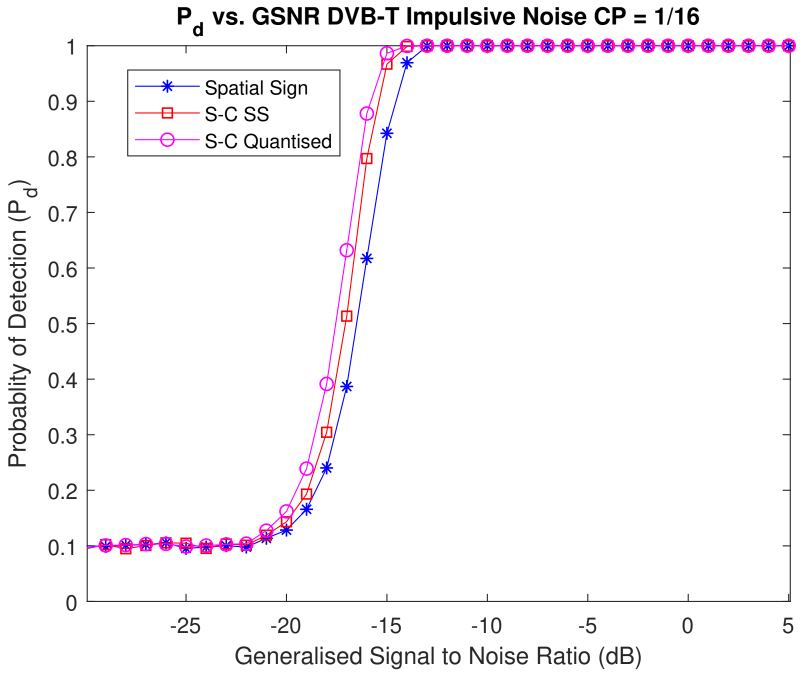


Figure A.11: $P_d$ vs. GSNR comparison DVB-T CP = 1/16 in Cauchy impulsive noise

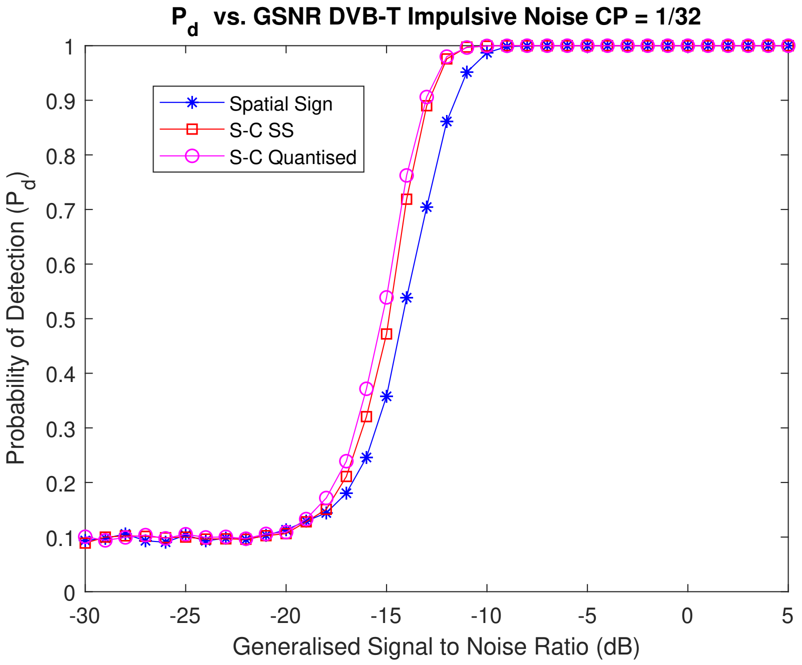


Figure A.12: $P_d$ vs. GSNR comparison DVB-T CP = 1/32 in Cauchy impulsive noise

# Bibliography

[1] D. Allan, L. Crockett, and R. Stewart, "A Low Complexity Cyclostationary Detector for OFDM Signals," in *Proc. New Generation of Circuits and Systems*, Genova, Italy, Sep. 6–9 2017.

[2] D. Allan, L. Crockett *et al.*, "FPGA implementation of a cyclostationary detector for OFDM signals," in *Proc. EUSIPCO '16*, Budapest, Hungary, Aug. 29–Sep.2 2016.

[3] A. Tanenbaum and D. Wetherall, *Computer Networks*, 5th ed. Boston, MA: Prentice Hall, 2011.

[4] R. Stewart, K. Barlee *et al.*, *Software Defined Radio using MATLAB and Simulink and the RTL-SDR*, 1st ed. Glasgow, UK: Strathclyde Academic Media, 2015.

[5] A. Goldsmith, *Wireless Communications*, 1st ed. Cambridge: Cambridge University Press, 2005.

[6] G. Rice, "On delay tracking strategies for WCDMA receivers," Ph.D. dissertation, Univ. of Strathclyde, Glasgow, 2003.

[7] I. Glover and P. Grant, *Digital Communications*, 2nd ed. Essex, UK: Prentice Hall, 2015.

[8] D. M. Barnes, "The introduction of cellular radio in the United Kingdom," in *Proc. 35th IEEE Conference on Vehicular Technology*, Colorado, Usa, May 21–23, 1985.

[9] S. Sesia, I. Toufik *et al.*, *LTE - The UMTS Long Term Evolution: From Theory to Practice*, 2nd ed. Chichester, UK: Wiley, 2011.

[10] *GSM 03.60 Release 1997: Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); Service Description Stage 2*, 3GPP Std., 1999.

[11] *3GPP TS 36.211 V8.4.0 Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 8)*, 3GPP Std., 2008/2009.

[12] *IEEE Standard for Information Technology-Telecommunications and Information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: WLAN MAC and PHY specifications*, IEEE Std., 2012.

[13] S. Haykin, "Cognitive Radio : Brain-Empowered Wireless Communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[14] What is Software Defined Radio? [Online]. Available: https://www.wirelessinnovation.org/Introduction_to_SDR

[15] L. Crockett, R. A. Elliot *et al.*, *The Zynq Book: Embedded Processing with the ARM Cortex - A9 on the Xilinx Zynq - 7000 All Programmable SoC*, 1st ed. Glasgow, UK: Strathclyde Academic Media, 2014.

[16] E. Hossain, D. Niyato, and Z. Han, *Dynamic Spectrum Access and Management in Cognitive Radio Networks*, 1st ed. NY, USA: Cambridge University Press, 2009.

[17] F. Harris and W. Lowdermilk, "Software Defined Radio: Part 22 in a Series of Tutorials on Instrumentation and Measurement," *IEEE Instrumentation and Measurement Magazine*, vol. 13, no. 1, pp. 23–32, 2010.

[18] A Software Driven Approach to SDR Design. [Online]. Available: https://web.archive.org/web/20110609125008/http://www.cotsjournalonline.com/articles/view/100056

[19] M. Song, C. Xin *et al.*, "Dynamic Spectrum Access: From Cognitive Radio to Network Radio," *IEEE Wireless Communications*, vol. 19, no. 1, pp. 23–29, 2012.

[20] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, 1st ed. NY, USA: John Wiley and Sons, 1996.

[21] Vivado High Level Synthesis. [Online]. Available: https://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html

[22] System Generator for DSP. [Online]. Available: https://www.xilinx.com/products/design-tools/vivado/integration/sysgen.html

[23] MathWorks. Hdl Coder. [Online]. Available: www.mathworks.com/products/hdl-coder.html

[24] UltraScale+ FPGAs Product Tables and Product Selection Guide. [Online]. Available: https://www.xilinx.com/support/documentation/selection-guides/ultrascale-plus-fpga-product-selection-guide.pdf

[25] UltraScale FPGA Product Tables and Product Selection Guide. [Online]. Available: https://www.xilinx.com/support/documentation/selection-guides/ultrascale-fpga-product-selection-guide.pdf

[26] All Programmable 7 Series Production Selection Guide. [Online]. Available: https://www.xilinx.com/support/documentation/selection-guides/7-series-product-selection-guide.pdf

[27] Zynq UltraScale+ MPSoC Product Tables and Product Selection Guide. [Online]. Available: https://www.xilinx.com/support/documentation/selection-guides/zynq-ultrascale-plus-product-selection-guide.pdf

[28] Zynq-7000 All Programmable SoC Family Product Tables and Product Selection Guide. [Online]. Available: https://www.xilinx.com/support/documentation/selection-guides/zynq-7000-product-selection-guide.pdf

[29] J. Heiskala and J. Terry, *OFDM Wireless LANs : A Theoretical and Practical Guide*, 1st ed. Indianapolis, USA: Sams Publishing, 2002.

[30] A. V. Oppenheim and A. S. Willsky, *Signals and Systems Second Edition*, 2nd ed. New Jersey, USA: Prentice Hall, 1997.

[31] J. D. Guffey, "OFDM Physical Layer Implementation for the Kansas University Agile Radio," University of Kansas, KSA, Tech. Rep. ITTC-FY2008-TR-31620-06, Feb. 2008.

[32] T. M. Schmidl and D. C. Cox, "Robust Frequency and Timing Synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.

[33] M. Sliskovic, "Sampling Frequency Offset Estimation and Correction in OFDM Systems," in *Proc. ICECS '01*, Malta, Sep. 2–5 2001.

[34] *ETSI EN 300 744 V1.6.1 : Digital Video Broadcasting; Framing structure, channel coding and modulation for digital terrestrial television*, European Telecommunications Standards Institute (ETSI) Std., 2009.

[35] (2018) pureLiFi. [Online]. Available: https://purelifi.com/

[36] T. Yucek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” *IEEE Communications Surveys and Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[37] A. Ali and W. Hamouda, “Advances on spectrum sensing for cognitive radio networks: Theory and applications,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 2, pp. 1277–1304, 2016.

[38] R. McDonough and A. Whalen, *Detection of Signals in Noise*, 2nd ed. California, USA: Academic Press, 1995.

[39] M. Kosunen, V. Turunen *et al.*, “Survey and analysis of cyclostationary signal detector implementations on FPGA,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 4, pp. 541–551, 2013.

[40] D. Cabric, S. Mishra, and R. Brodersen, “Implementation issues in spectrum sensing for cognitive radios,” in *Proc. Thirty-Eighth Asilomar Conference on Signals, Systems and Computers ’04*, California, USA, Nov. 7–10, 2004.

[41] Z. Xuping and P. Jianguo, “Energy-detection based spectrum sensing for cognitive radio,” in *Proc. CCWMSN ’07*, Shanghai, China, Dec. 12–14, 2007.

[42] F. F. Digham, M. Alouini, and M. Simon, “On the energy detection of unknown signals over fading channels,” *IEEE Transactions on Communications*, vol. 55, no. 1, pp. 21–24, 2007.

[43] K. Kim, Y. Xin, and S. Rangarajan, “Energy detection based spectrum sensing for cognitive radio: An experimental study,” in *Proc. IEEE GLOBECOM ’10*, Miami,Florida, USA, Dec. 6–10, 2010.

[44] N. Armi, B. Chaeriah *et al.*, “Performance of energy detector with certain and uncertain noise in OFDM cognitive radio system,” in *Proc. Int. Electronics Symposium ’16*, Denpasar, Indonesia, Sep. 29–30, 2016.

[45] D. Bhargavi and C. Murthy, “Performance comparison of energy, matched-filter and cyclostationary-based spectrum sensing,” in *Proc. IEEE SPAWC ’10*, Marrakech, Morocco, Jun. 20–23, 2010.

[46] F. Salahdine, H. Ghazi *et al.*, “Matched filter detection with dynamic threshold for cognitive radio networks,” in *Proc. WINCOM ‘15*, Marrakech, Morocco, Oct. 20–23, 2015.

[47] A. Eduardo and R. Caballero, “Experimental evaluation of performance for spectrum sensing: Matched filter vs energy detector,” in *Proc. IEEE. COLCOM ‘15*, Popayan, Colombia, May 13–15, 2015.

[48] T. Bourgeois and Y. Sanada, “Cyclostationarity-based detector for ofdm signals with correlated pilot subcarriers,” in *Proc. Int. Symp. WPMC ‘11*, Brest, France, Oct. 3–7, 2011.

[49] S. Sohn, N. Han *et al.*, “Ofdm signal sensing method based on cyclostationary detection,” in *Proc. CROWNCOM ‘07*, Orlando, Florida, USA, Aug. 1–3, 2007.

[50] A. Ghasabeh, A. Tarighat, and B. Daneshrad, “Spectrum sensing of OFDM waveforms using embedded pilot subcarriers,” in *Proc. IEEE. Int. Conf. Communications‘10*, Cape Town, South Africa, May 23–27, 2010.

[51] V. Turunen, M. Kosunen *et al.*, “Implementation of cyclostationary feature detection for cognitive radios,” in *Proc. CROWNCOM ’09*, Hannover, Germany, Jun. 22–24, 2009.

[52] E. April, “The advantages of cyclic spectral analysis,” Defence research establishment Ottawa, Ottowa, Tech. Rep. Technical Note 92-4, Oct. 1991.

[53] B. Agee, S. Schell, and W. Gardner, “Spectral self-coherence restoral: A new approach to blind adaptive signal extraction using antenna arrays,” *Proceedings of the IEEE*, vol. 78, no. 4, pp. 753–767, 1990.

[54] W. Gardner *et al.*, *Cyclostationarity in Communications and Signal Processing*, 1st ed. New York: IEEE Press, 1994.

[55] W. Gardner, “Measurement of spectral correlation,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 5, pp. 1111–1123, 1986.

[56] W. Gardner, W. Brown *et al.*, “Spectral correlation of Modulated Signals: Part II - Digital Modulation,” *IEEE Transactions on Communications*, vol. 35, no. 6, pp. 595–601, 1987.

[57] W. Gardner, “Signal interception: a unifying theoretical framework for feature detection,” *IEEE Transactions on Communications*, vol. 36, no. 8, pp. 897–906, 1988.

[58] J. Lunden, S. A. Kassam, and V. Koivunen, "Robust Non-parametric cyclic correlation-based spectrum sensing for Cognitive Radio," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 38–52, 2010.

[59] L. Izzo, L. Paura, and M. Tanda, "Signal interception in non-Gaussian noise," *IEEE Transactions on Communications*, vol. 40, no. 6, pp. 1030–1037, 1992.

[60] K. Kim, I. A. Akbar *et al.*, "Cyclostationary approaches to signal detection and classification in Cognitive Radio," in *Proc. Int. Symp. New Frontiers in Dynamic Spectrum Access Networks '09*, Dublin, Ireland, Apr. 17–20, 2007.

[61] Q. Yuan, P. Tao *et al.*, "Cyclostationarity-based spectrum sensing for wideband Cognitive Radio," in *Proc. Int. Conf. Communications and Mobile Computing '09*, Yunnan, China, Jan. 6–8, 2009.

[62] J. Lee, J. Yoon, and J. Kim, "A new spectral correlation approach to Spectrum Sensing for 802.22 WRAN system," in *Proc. IPC. '07*, Jeju City, South Korea, Oct. 11–13, 2007.

[63] A. Al-Dulaimi, N. Radhi, and H. Al-Raweshidy, "Cyclostationary detection of undefined secondary users," in *Proc. Int. Conf. Next Generation Mobile Applications, Services and Technologies '09*, Cardiff, Wales, Sep. 15–18, 2009.

[64] A. M. Mossaa and V. Jeoti, "Cognitive Radio: Cyclostationarity-based classification approach for analog TV and wireless microphone signals," in *Proc. Innovative Technologies in Intelligent Systems and Industrial Applications*, Monash, Malaysia, Jul. 25–26, 2009.

[65] S. Sohn, N. Han *et al.*, "OFDM signal sensing method based on cyclostationary detection," in *Proc. Int. Conf Cognitive Radio Oriented Wireless Networks and Communications*, Orlando,Florida, USA, Aug. 1–3, 2007.

[66] S. Da, G. Xiaoying *et al.*, "Significant cycle frequency based feature detection for cognitive radio systems," in *Proc. Int. Conf. Cognitive Radio Oriented Wireless Networks and Communications*, Hannover, Germany, Jun. 22–24, 2009.

[67] A. A. Thomas and T. Sudha, "Primary user signal detection in cognitive radio networks using cyclostationary feature analysis," in *Proc. NCCSN '14*, Palakkad, India, Oct. 10–12, 2014.

[68] A. Tkachenko, D. Cabric, and R. Brodersen, “Cyclostationary feature detector experiments using reconfigurable BEE2,” in *Proc. Int. Symp. New Frontiers in Dynamic Spectrum Access Networks*, Dublin, Ireland, Apr. 17–20 2007.

[69] A. Dandawate and G. Giannakis, “Statistical tests for presence of cyclostationarity,” *IEEE Transactions on Signal Processing*, vol. 42, no. 9, pp. 2355–2369, 1994.

[70] J. Lunden, V. Koivunen *et al.*, “Censoring for collaborative spectrum sensing in Cognitive Radios,” in *Proc. Conf. Forty-First Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 4–7 2007.

[71] Y. Qi, W. Wang *et al.*, “Spectrum sensing combining time and frequency domain in multipath fading channels,” in *Proc. Int. Conf. Communications and Networking in China*, Hangzhou, China, Aug. 25–27 2008.

[72] J. Lunden, V. Koivunen *et al.*, “Spectrum sensing in Cognitive Radios based on multiple cyclic frequencies,” in *Proc. Int. Conf. Cognitive Radio Oriented Wireless Networks*, Orlando, Florida, USA, Aug. 1–3 2007.

[73] K. Muraoka, M. Ariyoshi, and T. Fujii, “A novel spectrum-sensing method based on maximum cyclic autocorrelation selection for Cognitive Radio system,” in *Proc. IEEE. Symp. New Frontiers in Dynamic Spectrum Access Networks*, Chicago, Illinois, USA, Oct. 14–17 2008.

[74] J. Lunden, V. Koivunen *et al.*, “Collaborative cyclostationary spectrum sensing for Cognitive Radio systems,” *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4182–4195, 2009.

[75] V. Turunen, M. Kosunen *et al.*, “Spectrum estimator and cyclostationary detector for cognitive radio,” in *Proc. European Conference on Circuit Theory and Design*, Antalya, Turkey, Aug. 23–27 2009.

[76] ——, “Implementation of cyclostationary feature detector for Cognitive Radios,” in *Proc. Int. Conf. Cognitive Radio Oriented Wireless Networks and Communications*, Hannover, Germany, Jun. 22–24 2009.

[77] V. Sebesta, R. Marsalek, and Z. Fedra, “OFDM signal detector based on cyclic autocorrelation function and its properties,” *Radio Engineering*, vol. 20, no. 4, pp. 926–931, 2011.

[78] F. Paisana, N. Prasad *et al.*, “An alternative implementation of a cyclostationary detector,” in *Proc. Int. Symp. on Wireless Personal Multimedia Communications*, Taipei, Taiwan, Sep. 24–27 2012.

[79] G. Huang and J. Tugnait, “On cyclic autocorrelation based spectrum sensing for Cognitive Radio systems in Gaussian noise,” in *Proc. Int. Annual Allerton conference on Communication, Control and Computing*, Monticello, Illinois, USA, Sep. 28–30 2011.

[80] ——, “On cyclostationarity based spectrum sensing under uncertain Gaussian noise,” *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2042–2054, 2013.

[81] J. Tugnait and G. Huang, “Cyclic autocorrelation based spectrum sensing in colored Gaussian noise,” in *Proc. IEEE WCNC ’12*, Shanghai, China, Apr. 1–4 2012.

[82] A. Tani, R. Fantacci, and D. Marabissi, “A low-complexity cyclostationary spectrum sensing for interference avoidance in Femtocell LTE-A-Based networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2747–2753, 2016.

[83] J. Lunden, S. Kassam, and V. Koivunen, “Nonparametric cyclic correlation based detection for Cognitive Radio systems,” in *Proc. CROWNCOM ’08*, Singapore, May 15–17 2008.

[84] J. Lunden and V. Koivunen, “Spatial Sign and Rank Cyclic Detectors,” *IEEE Signal Processing Letters*, vol. 21, no. 5, pp. 595–599, 2014.

[85] J. Renard, L. Lampe, and F. Horlin, “Spatial sign cyclic-feature detection,” *IEEE Transactions on Signal Processing*, vol. 61, no. 18, pp. 4521–4531, 2013.

[86] V. Turunen, M. Kosunen *et al.*, “Correlation-based detection of OFDM signals in the angular domain,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 951–958, 2012.

[87] M. Tohyama, *Sound in the time domain*, 1st ed. NY, USA: Springer, 2017.

[88] J. Volder, “The Birth of CORDIC,” *Journal of VLSI Signal Processing*, vol. 25, pp. 101–105, 2000.

[89] S. V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction*, 2nd ed. NJ, USA: John Wiley and Sones Ltd, 2000.

[90] (2018) Stable distribution. [Online]. Available: https://uk.mathworks.com/help/stats/stable-distribution.html

[91] MathWorks. Matlab Coder. [Online]. Available: www.mathworks.com/products/matlab-coder.html

[92] Xilinx. Vivado Design Suite - HLx Editions. [Online]. Available: www.xilinx.com/products/design-tools/vivado.html

[93] J. E. Volder, "The cordic Trigonometric Computing Technique," *IRE Transactions on Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, 1959.