University of Southern Queensland

Faculty of Health, Engineering & Sciences

Cognitive Radio for Improved NBN Access in Regional Areas

A dissertation submitted by

B. Hardless

in fulfilment of the requirements of

ENG4112 Research Project

towards the degree of

Bachelor of Engineering (Honours), Electrical & Electronic Engineering

Submitted: October, 2021

Abstract

Cognitive Radio (CR) has emerged as a potential solution to spectrum scarcity, whereby time division of spectrum use could allow Secondary Users (SU) to leverage bands when unoccupied by a Primary User (PU). This requires accurate detection of PU signal presence at low Signal to Noise Ratio (SNR).

Eigenvalue-based detection schemes have the potential to detect low SNR signals with periodicity. All schemes reviewed in the literature use Random Matrix Theory and combinations of eigenvalues to develop a threshold for Boolean detection. This project extends early work in Maximum-Minimum Eigenvalue (MME) detection by using eigenvalues as a proxy for SNR, examines the factors affecting it, and develops a framework for optimised detection. Novel Maximum-Minimum Percentage Difference (MMPD) detection is applied to simulated and real SDR signals to examine whether the SNR proxy hypothesis is valid under a range of conditions. Simulations were used to develop a mathematical model for the MMPD scheme. It is refined and tested against generated and real signals, and pre-processing requirements for received signals are defined. An alternative detection scheme called Eigenvector Augmented Fast Fourier Transform (EVA-FFT) is also developed based on Principal Component Analysis (PCA) and tested extensively.

This project defines the relationship between the eigenvalues of the sample covariance matrix and SNR, as well as how it changes with antenna count and listening time. Low SNR limits are proposed for eigenvalue based detection schemes based on antenna count and cycles captured. Neither MMPD or EVA-FFT employ energy detection as most other eigenvalue/eigenvector based schemes do.

University of Southern Queensland

Faculty of Health, Engineering & Sciences

ENG4111/2 *Research Project*

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering & Sciences or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled "Research Project" is to contribute to the overall education within the student's chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Dean

Faculty of Health, Engineering & Sciences

Certification of Dissertation

I certify that the ideas, designs and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.
I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

B. Hardless

Acknowledgments

I would first like to thank the many staff at the University of Southern Queensland, who have given their time and energy to share their hard won knowledge, skills and experience with me. In particular I give thanks to my project supervisor, Dr John Leis, whose infectious enthusiasm, tremendous support and guidance spurred me forward in this endeavour.

To my three brothers and dear friends, thank you for your company and counsel over these past four years. I am truly fortunate to have you in my life.

Most of all I owe thanks to my parents, Michael and Lorna Hardless, who valued education and sacrificed much to provide me with opportunities they never had. They instilled in me a love of learning and the self-belief to strive toward my potential. I am forever grateful, and aspire to be such a fine example for my own son, Oliver, who fills me with joy and purpose. As I write this I smile gladly, for I know there is ever more to learn, ever a problem to solve, and that we might try together.

B. Hardless

Contents

Abstract **i**

Acknowledgments **vii**

List of Figures **xiii**

List of Tables **xvii**

Chapter 1 Literature Review **1**

1.1 Broadband in Australia 1

1.1.1 National Broadband Network (NBN) Background 1

1.1.2 Urban Strategy 2

1.1.3 Urban Fringe and Regional Strategy 2

1.2 Software Defined Radio and Cognitive Radio 3

1.2.1 Introduction 3

1.2.2 Spectrum Availability and Sensing Generally 5

1.2.3 Dynamic Reconfiguration 8

1.2.4 Dynamic Spectrum Access 10

1.2.5 Eigenvalue Based Spectrum Sensing 13

1.3 Conclusions 18

Chapter 2 Methodology **21**

2.1 Data Collection 21

2.2 Data Analysis 22

2.3 Tools and Materials 23

Chapter 3 Eigenvalues and Signal to Noise Ratio **25**

3.1 Max-Min Percentage Difference (MMPD) Eigenvalue Detection 25

3.2 Optimal MMPD 32

3.3 FFT Detection Effectiveness 34

3.3.1 Potential For Channel State Detection 34

3.3.2 Eigenvector Augmented Fast Fourier Transform (EVA-FFT) . . . 39

Chapter 4 Hardware Experiments **42**

4.1 Initial Data Capture and Analysis 42

4.1.1 Single SDR Receiver - FM Signals 42

4.2 MMPD Applied to FM Signal Data 48

4.2.1 Quadrupled Single Receiver Signal 48

4.2.2 Four Receiver Signals 54

4.3 EVA-FFT Applied to Multiple Receivers 55

4.3.1 Correlated Noise Vulnerability 60

4.3.2 EVA-FFT Applied to FM Signals 62

4.3.3 EVA-FFT Applied to 802.11n WiFi 66

Chapter 5 Conclusions **70**

5.1 General Summary 70

5.2 Performance of Detection Schemes 72

5.3 Questions Revisited and Further Research 74

References **77**

Appendix A Project Specification **83**

Appendix B Risk Assessment **87**

Appendix C Ethical Clearance **93**

Appendix D Code Listings **95**

D.1 Code Listing 1 96

D.2 Code Listing 2 98

D.3 Code Listing 3 99

D.4 Code Listing 4 104

D.5 Code Listing 5 110

D.6 Code Listing 6 115

D.7 Code Listing 7 116

List of Figures

2.1 Generalised testing and data collection process flow. 22

3.1 Eigenvalue % difference vs SNR for 4 antennas - 1500 cycles captured. . . 27

3.2 Logistic growth model for 4 antennas - 1500 cycles captured. 28

3.3 SNR cutoff vs number of cycles for varying number of antennas. 29

3.4 Probability of detection P_d at different SNR with varying inferred SNR threshold. 31

3.5 Probability of false alarm P_{fa} at different SNR with varying inferred SNR threshold. 31

3.6 Probability of detection P_d vs SNR for MMPD and MME detection. . . . 33

3.7 Effect of MMPD and PCA on FFT. 36

3.8 Eigenvector Augmented Fast Fourier Transform test method. 39

3.9 FFT for uncorrelated noise. 40

4.1 Single RTL-SDR for data capture. 43

4.2 SDR# Settings and spectrum for data capture. 44

4.3 Recovered intelligence from 100.7MHz FM radio station in Toowoomba. . 45

4.4 Baseband spectrum for broadcast FM (Murray n.d.). 46

4.5 IQ signal MMPD pre-processing for a single receiver. 48

4.6 Probability of detection P_d when uncorrelated noise is band-pass filtered before MMPD. 51

4.7 Probability of detection P_d when uncorrelated noise is not band-pass filtered before MMPD. 52

4.8 Probability of detection P_d vs SNR for MMPD and MME detection - 1 RX. 53

4.9 Probability of detection P_d vs SNR for MMPD and MME detection (FM 100.7 MHz) - 4 RX. 56

4.10 hackRF RF transceiver. 57

4.11 Detection for uncorrelated noise capture from 4 receivers. 59

4.12 Detection for correlated noise without pre-filtering. MMPD and MME falsely detected active signals. 60

4.13 Detection for correlated noise with pre-filtering. 61

4.14 Probability of detection P_d vs SNR for EVA-FFT (FM 100.7 MHz i) - 4 RX. 62

4.15 Probability of detection P_d vs SNR for EVA-FFT (FM 100.7 MHz i) - 4 RX. 63

4.16 Probability of detection P_d vs SNR for EVA-FFT (FM 100.7 MHz ii) - 4 RX. 63

4.17 Probability of detection P_d for different RX performance (FM 100.7 MHz i) - 4 RX. 64

4.18 Probability of detection P_d for perfectly correlated data - 1 RX x4. 65

4.19 Probability of detection P_d vs SNR for EVA-FFT (WiFi 2.42 GHz i) - 4 RX. 66

4.20 Probability of detection P_d vs SNR for EVA-FFT (WiFi 2.42 GHz ii) - 4 RX. 67

4.21 Probability of detection P_d vs SNR for EVA-FFT (WiFi 2.44 GHz) - 4 RX. 67

4.22 Unexpected spectral peak in WiFi data. 68

4.23 SDR# spectrum and waterfall for WiFi 2.42 GHz - RX1. 69

5.1 MMPD concept diagram showing the relationship between eigenvalue percentage difference, SNR and received signal correlation. 71

List of Tables

3.1 Parameters by number of antennas - 2000 cycles captured. 30

3.2 Multi-channel FFT effectiveness without PCA and MMPD. 37

3.3 Multi-channel FFT effectiveness using PCA and MMPD. 38

Chapter 1

Literature Review

1.1 Broadband in Australia

1.1.1 National Broadband Network (NBN) Background

The NBN in its current form was first proposed in 2009 by the Rudd Government as a faster alternative to existing copper infrastructure in Australia. The intent was to service residential, non-residential and non-premises end users in all parts of the country (Communications Alliance n.d.) using a combination of optical Fibre To The Premises (FTTP), fixed wireless and satellite communications. It was anticipated that FTTP internet speeds of up to 100 megabits per second would be available to 90% of Australian premises, with the remainder gaining access to speeds up to 12 megabits per second (Rudd 2009). At the time the NBN was the largest infrastructure project in Australian history (Egan 2008), with an initial estimated cost of $43B spanning eight years. Although the government sought private sector investment as part of the forecast project costs, there was a clear intent to implement ownership restrictions in order to promote an open-access network and encourage competition between telecom retailers. The Rudd government billed the NBN as productivity enabling infrastructure and an important driver of economic growth (Rudd 2009).

NBN Co Limited was subsequently established in April 2009 (Australian Business Register 2021) to provide a primarily hard-wired FTTP technology supplemented by fixed wireless and satellite technologies. The regulatory framework for the NBN is established under

the *National Broadband Network Companies Act 2011* (Cwlth) and *Telecommunications Legislation Amendment (National Broadband Network Measures - Access Arrangments) Act 2011* (Cwlth).

When Prime Minister Rudd left office in 2013 the NBN project was failing to meet targets and was significantly behind schedule, with only 154,000 people able to access the service. Since that time the NBN implementation has become a mix of FTTP and Fibre To The Node (FTTN) due to differing strategies between the Australian Liberal and Labor parties (Conifer 2016). FTTN installations connect fibre optic lines to node points, which are then connected to premises via copper lines, resulting in lower maximum speeds than FTTP (Australian Competition & Consumer Commission n.d.). The net cost for the project has also exceeded original estimates, reportedly reaching $57B in 2020 when the 'roll-out' phase was completed (Conifer 2020).

1.1.2 Urban Strategy

Premises in urban environments are equipped with either FTTP, FTTN, Hybrid Fibre Coaxial (HFC), Fibre To The Basement (FTTB) or Fibre To The Curb (FTTC). HFC involves using existing pay TV cabling to connect the fibre node to the premises (NBN Co Limited 2020*a*). FTTB and FTTC are similar to FTTN, in that the fibre terminates at a node and requires a copper line to the premises (Australian Competition & Consumer Commission n.d.*a*). FTTP is the fastest connection type in terms of data transfer rates for customers because the optical fibre transmission lines use light rather than electric pulses to transmit information, permitting higher propagation velocity without attenuation losses due to distributed electrical parameters.

The maximum data download and upload rates using FTTP technology via the NBN are presently 1000Mbps and 50Mbps respectively (NBN Co Limited 2020*b*).

1.1.3 Urban Fringe and Regional Strategy

Urban fringe areas are on the periphery of population centres and represent the 'middle ground' between distributed regional and high density urban populations. To service these areas NBN Co Limited uses a mix of FTTP/N/B/C, HFC as well as fixed wire-

less and satellite wireless. Fixed wireless installations consist of radio frequency (RF) transceiver installations that provide line of sight data transfer between customers and the NBN wide area network (WAN) using Time Division Duplex Long Term Evolution (TDD-LTE) protocols to minimise bandwidth requirements compared to the frequency division alternative (FDD-LTE) (Australian Competition & Consumer Commission n.d.*b*). TDD-LTE dedicates the majority of time in any sub-channel to downlink communications (download), while less time as allocated to uplink (upload) since users are typically consuming media hosted elsewhere. Satellite installations require less infrastructure than fixed wireless, since base stations are not required in the targeted regional locality. For the purposes of this project, investigations are targeted toward NBN fixed wireless, with a view to maximise bandwidth utility. As described subsequently in Section 1.2, time-sharing of bandwidth has the potential to improve spectral efficiency and maximise data rates in regional areas. This requires accurate detection of when bands are occupied. The specific focus of this project, therefore, is detection schemes that enable such time-sharing of the RF spectrum.

1.2 Software Defined Radio and Cognitive Radio

1.2.1 Introduction

The major premise giving rise to interest in and demand for Cognitive Radio (CR) is the scarcity of suitable and available RF spectrum for communications between wireless devices to support information transmission at high data rates. In the cellular specific context, over the last 15 years there has been drastic growth in mobile traffic and wireless devices, exacerbated due to the proliferation of smart devices. This presents a challenge in terms of capacity for existing cellular infrastructure (Yu, Xue, Bennis, Chen & Han 2018).

The capacity problem is that information transfer is demanded at very high volume flow rates which requires spectral use, while the available RF spectrum is insufficient to satisfy the anticipated future demand in the context of the prevailing regulatory approach by governments. The conventional regulatory approach of fixed spectrum assignment via licensing is limited by the finite RF bands suitable for particular wireless applications. Identifying new spectrum for use is considered “almost impractical” as a means to solve the capacity problem (Gupta & Dhurandher 2020). The only option, therefore, is to

make more efficient use of the existing spectra through a system of sharing (Chen, Chen & Meng 2014). CR seeks to exploit the fact that primary users (PU) holding licences for RF bands do not generally occupy those bands constantly, nor over the entirety of the regulated area to which the licence applies. The realisation of temporal and spatial diversity in the regulated RF environment permits the conception of the CR, which takes advantage of wide-band spectral sensing to detect unoccupied frequencies and use them for communications, without impacting other users (Wen, Tang & Ziolkowski 2019). This is known as Dynamic Spectrum Access (DSA) and is based on context-aware behaviour, underpinned by autonomous reconfiguration driven by learnings from sampling the spectral environment (Lopez-Benitez & Casadevall 2013).

Software Defined Radio (SDR) expands on conventional radio systems by using software to set transceiver characteristics and process signals, rather than using dedicated hardware for the purpose (Fisne & Ozsoy 2018). An SDR is a wireless device whose characteristics are modified using a computer terminal or embedded system (Regula, Gilbert & Sheikh 2020). This makes SDRs a flexible wireless solution for any enterprise that requires a configurable and/or bespoke network, and as such SDR is substantially used in military applications (Jondral, Elsner & Schwall 2012). For example, unmanned vehicles employ SDRs to enhance battlespace communications and extend effective range for ground vehicles where line of sight is impeded. This is accomplished by using unmanned aerial vehicles as a relay for re-transmission (Webb, Schwall, Elsner & Jondral 2012).

A CR is an SDR which senses its radio frequency (RF) environment, examines changes, and uses this information to dynamically reconfigure. Thus, SDR is an enabling and prerequisite technology for CR (Jondral et al. 2012). It may be said that all CR are SDR, but not vice versa. In contrast to conventional wireless transceivers that can only make use of preallocated bands, CR are able to utilise any available frequency within their bandwidth according to inferences about their environment, which has been described as an OODA (Observe, Orient, Decide, Act) loop (Zhu & Pan 2020). The flexibility of CR technology and techniques are foreseeably extensible, since multiple operating parameters such as frequency, transmission power and modulation technique can potentially be manipulated in real time and different combinations (Gupta & Dhurandher 2020). The fundamental elements of a CR (Fernando 2019) are:

- spectrum sensing - gathering real time spectral information;

- spectrum allocation - optimal channel assignment;
- spectrum access - transmit on the allocated channels at appropriate power levels;
- spectrum sharing - coordinate spectral use with other SUs and PUs; and
- spectrum mobility - detect PU channel reclamation, release the channel(s) and adopt an alternative channel or strategy.

This research project focusses primarily on spectrum sensing and investigates the different techniques for retrieving man-made signals that are buried in significant noise.

1.2.2 Spectrum Availability and Sensing Generally

Spectrum availability can be considered in terms of occupied and unoccupied frequency bands, with *availability* being a state determined by the CR that a particular band corresponding to one or more channels is not in use. Bands unused by the licensed PU at a specific time and geographic location are termed *spectrum holes* (Gupta & Dhurandher 2020). It is thought that spectral usage varies significantly with time and geographic location ranging from 15% to 85% (Guo 2010). For a CR to operate effectively it must be able to sense the PU occupied bands of the RF spectrum and dynamically access spectrum holes for opportunistic communications (Almalfouh & Stuber 2011). This must be done in a manner that is autonomously adaptive and does not affect PU communications, while seeking to maintain sufficient quality of service (QoS) to satisfy user demand (Regula et al. 2020). CR are thought to eventually provide meaningful utility across a wide range of frequencies e.g. tens of MHz to approximately 10 GHz, though efforts prior to 2010 substantially sought to leverage the bands below 1 GHz which are primarily used for television (Razavi 2010). Spectrum sensing requires the *reliable* detection of PU signals indicative of channel occupation *in real time* (Gupta & Dhurandher 2020), a problem statement which acknowledges the challenges of temporal and spatial diversity, as well as the need for the system to be adequately responsive such that vacant channels can be exploited before they are reclaimed by the PU.

There are three main modes of spectrum sensing described in the literature (Chen et al. 2014):

1. Overlay - The CR has knowledge of the PU RF codebook and is thus able to avoid

interference. This is generally difficult to attain since it would require formal agreements and potential sharing of intellectual property between the PU and SUs. If available, CR nodes can examine sensor information for signal properties or patterns indicative of PU channel access. This is known as *feature detection.*

2. Interweave - The CR network employs spectrum sensing to identify and exploit spectrum holes, accessing channels not claimed by the PU for its own communications.

3. Underlay - The CR network adaptively varies its operating parameters such as transmit power, direction and sub-carrier allocation in Orthogonally Frequency Division Multiplexed (OFDM) systems. This type of scheme tries to use channels by remaining below a pre-set *interference temperature* threshold, such that the CR network transmissions appear as tolerable noise to the PU.

It is conceivable that a complex CR network could switch between these spectrum sensing strategies depending on available information, however, such functional complexity in pursuit of optimal spectral utility drives hardware costs and computational overhead.

A typical approach to spectrum sensing is done in two broad stages. First the receiver is isolated from the receive antenna and the receiver's internal noise power is measured as the signal $P_S(t) = P_n(t)$. The RF environment is subsequently measured with the antenna connected and this results in any occupying signal power, multiplied by the channel gain, plus the receiver noise power $P_S(t) = h(t)P_{PU}(t) + P_n(t)$. The test for channel occupation becomes a question as to whether the difference between these two measurements represents a signal power indicative of an occupied band. Since it is not only the presence of energy at particular frequencies that is important, but how this varies with time, energy detection methods require a long sensing time. Feature detection is a different method that is not reliant on knowledge of the receiver noise power, but instead looks for known patterns in received waveforms that correspond to PU modulation scheme(s). This method has a comparable sensing time to the energy detection method, with the added disadvantage of requiring synchronisation between the receiver's baseband Analog to Digital Converter (ADC) clock and the received signal symbol rate, which is difficult to achieve (Razavi 2010).

Spectrum sensing is a challenging part of CR network design because propagation factors like multi-path fading and shadowing can significantly affect spectral component detection

at SU CR nodes. This could cause an occupied band to be improperly sensed as vacant or vice versa, causing either undue interference to PU communications or inefficient SU spectral utility respectively (Almalfouh & Stuber 2011). One may say, then, that the first challenge for a CR is to accurately sense the RF environment and ascribe meaning to the power densities detected. It follows that the use of multiple sensor nodes in a cooperative fashion to achieve spatial diversity in spectrum sensing will improve the accuracy of the inferences drawn, and offset the adverse effects of fading and shadowing (Chen et al. 2014). A consequence of shadowing and fading effects is that CRs require a highly sensitive RF front end, far exceeding those sensitivities specified in wireless standards. The difference between a genuine PU signal and noise at a CR node may be very small and difficult to resolve, emphasising the need for stable noise power in the CR receiver that does not drift (Razavi 2010). For example, a CR may determine channel availability based on an SNR threshold of -15 dB to -20 dB even though standards specify +8 dB to +25 dB depending on the modulation scheme in use (Razavi 2010).

Various cooperative spectrum sensing strategies are described in the literature which use multiple SU nodes to try to account for spatial and temporal diversity, as well as unknown PU antenna gain characteristics. The use of multiple channels per CR node adds to the complexity of spectrum sensing, and so too does the fact that most transceivers are not full duplex since they cannot maintain continuity in sensing while the CR is transmitting (Fernando 2019). The algorithms employed in the literature to enhance the reliability of sensor data are almost exclusively stochastic; deterministic methods would entail the processing of enormous data sets for multiple variables across vast areas including topography, soil type, vegetation and human structures to name just a few (Almalfouh & Stuber 2011).

Different techniques have been employed to monitor the RF environment including RADAR imaging and electromagnetic spectrum sensing. Different methods entail different hardware requirements which affects the size, power consumption and weight of CR installations. A typical CR transceiver contains an RF front end, Analog to Digital Converter (ADC), Digital Signal Processor (DSP) and Digital to Analog Converter (DAC) supplying RF output stages. Depending on the implementation, an array of frequency converters and narrow band filters might be used to cover the desired bandwidth. Tunable RF components might be used to make the system more flexible. These variations are usually seen in the analog RF front end of CR receivers after the antennas and initial amplifica-

tion stages (Zhu & Pan 2020). The speed and resolution demands on the digital hardware can vary significantly depending on the algorithms employed. For example, one suggested scheme requiring digital downconversion of a block of channels and then taking the Fast Fourier Transform (FFT) of the entire block would require high ADC speed and resolution compared to simpler energy detection methods (Razavi 2010).

Energy detection methods are quite straightforward, comparing received signal strength against a threshold to make a decision as to channel states. Drawing correct conclusions in such a scheme relies primarily on spatial diversity of nodes as previously mentioned, however, the accuracy of sensing can be reduced due to low SNR regions. An alternative in an overlay scenario is to employ feature or coherence detection using matched filters, though the requisite PU information may not be available to the SU, thus excluding this method. The use of cyclic autocorrelation is another alternative in a scheme known as cyclostationary detection, where the convolution of adjacent time series of signal data will tend to reject noise and reveal the presence of PU signals in a band. An FFT can then be constrained to lag windows flagged in the autocorrelation, thus reducing the computational burden for SU nodes. Although an autocorrelation approach can be optimised to some extent, the requirement to correlate at all lags for the sampling window introduces significant computation delay compared to energy detection (Chen et al. 2014).

Another scheme for networks with multiple antennas allows spectrum occupation to be detected without any prior knowledge of PU signals, by comparison of maximum and minimum eigenvalues of a received signal matrix (Chen et al. 2014). Since a multi-dimensional matrix can be representative of a linear transformation of space, most vectors subjected to the transform will end up moving off their initial axis. An eigenvector is a vector which remains on the same axis when the transform matrix is applied, and is scaled by an associated eigenvalue. Eigenvalue and eigenvector based detection methods are elaborated in Section 3, since they form the primary basis for this research project.

1.2.3 Dynamic Reconfiguration

Dynamic reconfiguration describes the ability of a CR to take advantage of different frequency bands by changing its operating parameters or hardware configuration in real time. Reconfigurable antennas, active devices and Multiple Input Multiple Output (MIMO) systems are all examples of hardware that facilitate dynamic reconfiguration to support the

flexible and wide-band operations of CR (Guo 2010). Reconfigurable antennas permit sensing and transmission for a larger variety of frequency bands, and studies have been conducted on various linearly polarised antenna variants and a circularly polarised variant (Wen et al. 2019). Reconfigurable active devices such as filters have the potential to reduce the hardware and power requirements for CR nodes. MIMO systems allow CR installations to take advantage of spatial diversity by enabling beamforming. This has the potential to improve spectrum sensing by taking advantage of multipath effects, as well as to avoid interference with PUs by controlling the direction of transmitted power (Chen et al. 2014).

Beamforming strategies have been investigated when PU Channel State Information (CSI) is available to the CR network in real time via a dedicated channel. When the CR network has the PU CSI, it can generate beamforming strategies between the CR base station and nodes, such that power is directed in a way that minimises the effect on the PU. This also allows the network to be more flexible and maintain a useable service level for CR nodes, while relieving the requirement for spectrum sensing at those nodes (Chen et al. 2014). Though this approach may sound attractive, PU CSI is not practically available in the manner described and there is no obvious incentive for a PU to share that information in the regulatory environment of exclusive licensing for RF bands.

The criteria for reconfiguration are based on switching criteria in a CR, using a scheme called Adaptive Modulation and Coding (AMC). AMC is an umbrella term for the influence of Medium Access Control (MAC) and physical layer parameters on SDR reconfiguration. Optimising this influence can yield performance improvements in SDRs generally and extends to their use in CR networks (Jondral et al. 2012). Reconfiguration during runtime with minimal disruption to communications or data processing is an obvious challenge. Block abstraction of signal processing chains is not unusual, with different waveform processing schemes selected by switch action. Dickens et al. (2012) describe a method of streamlining dynamic reconfiguration using dynamic waveform insertion and removal to minimise disruption to signal processing. The scheme is based on a block abstraction represented in a Graphical User Interface (GUI), and involves the concept of a *surfer* which is a resource control algorithm that queues blocks for processing by each thread. The surfer is able to improve block processing efficiency since normally each block as assigned a single thread or a single thread is used for all processing. The surfer concept leverages shared memory between different physical processors and is effectively

a sophisticated buffering scheme for process execution (Dickens, Laneman & Dunn 2012).

Some SDR devices have Digital Clock Managers (DCMs) which allow variation in operating clock frequency during runtime by the manipulation of Dynamically Reconfigurable Ports (DRPs) (Jondral et al. 2012). This is in keeping with the need to maximise hardware utility in a CR by making the existing hardware as flexible as possible, such that it can be used over a wide range of frequencies and adapt as required to its environment. Another scheme known as Partial Recognition (PR) uses time sharing for part of a Field Programmable Gate Array (FPGA) while the rest of the FPGA operates uninterrupted. This was combined with the concept of DRPs to significantly reduce hardware utilisation in SDR architecture compared to static FPGA design (He, Crockett & Stewart 2012).

Microwave photonics is an area of study that modulates optical signals with radio frequency signals using electro-optical modulators. Since optical signals are immune to electromagnetic interference and have a much shorter wavelength, there are advantages over RF solutions in terms of size, weight, propagation loss and interference. Hardware can operate with a wide bandwidth of tens or even hundreds of GHz. This presents a challenge in terms of dynamic reconfiguration because a CR needs to vary its operating frequency in a highly flexible manner, which in turn requires variable signal generation. Thus, if a CR network is to leverage microwave photonics in its architecture, the high frequencies are a limiting factor for electrical waveform generation via a conventional DAC. A solution has been proposed employing multiple lasers in a feedback loop, where a master laser is modulated using an electro-optical modulator until the output frequency settles at the desired value. Electrical conversion is then achieved by way of a photodetector. Receivers based on microwave photonics have adaptations of the superheterodyne principle to achieve flexibility in information recovery, where the local oscillator takes the form of a reference light source, and photodetectors act in the place of mixers (Zhu & Pan 2020).

1.2.4 Dynamic Spectrum Access

Once a CR network has knowledge of the RF environment through spectrum sensing, it must then determine how to utilise the parts of the spectrum deemed available (spectrum holes), and how it will behave as the environment changes. This system of adaptively using the available spectrum is known as Dynamic Spectrum Access (DSA) and relies on

spectrum sensing information and dynamic reconfiguration capabilities as prerequisites. One could say DSA is the set of protocols governing how the CR network assigns channels to nodes, based on its knowledge of spectrum holes (Zhu & Pan 2020). Provided the spectrum sensing information is accurate, improving DSA algorithms is an obvious way to enhance spectrum utilisation efficiency (Jondral et al. 2012, Gupta & Dhurandher 2020) by ascribing meaning to the sensed information and then deciding how the CR network should optimally access any available channels. This alludes to a complicating issue, being that DSA algorithms make judgements about the spectrum sensing information, since in reality there are inaccuracies. This is generally based on the application of weighting systems and/or the application of probability functions to assign meaning to the aggregated sensor data (Almalfouh & Stuber 2011). A significant barrier to DSA exists for portable consumer devices, where robust spectrum sensing may not be a viable option given size, weight and power constraints (Webb et al. 2012), exacerbated by the inherent wide-band nature of DSA (Fisne & Ozsoy 2018).

Opportunistic access of PU licensed spectrum by SUs during times of inactivity describes a hierarchical access scheme with at least two levels. The different types of opportunistic spectrum access were defined at Section 1.2.2 as overlay, interweave or underlay. In any of these schemes, DSA can produce interference for PUs when the spectrum is imperfectly sensed, interpreted, or out of band emissions occur. Out of band emissions are when part of a CR node's radiated power in a vacant channel/sub-band leaks into neighbouring ones occupied by the PU, generally as a result of pulse shaping at the transmitter (Almalfouh & Stuber 2011). A particular underlay CR network implementation has been simulated with the PU providing CSI via an RF feedback link, giving obvious performance advantages over conventional spectrum sensing due to accurate and timely CSI informing DSA strategy. Variations of this basic idea have been explored with differences in the manner of CSI feedback from the PU, or the number of PU nodes, though the major premise of direct contact between PU and the CR network is essentially unchanged (Chen et al. 2014, He & Dey 2011, He & Dey 2012).

A significant part of any DSA strategy for a CR network is what to do when a PU reclaims a channel. This is known as a *hand off* strategy and the common methods employed are as follows (Gupta & Dhurandher 2020):

1. Reactive-decision spectrum hand off: Searching for a target channel only begins after

interruption of channel use by a PU. Using this method an SU node must tolerate the delay between interruption and transmission resumption on an alternative channel. If there aren't any channels available then the SU node will need to wait in a queue.

2. Proactive-decision spectrum hand off: The SU node relies on a pre-prepared list of target channels for handoff. This can be done by regularly observing channels and populating a hand off list based on their usage history, or geolocation databases can be looked up. This helps to reduce sensing time and delays to service. A challenge associated with this method is that it cannot be known whether a PU will have occupied a handoff channel after it was deemed suitable and available, but before interruption of the SU. Depending on the implementation, an SU node may perform channel switching pre-emptively if it predicts PU interruption is imminent. The scheme might perform poorly if the predictions are wrong.

3. Non-switching hand off: An SU node prefers staying on the same channel rather than switching. If the PU reclaims a channel, the SU node will wait in the queue until the PU stops transmitting.

In multiple node CR networks the matter of dynamic access is closely tied to the problems of cooperative spectrum sensing and spectrum sharing, being how to adequately sense the spectrum for a spatially distributed set of nodes and how to optimally issue channel assignments to them. Cooperative communications techniques for CR networks including feedback of SU sensor information have been explored as a means to optimise decision making regarding spectrum access (Letaief & Zhang 2009, Chen et al. 2014). A clustering approach to cooperative sensing has been explored too, with the aim being to obtain sensing diversity via the spatial distribution of node clusters. A particular node in each cluster would be assigned as the 'cluster head' if it had the highest level energy detection residual energy, communicating cluster sensing information to a central 'fusion centre' through a common control channel (Sun, Zhang & Letaief 2007). Clustering approaches have also been modelled as a congestion game, analysing cluster formation to arrive at an optimal result by convergence toward a Nash equilibrium solution, where convergence speed and cluster size were the primary parameters of interest (Li, Fang & Gross 2018). In similar work, clustering has been used for node organisation into logical groups with a focus on scalability, sensing stability/accuracy and energy efficiency. A key part of the work was an algorithm for distributed load balancing. Cluster member nodes each calculated the residual energy and distance from neighbouring clusters, using this information

to participate in a process for selection of the cluster heads. Due to the discrete-time and stochastic nature of the control processes at play, the authors elected to use a Markov decision process for cluster size selection (Muthukkumar & Manimegalai 2017).

A non-cooperative game approach was used for the formation of coalitions, which are effectively groups of SU nodes, for cooperative spectrum sensing. The game was separated into two parts, the first being optimal coalition formation for spectrum sensing and the second being optimal channel allocation of transmit time. Nash equilibria were found for both games separately, with the results demonstrating the combination of the two Nash equilibria is equivalent to the Nash equilibrium of a single game encompassing the entire complex problem. In this way the researchers established that the multivariable complexity of CR network optimisation can be decomposed into separate, less complex games and solved separately to arrive at an overall optimal solution (Jiang, Yuan, Leung, You & Zheng 2017).

1.2.5 Eigenvalue Based Spectrum Sensing

Eigenvalue based detection systems are based on the conception of a matrix as a transformation of multi-dimensional space, which is a central idea in linear algebra. The eigenvalues of a matrix may be found by application of the following equation:

$$\mathbf{A}\vec{v} = \lambda\vec{v} \tag{1.1}$$

Where $\mathbf{A}$ is the matrix, in this case the covariance matrix formed from the signals of each antenna source, $\vec{v}$ and λ are an eigenvector and eigenvalue of matrix $\mathbf{A}$ respectively. Any eigenvector of the $\mathbf{A}$ will result in a scaled version of itself post-transformation by $\mathbf{A}$, that is, the eigenvector represents a consistent direction that exists before and after transformation of multi-dimensional space. Eigenvectors can be indicative of useful characteristics depending on the context. For example, in rotational transformations in three dimensional space an eigenvector may be indicative of axis of rotation.

In CR applications energy detection methods are simple but suffer from being ineffective at low SNR values, while IEEE 802.22 suggests that PU signals should be detectable at -21 dB SNR with a probability of detection greater than 90 percent (Mashta, Altabban & Wainakh 2020). Eigenvalue based detection schemes have been identified as a method of inferring PU signal presence without any prior knowledge of the nature of that signal,

noting that increasing the number of signal samples (longer sampling time) does not translate to improved probability of detection for energy detection schemes when noise uncertainty is present (Zeng & Liang 2009). The authors appear to employ techniques similar to Principal Component Analysis (PCA) to inform threshold setting for a Boolean decision as to PU signal presence. Although they do not explicitly identify PCA as being employed, the formation of a covariance matrix from received signals and the selection of particular eigenvectors/values obtained from it are consistent with PCA. Their work explored detection algorithms based on the ratio of maximum or average eigenvalue to minimum eigenvalue.

Principal Component Analysis (PCA) is a statistical processing technique where multiple sets of observations are captured, standardised, arranged into a matrix and then processed into a covariance matrix. The matrix $\mathbf{X}$ for a set of M observations of signal *x(t)* expressed in discrete time as *x(n)*, will have dimensions MxN where N is the number of samples. The covariance matrix $\mathbf{C}$ is square and formed from $\mathbf{X}$. It has the variance of each row of $\mathbf{X}$ as its diagonal elements, and the covariances between rows of $\mathbf{X}$ comprise the other elements. For example, if there are three signal observations (such as from three co-located antennas) as rows of $\mathbf{X}$ then the covariance matrix will be:

$$\mathbf{C} = \begin{bmatrix} Cov(\mathbf{X}_1, \mathbf{X}_1) & Cov(\mathbf{X}_1, \mathbf{X}_2) & Cov(\mathbf{X}_1, \mathbf{X}_3) \\ Cov(\mathbf{X}_2, \mathbf{X}_1) & Cov(\mathbf{X}_2, \mathbf{X}_2) & Cov(\mathbf{X}_2, \mathbf{X}_3) \\ Cov(\mathbf{X}_3, \mathbf{X}_1) & Cov(\mathbf{X}_3, \mathbf{X}_2) & Cov(\mathbf{X}_3, \mathbf{X}_3) \end{bmatrix} = \mathbf{X}\mathbf{X}^T \tag{1.2}$$

Where the subscripts indicate the applicable row of $\mathbf{X}$. In an M multi-antenna system then, the covariance matrix of the received signals will have dimensions MxM. The eigenvalues of the covariance matrix correspond to the eigenvectors along which most of the variation exists in the data. By selecting the dominant eigenvectors and forming a 'feature matrix', the original (standardised) data matrix can be transformed into a single vector of observation data with the useful variance retained and irrelevant variance minimised. For the purposes of CR we expect useful variance around a zero mean for most man-made signals that exhibit periodicity, while noise can be considered variance that is not useful.

The first algorithm employed was Maximum-Minimum Eigenvalue (MME) detection and simply computed the covariance matrix from the received signals, calculated the ratio of maximum and minimum eigenvalues, then compared that value to a pre-defined threshold as follows:

$$\frac{\lambda_{max}}{\lambda_{min}} > \gamma \tag{1.3}$$

Where λ_{max} is the maximum eigenvalue of the sample covariance matrix, λ^{min} is the minimum eigenvalue of the same, and γ is the threshold. If the expression evaluates TRUE then the presence of a PU signal is inferred, else the opposite conclusion is drawn. The second algorithm employed was Energy with Minimum Eigenvalue (EME) detection which was similar to the first, however, employed energy detection to calculate the average power of the received signal rather than using the eigenvalue ratio. The ratio of the signal power to the minimum eigenvalue was compared to a pre-defined threshold in the same way as the first algorithm. This approach is subtly but meaningfully different from conventional energy detection, because it uses the minimum eigenvalue as a proxy for noise power rather than relying on a pre-defined noise power that may not be temporally stable. Neither approach used the relationship between signal and noise to select an appropriate threshold, but rather used number of samples, probability of false alarm and a smoothing factor as inputs into a probabilistic model using multivariate Gaussian Wishart matrices according to Random Matrix Theory (RMT). (Zeng & Liang 2009).

The two eigenvalue based spectrum sensing algorithms discussed use a decision threshold that is arrived at in using a probabilistic approach. There is an implicit assumption that the eigenvectosr associated with the PU signal have a persistently dominant eigenvalue at SNR where energy detection is not viable. This could be true and the experimental results presented support the conclusion that MME is superior to energy detection in the range of approximately -20 dB to -8 dB, however, inferior below -20 dB. The relationship between the eigenvectors, eigenvalues and the original data set permits transformation of the signal data into a signal matrix containing only the desired principal components, however, this was not pursued in either algorithm. The capacity for noise rejection is a key advantage of PCA techniques, but neither algorithm took advantage of the opportunity to reject noise in this way prior to comparison against a threshold.

Noting the primary advantage of eigenvalue detection over energy detection techniques is an apparently greater resilience to noise uncertainty, further work has been done in recent years to optimise eigenvalue detection approaches. One study defined a Standard Condition Number (SCN) as the ratio of the maximum eigenvalue of the signal covariance matrix to the minimum eigenvalue of the same. The SCN was then compared to a 'noise

only' SCN resulting in a Boolean decision of PU signal presence or absence. This study may be distinguished from the aforementioned one because it considers the effects of correlated noise rather than white noise, which renders the RMT approach based on a Wishart random matrix not applicable. The authors state that filtering of received signals gives rise to a filter output signal containing correlated noise, which consequently degrades sensing performance because the detection threshold selection was premised on uncorrelated white noise. The practical value of SNR estimation for transmission strategy selection is discussed as a potential way to modulate CR transmit power in an underlay scheme (Sharma, Chatzinotas & Ottersten 2013).

So far the eigenvalue based detection schemes discussed only considered a subset of the eigenvalues of the signal covariance matrix, however, an eigenvalue weighting system has been explored to make use of all of the eigenvalues. The approach involves an adaptation of the Generalised Likelihood Ratio Test (GLRT) and is referred to as the Eigenvalue Likelihood Ratio Test (ELRT), once again determining a threshold based on noise power inferred from some arrangement of the eigenvalues of the covariance matrix. Optimal, semi-blind and blind models were proposed, with each requiring different input data. Each model computes a weighting factor for each eigenvalue to be multiplied by, sums them all, thus giving a threshold against which to compare received the signal energy calculation and obtain a Boolean result for signal presence (Liu, Li, Wang & Jin 2017). The threshold calculations are given as follows:

$$Optimal: \qquad T_{opt} = \sum_{i=1}^{M} \frac{\rho_s}{\rho_s + \sigma_n^2} \hat{\lambda_i} \tag{1.4}$$

$$Semi-blind: \quad T_{sb} = \sum_{i=1}^{M} \frac{(\hat{\lambda_i} - \sigma_n^2)}{\sigma_n^2} \hat{\lambda_i} \tag{1.5}$$

$$Blind: \qquad T_b = \sum_{i=1}^{M} \frac{(\hat{\lambda_i} - \hat{\sigma}_n^2)}{\hat{\sigma}_n^2} \hat{\lambda_i} \tag{1.6}$$

Where ρ_s is the diagonal matrix of eigenvalues of the PU signal, $\hat{\lambda_i}$ is the i'th eigenvalue of the sample covariance matrix, σ_n^2 is the noise power and $\hat{\sigma}_n^2$ is the estimated noise power calculated from the mean of the eigenvalues of the recieved signal covariance matrix.

Another adaptation of the GLRT has been used based on the ratio between arithmetic and geometric mean (AGM) of the eigenvalues of the sample covariance matrix (Zhang,

Lim, Liang & Zeng 2010).

$$T_{AGM} = \frac{\frac{1}{M}\sum_m \lambda_{m,x}}{(\prod_m \lambda_{m,x})^{1/M}} \tag{1.7}$$

Where M is the number of antennas, m is the number of the eigenvalue, and $\lambda_{m,x}$ is the m^{th} eigenvalue of the sample covariance matrix x. This approach required only the eigenvalues of the sample covariance matrix, however, selected a threshold based on a desired false alarm probability which was presumably tabulated from prior simulations. The same analytical selection of threshold based on false alarm probability was seen in an adaptation of this approach (Bouallegue, Dayoub, Gharbi & Hassan 2018), where the AGM concept was applied again, except using only the maximum and minimum eigenvalues. This approach was termed 'mean to square extreme eigenvalue' (MSEE), giving the threshold as:

$$T_{MSEE} = \frac{\lambda_M + \lambda_m}{2\sqrt{\lambda_M \lambda_m}} \tag{1.8}$$

Where λ_M is the maximum eigenvalue of the sample covariance matrix, and λ_m is the minimum eigenvalue.

Automatic Modulation Classification (AMC) is the process of analysing a signal and algorithmically determining its modulation scheme according to known features. AMC can be used to enable demodulation of a range of signals in the RF environment, and finds obvious application in a military setting. In the CR context, eigenvalue based AMC has been explored as a means to enable a CR to glean more information about its RF environment and the types of signals present, which provides the opportunity to prioritise the signals of different users when selecting transmission strategy. The researchers sought to classify 64QAM, FM and OFDM received signals by comparison of their feature vectors. When received signals are noise only, the feature vector has a random direction and the eigenvalues of the sample covariance matrix are approximately equal. When the received signals contain signals exhibiting periodicity, the feature vector will have a definite direction which is indicative of the signal type, thus enabling classification (Bhatti, da Silva, Rowe & Sowerby 2014).

1.3 Conclusions

The Australian land area is very large and sparsely populated in many regional areas, making wireless broadband services a cost effective solution compared to wired networks. Those wireless networks are data rate limited by bandwidth, in part owing to a regulatory regime of exclusive spectrum licensing. There is an opportunity to increase the efficiency of spectrum utility through a time sharing approach via CR networks. These networks would be formed from SDR that adapt to their spectral environment. NBN fixed wireless services could potentially leverage bands of spectrum reserved for other users, such as TV broadcasters, when they are not in use.

IEEE 802.22 Wireless Regional Area Networks (WRAN) requires that an SU must be able to detect a PU with a probability of detection P_d of 90% and probability of false alarm P_{fa} of 10% at -21 dB (*IEEE Std 802.22-2011* 2011). Many different approaches are evident in the literature, with most using energy detection as the primary scheme. The point of difference lies in determination of the threshold against which the average received signal energy is compared, which requires an estimation of noise power. The common approach to this estimation is to use a probabilistic approach, which is intuitive because Gaussian white noise is random and conforms to a normal probability distribution. Where multiple antennas are used to gather received signals, the noise becomes a multivariate Gaussian probability distribution, hence the common application of random matrix theory observed in the literature.

The multiple antenna case permits examination of the sample covariance matrix, whose eigenvectors are indicative of the principal signal components and whose eigenvalues are the corresponding magnitudes. Studies exploring eigenvalue based detection schemes widely use the minimum eigenvalue or a combination of the eigenvalues as a proxy for noise power, with this assumption the major premise of the application of the selected probability distribution. An interesting exception is in some of the earliest and most promising work using the ratio of maximum to minimum eigenvalues as a proxy for signal power (Zeng & Liang 2009). This approach avoided a conventional energy detection approach and suggested that sufficient information could be contained in the eigenvectors and eigenvalues of the sample covariance matrix to make an informed decision on the question of signal presence.

Further, the signal detection algorithms only seek to report the presence of man-made signals as a Boolean result, however, modern modulation schemes such as OFDM utilise a spread spectrum approach employing different sub-channels that are occupied at different times. This is true in the context of NBN fixed wireless services, which utilise a scheme similar to 4G TDD-LTE. Energy detection is not a viable option for such an application without extensive filtering, since it is unable to differentiate between signal components. The Fast Fourier Transform (FFT) seems an appropriate tool for channel state detection to decide which channels are occupied and which are available for use in a particular band. The central question then becomes what threshold to set for detection when considering the normalised FFT amplitudes. The problem is simplified when channel frequencies are known in advance, which is realistic for practical cases involving commercial PUs who employ standardised modulation schemes.

The eigenvalues of a sample covariance matrix will be equal to noise variance when there is no man-made signal component present i.e. they will be approximately equal. In the alternative case the variance/covariance due to the man-made signal will dominate and thus correspond to the maximum eigenvalue and feature vector, up to a point. Intuitively, then, at some SNR the noise variance will conceal the presence of the useful signal variance/covariance, meaning there is a limiting SNR below which eigenvalue/eigenvector detection methods will be ineffective. Some important questions not addressed in the literature are as follows:

1. Is there a limiting SNR for eigenvalue/eigenvector based detection schemes, and what are the driving factors?
2. Can the relationship between eigenvalues serve as a proxy for SNR, and therefore support Boolean detection?
3. Can PCA be applied to signals captured from different receivers in the same band, in order to decide signal presence without performing FFT on all received signals?
4. Can an eigenvalue informed approach to FFT threshold setting be used to maximise detection accuracy in a multi-antenna, multi-channel context?
5. Are the above items applicable to the Australian context described in Section 1.1?

The focus of this research project is to investigate these questions primarily in the context of a spectral band containing constantly present components at different SNR. These

signals will ultimately be received via multiple antennas attached to separate receivers. Since channel detection experiments would require prior knowledge of channel states at various times, this research project instead focuses on Boolean detection. A channel detection simulation employing eigenvalue inferred SNR in combination with FFT threshold detection is also presented.

Chapter 2

Methodology

2.1 Data Collection

Several questions arose after review of the literature in Section 1.2. Investigation of these areas requires the collection and analysis of data. Data is collected from simulations run in MATLAB using generated signals initially, then subsequently using imported signals captured via SDR. A Boolean detection algorithm is applied to a signal generated at varying SNR and the associated P_d and P_{fa} are tabulated against them. To demonstrate the potential application of developed algorithms to an FFT detection scenario, a multi-channel analysis is conducted using generated signals with varying active component frequencies. Detection results are recorded against multiple SNR levels, with P_d and P_{fa} recorded for each SNR across multiple iterations of channel state changes.

Since the algorithms are being refined as the project progresses, the scenarios begin as simple simulations, progress to a replicated single receiver signal and ultimately culminates in the multiple receiver case. Simulations and simulation results are detailed in Section 3, while hardware experiments are detailed in Section 4. The same process flow shown in Figure 2.1 applies to both simulation and hardware experiments, except the recording and pre-processing blocks are replaced with signals synthesised in code for simulations. When using SDR receivers, the associated software permits export of a band in **.wav* file format. The receivers are used to simultaneously capture signals in the same frequency band (dependent on the limitations of the software), with the bandwidth dependent on the type of signal. The use of multiple receivers is intended to simulate a single multi-

antenna receiver. FM signals are used to gauge algorithm performance and refine signal processing prior to applying the developed detection algorithm to spread spectrum 2.4 GHz WiFi signals. Computation time for the core operations of MMPD and MME are also captured.

The broad process flow for data capture is shown in Figure 2.1. The same process is repeated 10,000 times for each SNR integer level between -35 dB and 0 dB. N_T is the truncated number of samples after pre-processing, and is less than the captured number of samples N.

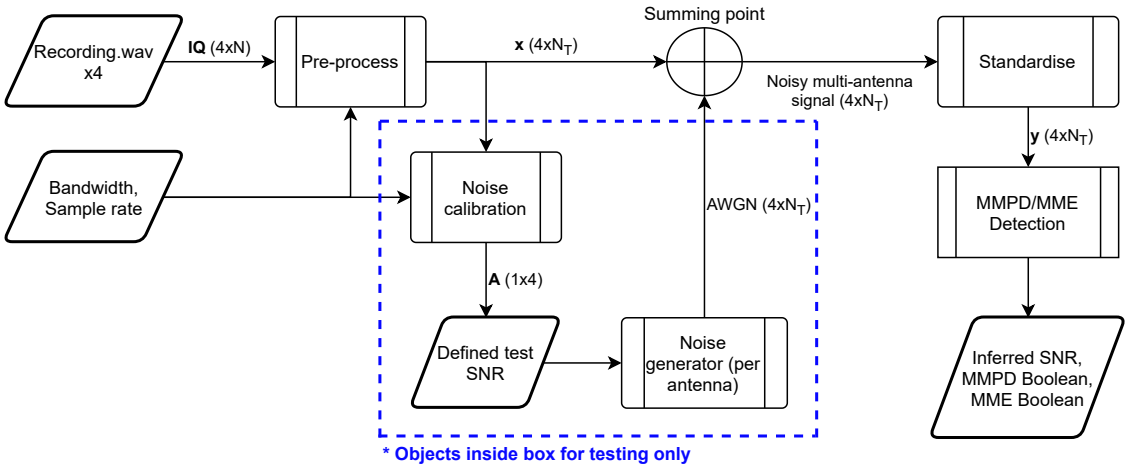

Figure 2.1: Generalised testing and data collection process flow.

2.2 Data Analysis

Quantitative data is collected based on the IEEE 802.22 standard requirements for probability of detection and probability of false alarm. The probability of detection is the number of correctly detected active signals out of the total number of active signals. The probability of false alarm is the number of incorrectly detected active signals out of the total noise only signals, providing worst case P_{fa}. This means the low SNR P_d will be equal to P_{fa}. The overall accuracy of detection algorithms will be captured through these two metrics, at different SNR for P_d. Varying SNR signals are obtained by evaluating the received signals and artificially adding white Gaussian noise. Detection algorithm performance is compared to maximum-minimum eigenvalue detection (MME), noting the different methods described in the literature are quite similar in many of their features.

To ensure the algorithms are compared in a meaningful way, they are applied to common scenarios i.e. using the same number of antennas, cycles captured, sampling frequency and signals. The P_d and P_{fa} are captured for multiple iterations at different SNR levels, and computation time is captured using the MATLAB 'run and time' feature. Since there is no ability to control the broadcasts captured using SDRs, only P_d is captured for the SDR scenarios. This is not problematic, since P_{fa} is equal to P_d at low SNR.

Ultimately, algorithm performance is based on a comparison of detection performance (P_d) at -21 dB. This value is chosen because it is a key performance metric defined in *IEEE Std 802.22-2011*. The developed algorithm will be deemed superior to MME if it satisfies both of the following criteria:

1. P_d is higher than MME at -21 dB, and this difference is statistically significant when averaged over 10,000 iterations; and
2. P_d at low SNR (-30dB and below) is greater than MME, but remains within the range $8\% \leq P_{fa} \leq 12\%$. This permits a $\pm 2\%$ tolerance around the 10% threshold defined in Section 3.2.

An average P_d difference greater than or equal to 5% at -21 dB will be deemed statistically significant *prima facie*, and will not be demonstrated through formal statistical testing. In either case, P_d vs SNR performance is reported graphically via MATLAB generated plots.

2.3 Tools and Materials

The broad resource requirements are identified in the table below

#	Resource	Quantity	Source
1	Computer and peripherals.	2	Existing asset and online purchase.
2	MATLAB software.	2	Existing assett.
3	RTL-SDR receiver	4	Nooelec online store.
4	HackRF SDR transceiver	1	Nooelec online store.
5	TexStudio Latex Editor	1	Existing asset.
6	Microsoft Office	2	Existing asset.
7	Printer and consumables	1	Existing asset.
8	Scientific calculator	1	Existing asset.
9	Office furniture	1	Existing asset.

Chapter 3

Eigenvalues and Signal to Noise Ratio

3.1 Max-Min Percentage Difference (MMPD) Eigenvalue Detection

If signal detection is to be accomplished by examining the eigenvectors and eigenvalues of the sample covariance matrix, then the relationship between the eigenvalues and SNR must be understood. The persistence of a signal eigenvalue compared to noise is logically related to how strong the variance/covariance distribution is, which is of course related to the number of data points. This is reflected in the literature where there is an improvement in detection probability with a higher number of samples or an increased number of receiving antennas. Intuitively, the frequency is also a contributing factor since the variance attributable to the signal is maximised when more cycles occur in the sample space. Consequently, the relationship between number of cycles captured and SNR detectability is a natural entry point for investigation.

To obtain meaningful results, SNR detectability needs to be properly defined and so too the eigenvalue proxy we wish to use. Since the maximum to minimum eigenvalue ratio has proven to be a reasonable proxy for signal strength, and the minimum eigenvalue has frequently been used as a proxy for noise power, we suspect the relationship between maximum and minimum eigenvalues is key. Since we know that there is a large difference

between the maximum and minimum eigenvalues of the sample covariance matrix when a strong signal is present, and that the difference between them diminishes as noise begins to dominate, one may reason that the percentage difference between the maximum and minimum eigenvalues makes a good proxy for SNR i.e. we might infer SNR via calculation of percentage difference in eigenvalues. Thus, a figure of merit is this percentage difference defined as:

$$D = 100\frac{\lambda_{max} - \lambda_{min}}{\lambda_{max}} \tag{3.1}$$

Where λ_{max} is the maximum eigenvalue of the sample covariance matrix, and λ_{min} is the minimum eigenvalue of the same.

To examine the relationship between D and SNR, a signal was generated in MATLAB, subjected to varying levels of additive white Gaussian noise (AWGN) corresponding to SNR levels from 10 dB to -40 dB. To initially examine the relationship, 1500 cycles C were generated as N samples at frequency f of 1 MHz for 4 (M) antennas. The sample space was therefore an $M \times N$ matrix $\mathbf{X}$. The sample covariance matrix was computed by application of Equation 1.2. The eigenvalues were calculated using the built in eigs() function in MATLAB, which is based on Equation 1.1. The maximum and minimum eigenvalue percentage difference D was then calculated for all SNRs and plotted from -25 dB to 10 dB:

From Figure 3.1 we see a clear relationship between D and SNR. The plot was re-generated with varying values of f with nil effect, indicating the eigenvalue to SNR relationship is independent of frequency provided a given number of cycles are captured in the sample space. This relationship underpins our novel eigenvalue based detection scheme, named Maximum-Minimum Percentage Difference (MMPD) Eigenvalue Detection. The code listing for the function used to obtain the above results is located in Appendix D.1.

By taking the average of the percentage difference values at low SNR of -40 dB to -30 dB, we may define the percentage difference for cutoff D_{co} and SNR cutoff SNR_{co}, which appears to be approximately -20 dB by visual inspection of the plot. Below SNR_{co} the signal cannot be reliably distinguished from the noise using the relationship between maximum and minimum eigenvalues. This is an interesting result because it corresponds to the SNR threshold where detection schemes significantly degrade as presented in the literature, noting that those schemes almost all use 4 antennas too. When less cycles

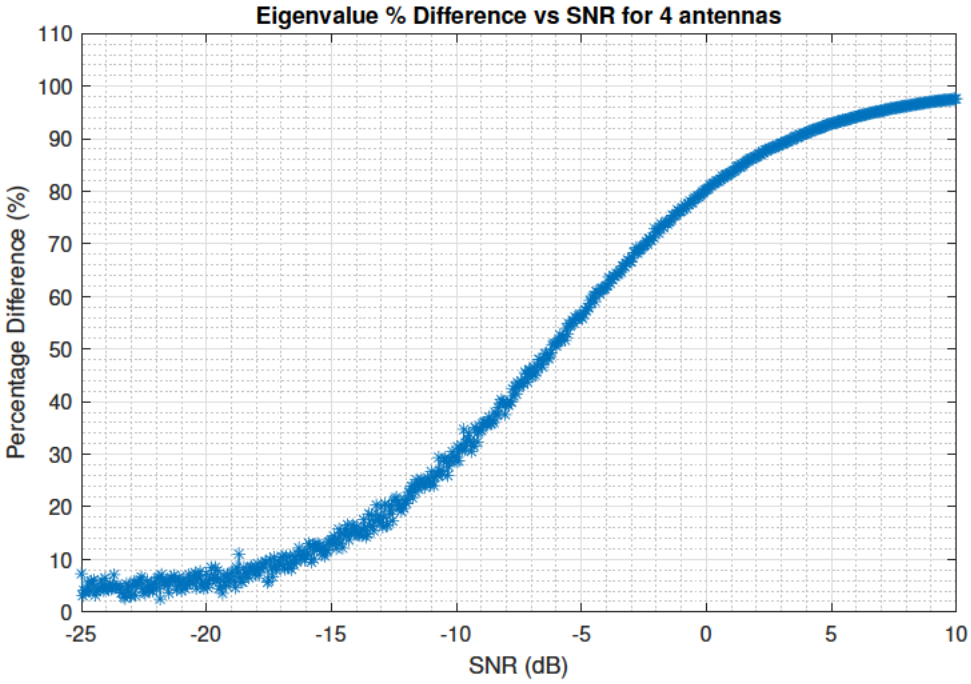

Figure 3.1: Eigenvalue % difference vs SNR for 4 antennas - 1500 cycles captured.

or less antennas were used in the simulations, the curve of Figure 3.1 flattened out and SNR_{co} became higher. To improve noise immunity we may choose a slightly higher percentage difference value for signal detection. Using the conventional corner approach, we shall define the percentage difference for degraded detection as $D_{deg} = 1.5D_{co}$, and the associated SNR as SNR_{deg}.

Using an adaptation of the logistic growth equation, the 4 antenna, 1500 cycle case is adequately modelled by the following equation where D_{co} is 4.3321 and SNR_{co} is -19.72.

$$\text{SNR} = -4.362 \times \ln\left(\frac{D_{co}(100 - D)}{D(100 - D_{co})}\right) + \text{SNR}_{co} \tag{3.2}$$

The relationship between simulated data and Equation 3.2 is shown in Figure 3.2. Although similarly good approximations are possible for different antenna counts using logistic growth models, some pre-tabulation of parameters is required and it is time consuming to compute them. Consequently, it is a simpler approach to take advantage of the curve fitting capabilities of MATLAB and pre-tabulate the polynomial coefficients. An important realisation at this point is that for a given number of cycles and antennas used, the characteristic curve for percentage difference does not change. This means that a value ϕ in the vicinity of D_{deg} can serve as a reliable threshold for Boolean signal de-

tection at SNRs greater than or equal to SNR_{deg}. To further examine this idea, one may consider the relationship between SNR_{co}, C and M.

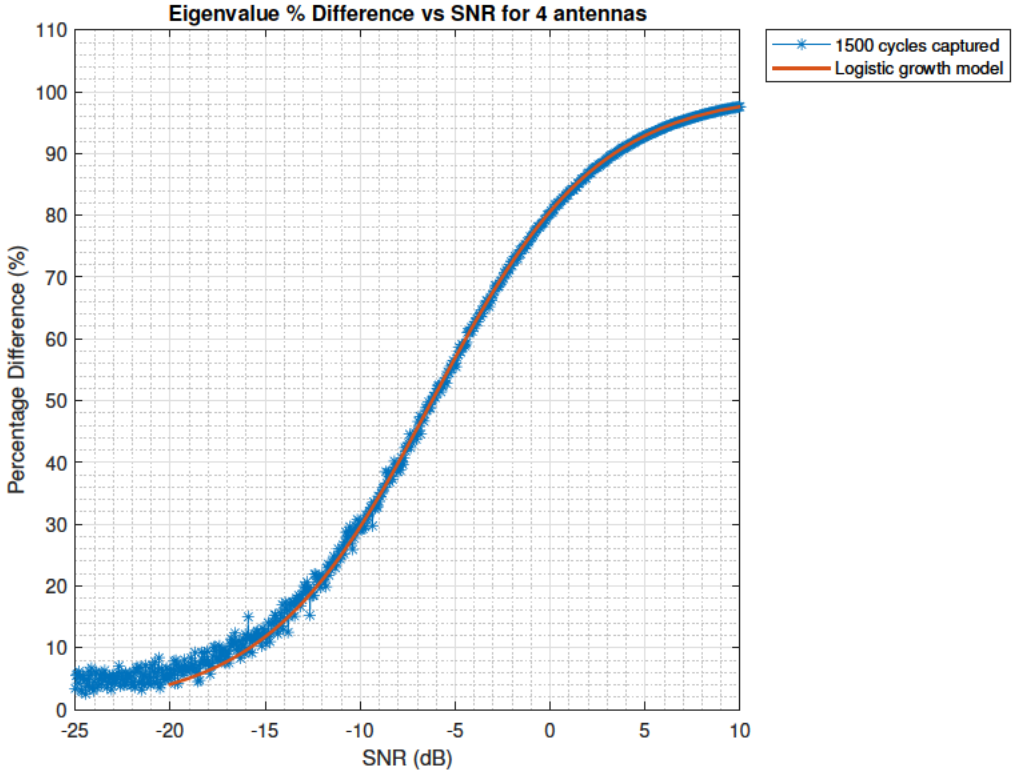

Figure 3.2: Logistic growth model for 4 antennas - 1500 cycles captured.

Figure 3.3 indicates an exponential decay relationship with diminishing improvements as the number of cycles captured increases. The magnitude of the SNR cutoff improvements (the lowest SNR signal able to be distinguished from noise) appears to be governed by the number of antenna elements used to receive the signal. The number of cycles captured simply moves along the curve defined by the antenna count, although this is for a fixed sampling frequency five times signal frequency f. If the sampling frequency is increased further, one would expect more variance/covariance attributable to the signal, and thus a lower SNR_{deg}. A consequence of these results are that a system designer can select C and M as a trade off between hardware cost/complexity and CR listening time. The duration that the CR monitors the environment before processing the information and making decisions is very important, since opportunistic channel use is time sensitive if transmission conflicts with a PU are to be avoided. The code listing for generation of Figure 3.3 is in Appendix D.2.

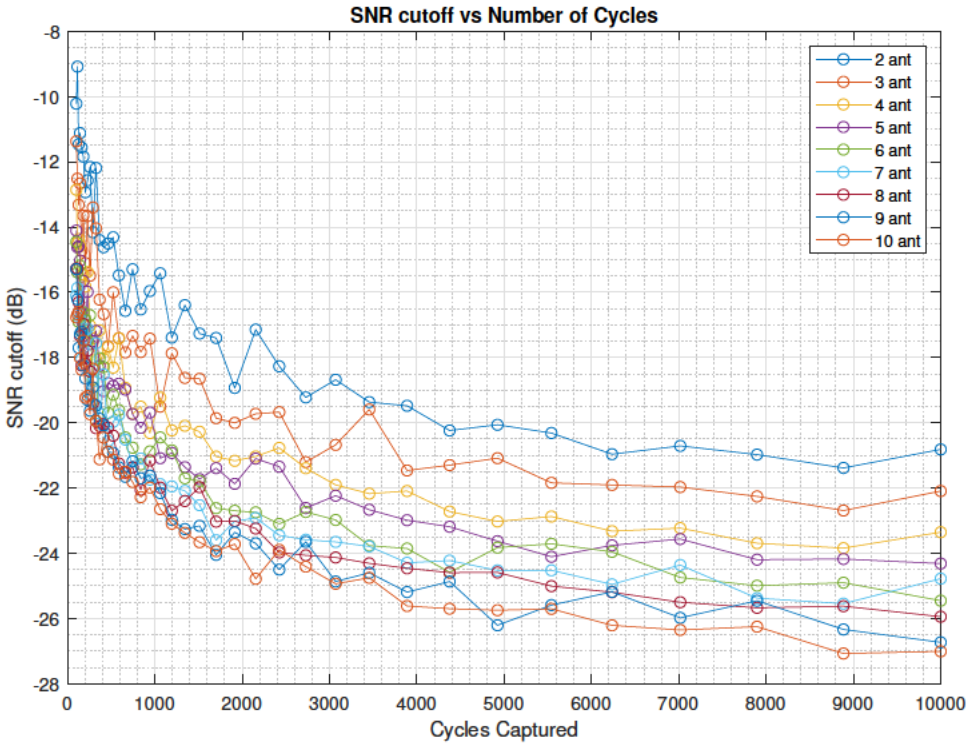

Figure 3.3: SNR cutoff vs number of cycles for varying number of antennas.

Tabulating the polynomial coefficients is a computationally efficient way to obtain a reasonable SNR inference from an unknown signal, for a given number of co-located antennas. Review of Figure 3.3 reveals that the majority of improvement from additional antennas occurs at 2000 cycles captured and below, thus lookup tables by M for D_{co}, SNR_{co}, and the polynomial coefficients for $SNR(D)$ calculation are given in the table below for the 2000 cycle case.

Table 3.1: Parameters by number of antennas - 2000 cycles captured.

			Polynomial Coefficients for **SNR(D)**					
M	D_{co}	SNR_{co}	**a**	**b**	**c**	**d**	**e**	**f**
2	1.7644	-17.701	1.993e-8	-4.818e-6	4.744e-4	-2.413e-2	8.177e-1	-1.894e1
3	2.9561	-18.989	3.247e-8	-8.1e-6	7.949e-4	-3.849e-2	1.111	-2.3e1
4	3.6597	-20.96	3.858e-8	-9.602e-6	9.315e-4	-4.427e-2	1.228	-2.53e1
5	4.5142	-21.007	5.106e-8	-1.293e-5	1.259e-3	-5.884e-2	1.513	-2.827e1
6	5.1995	-22.048	5.498e-8	-1.388e-5	1.345e-3	-6.237e-2	1.585	-2.973e1
7	5.6836	-22.345	5.056e-8	-1.249e-5	1.185e-3	-5.428e-2	1.412	-2.931e1
8	6.377	-22.849	6.204e-8	-1.563e-5	1.504e-3	-6.902e-2	1.717	-3.216e1
9	6.783	-23.529	6.099e-8	-1.520e-5	1.443e-3	-6.545e-2	1.631	-3.211e1
10	7.1653	-24.152	7.046e-8	-1.781e-5	1.712e-3	-7.825e-2	1.908	-3.477e1

The optimisation of the system is a question of deciding how close to SNR_{co} to position $SNR(\phi)$, where ϕ is the optimum percentage difference to serve as a detection threshold. Shifting the threshold ϕ down from 6.5 to 5.5, for example, improves P_d at -20 dB while increasing P_{fa}. Figures 3.4 and 3.5 illustrate this relationship averaged over 1000 iterations where $N = 10000$ samples, $f = 1$ MHz, $F_s = 5$ MSPS $M = 4$ and $C = 2000$. The code used to generate these plots is in Appendix D.3. Importantly, the standardisation method defined in Section 4.2.2 was applied prior to generation of these plots, since data from multiple receivers requires standardisation prior to forming the sample covariance matrix.

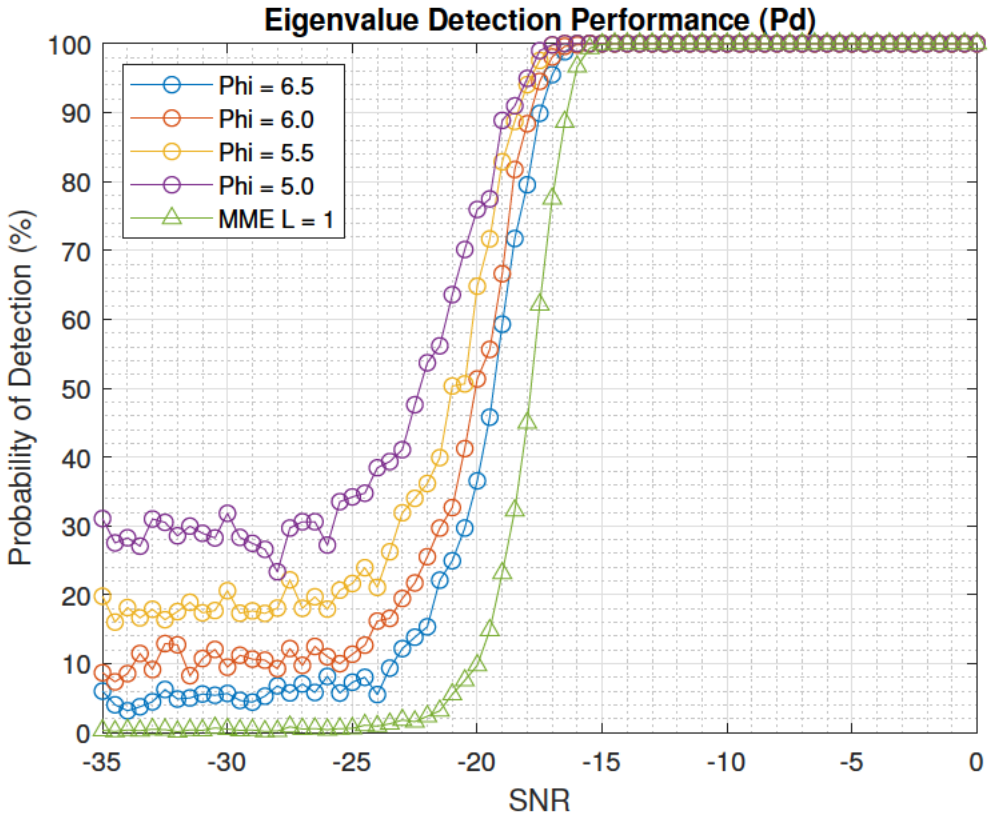

Figure 3.4: Probability of detection P_d at different SNR with varying inferred SNR threshold.

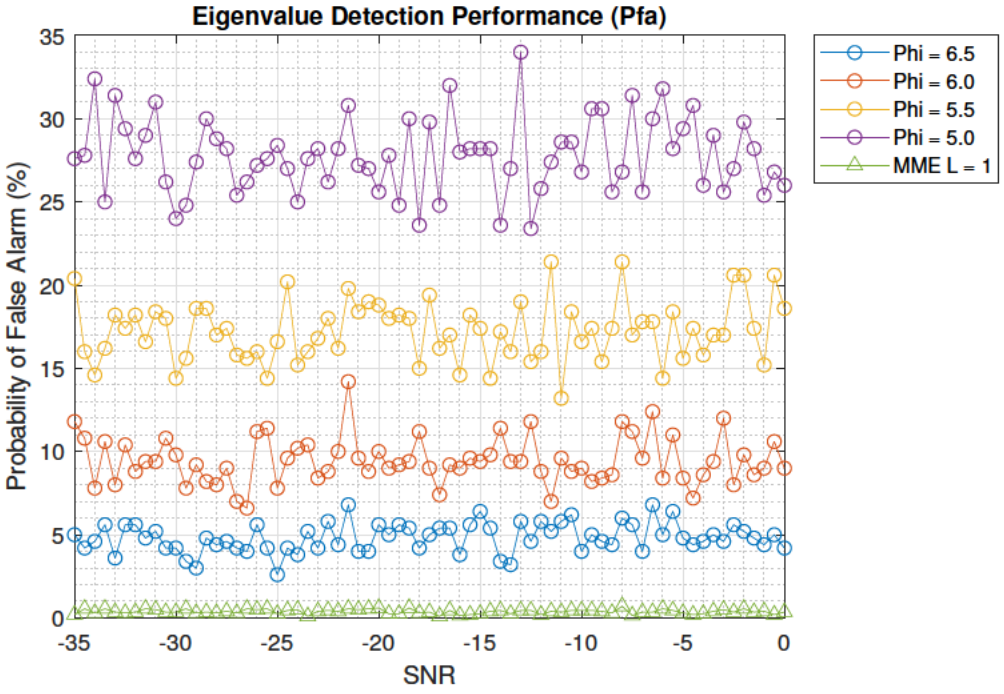

Figure 3.5: Probability of false alarm P_{fa} at different SNR with varying inferred SNR threshold.

An approach based on the percentage differences between maximum and minimum eigenvalues of the sample covariance matrix is not present in the literature to date, as far as the author is aware. It is distinguished from other eigenvalue based Boolean detection methods by the following features:

1. Only requires knowledge of bandwidth and the lowest frequency requiring detection.
2. Defines the low SNR limit of the scheme.
3. Does not employ energy detection as most other schemes do.
4. Does not employ probabilistic threshold setting.
5. Is far simpler to understand compared to random matrix theory approaches.
6. Informs CR listening time vs receive antenna count to achieve a desired performance.
7. Intuitive method to adjust performance in software, trading off P_d against P_{fa}.

3.2 Optimal MMPD

In the previous section the MMPD scheme was introduced and simulated results given for four different threshold settings. The thresholds selected were in the vicinity of D_{deg}, that is, $1.5D_{co}$. The optimum threshold is selected based on the 10% P_{fa} limit (*IEEE Std 802.22-2011* 2011), with the objective of remaining near to this limit since P_d and P_{fa} are a trade-off. We therefore define the optimum threshold ϕ when $9.5\% \leq P_{fa} \leq 10\%$. Analytical results indicate a reasonable approximation for $10\% \pm 2\%$ is given for the $C = 2000$ case by:

$$\phi \approx -0.0555M^2 + 1.561M + 0.5254 \pm \gamma \tag{3.3}$$

Adjusting γ permits fine P_{fa} control. Positive γ results in lower P_{fa}, while negative values have the opposite effect. For the four antenna case explored in this project, the optimal γ value is $+0.3$.

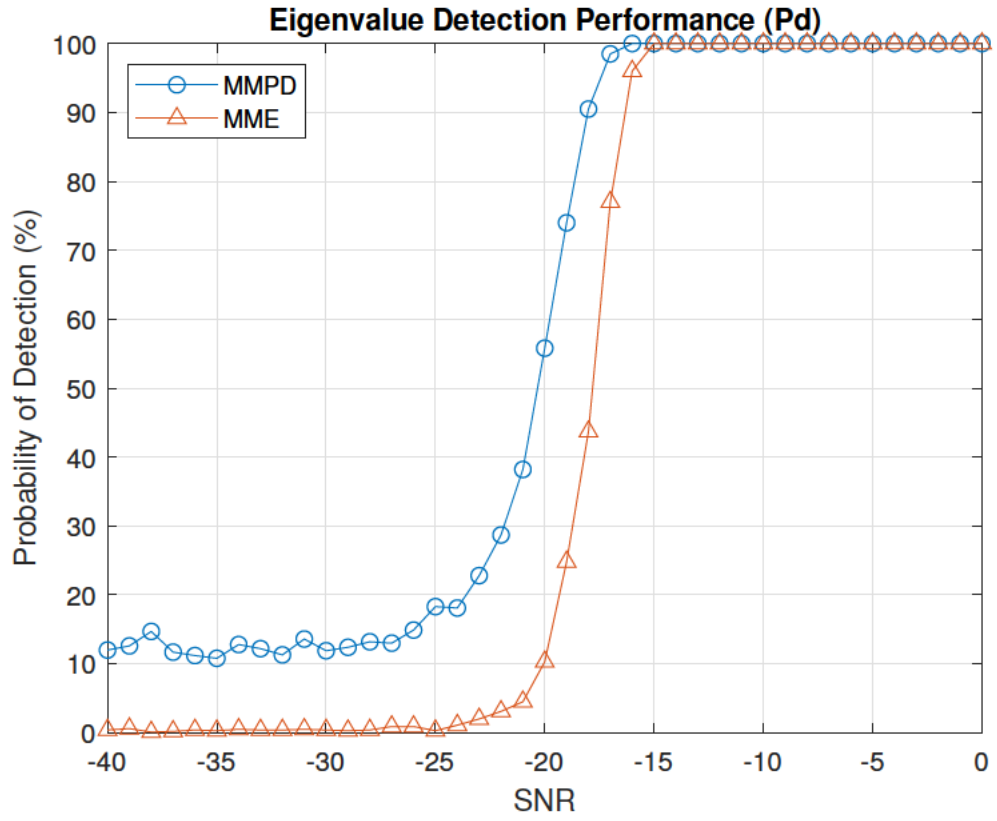

Figure 3.6: Probability of detection P_d vs SNR for MMPD and MME detection.

To examine how MMPD performs compared to MME, it is important first to note reliance of the MME threshold setting on a 'smoothing factor' L and some pre-tabulated values based on P_{fa}. The selection of the former is generally arbitrary since the number of signal sources for the received signal is typically unknown, which is problematic since L has a significant effect on performance of the MME detection scheme. The MME scheme and MMPD scheme both rely on an analytical approach to threshold setting to some extent. Here we have sought to inform the MMPD threshold setting in a pragmatic way based on simulation results, which yields superior results to MME at significantly lower sample counts to those reported in the literature. This translates to a theoretically shorter required listening time to decide the question of signal presence using eigenvalue based detection, maximising P_d while maintaining an average P_{fa} of $10\% \pm 2\%$.

Figure 3.6 shows average results over 1000 iterations, with $N = 10000$, $f = 1$ MHz, $F_s = 5$ MSPS, $M = 4$, $C = 2000$, $\phi = 6.3543$ (MMPD), $L = 1$ (MME). Although the MME performance could be improved if L were lower than 1.0, the scheme defines $L > 1$ (*IEEE Std 802.22-2011* 2011). Figure 3.6 also represents best case MMPD and MME performance, since it is based on perfectly correlated antenna inputs. Practical results

using signals from different receivers is expected to show the curves right-shifted, with P_d tapering off at higher SNR.

MMPD execution time averaged 407.56 μs while MME averaged 399.92 μs, a difference of 1.875%.

3.3 FFT Detection Effectiveness

3.3.1 Potential For Channel State Detection

While Boolean detection algorithms are useful for indicating signal presence, they are less useful for determining which channels are in use in a given band. A useful property of the eigenvectors of the sample covariance matrix is the ability to form a feature vector and transform the data set into a single vector with the noise information minimised. This is a specific application of PCA. A natural consequence of this noise suppression ought to be extension of the useful SNR range for the FFT, in a similar way to autocorrelating a single signal vector prior to FFT.

Another interesting idea is that a particular eigenvector is indicative of a particular modulation scheme or signal type (Bhatti et al. 2014). This prompts the idea of eigenvector persistence, where one might discover the PU eigenvector by survey at a known strong signal location, then use that information to inform detection at some far away SU location that has a poor or variable SNR.

From these ideas we conceive the following approach to signal detection in the single/multi-channel context employing FFT:

1. SU receives PU signal at unknown SNR.
2. SU uses MMPD algorithm developed in Section 3.1 to infer SNR.
3. If the inferred SNR is above $SNR(\phi)$ then transform the sample matrix using the feature vector of the sample covariance matrix, then apply the FFT.
4. If the SNR is below $SNR(\phi)$ then transform the sample matrix using the last calculated feature vector.

5. If the SNR is below $SNR(\phi)$ and no previously calculated feature vector is available, transform the sample matrix using the feature vector then apply the FFT. This rejects noise despite not being able to extract a meaningful feature vector.

6. Use threshold detection modulated by the inferred SNR i.e. when there is less noise it is safe to use a relatively low threshold, but when the SNR is low the threshold setting can be ramped or adjusted to minimise PU transmission conflicts or improve some other metric.

Transmission conflicts with PUs are highly undesirable because they lead to inefficient use of SU transmission power as well as having potential commercial implications. Consequently, we reason it is better to incorrectly deem a channel as occupied than to incorrectly deem it as free. For the FFT this leads to the somewhat counter-intuitive conclusion that a lower threshold is preferable at low SNR. Figure 3.7 demonstrates the basic advantage of PCA for noise rejection, noting the significant noise reduction at -25 dB even when $SNR(\phi)$ is -19 dB. This is only possible with the use of MMPD detection, since it yields the inferred SNR upon which feature vector selection is dependent. Here $N = 23,353$ samples, $F_s = 2^{22}$ SPS, $M = 4$, $C = 2000$ and the frequencies are channels in the 4G LTE band from 3.4025 GHz to 3.5945 GHz spaced at 8 MHz, down-shifted by a factor of 4. The results for FFT threshold detection without PCA or MMPD application are given in table 3.2. The FFT results after PCA transform, and where the threshold is modulated by the inferred SNR from MMPD, are given in table 3.3. The results are averaged over 1000 iterations of random channel states for each of the 25 channels. The code for this section is in Appendix D.4.

The tabled data indicates that the use of MMPD and PCA increase the accuracy of channel state detection at low SNR. The adjustment of the normalised peak detection threshold used was extremely simple, being 0.4 above $SNR(\phi)$ and 0.25 below. A more sophisticated approach could optimise the trade-off between minimising PU conflicts and maximising free channel utility, but here we consider it sufficient to broadly demonstrate the advantages of PCA and MMPD.

Another important point to consider here is standardisation. In the simulations thus far the observation sets (samples from each antenna) have been defined in such a way that no standardisation is required, however, this will not be the case when capturing real signals if the antenna and receiver gains are unequal.

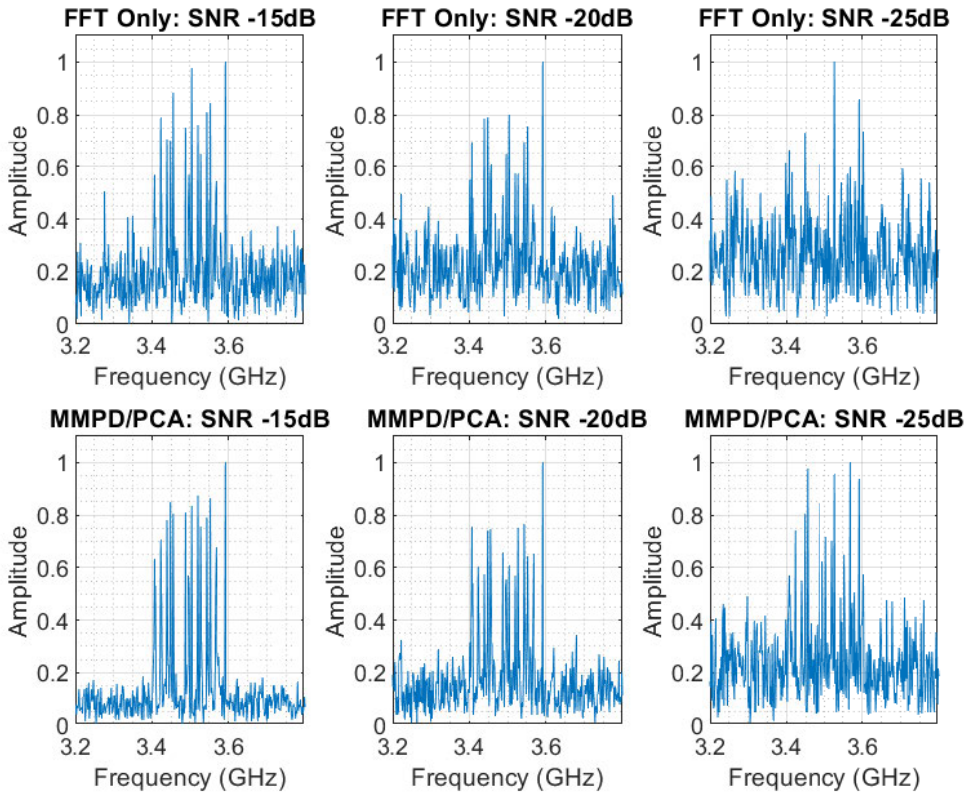

Figure 3.7: Effect of MMPD and PCA on FFT.

Table 3.2: Multi-channel FFT effectiveness without PCA and MMPD.

SNR (dB)	Channel Detection Accuracy (%)	PU Conflicts (%)	Free Channel Utility (%)
0	100	0	100
-2	100	0	100
-4	100	0	100
-6	99.984	0.016	100
-8	99.944	0.056	100
-10	99.9	0.1	100
-12	99.644	0.356	100
-14	99.18	0.76	99.862
-16	98.192	1.404	99.097
-18	95.584	2.728	96.357
-20	91.58	4.42	91.695
-22	85.628	7.496	85.862
-24	78.94	11.748	81.168
-26	71.592	18.244	79.513
-28	64.924	24.592	79.143
-30	60.332	29.148	79.856

Table 3.3: Multi-channel FFT effectiveness using PCA and MMPD.

SNR (dB)	**Channel Detection Accuracy (%)**	**PU Conflicts (%)**	**Free Channel Utility (%)**
0	100	0	100
-2	100	0	100
-4	100	0	100
-6	100	0	100
-8	100	0	100
-10	100	0	100
-12	99.996	0	99.987
-14	99.936	0	99.846
-16	99.676	0	99.26
-18	98.716	0.004	97.224
-20	96.336	0.024	92.317
-22	92.192	0.12	83.955
-24	86.748	0.416	73.703
-26	80.976	0.88	63.304
-28	74.464	2.172	53.151
-30	70.68	4.032	49.736

3.3.2 Eigenvector Augmented Fast Fourier Transform (EVA-FFT)

The hardware experiments conducted in Section 4.2.2 triggered the need for an alternative detection algorithm that could perform Boolean signal detection in a band of interest when the received signals have poor correlation and eigenvalue based methods are impaired.

EVA-FFT was thus developed as a simple detection scheme that takes advantage of the hallmark PCA technique of transforming the sample covariance matrix, while borrowing from the work done developing MMPD. In short, EVA-FFT has evolved from MMPD and its process flow is presented in Figure 3.8.

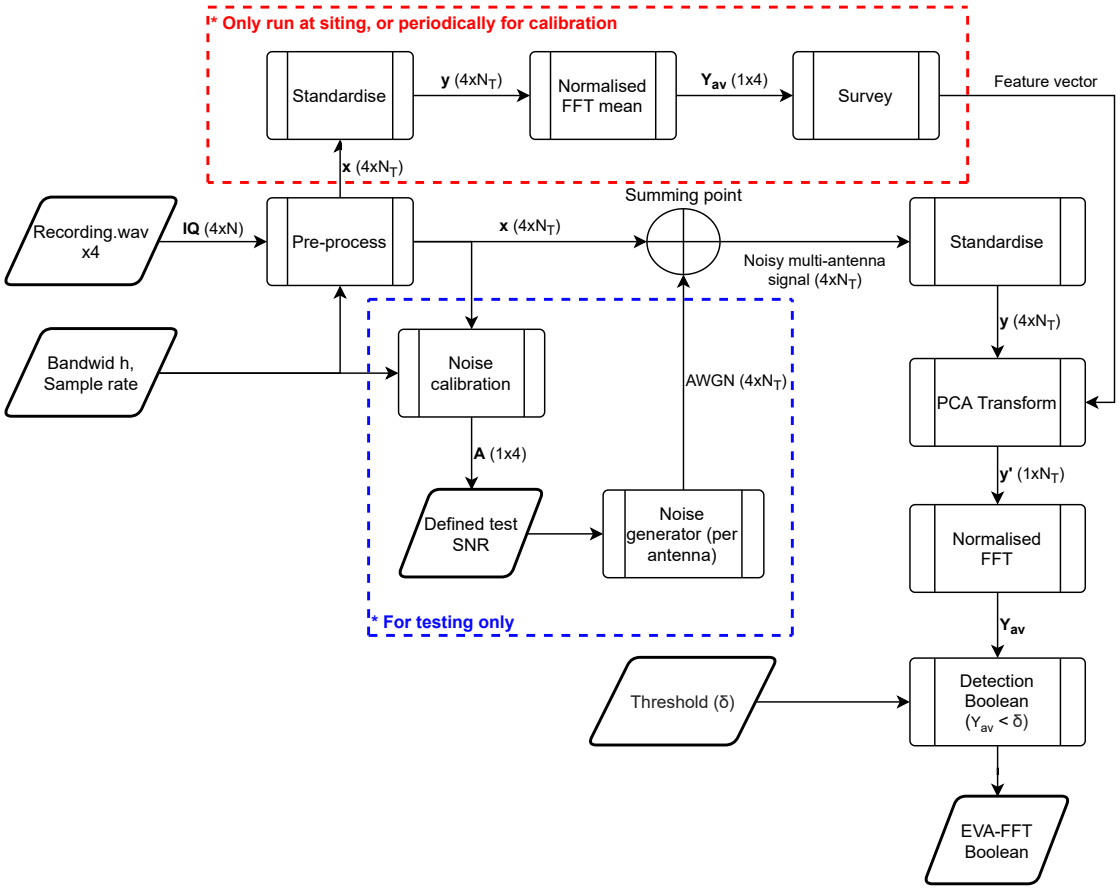

Figure 3.8: Eigenvector Augmented Fast Fourier Transform test method.

EVA-FFT transforms the sample covariance matrix using a surveyed feature vector. In this context, the survey operation is the assignment of weightings to the receiver signals for the purposes of signal detection. These weightings are the elements of a vector used to transform the standardised sample matrix $\mathbf{y}$. A normalised FFT is then applied to the result, yielding $\mathbf{Y}$. The average value of $\mathbf{Y}$ is $\mathbf{Y}_{av}$. The normalisation is useful if

one considers that the normalised spectral content of noise has a particular average level δ. Simulations using uncorrelated white Gaussian noise show δ is approximately 0.3. If some other signal is present in the band, even slightly above the noise floor, the average amplitude of the normalised spectrum will be below δ. This relies on pre-processing to shift the tuned center frequency to $Fs/4$ as described in Section 4.2.1. For this to be effective we must satisfy $F_s > 4B$ to ensure the noise floor is represented in the vicinity of $F_s/2$, where bandwidth B is the bandwidth centered on the signal of interest. The parameter δ serves as a viable threshold for Boolean signal detection, and can be adjusted to provide a desired P_{fa}. Analytical results indicate the optimum threshold for maintaining P_{fa} of 10% is 0.2685. Experimental results for EVA-FFT are given in Section 4.3.

The normalised FFT for uncorrelated noise, and the average level are shown in Figure 3.9.

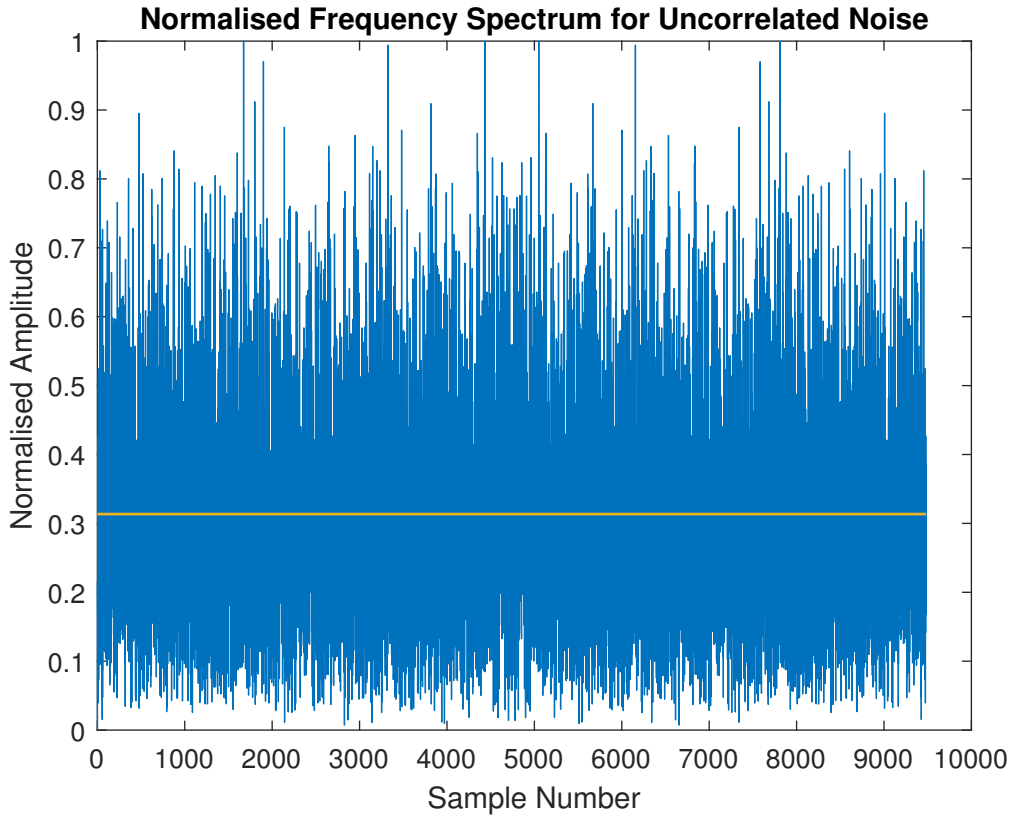

Figure 3.9: FFT for uncorrelated noise.

The red box in Figure 3.8 represents a self-survey function to select an appropriate feature vector. It only needs to be used for calibration, such as when the receiver changes location or orientation. An alternative is to use a surveyed feature vector obtained from MMPD.

The self-survey routine provides the system with different feature vector options depending on desired detection characteristics. The details of these options are fully elaborated in the hardware experimentation information at Section 4.3.

Chapter 4

Hardware Experiments

Chapter 3 defined MMPD as an eigenvalue based detection scheme and offered some mathematical methods to support algorithm implementation. Simulation results were collected and compared to sub-optimal MME detection for the Boolean detection case, and a methodology was proposed to extend the scheme to improve frequency domain peak detection post-FFT.

This chapter investigates the effectiveness of MMPD when dealing with real world signal information. The chapter follows initial equipment set-up, data capture and analytical trials, through to the final experiments.

4.1 Initial Data Capture and Analysis

4.1.1 Single SDR Receiver - FM Signals

To investigate the potential benefits of MMPD and PCA in processing RF derived signals, it is necessary to adequately capture such signals from the real world. This helps one to develop an understanding of the format of captured data and thus how it might be properly processed to yield some useful analysis. Initial data capture was performed using an RTL2832U/R820T2 receiver, commonly referred to as RTL-SDR. It uses a small telescopic antenna connected to the receiver via mini-coaxial connecion, with the receiver plugged into a USB Type A port on a computer. Initial data capture was done for various

wideband FM radio stations using a single RTL-SDR. This is because FM stations are easy to recognise and the decoded signal components can be verified by playback of the mono audio component. Figure 4.1 shows the blue RTL-SDR connected to a vertically polarized antenna. To reduce image height, the full length of the antenna is not shown.

Figure 4.1: Single RTL-SDR for data capture.

SDR Sharp (SDR#) software was used to examine the spectrum and define the settings for **.wav* file capture, which was subsequently processed in MATLAB. SDR# permits recording of demodulated audio, or baseband IQ signals, where I and Q represent the in-phase and quadrature components of the RF signal. The term baseband refers to the down-conversion of the RF signal from approximately 100 MHz to be in the vicinity of 0 Hz.

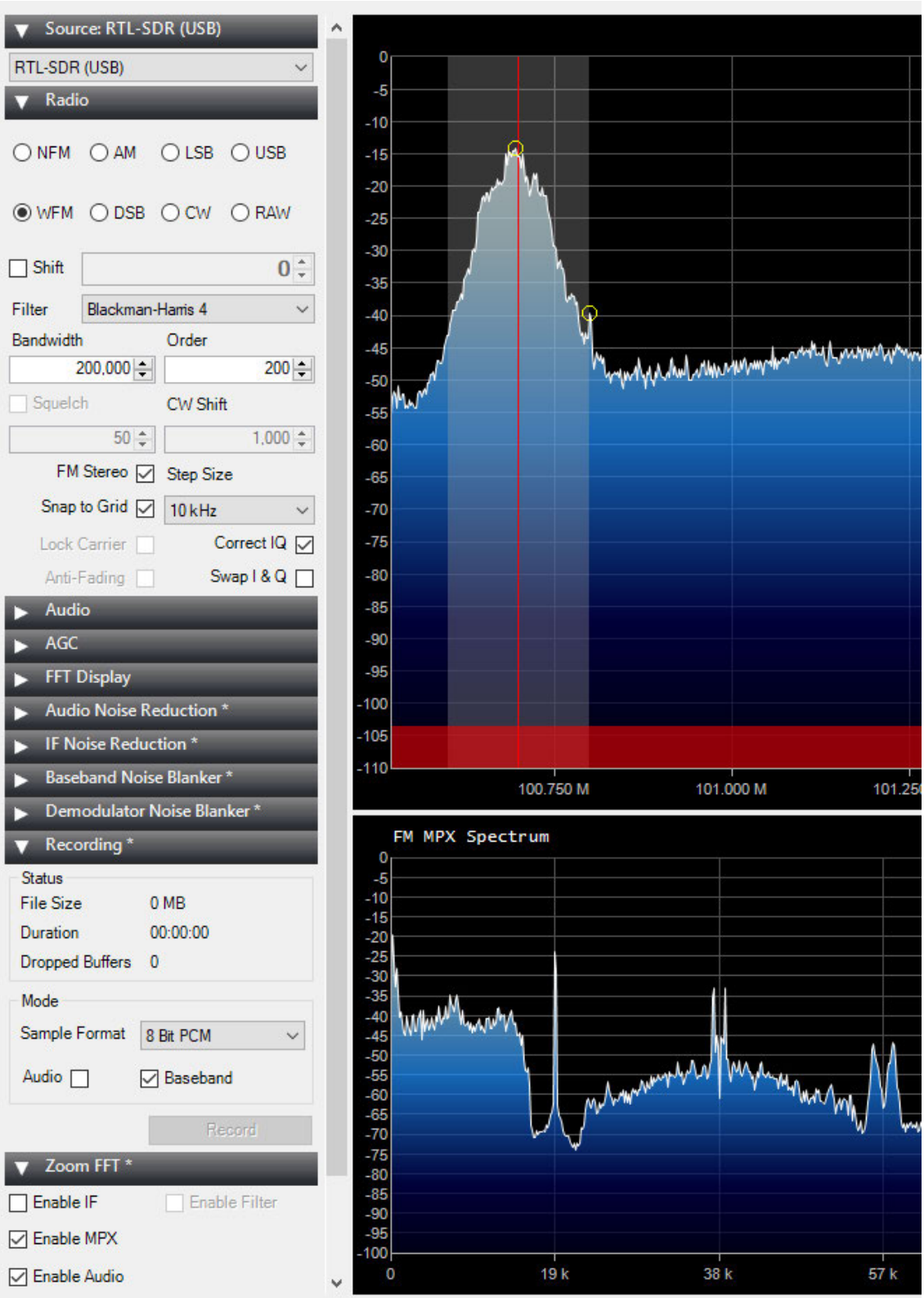

Figure 4.2: SDR# Settings and spectrum for data capture.

Figure 4.2 shows the SDR# software settings and selection of 100.7MHz for capture. The software limits capture to a maximum bandwidth of 250kHz in the wide-band FM mode, and here it was set to 200kHz. In the bottom right of the figure, the demodulated

information is clearly visible, particularly the 19kHz pilot tone. To the left of the pilot tone is the mono audio, which we obtain from the baseband IQ signal and play back in MATLAB to verify signal capture.

The baseband IQ recordings were imported into MATLAB at a sample frequency of 3.2 MSPS. The default import using the *audioread()* function yields a matrix with two columns which hold the I and Q components. First, the FFT was applied to a fraction of the signal matrix to determine what the baseband offset frequency was. The SDR baseband offset frequency could not be manually selected and it changed each time the device was power cycled. The signal was then band-pass filtered with the offset frequency being the centre of the pass-band of a 30th order filter, and the filter bandwidth being equal to the bandwidth selected in SDR#. The filter was constructed using a custom function, and applied to minimise any noise in the signal so that the FM MPX spectrum would be clearly visible. Once filtered, the columns for the I and Q components were combined as $x = I + jQ$ to give a single vector of complex numbers. The phase angle ϕ between samples was obtained as the angle between the real and imaginary parts of $x_n x^*_{n+1}$. Noting that the modulating signal amplitude is proportional to the instantaneous frequency, and the instantaneous frequency is equal to the rate of change of phase angle, it follows that $m(n) = \phi(n) - \phi(n-1)$.

The result for a typical music recording is shown in Figure 4.3.

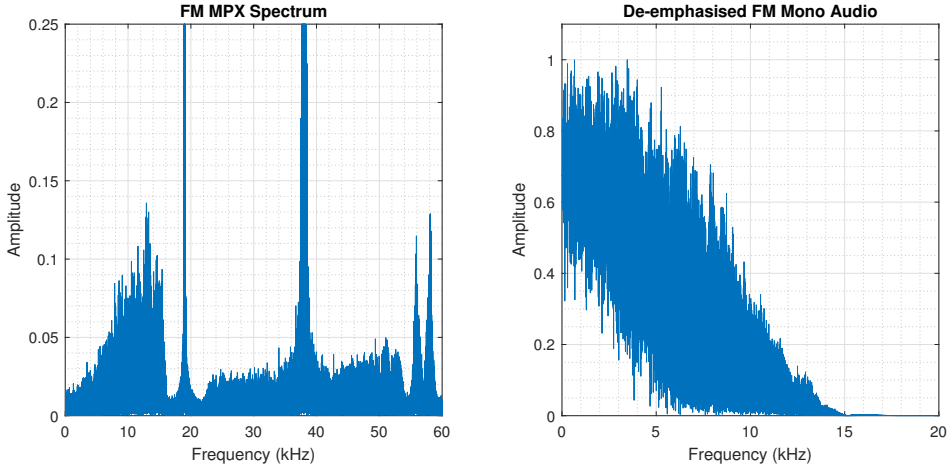

Figure 4.3: Recovered intelligence from 100.7MHz FM radio station in Toowoomba.

Appendix D.5 contains the code listing for this section. The MPX spectrum obtained can be compared to the typical spectrum for FM broadcasts shown in 4.4. Once the mono audio component was decimated to a 48 kSPS and a rudimentary approximation

of de-emphasis filtering was applied, the mono audio play back was loud and clear.

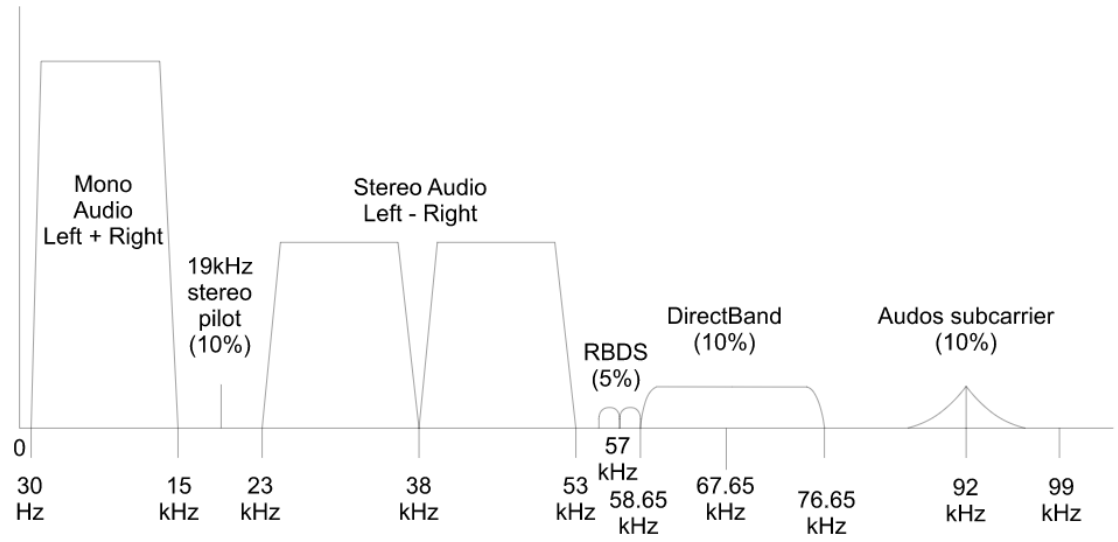

Figure 4.4: Baseband spectrum for broadcast FM (Murray n.d.).

The RTL-SDR receiver is well suited to capturing FM radio broadcasts which contain clearly defined peaks in the spectrum of the modulating signal. It should be noted that some broadcasts contain only mono audio, and the 19 kHz pilot tone and higher frequency components are not seen in the modulating signal spectrum. Comparing the performance of Boolean detection using a single receiver vs multiple receivers is a logical progression from here, and necessary to employ the MMPD techniques developed in the preceding sections.

The initial experience capturing signals using a single SDR receiver and processing them in MATLAB yielded some important results in terms of pre-processing SDR signals generally. Since the captured band is down-shifted to baseband and therefore centered at 0 Hz, any DC level presents as a significant 0 Hz spike at the center of the signal. Removal of a DC level requires application of a DC level shift in the time domain, which is a simple matter of finding the average amplitude.

Another important point is that the MMPD algorithm developed in the preceding sections assumes 2000 cycle capture based on the carrier frequency and bandwidth as viewed in the frequency domain. When the signal is centered at baseband it is not clear how many samples are required to achieve a 2000 cycle equivalent. The solution to this problem becomes simple when one considers the format of the data as imported into MATLAB i.e. an Nx2 matrix where the first column contains the I components and the second column contains the Q components. Since we require a time domain signal for application

of MMPD anyway, we may frequency up-shift the spectral components upwards during the conversion from IQ format to a real time domain signal.

The first question to be answered in that regard, then, is how far to up-shift the signal. Since the visible spectral response is limited to half the sample rate, it seems logical to center the band of interest in the visible window. This corresponds to $F_s/4$. Dimensional analysis reveals the appropriate number of samples is thus:

$$N_T = \frac{CF_s}{F_s/4 - B/2} \tag{4.1}$$

Where N_T is the number of samples for truncation, C is the number of cycles for capture, F_s is the sample rate and B is the bandwidth. This equation only yields a useful result if $F_s/4 > B$, therefore $N_T = 4C$ is used as an alternative. The IQ matrix is truncated to be a $N_T \times 2$ matrix. For reconstruction of the time domain waveform the desired carrier radian frequency is defined as $\omega_c = F_s/4$, the magnitude for each sample is equal to $|R(n)| = |IQ(n,1) + jIQ(n,2)|$, the phase angle is $\phi(n) = \arg(R(n))$, and the time domain waveform is thus:

$$x(n) = |R(n)| \cos(\omega_c t + \phi(n)) \tag{4.2}$$

Recalling that the algorithm developed to form the sample covariance matrix for eigenvalue and eigenvector calculation requires observations to be in the columns, with a set of observations per row, $x(n)$ is transposed appropriately here to be a row vector. Figure 4.5 shows the effect of the pre-processing operations described. The DC spike has been removed, the band is lower resolution due to sample truncation, and the band is centered at 50 kHz since the sample rate was 250 kHz in this example. MATLAB code for the pre-processing function is listed at Appendix D.6.

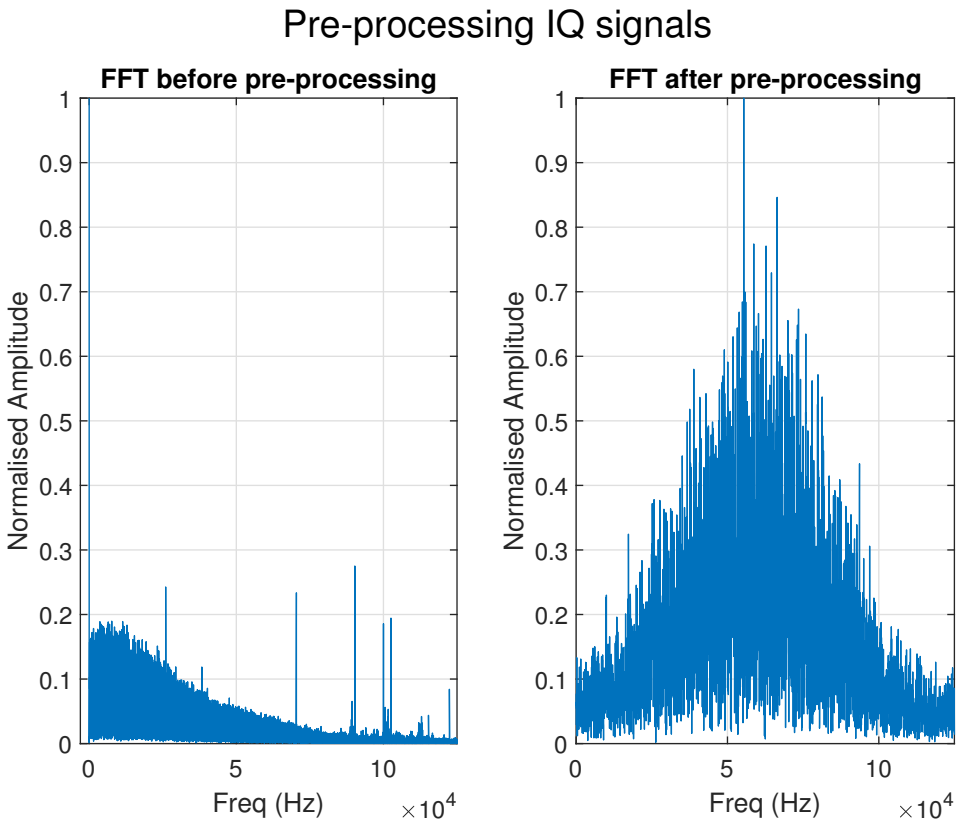

Figure 4.5: IQ signal MMPD pre-processing for a single receiver.

4.2 MMPD Applied to FM Signal Data

4.2.1 Quadrupled Single Receiver Signal

The SDR hardware available for experimentation in the project does not lend itself to synchronous recording. Multiple computer terminals are required, with each connected to an SDR and running appropriate software. Although multiple receiver recordings have been taken, as will be seen in subsequent sections, there are issues of spatial and temporal variation that could potentially affect the results. Temporal variation results from the inability to perfectly synchronise recording start and stop times on all computer terminals. Spatial variation refers to the physical dispersion of the receivers and associated antennas, causing variable SNR for each.

Noting these matters, it is prudent to attempt application of MMPD to data obtained from a single receiver, with that data replicated three times to simulate four received signals of comparable strength. Obviously, these signals will match exactly and correlation between them will be high. We therefore degrade the signals individually with Additive

White Gaussian Noise (AWGN) to artificially affect the eigenvalue variation in the sample covariance matrix. This allows preliminary testing of the MMPD algorithm on real SDR captured signal data without yet needing to address the temporal and spatial variation challenges. It also provides a benchmark for best-case detection performance, since the data is perfectly correlated. To be clear, eigenvalue based detection techniques cannot be used for a single data vector in the way shown in this section. For example, if we replicated AWGN and applied it to such a detection scheme, false positives would be obtained because the data is correlated. The approach is used here for the express purpose of developing the ideas discussed in the preceding sections.

To test the performance of the MMPD Boolean detection algorithm across a range of SNR, the received signals must be degraded with a specific amount of noise. To do this for all SNR values requires corresponding noise amplitude values, which in turn requires knowledge of how SNR changes with AWGN. By examining the spectral power density of the pre-processed IQ data before and after the addition of AWGN, approximate peak noise amplitude A_n can be determined according to Equation 4.3, where A_0 is the estimated signal amplitude and A_g is the estimated noise amplitude prior to the introduction of any additional noise.

$$A_n = \frac{A_0}{10^{\frac{SNR}{20}}} - A_g \tag{4.3}$$

A_0 and A_g were solved by simultaneous equations for known values of A_n and SNR. Hence, knowing A_0 and A_g to a reasonable approximation, we may determine the appropriate AWGN standard deviation A_n to obtain a desired SNR. Analytical results indicate this method is accurate within ± 5 dB which results in an offset compared to the SNR inferred by the MMPD algorithm.

The offset exists because Equation 4.3 relies on accurate calculation of initial SNR, as well as SNR after the introduction of test noise. That is, how the SNR changes with the introduction of test noise determines the result. SNR was estimated using the element-wise square of the FFT of the signal. The errors in SNR estimation were due to adjacent band interference and difficulty perfectly centering on the band of interest, since the center did not always correspond to the maximum peak. This inaccuracy caused imprecise AWGN introduction for testing and ultimately manifested in the afore-mentioned offset

which was different for each imported signal. The differences were due to the spectral peculiarities of each recording. This was problematic because evaluation of algorithm performance at specific SNR levels, of course, requires signals to be reliably degraded to those levels.

The first attempt to overcome the 'offset issue' was to introduce a band-pass digital pre-filter. This measure improved accuracy but inconsistent offset SNR values were still observed between different recordings, regardless of the filter order. If one recalls that an effective band-pass filter can eliminate all components outside the band of interest, it follows that the noise power has been almost entirely eliminated. A simple calculation is then performed to determine the average signal power P_s, which allows calculation of the required standard deviation in the noise distribution A_n for any desired SNR. In practice, A_n is a multiplier for the $randn(1, N)$ function in MATLAB to produce AWGN with average noise power A_n^2.

$$P_s = \frac{1}{N}\sum_{n=1}^{N} x(n)^2 \tag{4.4}$$

$$A_n = \sqrt{P_s 10^{\frac{-SNR}{10}}} \tag{4.5}$$

This approach eliminated the offset completely for all recordings, tested separately. Consequently, the MMPD inferred SNR exactly matched the expected SNR after the introduction of specific AWGN levels. It is essential that the pre-filtered signals be used only for the purposes of calculating the noise levels for testing, and not for filtering the received signals prior to applying the MMPD algorithm. This is because band-pass filtering gives rise to correlated noise, dramatically increasing P_{fa}. An example of this is shown in Figure 4.6, where uncorrelated white Gaussian noise only is received and subjected to pre-filtering, AWGN addition and then MMPD detection.

The right-most part of the plot shows the unwanted detection even though uncorrelated noise was received at each antenna. This represents a potential problem in radio receivers where band-pass filtering is a common operation, either in hardware or software. Care should be taken to avoid the problem of unintentionally causing noise correlation. Figure 4.7 shows the result of MMPD on the same uncorrelated noise, without applying band-pass filtering first.

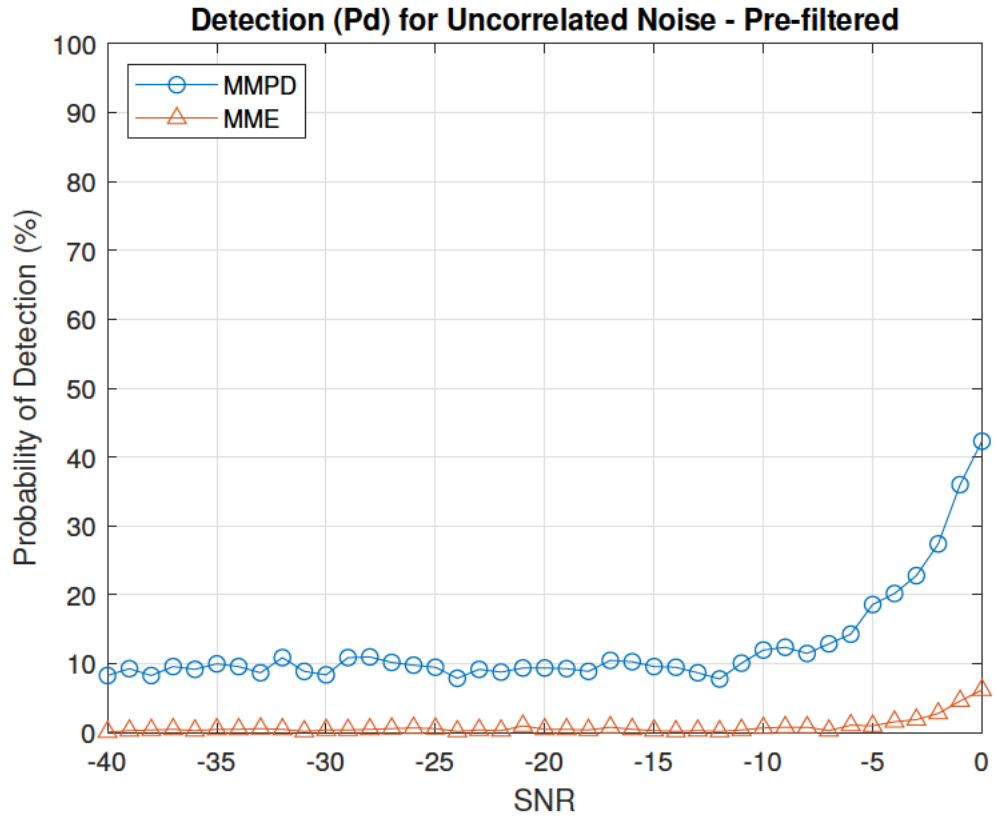

Figure 4.6: Probability of detection P_d when uncorrelated noise is band-pass filtered before MMPD.

Since the AWGN levels were determined by direct calculation from the captured signal data, it follows that the primary hypothesis of this research project remains valid; the percentage difference between the maximum and minimum eigenvalues of the sample covariance matrix is a valid proxy for SNR where the noise is AWGN. Figure 4.8 shows the probability of detection for MMPD and MME at various levels of signal degradation. These results are promising and constitute a logical progression toward the multiple receiver case, where the challenges of temporal and spatial diversity must be overcome. Comparing Figure 4.8 to Figure 3.6, we see comparable MMPD and MME performance to the simulated performance.

The MMPD result is remarkable, especially considering the significant differences in sample count and sample rate between the simulated and single receiver scenarios. Simulations were conducted using a sample rate of 5 MSPS and 10,000 samples captured, while the SDR hardware is limited to 3.2 MSPS. The 2000 cycle equivalent after pre-processing was 9481 samples. The low SNR P_d is approximately 10%, indicating optimised detection as described in Section 3.2 for a 10% P_{fa}.

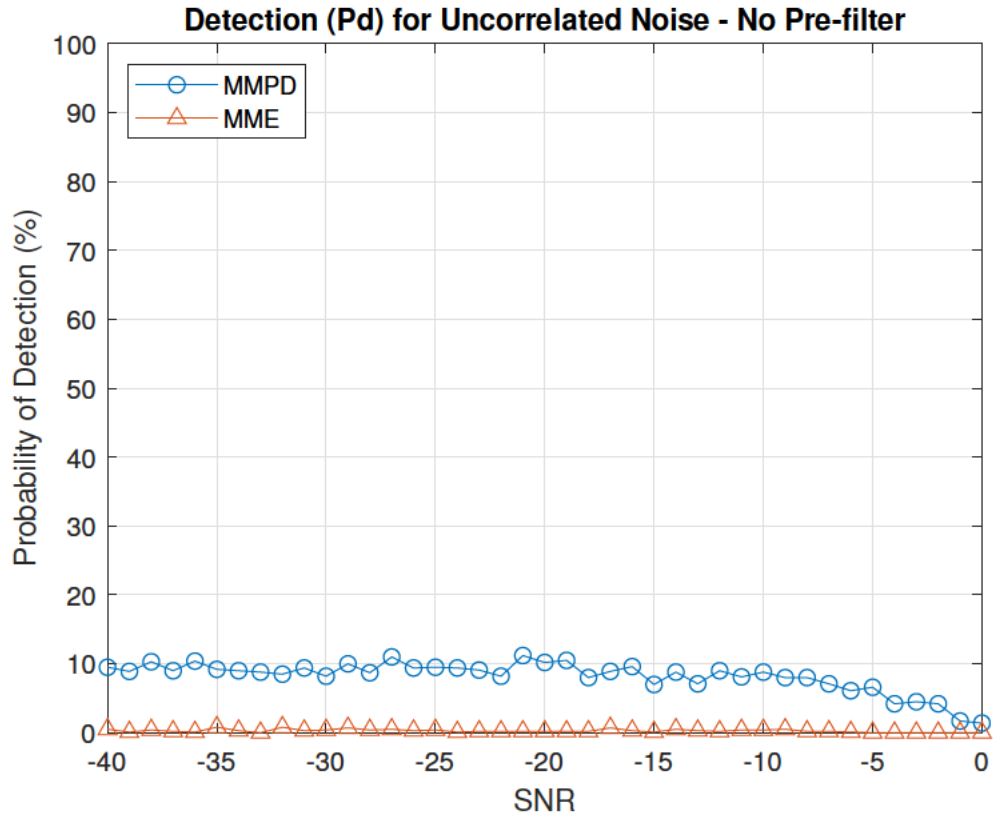

Figure 4.7: Probability of detection P_d when uncorrelated noise is not band-pass filtered before MMPD.

The detection performance closely matches the simulated results. Comparable detection performance supports the hypothesis developed in chapter 3 that the number of cycles captured has a significant influence over eigenvalue spread in the sample covariance matrix. This in turn supports the proposition that the MMPD scheme can inform required listening time for a CR node. The code for this section makes use of a function adapted from the single channel MMPD/MME detection code developed in chapter 3. The code listing with this function are included at Appendix D.7. The results for this section are averaged for 10,000 iterations and were repeatable using different SDRs and computer terminals.

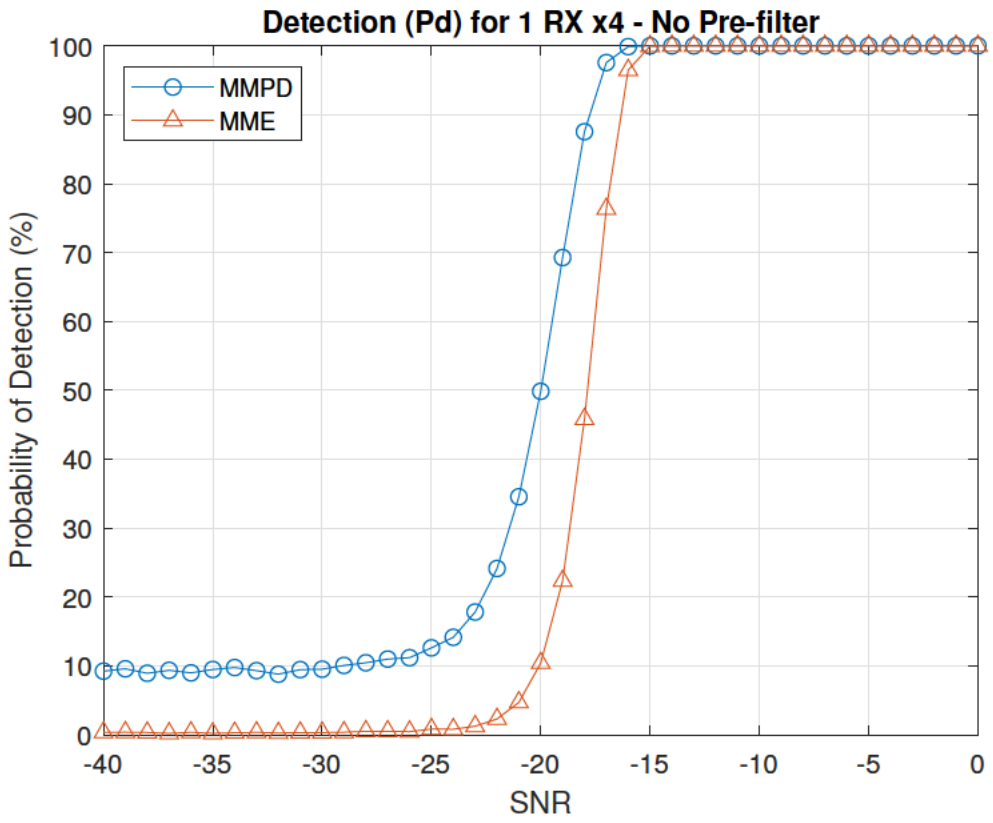

Figure 4.8: Probability of detection P_d vs SNR for MMPD and MME detection - 1 RX.

4.2.2 Four Receiver Signals

As Section 4.2.1 alluded, the multiple receiver case introduces the problems of temporal and spatial diversity which must be overcome to obtain meaningful results from the sample covariance matrix of the received signal data. Temporal diversity is a problem of synchronisation, where the recording start and stop times vary which means that corresponding samples are offset in terms of their matrix position. The synchronisation problem also means the different recordings contain a different number of samples. Spatial diversity refers to the physical differences in receiver and antenna positioning. The data from each receiver is scaled differently due to the variation in received signal strength. Another issue is that separate receivers apply slightly different offsets when shifting the band of interest to baseband. This results in different frequency components between recordings. While pre-processing improves the alignment of spectral components, there is still some difference between them.

Spatial diversity necessitates standardisation of the data, which scales the sets evenly so that the sample covariance matrix provides meaningful results, and therefore yields meaningful eigenvectors. Standardisation has been achieved by converting all data to Z scores $z(n)$ according to Equation 4.6, where $x(n)$ is the sample value, $\hat{x}$ is the mean and σ is the standard deviation for the data from a particular antenna. Applying this equation for each antenna after pre-processing and noise addition, then concatenating the vectors, effectively addresses the issue of spatial diversity and better simulates co-located antennas with comparable gain.

$$z(n) = \frac{x(n) - \hat{x}}{\sigma} \tag{4.6}$$

Considering the temporal diversity problem, the start time offset is very small compared to the total recording time for the signals. Since the signal in the band of interest is not intermittent in our case, we may conclude that there is sigificant overlap of samples exhibiting periodicity. We therefore subject the received signals to the detection algorithm without undertaking to resolve the temporal offset known to exist in the data. It is expected that MMPD performance will be reduced as a result of this decision. The difference in sample count is naturally resolved by the pre-processing function, which truncates the data for each receiver to be the same number of samples N_T.

AWGN with specific peak amplitudes were introduced to evaluate MMPD and MME algorithm performance for different SNR for FM signals at 100.7 MHz with bandwidth 250 kHz. This was accomplished per antenna in the same manner described in Section 4.2.1. This approach ensures the signals are approximately equal SNR for testing purposes, however, the temporal offset and slight difference in frequency components affects the sample covariance matrix and therefore also affects the eigenvalue spread. This resulted in the eigenvalue based detection methods, MMPD and MME, failing to detect at all when multiple receivers were used. The probability of detection versus SNR is shown in Figure 4.9. This may not be overly problematic, since in a real multi-antenna receiver the synchronisation and frequency offset difference will be negligible. This situation is likely, however, to cause significantly reduced detection performance if eigenvalue based detection methods are applied to signals from separate receivers. The problem is potentially exacerbated if the receivers are in motion e.g. mounted to a vehicle.

Eigenvalue based detection schemes in the literature and simulated scenarios assume perfectly correlated data prior to the addition of AWGN. The results of this experiment suggests that the relationship between SNR and eigenvalues relied upon in these schemes is not practically viable for detection of low SNR RF signals.

An important conclusion arising from the results obtained is that the relationship between SNR and the maximum/minimum eigenvalues of the sample covariance matrix is not evident for the multiple receiver case. The reduced performance compared to the single receiver case has an intuitive explanation; capture synchronisation and frequency up/down shifting affects the performance of any eigenvalue based detection scheme. These inferences are reasonable at face value, but require further investigation to establish with any rigour. These questions are left open for future research.

The code for this section is listed at Appendix D.7. The pre-processing algorithm is listed at Appendix D.6, having been re-used here. The results for this section are averaged for 10,000 iterations. The results were repeatable with different sets of SDR captured signals.

4.3 EVA-FFT Applied to Multiple Receivers

Since eigenvalue based detection performance was reduced compared to the theoretical performance when multiple receivers were used, the detection scheme defined in Section

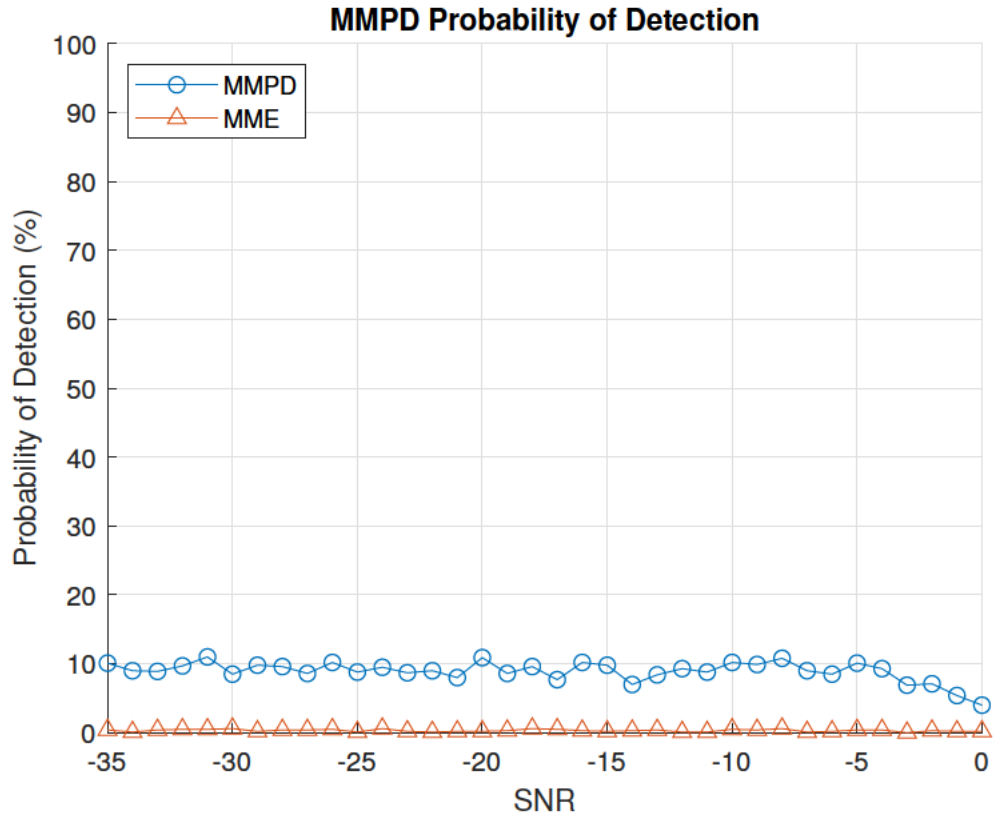

Figure 4.9: Probability of detection P_d vs SNR for MMPD and MME detection (FM 100.7 MHz) - 4 RX.

3.3.2 was developed and applied for the remaining experiments. The EVA-FFT algorithm is demonstrated for both a self-surveyed feature vector and an MMPD continuously obtained feature vector.

EVA-FFT initially relied on pre-existing knowledge of the feature vector for the signal of interest. This was obtained using a hackRF radio to capture strong signals of interest at 20 MSPS. This radio has an effective Low Noise Amplifier (LNA) and Variable Gain Amplifier (VGA), which allowed recording of high SNR signal data. The hackRF was also used because it permits complete independence from the four RTL-SDR receivers used for capturing the signals of interest.

An unexpected result was obtained when the feature vector was examined. Each of the values was approximately equal. This compels one to query whether the feature vector may be used as a weighting scheme for receiver P_d. When a sample covariance matrix is formed from identical observation sets, as done here with the hackRF, the feature vector contains equal values. The answer becomes obviously affirmative when

Figure 4.10: hackRF RF transceiver.

considering the basic PCA transform operation. For example, if the feature vector is $[1, 0, 0, 0]$, then pre-multiplying by the $M \times N$ receiver data will only include data from receiver 1. Since eigenvalue based detection has failed to operate using multiple receivers, the sample covariance matrix cannot yield information for the purposes of EVA-FFT i.e.

the feature vector. An alternative method was needed to determine the feature vector.

In the context of our experiments we consider the following facts:

1. The feature vector represents receiver weighting;
2. In PCA theory the dominant eigenvalue of the sample covariance matrix indicates the dominant eigenvector;
3. In PCA theory the dominant eigenvector is used as the feature vector; and
4. An eigenvalue is the magnitude of an eigenvector.

By deductive reasoning we conclude the following:

1. The elements of the dominant eigenvector of the sample covariance matrix, in a multiple receiver/antenna system, represents the relative performance of the receivers/antennas;
2. A dominant eigenvector will only present if the received signals have matching frequency and phase; and
3. If the appropriate feature vector can be deduced by using the FFT, then we have formed the dominant eigenvector.

In an attempt to optimise feature vector selection, a function was developed to performs automatic feature vector survey based on the normalised FFT of a block of the received signal. The function permits three options based on the inverse of the average normalised amplitude of each receivers spectral content, which is used as a proxy for signal power. The options are:

1. Weight the receiver data according to their relative signal power;
2. Use only the receiver with the highest signal power; or
3. Weight all receivers evenly.

A balanced effect would be obtained by setting all feature vector elements equal, however, this potentially under-utilises a particularly strong receiver. The strongest receiver might be selected to maximise the probability of detection. A compromise position is to use all receivers, but weight them in the feature vector according to their expected individual detection performance.

The self-survey function runs once at siting to perform a partial EVA-FFT on a block of each received signal. The mean of the normalised spectral amplitudes is calculated for each receiver and inverted. These values form the elements of the feature vector per item (1) in the list above. This option has been selected for EVA-FFT and is presented in all subsequent figures and discussion.

To ensure the results do not contain false positives in terms of P_d, baseline detection of the noise floor was attempted using captured signals from each RTL-SDR receiver at 3.2 MSPS. The results are shown in Figure 4.11 and establish that false detection does not occur under noise only conditions. It also establishes the worst case P_{fa} for EVA-FFT is 11.46%, which is within the limits established in Section 2.2.

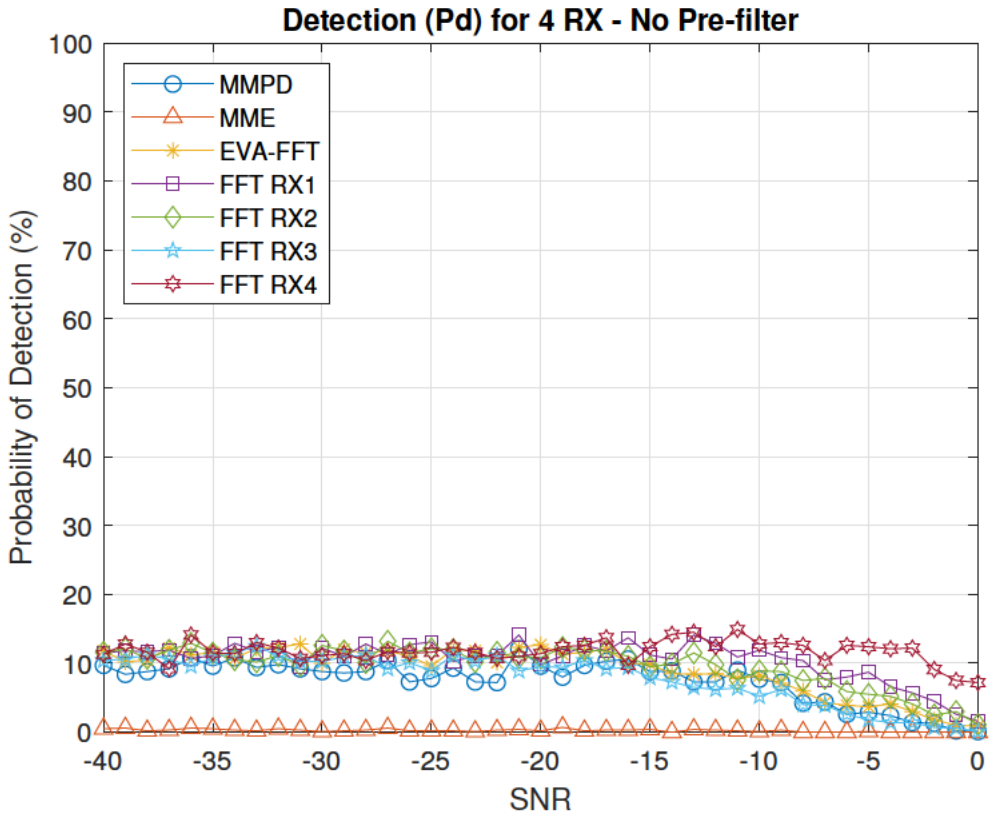

Figure 4.11: Detection for uncorrelated noise capture from 4 receivers.

4.3.1 Correlated Noise Vulnerability

EVA-FFT does not suffer from the same vulnerability to correlated noise that eigenvalue based detection schemes do. This is shown in Figure 4.12 where MMPD and MME have falsely detected active signals, while the FFT based methods have not.

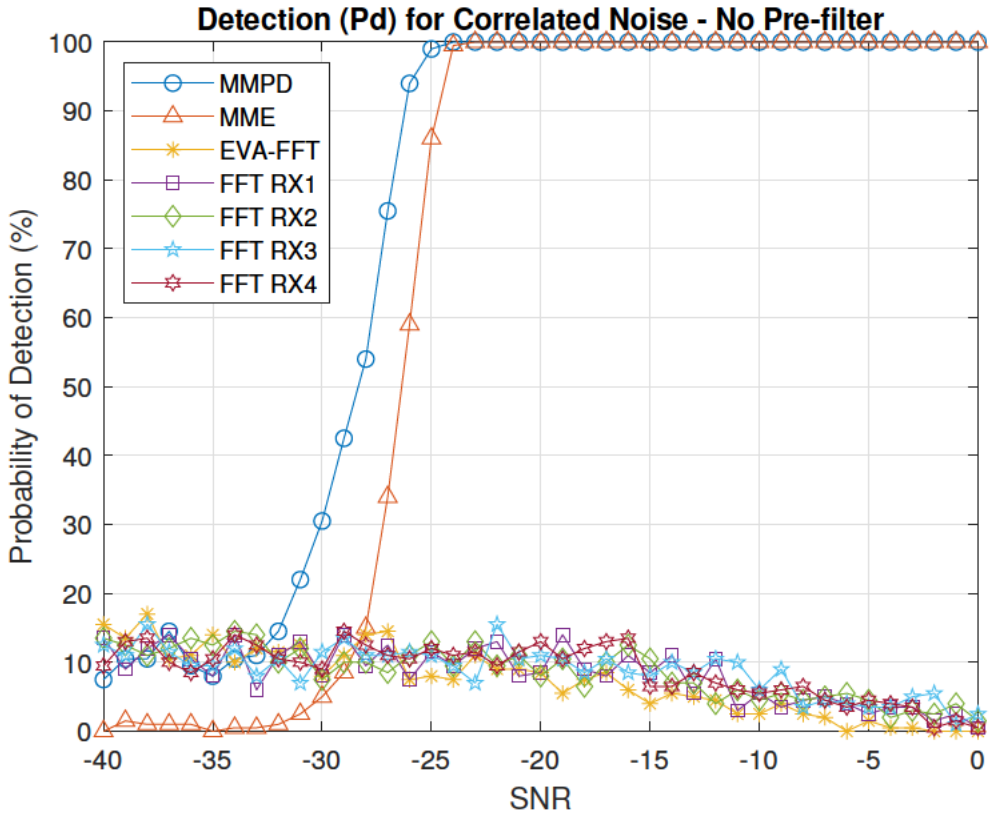

Figure 4.12: Detection for correlated noise without pre-filtering. MMPD and MME falsely detected active signals.

Band-pass pre-filtering introduces a higher amplitude in the band of interest when considering the frequency spectrum. As a result, pre-filtering still results in detection when no active signal is actually present, as shown in Figure 4.13. The remedy to this is the same as previously implemented for MMPD, being to use pre-filtering to calculate the appropriate AWGN levels for testing, but not to pre-filter the signals prior to applying the PCA transform.

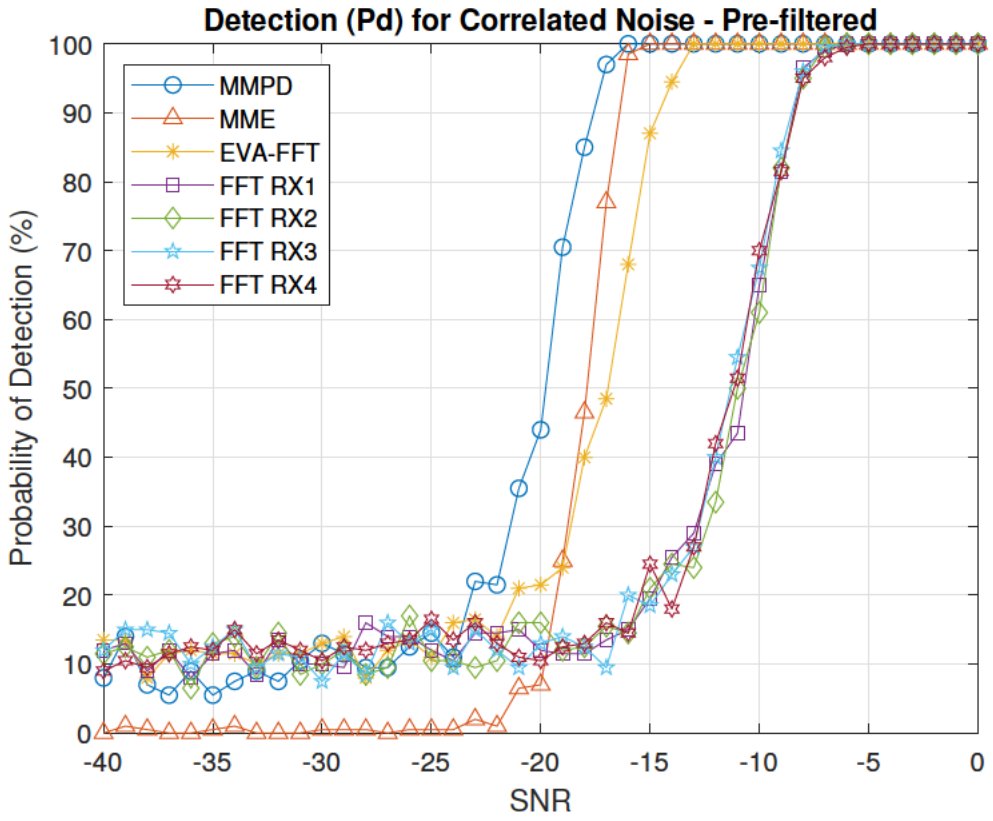

Figure 4.13: Detection for correlated noise with pre-filtering.

4.3.2 EVA-FFT Applied to FM Signals

The same signals were tested here as in Section 4.2.2, with the results given in Figure 4.14.

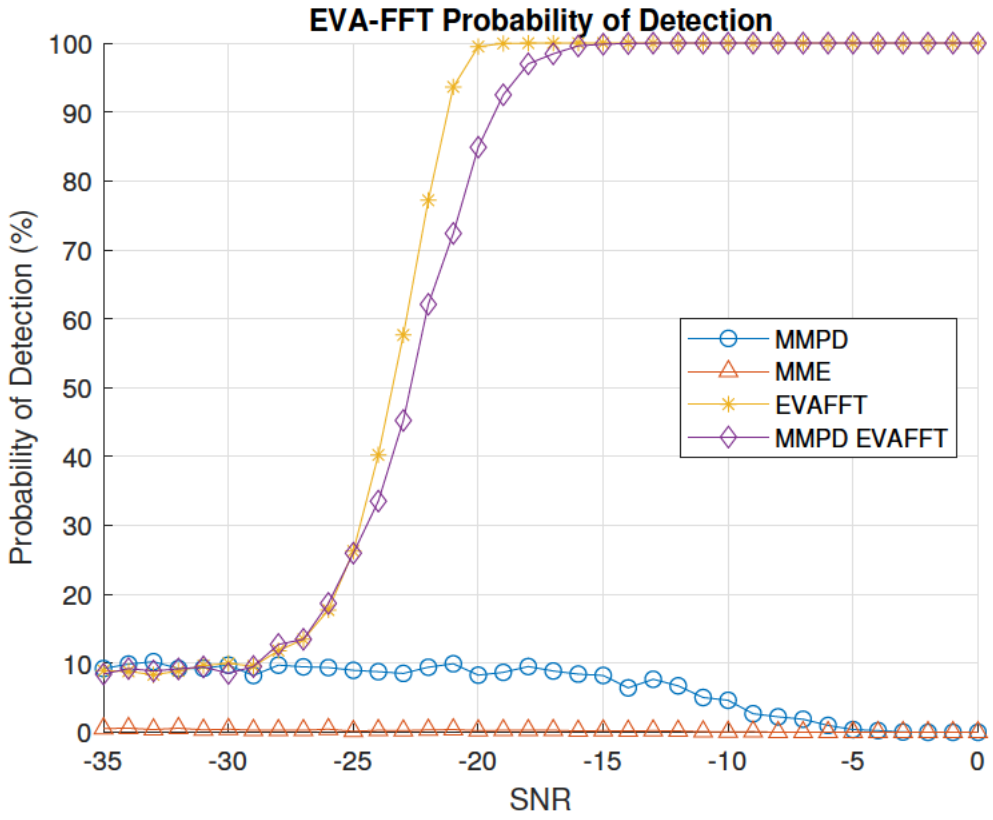

Figure 4.14: Probability of detection P_d vs SNR for EVA-FFT (FM 100.7 MHz i) - 4 RX.

Both EVA-FFT detection results in Figure 4.14 indicate detection of commercial FM signals at $SNR < 0$ dB. The results do not yet establish that the survey methods used to determine the feature vector provide any advantage in terms of detection performance for multiple receiver/antenna systems. Since the MMPD and MME detection remained stagnant, they are omitted from subsequent plots. Figure 4.15 shows the same FM signal capture, but includes the detection performance using the same FFT method for each receiver. Figure 4.16 shows the results for another FM capture, with the antennas moved to different positions. The results suggest that EVA-FFT and MMPD EVA-FFT perform comparably to individual receiver FFT detection when each receiver provides a comparable signal. This is because the automated survey routines recognise that each receiver has a comparably strong signal, thus weighting them approximately equally in the feature vector.

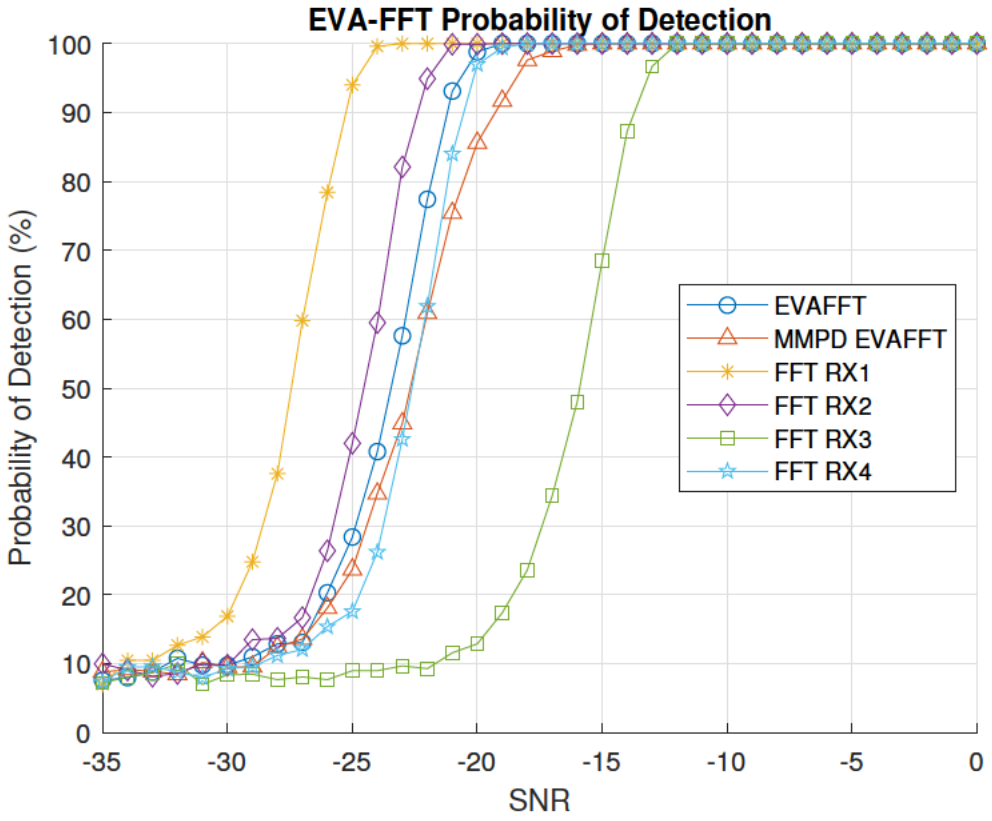

Figure 4.15: Probability of detection P_d vs SNR for EVA-FFT (FM 100.7 MHz i) - 4 RX.

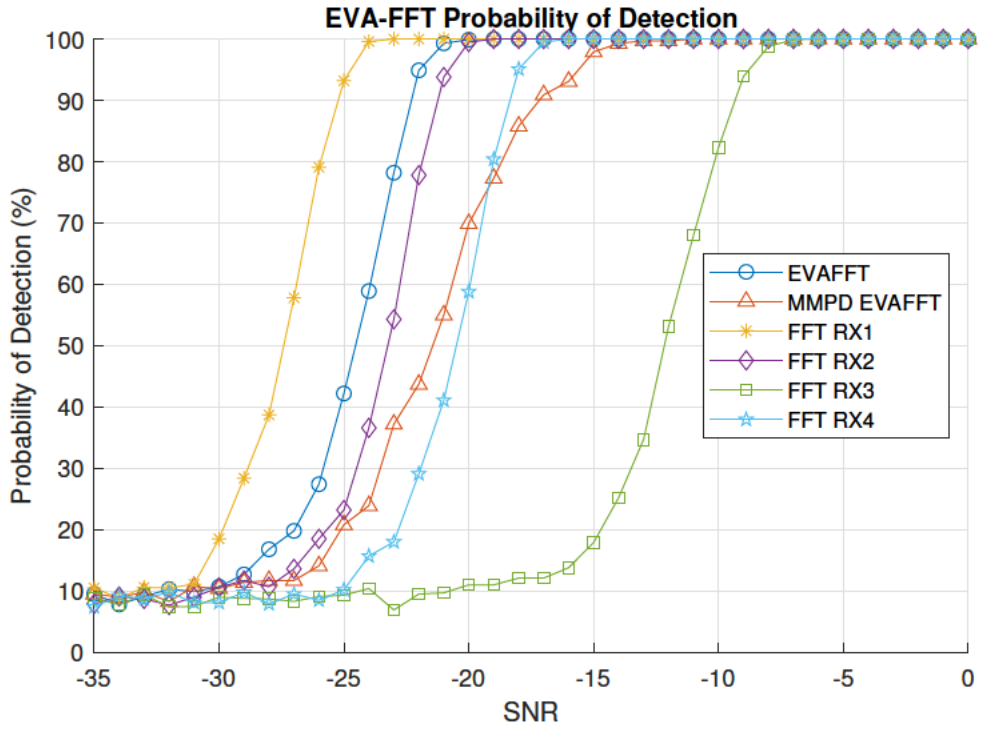

Figure 4.16: Probability of detection P_d vs SNR for EVA-FFT (FM 100.7 MHz ii) - 4 RX.

The real advantages of performing the PCA transform will only be seen when there is differential detection performance for individual receivers. For this reason, each received signal is intentionally degraded with AWGN to different SNR. Figure 4.17 shows the results. When considering this figure, the different probability of detection at different degradation levels is intentional, simulating some receivers collecting stronger signals than others or the received spectrum having a different peak geometry.

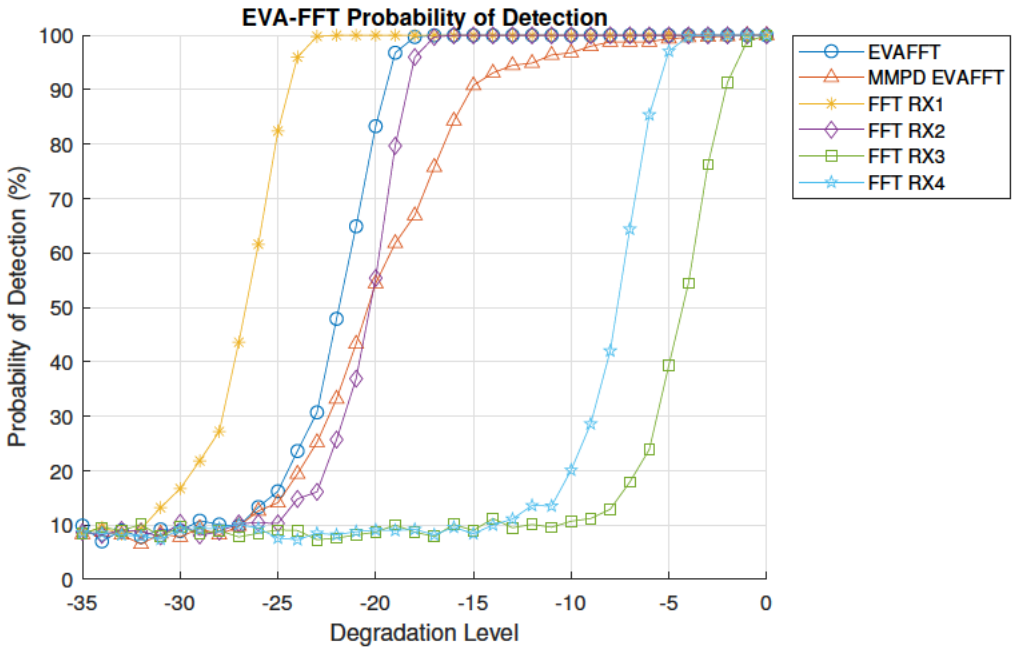

Figure 4.17: Probability of detection P_d for different RX performance (FM 100.7 MHz i) - 4 RX.

The results in Figure 4.17 show the advantage of EVA-FFT, with the 'relative strength' survey method providing better performance than most individual receivers, clearly favouring the receivers with stronger received signals. The MMPD survey also performs well, demonstrating that the eigenvalues of the sample covariance matrix of the received signal can assist in spectral peak detection schemes. It is noted that the automatic survey routine of EVA-FFT can be configured to select the 'best' receiver if desired.

If highly correlated signals are received (as assumed in eigenvalue based detection schemes) then EVA-FFT performs as shown in Figure 4.18, which was generated using a replicated signal from a single receiver. The uncorrelated noise tends to cancel, providing better detection performance than any individual receiver. If the problems of frequency and phase shift can be avoided, this is a compelling reason not to simply select the results from one receiver or antenna, but rather to weight them using the 'relative strength'

option.

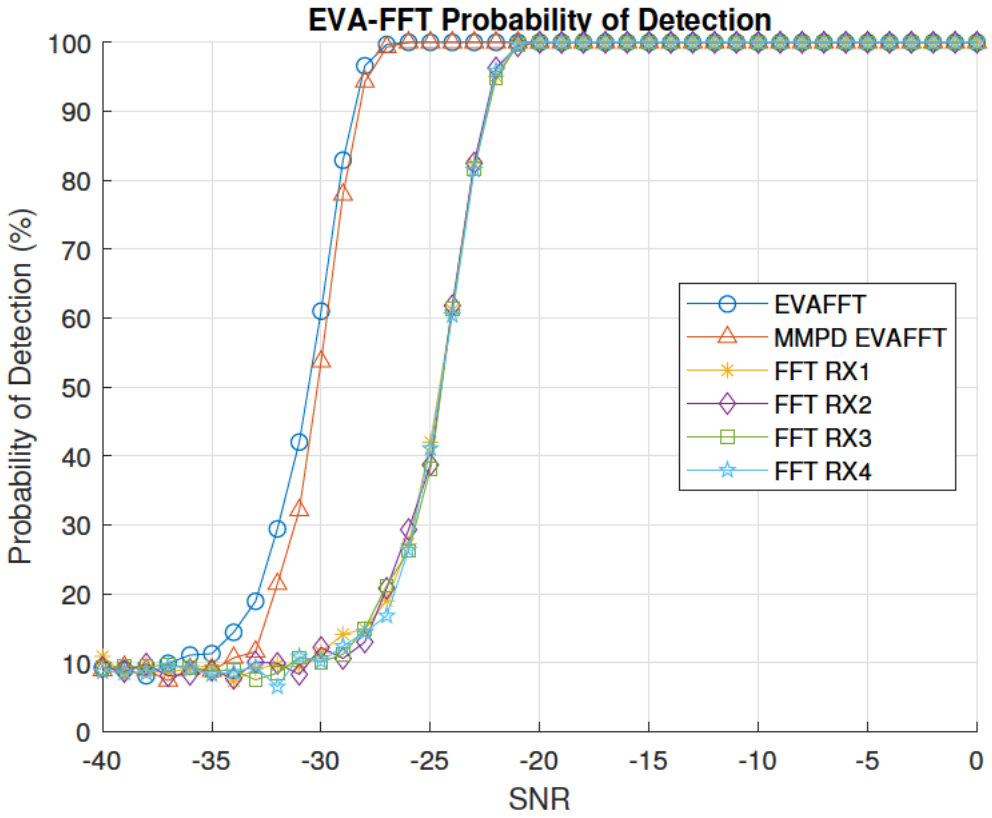

Figure 4.18: Probability of detection P_d for perfectly correlated data - 1 RX x4.

The code for this section is the same used in Section 4.2.1, and is listed at Appendix D.7.

4.3.3 EVA-FFT Applied to 802.11n WiFi

Four receiver signals were captured from RTL-SDR radios connected to four separate computer terminals. Each were set up with the same settings in the SDR# software, capturing at a center frequency of 2.42 GHz with a bandwidth of interest 250 kHz wide. The captured bandwidth is equal to half the sample rate. The results are shown in Figure 4.19 and again in 4.20 for adjusted antenna locations. The same tests were repeated with a center frequency of 2.44 GHz, and the results are given in Figure 4.21.

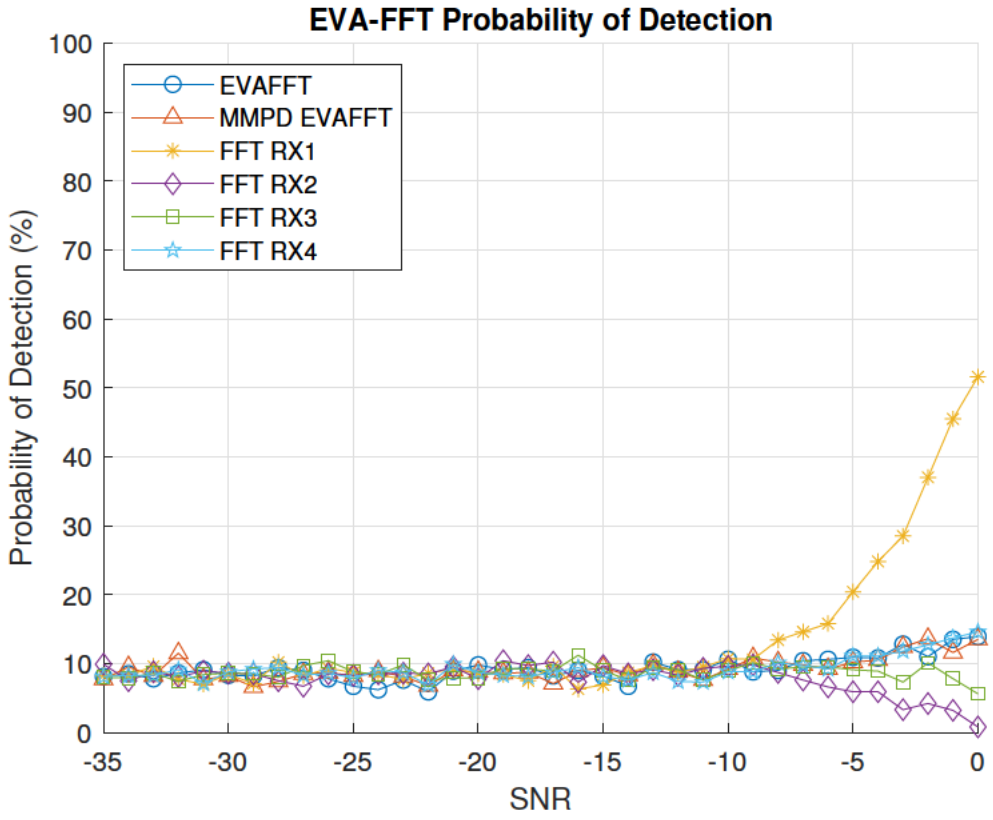

Figure 4.19: Probability of detection P_d vs SNR for EVA-FFT (WiFi 2.42 GHz i) - 4 RX.

The results show that EVA-FFT is ineffective for detecting spread spectrum signals. The live spectrum display in SDR# showed the band was flat but 'bouncing' slightly and contrast changes were seen in the waterfall display shown at Figure 4.23. If a clearly defined peak, such as a channel boundary can be captured, then EVA-FFT may be able to detect it. This is what occured in Figures 4.20 and 4.21 with RX2 and RX4, so the result is considered a false positive for spread spectrum detection. The unwanted peak is shown in Figure 4.22. MMPD and MME were also unable to detect the spread spectrum signals, and a brief theoretical explanation is given for this in the conclusions at Section 5.

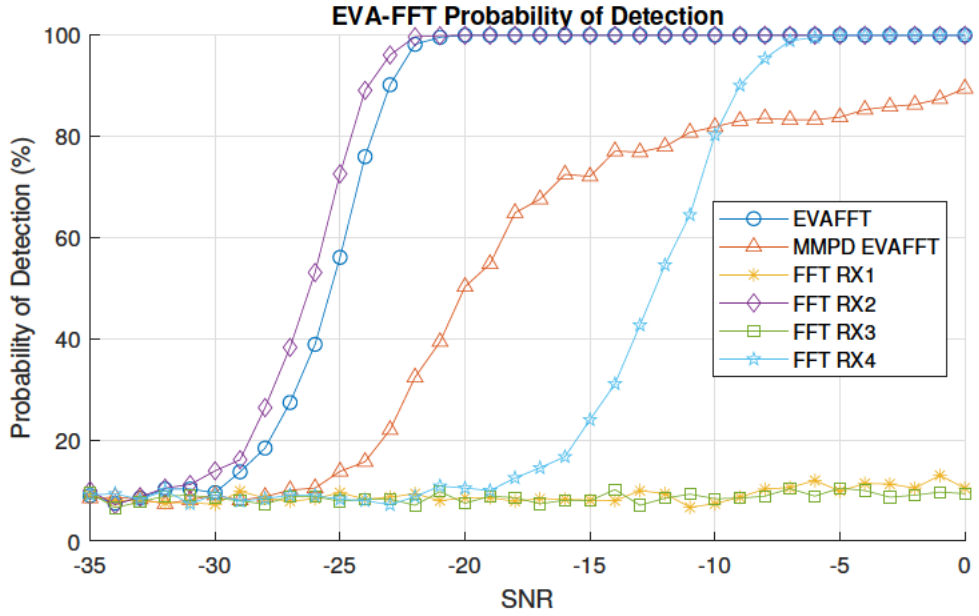

Figure 4.20: Probability of detection P_d vs SNR for EVA-FFT (WiFi 2.42 GHz ii) - 4 RX.

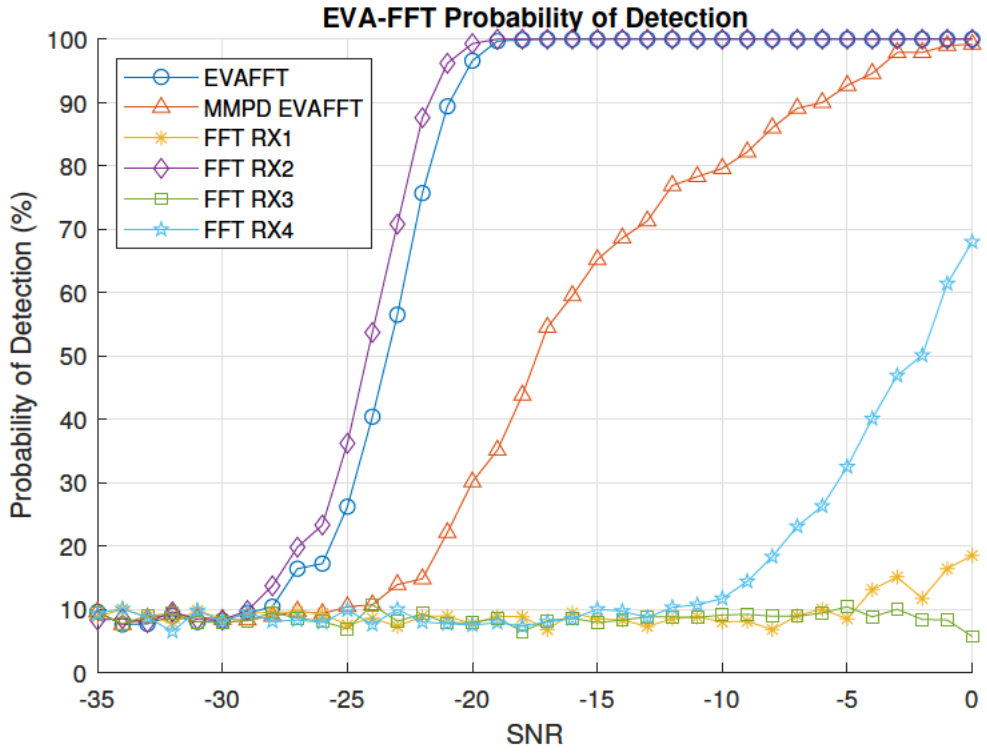

Figure 4.21: Probability of detection P_d vs SNR for EVA-FFT (WiFi 2.44 GHz) - 4 RX.

The code for this section is the same used in Section 4.2.1, and is listed at Appendix D.7.

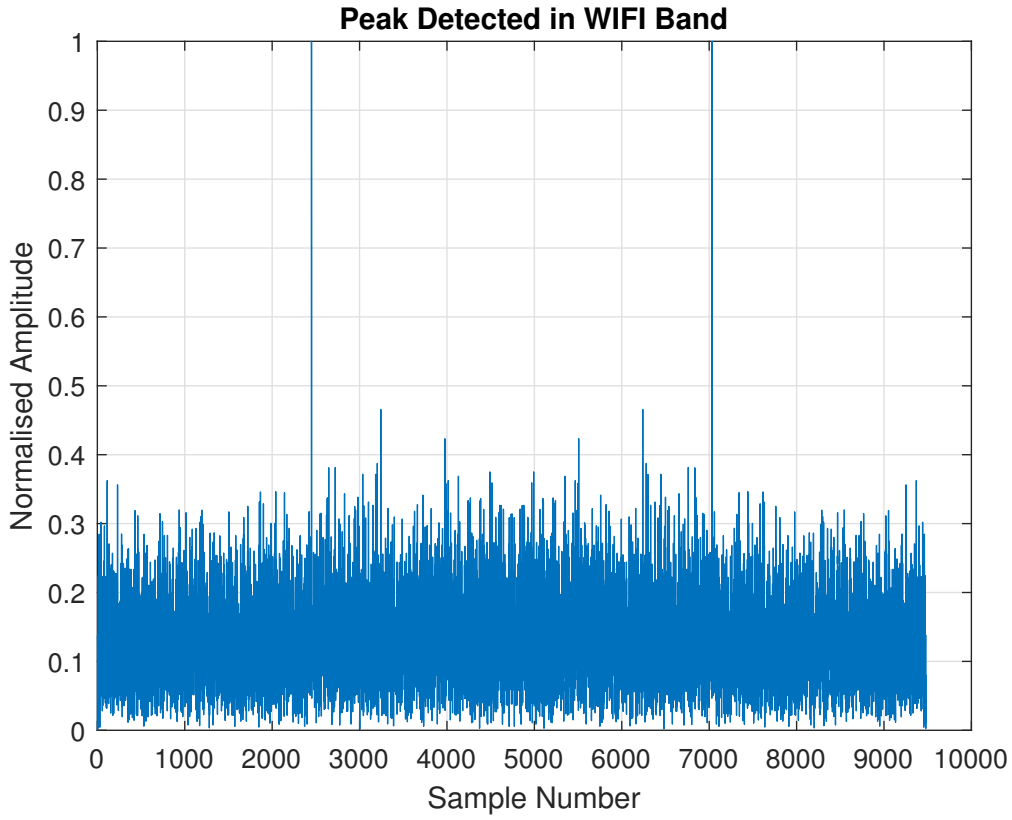

Figure 4.22: Unexpected spectral peak in WiFi data.

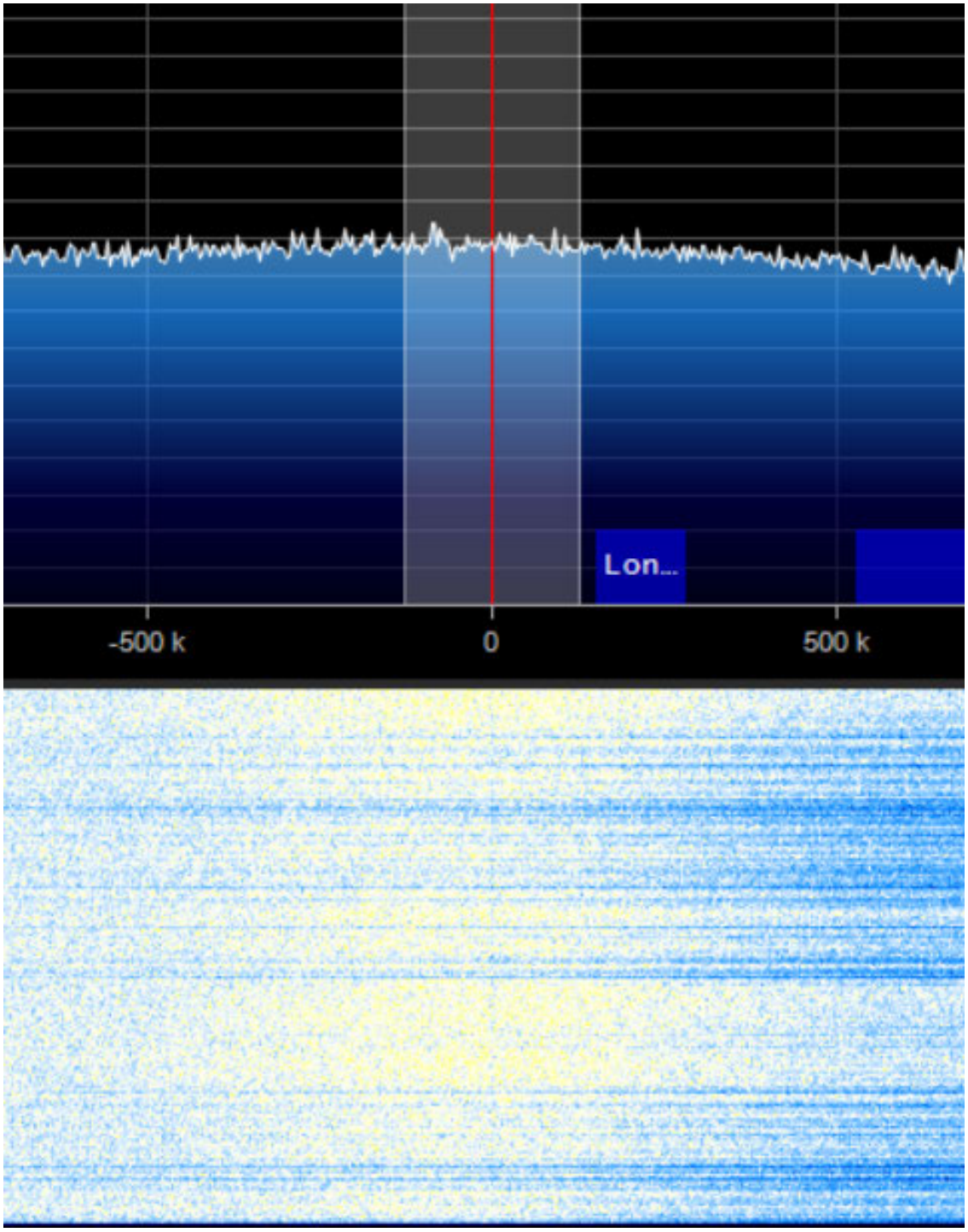

Figure 4.23: SDR# spectrum and waterfall for WiFi 2.42 GHz - RX1.

Chapter 5

Conclusions

5.1 General Summary

MMPD and MME performed well when received signals were perfectly correlated prior to AWGN addition. When separate receivers were used, however, these eigenvalue based detection schemes performed poorly. Nevertheless, these investigations informed the development of pre-processing and standardisation algorithms needed for development of a sample covariance matrix that is primed for PCA transform. The process of obtaining a dominant eigenvalue from the sample covariance matrix, and using it to select a feature vector constitutes a rapid survey of relative signal strengths between receivers. This permits the system to give preference to the best received signal(s) at any time.

This enabled the evolution of MMPD into EVA-FFT, which was optimised for P_{fa} of 10% and processes a defined number of samples as a block. Since meaningful eigenvalue variation was not able to be captured for the multiple receiver case, EVA-FFT was tested against FM signals from multiple receivers, and this established that EVA-FFT is superior to eigenvalue based detection in narrow-band applications. EVA-FFT was also tested against WiFi signals in the vicinity of 2.4 GHz, where the captured signal was difficult to visually distinguish from the noise floor. The small variations in the spectral content captured were not sufficient to present a dominant eigenvalue or associated eigenvector for MMPD or MME detection. EVA-FFT appeared to have detected the WiFi signals, but upon further inspection it had detected nearby peaks in the spectrum.

PCA provides a computationally efficient way to combine the received signals into a single data vector for application of the FFT, providing high detection performance even if most of the receivers do not capture noteworthy spectral peaks. This is because of the automatic surveying ability of the EVA-FFT algorithm, which allows rapid assessment of relative receiver/antenna detection performance.

A great deal of time was spent on MMPD during this project, and it is worth mentioning why it ultimately failed when attempted with multiple receivers.

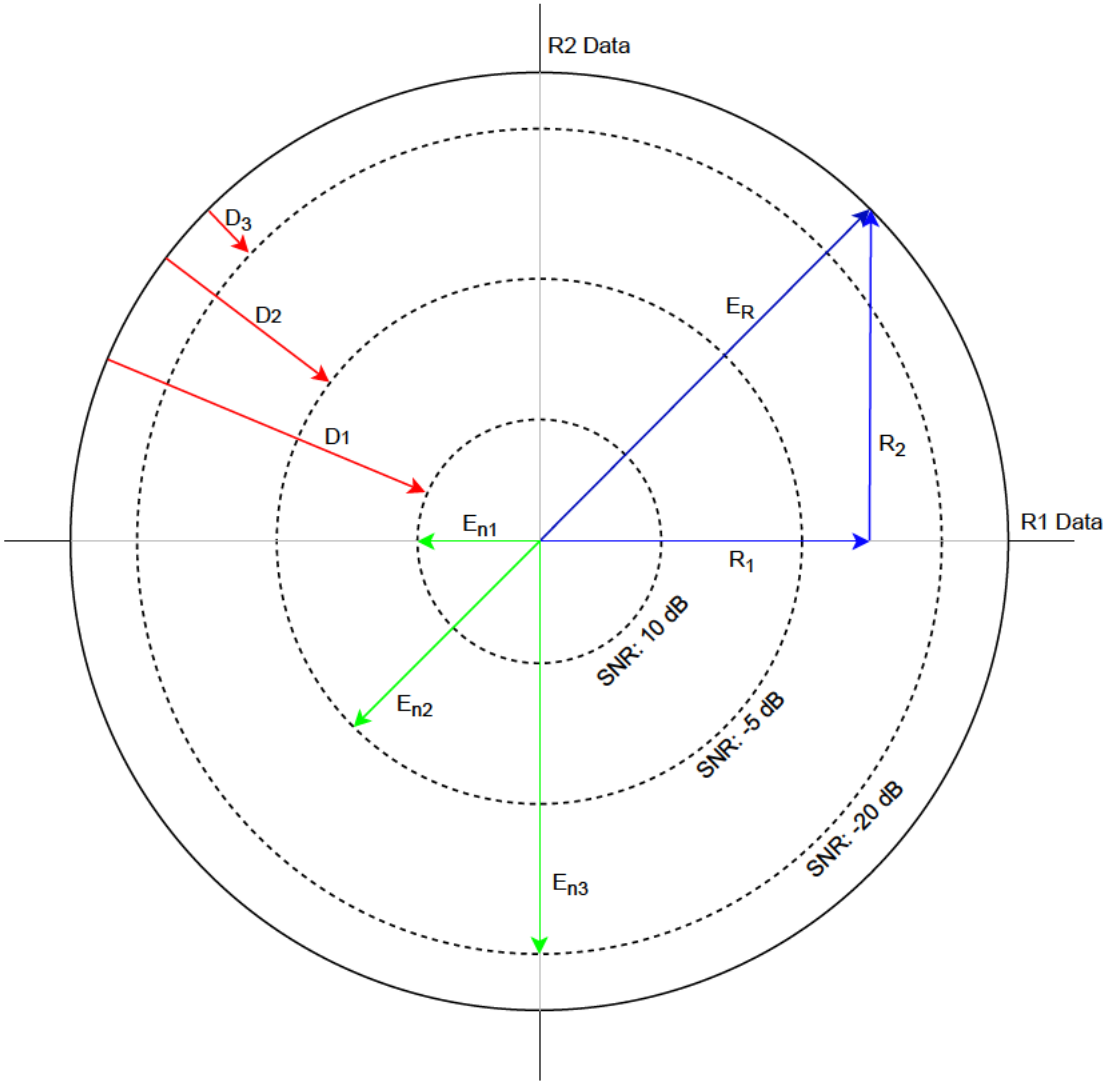

Figure 5.1: MMPD concept diagram showing the relationship between eigenvalue percentage difference, SNR and received signal correlation.

Figure 5.1 shows a concept diagram to assist the reader to understand MMPD. It uses a two receiver example so that discussion can be confined to the two dimensions available. The dimensions are not to scale so that the salient parts are visible.

The solid outer circle has a radius defined by the dominant eigenvector E_R, which arises due to received signals R_1 and R_2. These are the blue arrows. One can imagine the oscillating data points left and right for R_1, and up and down for R_2. If the receivers are synchronised in frequency and phase, and have comparable magnitudes, then the radius of the outer solid circle is maximum. That is, E_R is the dominant eigenvector whose magnitude is the dominant eigenvalue.

The green arrows represent the eigenvalues E_{n1}, E_{n2} and E_{n3}, which correspond to the noise levels 10 dB, -5 dB and -20 dB respectively. White Gaussian noise is distributed evenly, hence the dashed circles, and one can imagine the inner noise circle expanding outward toward the perimeter as more noise is added. Eventually, the noise circle becomes so large that it conceals the eigenvector E_R associated with the received signals. This is why the difference between maximum and minimum eigenvalues diminishes as SNR gets lower. Eventually all the eigenvalues become equal, as the 'noise circle' is approximately equal in all directions and hides the desired signal.

The percentage difference between the maximum and minimum eigenvalues is shown by the red arrows for different SNR. When SNR is high, the difference D_1 is large and easy to detect. When SNR is low, the difference D_3 is small and difficult to detect.

All the theoretical experiments and literature reviewed focused on what happens when the noise level changes i.e. the dashed circle expanding out toward the perimeter and concealing E_R. In the multiple receiver case, not only did the noise circle expand outward, but the solid outer circle also moved inward. This is because R_1 and R_2 could not be precisely synchronised in frequency and phase, resulting in a more circular and noise-like distribution when the sample covariance matrix was calculated.

5.2 Performance of Detection Schemes

P_d was collected for MMPD, EVA-FFT and MME detection schemes, where MME was the baseline. In Section 2.2 the following criteria were established for determining whether a detection algorithm was superior to MME:

1. P_d is higher than MME at -21 dB, and this difference is statistically significant when averaged over 10,000 iterations; and

2. P_d at low SNR (-30dB and below) is greater than MME, but remains within the range $8\% \leq P_{fa} \leq 12\%$. This permits a $\pm 2\%$ tolerance around the 10% threshold defined in Section 3.2.

It was noted that an average P_d difference greater than or equal to 5% at -21 dB will be deemed statistically significant *prima facie*, and will not be demonstrated through formal statistical testing.

MMPD is dependent on being able to detect some difference between the maximum and minimum eigenvalues of the sample covariance matrix of a set of received signals. As such it will work in any scenario where MME does. The best case performance was simulated using perfectly correlated synthesised and captured signals. In this context MMPD showed a $P_d \approx 30\%$ at -21 dB, while MME $P_d < 5\%$, satisfying criterion 1. The low SNR P_d represented the worst case P_{fa} of 9.96%, satisfying criterion 2. By these metrics MMPD is deemed superior to MME *prima facie* in theory only, since the theoretical performance could not be validated using multiple receivers.

EVA-FFT relies on a feature vector for the band of interest, surveyed prior to detection. It evolved from MMPD and uses the same pre-filtering and standardisation routines, however, does not require a meaningful difference in eigenvalues of the sample covariance matrix. Instead, EVA-FFT aggregates data from multiple receivers and transforms it using the surveyed feature vector, permitting multiple received signals to be combined into a single vector. Considering FM signals, EVA-FFT demonstrated $P_d > 90\%$ at -21 dB, while MME failed to exceed 2%. Even the theoretical maximum MME P_d of 5% at -21 dB is significantly less than EVA-FFT. Criterion 1 is therefore satisfied for EVA-FFT. P_{fa} for EVA-FFT was 11.46%, satisfying criterion 2. By these metrics EVA-FFT is deemed superior to MME *prima facie*.

MMPD EVA-FFT did not perform an FFT based survey to obtain receiver weightings, but rather used the dominant eigenvector of the sample covariance matrix for weighting. P_d for FM signals exceeded 50% at -21 dB, also satisfying criterion 1. P_{fa} was also 11.46%, satifying criterion 2. By these metrics MMPD EVA-FFT is deemed superior to MME *prima facie*.

5.3 Questions Revisited and Further Research

The following questions were raised at the end of the literature review at Section 1.3:

1. Is there a limiting SNR for eigenvalue/eigenvector based detection schemes, and what are the driving factors?

   (a) Yes, the limiting SNR are defined as SNR_{co} in table 3.1. The most influential factors appear to be antenna count and number of cycles captured.

2. Can the relationship between eigenvalues serve as a proxy for SNR, and therefore support Boolean detection?

   (a) Theoretically, yes. This relies on the same theory underpinning the MME scheme which is widely accepted as a baseline detection method in the literature. Chapter 3 details the relationship between eigenvalues and SNR at length. This project was unable to validate this theory through hardware experiments. Further work is needed in this area.

3. Can PCA be applied to signals captured from different receivers in the same band, in order to decide signal presence without performing FFT on all received signals?

   (a) Yes. This was established concretely for FM signals as shown in Figure 4.14, however, negative results were obtained for spread spectrum signals as shown in Figures 4.19, 4.20 and 4.21.

4. Can an eigenvalue informed approach to FFT threshold setting be used to maximise detection accuracy in a multi-antenna, multi-channel context?

   (a) Theoretically, yes. This is demonstrated in principle in Section 3.3.1, but this project was unable to validate the theory through hardware experiments. Further work is needed in this area.

5. Are the above items applicable to the Australian context described in Section 1.1?

   (a) Yes, the Australian context does not change the fact that more available bandwidth facilitates higher data transfer rates. The vast land area of the country means that regional locations serviced by NBN fixed wireless could make use of spectral holes when licenced users are not occupying particular bands. The

above-listed items highlight some potential techniques to aid detection of spectral holes.

References

Almalfouh, S. M. & Stuber, G. L. (2011), 'Interference-aware radio resource allocation in ofdma-based cognitive radio networks', *IEEE transactions on vehicular technology* **60**(4), 1699–1713.

Australian Business Register (2021), 'Current details for abn 86 136 533 741', https://abr.business.gov.au/ABN/View/86%20136%20533%20741. [Online; accessed March-2021].

Australian Competition & Consumer Commission (n.d.), 'Broadband speeds', https://www.accc.gov.au/consumers/internet-landline-services/broadband-speeds#accc-enforcement-action-on-broadband-speeds. [Online; accessed March-2021].

Australian Competition & Consumer Commission (n.d.*a*), 'National broadband network (nbn)', https://www.accc.gov.au/consumers/internet-landline-services/national-broadband-network-nbn#different-nbn-broadband-technologies. [Online; accessed March-2021].

Australian Competition & Consumer Commission (n.d.*b*), 'Using nbn fixed wireless', https://www.accc.gov.au/consumers/internet-landline-services/broadband-speeds/using-nbn-fixed-wireless#more-information. [Online; accessed April-2021].

Bhatti, F. A., da Silva, C. R. C. M., Rowe, G. B. & Sowerby, K. W. (2014), On the use of eigenvectors for signal detection and classification in multiple antenna cognitive radios, *in* '2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)', IEEE, pp. 753–757.

Bouallegue, K., Dayoub, I., Gharbi, M. & Hassan, K. (2018), 'Blind spectrum sensing

using extreme eigenvalues for cognitive radio networks', *IEEE communications letters* **22**(7), 1386–1389.

Chen, X., Chen, H.-H. & Meng, W. (2014), 'Cooperative communications for cognitive radio networks - from theory to applications', *IEEE Communications surveys and tutorials* **16**(3), 1180–1192.

Communications Alliance (n.d.), 'NBN project history', https://commsalliance.com.au/Activities/nbn/history. [Online; accessed March-2021].

Conifer, D. (2016), 'Federal election: Nbn promises past and present explained, abc', https://www.abc.net.au/news/2016-06-13/federal-election-nbn-promises-past-and-present/7506714. [Online; accessed March-2021].

Conifer, D. (2020), 'Federal election: Nbn promises past and present explained, the canberra times', https://www.canberratimes.com.au/story/6978625/nbn-rollout-network-upkeep-at-57-billion/. [Online; accessed March-2021].

Dickens, M., Laneman, J. N. & Dunn, B. P. (2012), 'Seamless dynamic runtime reconfiguration in a software-defined radio', *Journal of signal processing systems* **69**(1), 87–94.

Egan, M. (2008), 'A separate company for a broadband network', https://www.theage.com.au/business/a-separate-company-for-a-broadband-network-20080904-49zt.html. [Online; accessed March-2021].

Fernando, X. (2019), *Cooperative Spectrum Sensing and Resource Allocation Strategies in Cognitive Radio Networks*, SpringerBriefs in Electrical and Computer Engineering, 1st ed. 2019. edn, Springer International Publishing, Cham.

Fisne, A. & Ozsoy, A. (2018), 'Design and implementation of real-time wideband software-defined radio applications with gpgpus', *Concurrency and computation* **30**(21), e4791–n/a.

Guo, Y. J. (2010), Keynote speech 4: Cognitive radio and australia's national broadband network, *in* '2010 10th International Symposium on Communications and Information Technologies', IEEE, pp. XXI–XXII.

Gupta, N. & Dhurandher, S. K. (2020), 'Cross-layer perspective for channel assignment in cognitive radio networks: A survey', *International journal of communication systems* **33**(5), e4261–n/a.

He, K., Crockett, L. & Stewart, R. (2012), 'Dynamic reconfiguration technologies based on fpga in software defined radio system', *Journal of signal processing systems* **69**(1), 75–85.

He, Y. & Dey, S. (2011), 'Power allocation in spectrum sharing cognitive radio networks with quantized channel information', *IEEE transactions on communications* **59**(6), 1644–1656.

He, Y. & Dey, S. (2012), 'Throughput maximization in cognitive radio under peak interference constraints with limited feedback', *IEEE transactions on vehicular technology* **61**(3), 1287–1305.

IEEE Std 802.22-2011 (2011), IEEE.

Jiang, Z., Yuan, W., Leung, H., You, X. & Zheng, Q. (2017), 'Coalition formation and spectrum sharing of cooperative spectrum sensing participants', *IEEE transactions on cybernetics* **47**(5), 1133–1146.

Jondral, F., Elsner, J. & Schwall, M. (2012), 'Software defined radio - guest editorial', *Journal of signal processing systems* **69**(1), 1–3.

Letaief, K. B. & Zhang, W. (2009), 'Cooperative communications for cognitive radio networks', *Proceedings of the IEEE* **97**(5), 878–893.

Li, D., Fang, E. & Gross, J. (2018), 'Robust clustering for ad hoc cognitive radio network', *Transactions on emerging telecommunications technologies* **29**(5), e3285–n/a.

Liu, C., Li, H., Wang, J. & Jin, M. (2017), 'Optimal eigenvalue weighting detection for multi-antenna cognitive radio networks', *IEEE transactions on wireless communications* **16**(4), 2083–2096.

Lopez-Benitez, M. & Casadevall, F. (2013), 'Time-dimension models of spectrum usage for the analysis, design, and simulation of cognitive radio networks', *IEEE transactions on vehicular technology* **62**(5), 2091–2104.

Mashta, F., Altabban, W. & Wainakh, M. (2020), 'Two-stage spectrum sensing for cognitive radio using eigenvalues detection', *International journal of interdisciplinary telecommunications and networking* **12**(4), 18–36.

Muthukkumar, R. & Manimegalai, D. (2017), 'Enhanced cooperative spectrum sensing in crahns using distributed dynamic load-balanced clustering scheme', *Wireless personal communications* **94**(4), 2513–2531.

NBN Co Limited (2020*a*), 'nbn hfc made easy', https://www.nbnco.com.au/learn/network-technology/hybrid-fibre-coaxial-explained-hfc-3. [Online; accessed March-2021].

NBN Co Limited (2020*b*), 'Nbn launches three new residential wholesale higher speed tiers', https://www.nbnco.com.au/corporate-information/media-centre/media-statements/nbn-launches-three-new-residential-wholesale-higher-speed-tiers. [Online; accessed May-2021].

Razavi, B. (2010), 'Cognitive radio design challenges and techniques', *IEEE journal of solid-state circuits* **45**(8), 1542–1553.

Regula, W. M., Gilbert, J. M. L. & Sheikh, W. A. (2020), 'Dynamic wireless spectrum access using gnu radio and software-defined radios', *International journal of communication systems* **33**(4), e4233–n/a.

Rudd, K. (2009), 'New national broadband network', https://parlinfo.aph.gov.au/parlInfo/download/media/pressrel/PS8T6/upload_binary/ps8t60.pdf;fileType=application%2Fpdf#search=%22media/pressrel/PS8T6%22. media release, [Online; accessed March-2021].

Sharma, S. K., Chatzinotas, S. & Ottersten, B. (2013), 'Eigenvalue-based sensing and snr estimation for cognitive radio in presence of noise correlation', *IEEE transactions on vehicular technology* **62**(8), 3671–3684.

Sun, C., Zhang, W. & Letaief, K. B. (2007), Cluster-based cooperative spectrum sensing in cognitive radio systems, *in* '2007 IEEE International Conference on Communications', IEEE, pp. 2511–2515.

Webb, W., Schwall, M., Elsner, J. & Jondral, F. K. (2012), 'Dynamic spectrum access is the solution: What's the problem?', *Journal of signal processing systems* **69**(1), 5–9.

Wen, Z., Tang, M.-C. & Ziolkowski, R. W. (2019), 'Band- and frequency-reconfigurable circularly polarised filtenna for cognitive radio applications', *IET microwaves, antennas propagation* **13**(7), 1003–1008.

Yu, R., Xue, G., Bennis, M., Chen, X. & Han, Z. (2018), 'Hsdran: Hierarchical software-defined radio access network for distributed optimization', *IEEE transactions on vehicular technology* **67**(9), 8623–8636.

Zeng, Y. & Liang, Y.-C. (2009), 'Eigenvalue-based spectrum sensing algorithms for cognitive radio', *IEEE transactions on communications* **57**(6), 1784–1793.

Zhang, R., Lim, T., Liang, Y.-C. & Zeng, Y. (2010), 'Multi-antenna based spectrum sensing for cognitive radios: A glrt approach', *IEEE transactions on communications* **58**(1), 84–88.

Zhu, D. & Pan, S. (2020), 'Broadband cognitive radio enabled by photonics', *Journal of lightwave technology* **38**(12), 3076–3088.

Appendix A

Project Specification

ENG4111 Research Project
Project Specification

For: Mr Benedict Samuel Hardless[redacted]
Title: Cognitive Radio for Improved NBN Access in Regional Areas
Major: Electrical and Electronic Engineering
Supervisors: Dr John Leis
Enrollment: ENG4111 – EXT S1, 2021
Project Aim: To investigate cognitive radio systems and develop a computer model that simulates deployment in a regional setting.

Programme: Version 1, 13th March 2021

1. Conduct initial background research on cognitive radio fundamentals. Develop an understanding of regional broadband arrangements both in Australia and abroad.
2. Research licensing and RF spectral allocation, as well as channel assignment and typical power levels in Australia.
3. Review signal processing and control system fundamentals to support programming efforts.
4. Develop a preliminary concept design for a regional cognitive radio installation and what types of information would need to be exchanged between nodes. Acquire Software Defined Radios and appropriate peripherals in parallel.
5. Develop initial object layout in simulation environment (MATLAB), including the key properties and methods needed to describe those objects and their functions. Mathematical operations should be described in tandem as this phase progresses.
6. Develop a plan to target, capture and manage information from the simulation environment to support dissertation construction. Use these to establish SDR parameters of interest and performance metrics.
7. Define object interactions in the simulation environment, test, evaluate and refine approach repeatedly in an iterative development cycle. Develop understanding of SDR set up and deployment in parallel and attempt experiments.
8. Process and evaluate experimental data.
9. Quantify computational burden and evaluate how this might affect system costs compared to existing regional broadband solutions.

If time and resource permit:

10. Develop more sophisticated algorithms for the management of node interaction, spectral hole identification, channel selection and the temporal and spatial effects of channel utility.
11. Generalise algorithms to better cope with additional nodes at various locations.
12. Develop recommended rules for the proper spatial dispersion of nodes and transceiver operations in a regional setting.

Appendix B

Risk Assessment

University of Southern Queensland

USQ Safety Risk Management System

Print View

Version 2 0

Safety Risk Management Plan

Risk Management Plan ID:	Status:	Current User:	Author:	Supervisor:	Approver:
RMP_2021_5698	*Approve*	[redacted]	[redacted]	[redacted]	[redacted]

Assessment Title:	Engineering Research Project 2021 - Hardless	Assessment Date:	21/06/2021
Workplace (Division/Faculty/Section):	204010 - Faculty of Health, Engineering and Sciences	Review Date:	21/06/2025 (5 years maximum)

Approver:	*Supervisor: (for notification of Risk Assessment only)*
John Leis	*John Leis*

Context

DESCRIPTION:

What is the task/event/purchase/project/procedure?	Engineering Research Project 2021 - Hardless
Why is it being conducted?	Engineering Research Project 2021
Where is it being conducted?	Residential premises
Course code (if applicable)	ENG4111/4112
Chemical Name (if applicable)	

WHAT ARE THE NOMINAL CONDITIONS?

Personnel involved	Benedict Hardless
Equipment	Computers, software, office equipment
Environment	Domestic home.
Other	
Briefly explain the procedure/process	Office work at a computer and experiments with low power radios.

Assessment Team - who is conducting the assessment?

Assessor(s):	Benedict Hardless
Others consulted: (eg elected health and safety representative, other personnel exposed to risks)	

Risk Matrix

Probability	Consequence				
	Insignificant No Injury 0-$5K	Minor First Aid $5K-$50K	Moderate Med Treatment $50K-$100K	Major Serious Injury $100K-$250K	Catastrophic Death More than $250K
Almost Certain 1 in 2	M	H	E	E	E
Likely 1 in 100	M	H	H	E	E
Possible 1 in 1,000	L	M	H	H	H
Unlikely 1 in 10,000	L	L	M	M	M
Rare 1 in 1,000,000	L	L	L	L	L

Recommended Action Guide	
Extreme:	**E**= Extreme Risk – Task ***MUST NOT*** proceed
High:	**H** = High Risk – Special Procedures Required (Contact USQSafe) Approval by VC only
Medium:	**M**= Medium Risk - A Risk Management Plan/Safe Work Method Statement is required
Low:	**L**= Low Risk - Manage by routine procedures.

Risk Register and Analysis

Step 1	Step 2	Step 2a	Step 2b	Step 3	Step 4

Risk Register and Analysis

	Hazards: From step 1 or more if identified Step 1	The Risk: What can happen if exposed to the hazard without existing controls in place? Step 2	Consequence: What is the harm that can be caused by the hazard without existing controls in place? Step 2b	Existing Controls: What are the existing controls that are already in place? Step 2b	Risk Assessment: Consequence x Probability = Risk Level Step 3			Additional Controls: Enter additional controls if required to reduce the risk level	Risk assessment with additional controls: Step 4 Has the consequence or probability changed?			
					Probability	Risk Level	ALARP		Consequence	Probability	Risk Level	ALARP
	Example Working in temperatures over 35^{0} C	Heat stress/heat stroke/exhaustion leading to serious personal injury/death	catastrophic	Regular breaks, chilled water available, loose clothing, fatigue management policy.	possible	high	No	temporary shade shelters, essential tasks only, close supervision, buddy system	catastrophic	unlikely	mod	Yes
1	Computer vir...	Total inability to complete the project.	Catastrophic	Anti-virus software	Rare	Low	☑					☐
2	Office ergon...	physical injury requiring non-emergency medical intervention	Moderate	Adjustable chair, monitors and keyboard.	Rare	Low	☑					☐
3	Data loss	Total inability to complete the project.	Catastrophic	Nil	Almost	Extr...	☐	Cloud stored data, USB backup each time project work is done.	Minor	Unlikely	Low	☑
4	Scope creep ...	delay less than one week but greater than one day, still permitting completion to a passable standard.	Moderate	Nil	Possible	High	☐	Communicate regularly with project supervisor, with a focus on defined timelines. Clearly defined objectives and methodology.	Minor	Rare	Low	☑
5	Unable to pr...	one month or greater delay to project progress, but still permits completion to a passable standard.	Major	Nil	Possible	High	☐	Order early (done 30-May-21). Download real signals rather than capturing with SDR.	Major	Rare	Low	☑
6	Computer fai...	Total inability to complete the project.	Catastrophic	Nil	Unlikely	Me...	☐	Obtain backup computer with MATLAB software and access to cloud storage.	Insignifica	Rare	Low	☑
7	Estimating/s...	one month or greater delay to project progress, but still permits completion to a passable standard.	Major	Nil	Possible	High	☐	Use conservative estimates and don't try to do too much. Stick to the core objectives and track key dates in calendar.	Minor	Unlikely	Low	☑
8	Unforeseen t...	one month or greater delay to project progress, but still permits completion to a passable standard.	Major	Nil	Likely	Extr...	☐	Detailed planning and regular communication with supervisor.	Minor	Unlikely	Low	☑
9	Extreme wea...	Total inability to complete the project.	Catastrophic	Nil	Rare	Low	☐	Consult with university and seek support regarding events that are likely to affect large numbers of students.	Moderate	Rare	Low	☑

Step 5 - Action Plan (for controls not already in place)

	Additional Controls:	Exclude from Action Plan: (repeated control)	Resources:	Persons Responsible:	Proposed Implementation Date:
3	Cloud stored data, USB backup each time project work is done.	☐	Google drive, USB storage device.	Benedict Hardless	21/06/2021
4	Communicate regularly with project supervisor, with a focus on defined timelines. Clearly defined objectives and methodology.	☐	Email.	Benedict Hardless	21/06/2021
5	Order early (done 30-May-21). Download real signals rather than capturing with SDR.	☐	Internet.	Benedict Hardless	21/06/2021
6	Obtain backup computer with MATLAB software and access to cloud storage.	☐	Google drive, spare computer.	Benedict Hardless	21/06/2021
7	Use conservative estimates and don't try to do too much. Stick to the core objectives and track key dates in calendar.	☐	MS office.	Benedict Hardless	21/06/2021
8	Detailed planning and regular communication with supervisor.	☐	MS office and email.	Benedict Hardless	21/06/2021
9	Consult with university and seek support regarding events that are likely to affect large numbers of students.	☐	Email and telephone.	Benedict Hardless	21/06/2021

Supporting Attachments

Supporting Attachments

No file attached

Step 6 – Request Approval

Drafters Name:	*Benedict Hardless*	*Draft Date:*	*21/06/2021*
Drafters Comments:			

Assessment Approval: **All risks are marked as ALARP** 0

Maximum Residual Risk Level **Low - Manager/Supervisor Approval Required** 1

Document Status:	*Approve*

Step 6 – Approval

Approvers Name:	*John Leis*	*Approvers Position Title:*	*Project Supervisor*
Approvers Comments:	*very comprehensive. covers all forseeable risks and has appropriate plan to deal with eventualities.*		

I am satisfied that the risks are as low as reasonably practicable and that the resources required will be provided.

Approval Decision: Approve	*Approve / Reject Date:* *21/06/2021*	*Document Status:* *Approve*

Appendix C

Ethical Clearance

There are no Ethical Clearances applicable to this project.

Appendix D

Code Listings

D.1 Code Listing 1

```
%{
AUTHOR:     Benedict Hardless
CREATED:    25-May-2021
MODIFIED:   May-21, Jun-21
TESTED:     Jun-21
FILE NAME:  PDtoSNR.m
PURPOSE:    Based on the arguments, simulates a periodic signal with
            varying SNR, then examines the eigenvalue relationship to SNR.
            It also determines a lower limit for determining SNR from the
            eigenvalues.
ARGS:       NUM_CYCLES is the number of cycles define how many cycles are
                simulated as captured by the receiver, used for calculating
                the eigenvalues.
            NUM_ANT is the number of co-located antenna elements used for
                capture, used for calculating the covariance signal matrix.
RETURNS:    p is a polynomial to infer SNR from max/min eigenvalue
                percentage difference.
            SNRcutoff is the lowest possible inferrable SNR from the
                max/min eigenvalue percentage difference.
            SNRdegrade is 1.5 times SNRcutoff.
            pdCutoff is the max/min eigenvalue percentage difference
                corresponding to SNRcutoff.
            pdMax is the maximum calculated percentage difference between
                max/min eigenvalues. It is used for developing the
                logistic growth model in the dissertation.
%}
function [p,SNRdegrade,SNRcutoff, pdCutoff, pdMax] = ...
    PDtoSNR(NUM_CYCLES,NUM_ANT)

    f = [1E6];                          % Define frequencies to consider
    w = 2*pi*f;                         % Radian frequencies
    % Use different number of cycles for capture, so we can compare how the
    % max/min eigenvalue relationship changes. Capturing less cycles makes
    % eigenvalue based signal/noise discrimination less effective at ...
        low SNR.
    SNR_ITVL = 0.1;
    SNR = 10:-SNR_ITVL:-40;         % SNR levels to map to eigval % diff
    Fs = f*10;                      % Sampling frequency
    T = 1/Fs;                       % Sampling interval
```

```
38      DUR = NUM_CYCLES/f;             % Duration for the number of cycles
39      n = 1:Fs*DUR;                   % Vector of sample numbers
40      t = (n-1)*T*1000;               % Vector of times in ms
41      N = length(n);                  % Number of samples
42      w0 = 2*pi*f/Fs;                 % Radians per sample
43      X = sin(w0*n);                  % Locally generated signal
44      Ps = sum(X.^2)/length(X);       % Signal power
45      An = sqrt(Ps*10.^-(SNR/10));     % Noise scaling amplitudes to get SNRs
46      for i_snr = 1:length(SNR)
47          for i = 1:NUM_ANT
48             noise = An(i_snr)*randn(1,length(X(1,:)));   % Generate noise
49             x(i,:) = X + noise;                   % Generate noisy signal
50          end
51          C = x*x';                       % Covariance matrix
52          eigs=flipud(sort(eig(C)));      % Eigenvalues largest to smallest
53          % Percentage difference betweeen main and weakest eigenvalues
54          pd(i_snr) = 100*(eigs(1)-eigs(end))/eigs(1);
55      end
56      %{
57      Use lowest percentage difference to decide on a cutoff value, below
58      which the percentage difference in eigenvalues is not useful for
59      determining SNR and therefore not useful for determining signal
60      presence. The eigenvalue corresponding to the man-made signal can't
61      be effectively distinguished from the noise below, so eigenvalue
62      based detection becomes significantly less effective.
63      %}
64      clear x X noise;
65      % The % difference threshold is the mean PD at low SNR (-30:-40dB)
66      pdCutoff = mean(pd(end-10/SNR_ITVL:end));
67      %pdCutoff = max(pd(end-10/SNR_ITVL:end));
68      % Convert threshold to index number so it can be matched to cutoff SNR
69      cutoffIdx = min(find(pd < pdCutoff));
70      % Polynomial for above the cutoff PD - gives SNR from PD
71      p = polyfit(pd(1:cutoffIdx),SNR(1:cutoffIdx),5);
72      % The low SNR limit of eigenvalue detection effectiveness
73      SNRcutoff = polyval(p,pdCutoff);
74      % The SNR where eigenvalue detection begins to significantly degrade
75      % Choose degrade point based on 3dB (1.5* mean PD at low SNR)
76      SNRdegrade = [polyval(p,pdCutoff*1.5); polyval(p,pdCutoff*1.05);...
77                    polyval(p,pdCutoff*1.02)];
78      %SNRdegrade = polyval(p,pdCutoff*1.05);
79      %fprintf('Cutoff SNR: %.2f\n',SNRcutoff);
80      %fprintf('SNR degrade from: %.2f\n',SNRdegrade(1));
```

```
81      pdMax = max(pd);
82  end
```

D.2 Code Listing 2

```
1  %{
2  AUTHOR:      Benedict Hardless
3  CREATED:     27-May-2021
4  MODIFIED:    May-21, Jun-21
5  TESTED:      Jun-21
6  FILE NAME:   PDtoSNRrelations.m
7  PURPOSE:     Plots SNR cutoff vs cycles captured for various antenna counts.
8  FILES REQ:   Must be in the same directory as PDtoSNR.m
9  %}
10 clc; clear; close all;
11 % number of data points
12 N = 40;
13 NUM_CYCLES = exp(linspace(log(100),log(10000),N));
14 figure(1);
15 for NUM_ANT = 2:10
16     p = nan(N,6);
17     SNRdeg = nan(N,1); SNRcut = SNRdeg;
18     for i_cyc = 1:length(NUM_CYCLES)
19         [p(i_cyc,:),SNRdeg(i_cyc),SNRcut(i_cyc)] =...
20             PDtoSNR(NUM_CYCLES(i_cyc),NUM_ANT);
21     end
22 plot(NUM_CYCLES,SNRdeg,'-o'); hold on;
23 end
24 grid on; grid minor;
25 title('SNR cutoff vs Number of Cycles');
26 xlabel('Cycles Captured');
27 ylabel('SNR cutoff (dB)');
28 legend('2 ant', '3 ant', '4 ant', '5 ant', '6 ant',...
29        '7 ant', '8 ant', '9 ant', '10 ant');
30 xlim([0 10000]);
31 % Uncomment for printing figure to .eps file
32 %print('-f1','SNRtoNumCyc','-depsc');
```

D.3 Code Listing 3

```
%{
AUTHOR:     Benedict Hardless
CREATED:    27-May-2021
MODIFIED:   May-21, Jun-21
TESTED:     Jun-21
FILE NAME:  MMPD_singleChan.m
PURPOSE:    Plots probability of detection and probability of false alarm
            against SNR. User has the option of examining multiple custom
            thresholds or a single optimised threshold, compared against
            sub-optimal MME performance.
INFO:       Uncomment line 18 if user wants optimum
            threshold. Comment out for custom.
%}
clc; clear; close all;
fprintf('\n-----------------------------------------------\n');
fprintf('BOOLEAN MMPD EIGENVALUE DETECTION\n\n');
OPTION = 0; % 1 for optimum MMPD threshold. 0 for custom thresholds.
LOOPS = 500;                    % Times to loop the simulation
%% User Settings
f = 1E2;                         % Frequencies
ITER_CHAN = 1;                   % Number of channel change iterations
Fs = f*5;                        % Sampling frequency
T = 1/Fs;                        % Sampling interval
M = 10;                           % Number of antennas
NUM_CYCLES = 2000;               % Number of cycles to capture
DUR = NUM_CYCLES/f;              % Duration for the number of cycles
n = 1:Fs*DUR;                    % Vector of sample numbers
t = (n-1)*T*1000;                % Vector of times
N = length(n);                   % Number of samples
w0 = 2*pi*f/Fs;                  % Radians per sample
A = 1;                           % Signal amplitude
SNR_ITVL = 0.5;                  % Spacing for SNR values
SNR_LVLS = 0:-SNR_ITVL:-35;      % SNR levels for consideration
POLYS = [1.993E-8, -4.818E-6, 4.744E-4, -2.413E-2, 8.177E-1, -1.894E1;...
         3.247E-8, -8.1E-6,   7.949E-4, -3.849E-2, 1.111,    -2.3E1;...
         3.858E-8, -9.602E-6, 9.315E-4, -4.427E-2, 1.228,    -2.53E1;...
         5.106E-8, -1.293E-5, 1.259E-3, -5.884E-2, 1.513,    -2.827E1;...
         5.498E-8, -1.388E-5, 1.345E-3, -6.237E-2, 1.585,    -2.973E1;...
         5.056E-8, -1.249E-5, 1.185E-3, -5.428E-2, 1.412,    -2.931E1;...
```

```
          6.204E-8, -1.563E-5, 1.504E-3, -6.902E-2, 1.717,    -3.216E1;...
          6.099E-8, -1.52E-5,  1.443E-3, -6.545E-2, 1.631,    -3.211E1;...
          7.046E-8, -1.781E-5, 1.712E-3, -7.825E-2, 1.908,    -3.477E1;];
SNRCUTS = -[17.701, 18.989, 20.96, 21.007, 22.048, 22.345, 22.849, ...
    23.529,...
           24.152];
DCUTS = [1.7644 2.9561 3.6597 4.5142 5.1995 5.6836 6.377 6.783 7.1653];
% Define key parameters for the desired cutoff points
% Characteristic polynomial per table 3.1
P = POLYS(M-1,:);
% SNR degraded values
Dco = DCUTS(M-1);
% MMPD Threshold(s)
if (OPTION == 1)
    phi = -0.0609*M^2+1.6506*M+0.34112;
else
    phi = [10.73:-0.0125:3];
end
SNRphi = polyval(P,phi);
SNRcut = SNRCUTS(M-1);
% MME threshold
L = 1;  % Smoothing factor for MME
a = sqrt(N); b = sqrt(M*L); % These to shorten the equation next line
gamma = ((a+b)^2/(a-b)^2)*(1+((a+b)^(-2/3))/((N*M*L)^(1/6))*0.45);
%% Produce the signal and associated noise amplitudes
% Produce a signal
x = A*sin(w0*n);
% Get the signal power
Ps = sum(x.^2)/length(x);
% Noise amplitude for each SNR
An = sqrt(Ps*10.^-(SNR_LVLS/10));
%% Receive the signal with noise
% Pre-allocate matrices for channel detection
res = nan(LOOPS,length(SNR_LVLS));
% Pre-allocate matrices for MME detection
MMEres = res; MMEacc1 = res; MMEdetacc1 = res; MMEfa_acc1 = res;
MMEacc = nan(1,length(SNR_LVLS)); MMEdet_acc = MMEacc;
MMEfa_acc = MMEacc;
% Pre-allocate received signal matrix
rx = nan(M,N,length(SNR_LVLS));
% Pre-allocate accuracy matrices
acc1 = res; detacc1 = res; fa_acc1 = res;
acc = nan(length(phi),length(SNR_LVLS));
```

```
82 det_acc = acc;
83 fa_acc = acc;
84 % Pre-allocate for timers
85 tmrMMPD = nan(LOOPS,1); tmrMME = tmrMMPD; tmrEigs = tmrMMPD;
86 for i_thresh = 1:length(phi)
87     % For each SNR level
88     for i_sim = 1:LOOPS
89         for i_snr = 1:length(SNR_LVLS)
90             % Randomly decide the channel state
91             chanState = randi([0,1]);
92             % For each RX node
93             for i_node = 1:M
94                 noise = An(i_snr)*randn(1,N);
95                 % There is a RX signal with different noise (diff SNR ...
                       each frame)
96                 rx(i_node,:,i_snr) = x*chanState + noise;
97                 rx(i_node,:,i_snr) = stand(rx(i_node,:,i_snr));
98             end
99             %% Process using MMPD and MME
100            tEigs = tic;           % Start timer for obtaining eigenvalues
101            C = rx(:,:,i_snr)*rx(:,:,i_snr)'/N;     % Form covariance ...
                   matrix
102            [Vx,Dx] = eig(C);               % Get max eigenvectors and ...
                   values
103            eigs = diag(Dx);                % Isolate the eigenvalues
104            [eigs, eigIdx] = sort(eigs);    % Sort eigenvalues low to high
105            % change to be high to low
106            eigs = flipud(eigs); eigIdx = flipud(eigIdx);
107            tmrEigs(i_sim) = toc(tEigs);
108            tMMPD = tic;     % Start timer for MMPD
109            % Get percent difference between strongest/weakest eigenvalues
110            percDiff = 100*(eigs(1)-eigs(end))/eigs(1);
111            % Convert to SNR estimate
112            SNRinfer = polyval(P,percDiff);
113            % If the MMPD inferred SNR is above threshold, signal is ...
                   present
114            if (SNRinfer > SNRphi(i_thresh))
115               res(i_sim,i_snr) = 1;
116            % If MMPD inferred SNR is below threshold, signal is not ...
                   present
117            else
118               res(i_sim,i_snr) = 0;
119            end
```

```
120             tmrMMPD(i_sim) = toc(tMMPD);     % Stop timer for MMPD
121             tMME = tic;                % Start timer for MME
122             % If the MME ratio is above the threshold, signal is present
123             MMEratio = eigs(1)/eigs(end);
124             if (MMEratio > gamma) MMEres(i_sim,i_snr) = 1;
125             else MMEres(i_sim,i_snr) = 0;
126             end
127             tmrMME(i_sim) = toc(tMME);       % Stop timer for MME
128             % Test if the result matches the actual channel state for MMPD
129             acc1(i_sim,i_snr) = (res(i_sim,i_snr) == chanState);
130             % Do the same for MME
131             MMEacc1(i_sim,i_snr) = (MMEres(i_sim,i_snr) == chanState);
132             % Test if the result matches when the channel state is 1
133             if (chanState)
134                 detacc1(i_sim,i_snr) = (res(i_sim,i_snr) == chanState...
135                                         & chanState == 1);
136                 MMEdetacc1(i_sim,i_snr) = (MMEres(i_sim,i_snr) == ...
                        chanState...
137                                         & chanState == 1);
138             end
139             % Test if the result is a 1 when the channel state is 0
140             fa_acc1(i_sim,i_snr) = (res(i_sim,i_snr) > chanState);
141             MMEfa_acc1(i_sim,i_snr) = (MMEres(i_sim,i_snr) > chanState);
142         end
143     end
144     %% Determine accuracy
145     % For each SNR
146     for i_snr = 1:length(SNR_LVLS)
147         % The accuracy is the percentage of ones in the column
148         acc(i_thresh,i_snr) = 100*sum(acc1(:,i_snr))/LOOPS;
149         MMEacc(i_thresh,i_snr) = 100*sum(MMEacc1(:,i_snr))/LOOPS;
150         % Probability of detection is percentage of ones in the column
151         % out of all non-nan results
152         det_acc(i_thresh,i_snr) = 100*sum(detacc1(:,i_snr)==1)/...
153                         sum(~isnan(detacc1(:,i_snr)));
154         MMEdet_acc(i_thresh,i_snr) = 100*sum(MMEdetacc1(:,i_snr)==1)/...
155                         sum(~isnan(MMEdetacc1(:,i_snr)));
156         % Probability of false alarm is percentage of ones in the column
157         fa_acc(i_thresh,i_snr) = 200*sum(fa_acc1(:,i_snr))/LOOPS;
158         MMEfa_acc(i_thresh,i_snr) = 200*sum(MMEfa_acc1(:,i_snr))/LOOPS;
159     end
160     detacc1 = nan(LOOPS,length(SNR_LVLS));
161     MMEdetacc1 = detacc1;
```

```
162     if (OPTION == 0)
163         fprintf('Phi: %.4f \tPfa: %.2f\n', phi(i_thresh), ...
                mean(fa_acc(i_thresh,:)));
164         if (mean(fa_acc(i_thresh,:)) > 10); break; end
165     end
166 end
167 fprintf('\n-------------------------------------------------\n');
168 % Average the results for the MME scheme
169 if (length(phi) > 1)
170     MMEdet_accAvg = mean(MMEdet_acc);
171     MMEfa_accAvg = mean(MMEfa_acc);
172 else
173     MMEdet_accAvg = MMEdet_acc;
174     MMEfa_accAvg = MMEfa_acc;
175 end
176 figure()
177 for i = 1:length(phi)
178     plot(SNR_LVLS, det_acc(i,:),'-o'); hold on;
179 end
180 plot(SNR_LVLS, MMEdet_accAvg,'-^');
181 title('Eigenvalue Detection Performance (Pd)');
182 xlabel('SNR');
183 ylabel('Probability of Detection (%)');
184 grid on; grid minor;
185 if (OPTION == 1)
186     legend('Opt MMPD', 'MME L = 1');
187 else
188     legend('Phi = 1.8','Phi = 1.5', 'Phi = 1.2','MME L = 1',...
189             'Location','northwest');
190 end
191 figure()
192 for i = 1:length(phi)
193     plot(SNR_LVLS, fa_acc(i,:),'-o'); hold on;
194 end
195 plot(SNR_LVLS, MMEfa_accAvg,'-^');
196 title('Eigenvalue Detection Performance (Pfa)');
197 xlabel('SNR');
198 ylabel('Probability of False Alarm (%)');
199 grid on; grid minor;
200 if (OPTION == 1)
201     legend('Opt MMPD', 'MME L = 1');
202 else
203     legend('Phi = 1.8','Phi = 1.5', 'Phi = 1.2','MME L = 1',...
```

```
204             'Location','northwest');
205 end
206 % Display execution time results
207 MMPDtime = (mean(tmrEigs)+mean(tmrMMPD))*1E6;
208 MMEtime = (mean(tmrEigs)+mean(tmrMME))*1E6;
209 fprintf('MMPD execution time: %.3f us\n', MMPDtime);
210 fprintf('MME execution time: %.3f us\n', MMEtime);
```

D.4 Code Listing 4

```
1  %{
2  AUTHOR:     Benedict Hardless
3  CREATED:    25-May-2021
4  MODIFIED:   May-21, Jun-21
5  TESTED:     Jun-21
6  FILE NAME:  MMPD_multiChan.m
7  PURPOSE:    Compares FFT performance with and without MMPD and PCA.
8  INFO:       Uncomment 'Plot Results' section to obtain comparison plot.
9              Regarding tables for examining FFT performance and the trade
10             off between TX conflicts and Free channel utility, this
11             requires multiple executions to obtain. For MMPD/PCA enabled,
12             set OPTION equal to 1. For baseline FFT set OPTION to 0. You
13             can adjust the MMPD/PCA low SNR threshold at line 144.
14 %}
15 clc; clear; close all;
16 fprintf('\n-----------------------------------------------\n');
17
18 %% User Settings
19 OPTION = 0; % Set to 1 for MMPD and PCA enabled. Else set to 0.
20 f = 3400E6:8E6:3592E6+2.5E6;    % 4G LTE band frequencies
21 DWN_SHFT_FAC = 4;
22 fd = f*10^-DWN_SHFT_FAC;        % Down shift factor for frequencies
23 NUM_CHANS = length(fd);         % Number of channels
24 ITER_CHAN = 1;                  % Number of channel change iterations
25 Fs = 2^22;                      % Sampling frequency
26 T = 1/Fs;                       % Sampling interval
27 M = 4;                          % Number of antennas
28 NUM_CYCLES = 2000;              % Number of cycles to capture
29 DUR = NUM_CYCLES/fd(end);       % Duration for the number of cycles
```

```
n = 1:Fs*DUR;                   % Vector of sample numbers
t = (n-1)*T*1000;               % Vector of times
N = length(n);                  % Number of samples
w0 = 2*pi*fd/Fs;                % Radians per sample
A = 1;                          % Signal amplitude
% Spacing for SNR values
SNR_itvl = 2;
% Times to loop the simulation to get averaged results
LOOPS = 1000;
TrLast = nan; Tr = nan;         % Pre-allocate for prior feature matrix
% Variables for averaging
sumAcc = 0;
sumErrorCount = 0;
sumUtilRes = 0;
%% Establish decision threshold and cutoff for signal detection
% Polynomials for SNR inference
POLYS = [1.993E-8, -4.818E-6, 4.744E-4, -2.413E-2, 8.177E-1, -1.894E1;...
         3.247E-8, -8.1E-6,   7.949E-4, -3.849E-2, 1.111,    -2.3E1;...
         3.858E-8, -9.602E-6, 9.315E-4, -4.427E-2, 1.228,    -2.53E1;...
         5.106E-8, -1.293E-5, 1.259E-3, -5.884E-2, 1.513,    -2.827E1;...
         5.498E-8, -1.388E-5, 1.345E-3, -6.237E-2, 1.585,    -2.973E1;...
         5.056E-8, -1.249E-5, 1.185E-3, -5.428E-2, 1.412,    -2.931E1;...
         6.204E-8, -1.563E-5, 1.504E-3, -6.902E-2, 1.717,    -3.216E1;...
         6.099E-8, -1.52E-5,  1.443E-3, -6.545E-2, 1.631,    -3.211E1;...
         7.046E-8, -1.781E-5, 1.712E-3, -7.825E-2, 1.908,    -3.477E1;];
SNRCUTS = -[17.701, 18.989, 20.96, 21.007, 22.048, 22.345, 22.849, ...
    23.529,...
          24.152];
DCUTS = [1.7644 2.9561 3.6597 4.5142 5.1995 5.6836 6.377 6.783 7.1653];
% Define key parameters for the desired cutoff points
% Characteristic polynomial per table 3.1
P = POLYS(M-1,:);
% SNR degraded values
Dco = DCUTS(M-1);
phi = Dco*(0.0105*M^2-0.1783*M+2.2815); % MMPD Threshold
SNRphi = polyval(P,phi);
SNRcut = SNRCUTS(M-1);
    %% Produce a composite freq signal
for i_sim = 1:LOOPS
    chanStatePU = randi([0,1],1,NUM_CHANS);  % Random PU channel states
    SNR_LVLS = 0:-SNR_itvl:-30;         % SNR levels for consideration
    % Develop TX signal components as channel frequencies
    for i_chan = 1:NUM_CHANS
```

```
72              % Produce signal matrix. Row => channel
73              xChan(i_chan,:) = A*sin(w0(i_chan)*n)*chanStatePU(i_chan);
74      end
75      % Sum the signals together to get the TX band as one signal
76      x = sum(xChan,1);
77      % Get the signal power
78      Ps = sum(x.^2)/length(x);
79      % Noise amplitude for each SNR
80      An = sqrt(Ps*10.^-(SNR_LVLS/10));
81      %% Receive the signal with noise
82      % Pre-allocate magnitude matrix for channel detection
83      A_chan = nan(length(SNR_LVLS),NUM_CHANS);
84      % Pre-allocate for timer
85      tMMPD = zeros(1,21);
86      % For each SNR level
87      for i_snr = 1:length(SNR_LVLS)
88          % For each antenna
89          for i_M = 1:M
90              noise = An(i_snr)*randn(1,N);
91              % There is a RX signal with different noise (diff SNR each ...
                    frame)
92              rx(i_M,:,i_snr) = x + noise;
93          end
94          %% Apply MMPD
95          tStartMED = tic;                          % Start timer
96          C = rx(:,:,i_snr)*rx(:,:,i_snr)'/N;       % Form covariance matrix
97          [Vx,Dx] = eig(C);                % Get max eigenvectors and values
98          eigs = diag(Dx);                 % Isolate the eigenvalues
99          [eigs, eigIdx] = sort(eigs);     % Sort eigenvalues low to high
100         % change to be high to low
101         eigs = flipud(eigs); eigIdx = flipud(eigIdx);
102         % Get percent difference between strongest/weakest eigenvalues
103         percDiff = 100*(eigs(1)-eigs(end))/eigs(1);
104         if (¬isnan(percDiff))
105             % Convert to SNR estimate if there is a valid percentage diff
106             SNRinfer = polyval(P,percDiff);
107         else
108             SNRinfer = SNRcut;
109         end
110         % If the inferred SNR is above threshold, use calculated dominant
111         % eigenvector for transform. Construct feature vector
112         if (SNRinfer > SNRphi) Tr = Vx(:,eigIdx(1))';
113         % But if inferred SNR is below threshold, use the previously
```

```
114         % calculated eigenvector for transform if there is one
115         elseif (¬isnan(TrLast)) Tr = TrLast; % Alternative feature vector
116         else % PUT THE SURVEYED FEATURE VECTOR HERE for Tr
117         end
118         TrLast = Tr;
119         % Obtain transformed data matrix (eig-vectors must be the rows)
120         x1 = rx(1,:,i_snr);
121         x2 = Tr*rx(:,:,i_snr);
122         % In the case of a single channel, the PCA will not work. You will
123         % get errors when the channel state is zero. Use the single channel
124         % script instead.
125         % Perform FFT for signal without any MMPD or PCA transform
126         x_fft_basic = fft(x1);
127         Ax1 = abs(x_fft_basic)*2/N;
128         Ax1 = Ax1/max(Ax1);
129         fScale1 = (n-1)/(N*T);
130         % Perform FFT for principle component(s) of signal
131         x_fft = fft(x2); % CHANGE BACK TO X@
132         Ax = abs(x_fft)*2/N;
133         Ax = Ax/max(Ax);
134         fScale = (n-1)/(N*T);
135         % If SNR is greater than intersection point between channel
136         % accuracy and preventing TX conflicts
137         if (SNRinfer > phi)
138             % Then use P3 polynomial
139             THRESH(i_snr) = 0.4;
140         else
141             % But if SNR is worse, then use more conservative model to
142             % minimise TX conflicts. Remain fixed if MMPD/PCA is disabled.
143             if (OPTION ==1)
144                 THRESH(i_snr) = 0.25;
145             else
146                 THRESH(i_snr) = 0.4;
147             end
148         end
149         tMED(i_snr) = toc(tStartMED);   % Elapsed time for MED section
150         %% Plot results
151         %{
152         % FFT only on received signal
153         subplot(2,length(SNR_LVLS),i_snr);
154         plot(fScale1/1E5,Ax1); grid on; grid minor;
155         xlabel('Frequency (GHz)');
156         ylabel('Amplitude');
```

```
157         txt = sprintf('FFT Only: SNR %ddB',SNR_LVLS(i_snr));
158         title(txt);
159         axis([3.2 3.8 0 max(Ax)*1.1]);
160 
161         % MMPD informed FFT using PCA
162         subplot(2,length(SNR_LVLS),i_snr+length(SNR_LVLS));
163         plot(fScale/1E5,Ax); grid on; grid minor;
164         xlabel('Frequency (GHz)');
165         ylabel('Amplitude');
166         txt = sprintf('MMPD/PCA: SNR %ddB',SNR_LVLS(i_snr));
167         title(txt);
168         axis([3.2 3.8 0 max(Ax)*1.1]);
169         %}
170 
171         %% Detect channel states
172         F = fd;                     % Frequencies to search for
173         S_tone = round(F*N*T)+1;    % Sample number of those frequencies
174         sideSamp = floor(350*N*T);  % Samples either side for the search
175         % Look up the corresponding amplitudes for the sample nums
176         for i_fSamp = 1:NUM_CHANS
177             if (OPTION == 1)
178                 A_chan(i_snr,i_fSamp) = Ax(S_tone(i_fSamp));
179             else
180                 A_chan(i_snr,i_fSamp) = Ax1(S_tone(i_fSamp));
181             end
182         end
183     end
184     % Compare the amplitude for each channel to a threshold
185     detChanStates = A_chan > THRESH';    % Is it above the threshold?
186     % Quantify over-all channel state accuracy
187     % For each SNR
188     for i_acc = 1:length(SNR_LVLS)
189         % Which channel states are equal to the defined PU channel states?
190         res = detChanStates(i_acc,:) == chanStatePU;
191         % Count the number of correct states and convert to % of channels
192         acc(i_acc) = sum(res)*100/NUM_CHANS;
193     end
194     %% Gather results
195     % Construct SNR and channel state matrix for manual review
196     SNRcol = [nan; SNR_LVLS']; chanStateTable = [chanStatePU; ...
            detChanStates];
197     reviewMat = [SNRcol, chanStateTable];
198     % For each SNR level
```

```
199     for i_snr = 1:length(SNR_LVLS)
200         % Count channel state errors where SU thinks channel free but ...
                it isn't
201         errorCount(i_snr) = sum(reviewMat(i_snr+1,2:end) < ...
                reviewMat(1,2:end));
202         utilRes(i_snr) = sum(reviewMat(i_snr+1,2:end) == 0 & ...
                reviewMat(1,2:end) == 0);
203     end
204     % Convert SU/PU conflict channel state errors to percentage for ...
            each SNR
205     errorCount = errorCount*100/NUM_CHANS;
206     if (NUM_CHANS > 1)
207         utilRes = utilRes*100/(sum(reviewMat(1,2:end) == 0));
208     else
209         utilRes = zeros(1,length(SNR_LVLS));
210     end
211     % Need this if statement to stop sim breaking if there are no chan ...
            state
212     % errors found.
213     if (sum(errorCount > 0) > 0)
214         % Identify a cutoff SNR based on where SU/PU conflicts occur
215         SNRcutOff = max(SNR_LVLS(find(errorCount > 0)))+SNR_itvl;
216     else
217         SNRcutOff = SNR_LVLS(end);
218     end
219     cutoffIdx = min(find(SNR_LVLS ≤ SNRcutOff));
220     util = sum(utilRes(1:cutoffIdx))/cutoffIdx;
221     % Results averaging across simulation loops
222     sumAcc = sumAcc + acc;
223     sumErrorCount = sumErrorCount + errorCount;
224     sumUtilRes = sumUtilRes + utilRes;
225 end
226 meanAcc = sumAcc/LOOPS;
227 meanErrorCount = sumErrorCount/LOOPS;
228 meanUtilRes = sumUtilRes/LOOPS;
229 %% Present results
230 % Show computation time
231 MEDtime = sum(tMED)*1000;
232 fprintf('MMPD & FFT computation time: %.4fms (all SNR)\n\n',MEDtime);
233 resTable = table(SNR_LVLS',acc',errorCount',utilRes');
234 resTable.Properties.VariableNames = {'SNR (dB)',...
235     'Channel Detection Accuracy (%)', 'TX conflicts (%)',...
236     'Free Channel Utility (%)'};
```

```
237 disp(resTable);
238 fprintf('Recommended SNR cutoff:%.2f dB\n', SNRcutOff);
239 fprintf('Avg channel prediction accuracy above SNR cutoff: %.3f%%\n',...
240     mean(acc(1:cutoffIdx)));
241 fprintf('Average free channel utility above SNR cutoff: %.3f%%\n\n',util);
242 %% Present averaged results over all LOOPS
243 resTable = table(SNR_LVLS',meanAcc',meanErrorCount',meanUtilRes');
244 resTable.Properties.VariableNames = {'SNR (dB)',...
245     'Channel Detection Accuracy (%)', 'TX conflicts (%)',...
246     'Free Channel Utility (%)'};
247 disp(resTable);
248 fprintf('Recommended SNR cutoff:%.2f dB\n', SNRcutOff);
249 fprintf('Avg channel prediction accuracy above SNR cutoff: %.3f%%\n',...
250     mean(meanAcc(1:cutoffIdx)));
```

D.5 Code Listing 5

```
1  %{
2  AUTHOR:      Benedict Hardless
3  CREATED:     11-Jul-2021
4  MODIFIED:    11,12,13 Jul-21
5  TESTED:      13 Jul-21
6  FILE NAME:   FMdemod.m
7  PURPOSE:     Import baseband IQ *.wav file and play back mono audio. Display
8               the MPX spectrum, which is the FM intelligence imposed on the
9               carrier. The modulation signal always contains mono audio,
10              sometimes a 19kHz pilot tone, sometimes stereo audio centered
11              at 38kHz, and other channel information.
12 INFO:        Uncomment time varying FFT and vary block size to see how
13              MPX spectrum changes with time.
14 %}
15 clc; clear; close all;
16 %% User defined information
17 FILENAME = 'SDRSharp_20210707_030954Z_101193000Hz_IQ.wav';
18 BW = 200E3; hBW = BW/2;      % Bandwidth for baseband RF filter
19 D_EMPH_ORDER = 8;            % Order of de-emphasis filter
20 VOL_SCALE = 10;              % Audio volume scaler
21 %% Import IQ Signal
22 [xIQ,Fs] = audioread(FILENAME);
```

```
23 % Combine default import into single vector of complex numbers
24 N = length(xIQ); n = 1:N; T = 1/Fs;
25 % Perform FFT on part of IQ signal to examine spectrum
26 N2 = floor(N/500); n2 = 2:N2;
27 IQfft = fft(xIQ(2:N2));
28 IQfft = abs(IQfft)*2/N2;
29 f = ((n2-1)/(N2*T));
30 %figure(); plot(f,IQfft);
31 % Find the maximum frequency component i.e. baseband offset
32 [M, I] = max(IQfft(2:end)); OFFSET = f(I+1);
33 fprintf('OFFSET: %4.2f kHz\n',OFFSET/1000);
34 % Filter it
35 b = getFilterTF(Fs,OFFSET-hBW,OFFSET+hBW,30);
36 xIQ = filter(b,1,xIQ);
37 %% Demodulate FM Signal from baseband IQ
38 x = xIQ(:,1)+j*xIQ(:,2);
39 % Modulating signal amplitude is proportional to inst freq
40 % Inst freq of signal is rate of change of phase angle
41 xA = x(1:end-1); xB = conj(x(2:end));
42 phi = angle(xA.*xB);
43 m = diff(phi);
44 %% Frequency shift signal to remove offset introduced by SDR#
45 % Filter for FM MPX info
46 b = getFilterTF(Fs,0,110E3,30);
47 MPX = filter(b,1,m);
48 % Isolate audio and apply de-emphasis filter
49 scale = round(Fs/48E3);
50 Fs2 = Fs/scale;
51 mDec = decimate(m,scale);
52 N2 = length(mDec); n2 = 1:N2; T2 = 1/Fs2;
53 b = getFilterTF(Fs2,0,16E3,30);
54 mMono = filter(b,1,mDec);
55 b = getFilterTF(Fs2,0,5000,D_EMPH_ORDER);
56 mMonoAudio = filter(b,1,mMono);
57 mMonoAudio = (mMonoAudio/max(mMonoAudio))*VOL_SCALE;
58 sound(mMonoAudio,Fs2);
59 %% Analyse FM MPX spectrum in single block
60
61 xFFT = fft(MPX);
62 xFFT = abs(xFFT)*2/N;
63 xFFT = xFFT/max(xFFT);
64 fScale = ((n-1)/(N*T))/1E3;
65 figure();
```

```
66 subplot(1,2,1);
67 plot(fScale(1:N/2), xFFT(1:N/2)); grid on; grid minor;
68 xlabel('Frequency (kHz)');
69 ylabel('Amplitude');
70 title('FM MPX Spectrum');
71 xlim([0 60]);
72 ylim([0 0.25]);
73
74 %% Analyse FM mono audio spectrum in single block after de-emphasis
75 xFFT = fft(mMonoAudio);
76 xFFT = abs(xFFT)*2/N2;
77 xFFT = xFFT/max(xFFT);
78 fScale = ((n2-1)/(N2*T2))/1E3;
79 subplot(1,2,2);
80 plot(fScale(1:N2/2), xFFT(1:N2/2)); grid on; grid minor;
81 xlabel('Frequency (kHz)');
82 ylabel('Amplitude');
83 title('De-emphasised FM Mono Audio');
84 xlim([0 20]);
85 ylim([0 1.1]);
86
87 %% Time varying FFT
88 %{
89 % Define block size, overlap and number of blocks
90 N = N-1;
91 bsize = floor(N/50);
92 lapFrac = 2; lap = bsize/lapFrac;
93 numBlks = floor(1+(N-bsize)/(bsize-lap));
94 X = nan(numBlks,floor(N/50));
95 % Iterate through the blocks
96 for i = 1:numBlks
97     % Set first block start and end. No overlap setting needed
98     if i==1
99         startN = 1;
100        endN = bsize;
101    else
102        % Set block start to create overlap with last block
103        startN = endN-lap+1;
104        % Set block end
105        endN = startN+bsize-1;
106    end
107    % Perform fft on block
108    X(i,:) = abs(fft(MPX(startN:endN)))*2/N;
```

```
end

N1 = size(X,2);
n1 = 1:N1;
f = ((n1-1)/(N1*T))/1E3;
X = X/max(max(X));

figure();

for i_blk = 1:numBlks
    plot(f, X(i_blk,:));
    grid on; grid minor;
    xlim([0 270]);
    ylim([0 0.4]);
    xlabel('Frequency (kHz)');
    pause(0.1);
end
%}
```

```
%{
returns b coefficients for use in filter(b,1,x) for a bandpass
filter with limits fL and fH. Higher order filters are more defined.
%}
function h = getFilterTF(Fs,fL,fH,order)

a = 2*pi*fL/Fs;
% Set upper filter pass freq
b =2*pi*fH/Fs;
% Filter order
M = order;
% Number of coefficients
N = M+1;
% Span of n values for our filter. More is more precise.
n = -M/2:M/2;
% Get the coefficients for the TF by solving for values of n.
h = 1./((n+eps)*pi).*(sin(b*(n+eps))-sin(a*(n+eps)));
% Include if you want a hamming window
n = 0:M;
hamwin = 0.54-0.46*cos(2*n*pi/M);
h = h.*hamwin;
%{
```

```
UNCOMMENT TO SEE FILTER RESPONSE PLOTTED
% Set frequency range to look at for filter response
dw = 1/400;
w = 0:dw:pi;
% Define z and get filter response
z = exp(j*w);
resp = 0;
for idx = 1:N
    if i < M
        resp = resp + h(idx).*z.^(-idx);
    end
end
% Scale the frequency for the defined sampling frequency
f = w*Fs/(2*pi);
% Plot result
plot(f,abs(resp)); grid on; grid minor;
title('Filter Frequency Response');
ylabel('Gain'); xlabel('Frequency (Hz)');
%}

end
```

```
%*********************************************************************
%{
Author:     Benedict Hardless
Created:    18-Jul-2021
Name:       getFFT
Purpose:    Simplify application of fft by scaling the values.
Arguments:  X is the signal for FFT.
            N is number of samples.
            T is sampling interval.
            NORM is 0, but change to 1 for normalised amplitudes.
Returns:    Y is vector of amplitudes
            f is vector of corresponding frequencies
%}
function [Y, f] = getFFT(X,N,T,NORM)
    n = 1:N;
    Y = fft(X);
    Y = abs(Y)*2/N;
    if (NORM > 0)
        Y = Y/max(Y);
```

```
    end
    f = ((n-1)/(N*T));
end
%**********************************************************************
```

D.6 Code Listing 6

```
%{
AUTHOR:     Benedict Hardless
CREATED:    14-Aug-2021
MODIFIED:   14-Aug-21
TESTED:     14-Aug-21
FILE NAME:  preProcess.m
PURPOSE:    Up-shifts band to center in FFT window. Truncates to 2000 cycle
            equivalent number of samples. Converts IQ format to time
            domain signal. Eliminates DC offset.
ARGUMENTS:  IQ is Nx2 matrix of I and Q components.
            Fs is sample frequency.
RETURNS:    x is row vector of pre-processed time domain waveform.
%}
function [x,I] = preProcess(IQ,Fs,BW)
      % Up-shift to center in FFT window
    C = 2000;
    % Number of samples for 2000 cycles. Required for MMPD.
    if (Fs/4 > BW)
        N = C*Fs/(Fs/4-BW/2);
    else
        N = 4*C;
    end
    N = round(N);
    % Truncate sample count
    IQ1 = IQ(1:N,:);
    T = 1/Fs;
    t = 0:T:(N-1)*T;
    % Convert IQ to complex number
    R = IQ1(:,1)+j*IQ1(:,2);
    % Remove DC offset
    Ravg = mean(R);
    if (Ravg > 0)
```

```
33          R = R-Ravg;
34      elseif (Ravg < 0)
35          R = R+Ravg;
36      end
37      phi = angle(R);
38      a = (abs(R).*cos(phi))';
39      [X f]= getFFT(a,N,T,1); %figure(); plot(f,X);
40      [M idx] = max(X); OFFSET = Fs/round(idx-N/4);
41      if (OFFSET > 0)
42          SHIFT = Fs/4+OFFSET*1E2;
43      else
44          SHIFT = Fs/4-OFFSET*1E2;
45      end
46      % Construct time domain waveform
47      wc = 2*pi*SHIFT;
48      % Return time domain waveform
49      x = (abs(R).*cos(wc*t'+phi))';
50      [X,f] = getFFT(x,length(x),T,1);
51      [M I] = max(X);
52      %figure(); plot(f,X);
53  end
```

D.7 Code Listing 7

```
1  %{
2  AUTHOR:      Benedict Hardless
3  CREATED:     16-Aug-2021
4  MODIFIED:    16:30-Aug-21
5  TESTED:      16:30-Aug-21
6  FILE NAME:   BoolDetectMulti_RevD.m
7  PURPOSE:     Attempts Boolean detection of baseband FM signals. This
8               script was developed in the early stages of SDR testing, and
9               has been refined and revised to use PCA and FFT for detection
10              following MMPD.
11 %}
12 %% Preliminary & User Settings
13 clc; clear all; close all;
14 LOWLVL = -35;                % SNR to start from
15 LOOPS = 100;                 % Times to iterate for averaging results
```

```
16 BW = 250E3;                  % Bandwidth of baseband RF
17 FFT_THRESH = 0.2685;         % Threshold for FFT detection
18 DIFF_PERF = 0;               % Different SNR for each receiver?
19                              % 1 for yes, 0 for no.
20 %{
21 Define the type of simulation. 0 for correlated noise. 1 is for
22 un-correlated noise. 2 is for separately recorded SDR signals.
23 %}
24 TYPE = 2;
25 %{
26 Activate pre-filtering. Important because it can give rise to correlated
27 noise. 0 to disable. 1 to enable.
28 %}
29 PRE_FILT = 0;
30 % Enable/disable standardisation
31 STAND = 1;
32 % Test signal present or absent. 1 for signal.
33 SIG = 1;
34 %{
35 Feature vector for signal of interest. 2 is to weight all receivers evenly.
36 1 is to select best receiver. 0 is to use all receivers, but favour those
37 with better received signal power.
38 %}
39 SINGLE = 0;
40 %% Replicate SDR IQ File and Apply MMPD and MME
41 if (SIG == 1)
42    %{
43     FILENAME1 = 'DTS_FM_100.7a.wav';
44     FILENAME2 = 'DTS_FM_100.7a.wav';
45     FILENAME3 = 'DTS_FM_100.7a.wav';
46     FILENAME4 = 'DTS_FM_100.7a.wav';
47     %}
48
49     FILENAME1 = 'DTN_WIFI_2.42Gc.wav';
50     FILENAME2 = 'DTS_WIFI_2.42Gc.wav';
51     FILENAME3 = 'TP_WIFI_2.42Gc.wav';
52     FILENAME4 = 'YOGA_WIFI_2.42Gc.wav';
53
54 else
55     FILENAME1 = 'Nfloor232.wav';
56     FILENAME2 = 'Nfloor232.wav';
57     FILENAME3 = 'Nfloor232.wav';
58     FILENAME4 = 'Nfloor232.wav';
```

```
end
if (TYPE == 2)
    [IQ1, Fs] = audioread(FILENAME1);
    [IQ2, Fs] = audioread(FILENAME2);
    [IQ3, Fs] = audioread(FILENAME3);
    [IQ4, Fs] = audioread(FILENAME4);
elseif (TYPE == 1)
    IQ1 = randn(1E6,2);
    IQ2 = randn(1E6,2);
    IQ3 = randn(1E6,2);
    IQ4 = randn(1E6,2);
elseif (TYPE == 0)
    IQ1 = randn(1E6,2);
    IQ2 = IQ1;
    IQ3 = IQ1;
    IQ4 = IQ1;
end
% Pre-process IQ signals. Filter the signals to eliminate noise outside BW.
% Allows better calculation of average signal power. You may need to look
% at the pre-process.m file and see where the spectrum sits at the end. It
% can change a bit between files.
%b = getFilterTF(Fs,-BW/2,BW/2,100);
[x1,I1] = preProcess(IQ1,Fs,BW);
[x2,I2] = preProcess(IQ2,Fs,BW);
[x3,I3] = preProcess(IQ3,Fs,BW);
[x4,I4] = preProcess(IQ4,Fs,BW);
b1 = getFilterTF(Fs,I1-BW/2,I1+BW/2,100);
b2 = getFilterTF(Fs,I2-BW/2,I2+BW/2,100);
b3 = getFilterTF(Fs,I3-BW/2,I3+BW/2,100);
b4 = getFilterTF(Fs,I4-BW/2,I4+BW/2,100);
xa = x1; xb = x2; xc = x3; xd = x4;
x1 = filter(b1,1,x1); x2 = filter(b2,1,x2);
x3 = filter(b3,1,x3); x4 = filter(b4,1,x4);
x = [x1; x2; x3; x4];
N = size(x,2); T = 1/Fs;
% Calculate average signal power to inform noise generation for SNR levels
for i = 1:4
    Ps(i) = sum(x(i,:).^2)/N;
end
if (DIFF_PERF > 0)
    Ps = max(Ps)*([0.2 1 15 15]);
end
% If pre-filtering is disabled then the filtering was only used to
```

```
% determine the AWGN levels.
if (PRE_FILT == 0)
    x = [xa; xb; xc; xd];
end
% Perform survey to obtain feature vector
FEAT = survey(stand(x),T,SINGLE);
% Pre-allocate for noisy matrix of signals
x4 = nan(4,N);
detlvl = nan;
det = zeros(LOOPS,length(LOWLVL:0));
for SNR = LOWLVL:0
    idx = SNR+abs(LOWLVL)+1;
    for i = 1:4
        An(i) = sqrt(Ps(i)*10^-(SNR/10));
    end
    for k = 1:LOOPS
        % Add the noise
        for i = 1:4
            awgn = An(i)*randn(1,N);
            x4(i,:) = x(i,:) + awgn;
        end
        % Standardise the data
        if (STAND > 0)
            y = stand(x4);
        else
            y = x4;
        end
        [SNRinfer(idx,k), MMPDres(idx,k), MMEres(idx,k),FEAT2] = ...
            boolMMPD(Fs,y,0.3);
        % Transformed data matrix obtained
        yTr = FEAT*y;
        yTr2 = FEAT2*y;
        % Perform FFT on transformed data
        YTr = getFFT(yTr,N,T,1);
        YTr2 = getFFT(yTr2,N,T,1);
        if (SNR == 0 && k == LOOPS)

        end
        % Test transformed data against threshold
        det(k,idx) = mean(YTr) < FFT_THRESH;
        det2(k,idx) = mean(YTr2) < FFT_THRESH;
        % Perform FFT separately for all receivers for comparison
        Y1 = getFFT(x4(1,:),N,T,1);
```

```
144         Y2 = getFFT(x4(2,:),N,T,1);
145         Y3 = getFFT(x4(3,:),N,T,1);
146         Y4 = getFFT(x4(4,:),N,T,1);
147         % Test individual RX FFT outputs against threshold
148         det3(k,idx) = mean(Y1) < FFT_THRESH;
149         det4(k,idx) = mean(Y2) < FFT_THRESH;
150         det5(k,idx) = mean(Y3) < FFT_THRESH;
151         det6(k,idx) = mean(Y4) < FFT_THRESH;
152     end
153     % Calculate detection probability Pd for MMPD and MME
154     perfMMPD(idx) = 100*sum(MMPDres(idx,:))/size(MMPDres,2);
155     perfMME(idx) = 100*sum(MMEres(idx,:))/size(MMEres,2);
156     % Show animation during low iteration testing
157     if (LOOPS ≤ 100)
158         txtDet = sprintf('MMPD DETECT rate: %.2f %%',perfMMPD(idx));
159         figure(1); plot(YTr); title(['SNR: ',num2str(SNR),' dB']);
160         pause(0.1);
161     end
162 end
163 % Report performance at SNR
164 fprintf('---------- DETECTION PERFORMANCE ----------\n\n')
165 % Calculate detection probability Pd for all FFT detections
166 EVAFFT = sum(det,1)*100/LOOPS;
167 EVAFFT2 = sum(det2,1)*100/LOOPS;
168 FFT1 = sum(det3,1)*100/LOOPS;
169 FFT2 = sum(det4,1)*100/LOOPS;
170 FFT3 = sum(det5,1)*100/LOOPS;
171 FFT4 = sum(det6,1)*100/LOOPS;
172 xvals = (LOWLVL:0);
173 % Plot the results
174
175 figure();
176 xlim([LOWLVL 0]);
177 xlabel('SNR');
178 ylabel('Probability of Detection (%)');
179 xlim([LOWLVL 0]); ylim([0 100]);
180 title('EVA-FFT Probability of Detection');
181 hold on; grid on;
182 %plot(xvals,perfMMPD,'-o'); plot(xvals,perfMME,'-^');
183 plot(LOWLVL:0,EVAFFT,'-o'); plot(LOWLVL:0,EVAFFT2,'-^');
184 plot(LOWLVL:0,FFT1,'-*'); plot(LOWLVL:0,FFT2,'-d');
185 plot(LOWLVL:0,FFT3,'-s'); plot(LOWLVL:0,FFT4,'-p');
186 legend('EVAFFT','MMPD EVAFFT','FFT RX1','FFT RX2',...
```

```
187         'FFT RX3','FFT RX4','location','east');
188 if (DIFF_PERF > 0)
189     xlabel('Degradation Level');
190 else
191     xlabel('SNR');
192 end
193
194 % ************************************************************************
195 %{
196 AUTHOR:     Benedict Hardless
197 CREATED:    31-Aug-2021
198 MODIFIED:   31-Aug-21
199 TESTED:     31-Aug-21
200 FILE NAME:  preProcess.m
201 PURPOSE:    Provides a feature vector based on a block of pre-processed
202             and standardised receiver signals.
203 ARGUMENTS:  x is the pre-processed and standardised received signals block.
204                 it is an MxN matrix where M is number of receivers.
205             T is sample rate
206             SINGLE decides type of weighting scheme. 1 is best receive
207             signal only. 2 is weighted based on relative received signal
208             power. 0 is weighted evenly.
209 RETURNS:    feat is the feature vector 1xN.
210 %}
211 function [feat] = survey(x,T,SINGLE)
212     N = size(x,2);
213     feat = zeros(1,size(x,1));
214     if (SINGLE ~= 2)
215         for i = 1:size(x,1)
216             % Perform normalised FFT for each RX
217             X(i,:) = getFFT(x(i,:),N,T,1);
218             % Get the means
219             Xav(i) = mean(X(i,:));
220         end
221         % Find the lowest mean
222         [low idx] = min(Xav);
223         if (SINGLE == 1)
224             feat(idx) = 1;
225             elseif (SINGLE == 0)
226             feat = 1./Xav;
227         end
228     else
229         feat = [1 1 1 1];
```

```
230     end
231 end
```

```
1  %{
2  ******************************* FUNCTION *********************************
3  AUTHOR:      Benedict Hardless
4  CREATED:     24-Jul-2021
5  MODIFIED:    Jul-21, 29:30-Aug-21
6  TESTED:      Jul-21, 29:30-Aug-21
7  FILE NAME:   boolMMPD.m
8  PURPOSE:     Takes a pre-processed SDR IQ recording (converted to real time
9               domain format etc.) with a known sample rate and antenna count.
10              Calculates the inferred SNR and provides a Boolean result for
11              both MMPD and MME detection.
12 ARGS:        Fs is sample rate.
13              IQ is the pre-processed SDR signal matrix with dims NUM_ANT*N.
14              FINE is gamma from Ch 3, s3.2, equation 3.3.
15 RETURNS:     SNRinfer is MMPD inferred SNR.
16              MMPDres is Boolean MMPD detection result.
17              MMEres is Boolean MME detection result.
18              feat is the feature vector
19 %}
20
21 function [SNRinfer, MMPDres, MMEres, feat] = boolMMPD(Fs, IQ, FINE)
22     % 1 for optimum MMPD threshold. 0 for custom thresholds.
23     OPTION = 1;
24     EXACT = 0;
25     M = size(IQ,1);                 % Number of antennas
26     N = length(IQ);                 % Number of samples
27     POLYS = [1.993E-8,-4.818E-6,4.744E-4,-2.413E-2,8.177E-1,-1.894E1;...
28              3.247E-8,-8.1E-6,7.949E-4,-3.849E-2,1.111,-2.3E1;...
29              3.858E-8,-9.602E-6,9.315E-4,-4.427E-2,1.228,-2.53E1;...
30              5.106E-8,-1.293E-5,1.259E-3,-5.884E-2,1.513,-2.827E1;...
31              5.498E-8,-1.388E-5,1.345E-3,-6.237E-2,1.585,-2.973E1;...
32              5.056E-8,-1.249E-5,1.185E-3,-5.428E-2,1.412,-2.931E1;...
33              6.204E-8,-1.563E-5,1.504E-3,-6.902E-2,1.717,-3.216E1;...
34              6.099E-8,-1.52E-5,1.443E-3,-6.545E-2,1.631,-3.211E1;...
35              7.046E-8,-1.781E-5,1.712E-3,-7.825E-2,1.908,-3.477E1];
36     SNRCUTS = -[17.701,18.989,20.96,21.007,22.048,22.345,22.849,23.529,...
37                 24.152];
38     DCUTS = [1.7644 2.9561 3.6597 4.5142 5.1995 5.6836 6.377 6.783 7.1653];
```

```
39      % Define key parameters for the desired cutoff points
40      % Characteristic polynomial per table 3.1
41      P = POLYS(M-1,:);
42      % SNR degraded values
43      Dco = DCUTS(M-1);
44      % MMPD Threshold(s)
45      if (OPTION == 1)
46          if (EXACT == 0)
47              % For Pfa ≤ 10% when no active signal present
48              phi = -0.0555*M^2+1.561*M+0.5254+FINE;
49          else
50              % Exact for Pfa ≤ 10% when 50% active signal present
51              phiAll = ...
                    [nan,3.275,4.84,5.975,6.9875,7.875,8.67,9.35,10.05,10.68];
52              phi = phiAll(M);
53          end
54      else
55          phi = Dco*[1.8 1.5 1.2];
56      end
57      SNRphi = polyval(P,phi);
58      SNRcut = SNRCUTS(M-1);
59      % MME threshold
60      L = 1;  % Smoothing factor for MME
61      a = sqrt(N); b = sqrt(M*L); % These to shorten the equation next line
62      gamma = ((a+b)^2/(a-b)^2)*(1+((a+b)^(-2/3))/((N*M*L)^(1/6))*0.45);
63      % Process using MMPD and MME
64      C = IQ*IQ'/N;                   % Form covariance matrix
65      [Vx,Dx] = eig(C);               % Get max eigenvectors and values
66      eigs = diag(Dx);                % Isolate the eigenvalues
67      [eigs, eigIdx] = sort(eigs);    % Sort eigenvalues low to high
68      % Change to be high to low
69      eigs = flipud(eigs); eigIdx = flipud(eigIdx);
70      % Get percent difference between strongest/weakest eigenvalues
71      D = 100*((eigs(1)-eigs(end))/eigs(1));
72      % Convert to SNR estimate
73      SNRinfer = polyval(P,D);
74      % If the MMPD inferred SNR is above threshold, signal is present
75      if (SNRinfer > SNRphi)
76         MMPDres = 1;
77      % If MMPD inferred SNR is below threshold, signal is not present
78      else
79         MMPDres = 0;
80      end
```

```
81      feat = Vx(:,eigIdx(1))';
82      % If the MME ratio is above the threshold, signal is present
83      MMEratio = abs(eigs(1)/eigs(end));
84      if (MMEratio > gamma) MMEres = 1;
85      else MMEres = 0;
86      end
87  end
```

```
1   %**********************************************************************
2   %{
3   Author:     Benedict Hardless
4   Created:    18-Jul-2021
5   Name:       getFFT
6   Purpose:    Simplify application of fft by scaling the values.
7   Arguments:  X is the signal for FFT.
8               N is number of samples.
9               T is sampling interval.
10              NORM is 0, but change to 1 for normalised amplitudes.
11  Returns:    Y is vector of amplitudes
12              f is vector of corresponding frequencies
13  %}
14  function [Y, f] = getFFT(X,N,T,NORM)
15      n = 1:N;
16      Y = fft(X);
17      Y = abs(Y)*2/N;
18      if (NORM > 0)
19          Y = Y/max(Y);
20      end
21      f = ((n-1)/(N*T));
22  end
23  %**********************************************************************
```

```
1   %{
2   returns b coefficients for use in filter(b,1,x) for a bandpass
3   filter with limits fL and fH. Higher order filters are more defined.
4   %}
5   function h = getFilterTF(Fs,fL,fH,order)
6
7   a = 2*pi*fL/Fs;
8   % Set upper filter pass freq
```

```
b =2*pi*fH/Fs;
% Filter order
M = order;
% Number of coefficients
N = M+1;
% Span of n values for our filter. More is more precise.
n = -M/2:M/2;
% Get the coefficients for the TF by solving for values of n.
h = 1./((n+eps)*pi).*(sin(b*(n+eps))-sin(a*(n+eps)));
% Include if you want a hamming window
n = 0:M;
hamwin = 0.54-0.46*cos(2*n*pi/M);
h = h.*hamwin;
%{
UNCOMMENT TO SEE FILTER RESPONSE PLOTTED
% Set frequency range to look at for filter response
dw = 1/400;
w = 0:dw:pi;
% Define z and get filter response
z = exp(j*w);
resp = 0;
for idx = 1:N
    if i < M
        resp = resp + h(idx).*z.^(-idx);
    end
end
% Scale the frequency for the defined sampling frequency
f = w*Fs/(2*pi);
% Plot result
plot(f,abs(resp)); grid on; grid minor;
title('Filter Frequency Response');
ylabel('Gain'); xlabel('Frequency (Hz)');
%}

end
```

```
%*****************************************************************
%{
Author:     Benedict Hardless
Created:    18-Aug-2021
Name:       stand
```

```
6  Purpose:    Standardise a data set for PCA or MMPD.
7  Arguments:  x is the data set. Rows per receive antenna, columns for
8              observations.
9  Returns:    y is the standardised data.
10 %}
11 function y = stand(x)
12     % Standard deviation for each antenna's observation set
13     S = std(x,0,2);
14     % Mean for each antenna's observation set
15     M = mean(x,2);
16     % Standardised results are z scores
17     for i = 1:size(x,1)
18         y(i,:) = (x(i,:) - M(i))/S(i);
19     end
20 end
```

```
1  function [feat] = survey(x,N,T,SINGLE)
2      for i = 1:size(x,1)
3          % Perform normalised FFT for each RX
4          X(i,:) = getFFT(x(i,:),N,T,1);
5          % Get the means
6          Xav(i) = mean(X(i,:));
7      end
8      % Find the lowest mean
9      [low idx] = min(Xav);
10     feat = zeros(1,size(x,1));
11     % Select receiver or combination of receivers and return weighting
12     if (SINGLE == 1)
13         feat(idx) = 1;
14     elseif (SINGLE == 0)
15         feat = 1./Xav;
16     else
17         feat = [1 1 1 1];
18     end
19 end
```